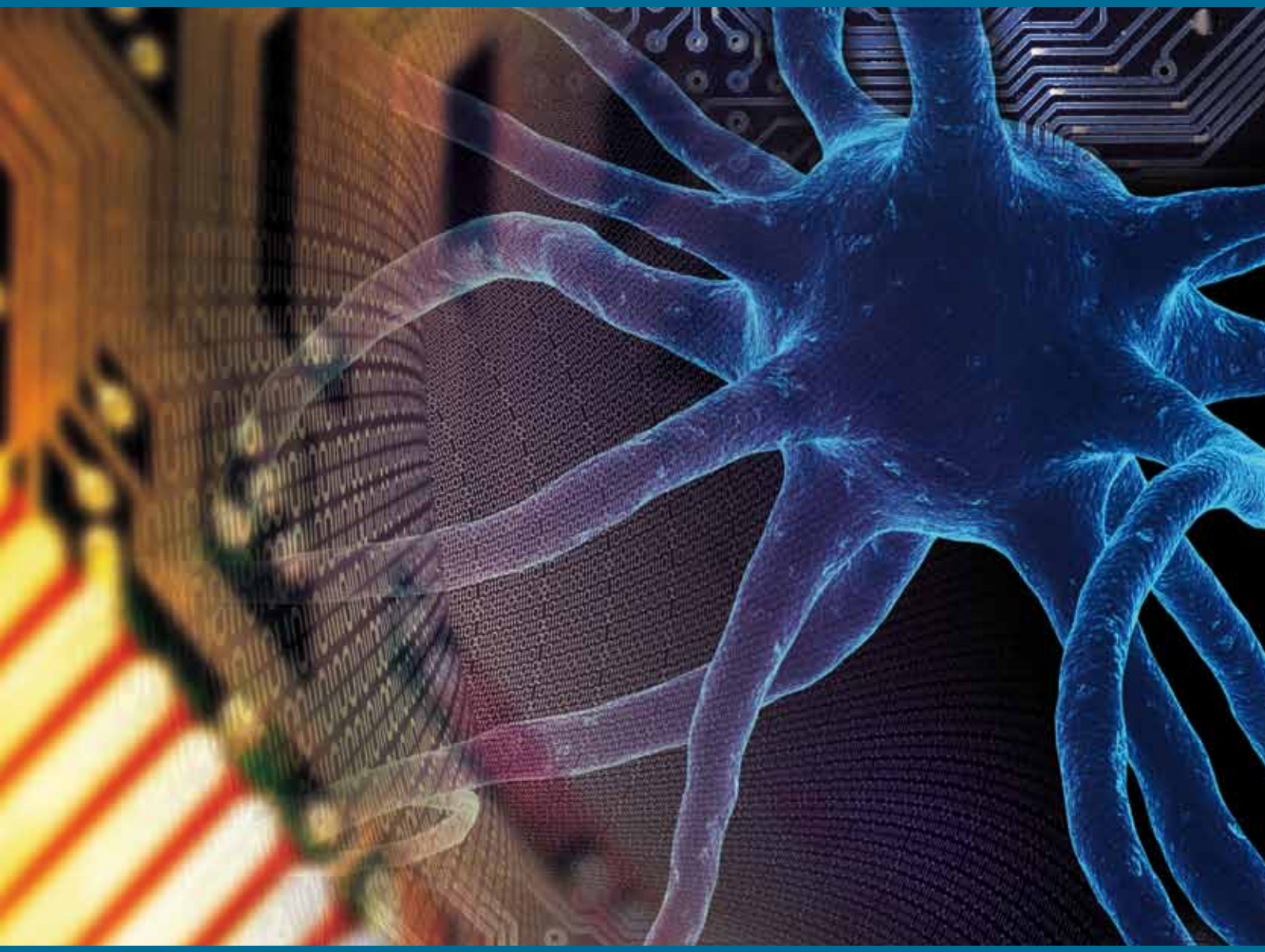
Advances in Artificial Neural Systems

Advances in Unsupervised Learning Techniques Applied to Biosciences and Medicine

Guest Editors: Anke Meyer-Baese, Sylvain Lespinats, Juan Manuel Gorriz Saez, and Olivier Bastien



Advances in Unsupervised Learning Techniques Applied to Biosciences and Medicine

Advances in Unsupervised Learning Techniques Applied to Biosciences and Medicine

Guest Editors: Anke Meyer-Baese, Sylvain Lespinats, Juan Manuel Gorriz Saez, and Olivier Bastien

Editorial Board

Contents

Hindawi Publishing Corporation
Advances in Artificial Neural Systems
Volume 2012, Article ID 219860, 2 pages
doi:10.1155/2012/219860

Editorial

Advances in Unsupervised Learning Techniques Applied to Biosciences and Medicine

Anke Meyer-Baese,[1] Sylvain Lespinats,[2] Juan Manuel Gorriz Saez,[3] and Olivier Bastien[4]

[1] *Department of Scientific Computing, Florida State University, Tallahassee, FL 32306-4120, USA*
[2] *Laboratoire des Systemes Solaires (L2S), Institut National de l'Energie Solaire (CEA/INES), BP 332, 73377 Le Bourget du Lac, France*
[3] *Department of Signal Theory, Telematics and Communications, Facultad de Ciencias, Universidad de Granada Fuentenueva, s/n, 18071 Granada, Spain*
[4] *Laboratoire de Physiologie Cellulaire Végétale, UMR 5168 CEA-CNRS-INRA-Université Joseph Fourier, CEA Grenoble, 38054 Grenoble Cedex 09, France*

Correspondence should be addressed to Anke Meyer-Baese, ameyerbaese@fsu.edu

Received 29 August 2012; Accepted 29 August 2012

1. Introduction

Artificial neural systems have been for the past thirty years a fascinating research topic with contributions from both theoreticians as well as from researchers implementing novel computational learning techniques into numerous application fields. Starting as an attempt to emulate human brain processing and cognition, supervised and unsupervised learning techniques as well as hybrid concepts have emerged for information visualization and discovery mining in high dimensional spaces. The properties of these spaces are very different from what we usually encounter in the more intuitive low dimensional spaces; the "curse of dimensionality," for example, often impacts the performance of data-mining tools. However, with the increasing demand of information and knowledge processing, and integration of information from heterogeneous sources into biomedical decision tools and resources for health care, innovative learning approaches become imperative. Especially, when facing a huge number of data descriptors as it is the case in almost all applications in life sciences.

The aim of this special issue is to present the current state of the art in the theory of unsupervised learning and applications by active experts researching in the vast area of biosciences and medicine.

2. Novel Learning Algorithms

The paper by X. Zhang et al. presents an indirect learning method that changes the synaptic weights by modulating spike-timing-dependent plasticity based on controlled input spike trains. V. Norris et al. propose a novel learning concept—competitive coherence—that appears to be relevant to a large group of biological systems and describe the differences to other existing content-addressable systems.

3. Data Compression of Non-Gaussian Signals

C. Plant et al.'s paper introduces a very general technique for evaluating ICA (Independent Component Analysis) results rooted in information-theoretic model selection and has potential applications to data compression.

4. Applications in Brain Research

A. Ortiz et al. proposes alternatives to MR (Magnetic Resonance) brain image segmentation algorithms based on self-organized maps and does not use any a priori information about the voxels in order to classify different tissue classes. R. Duan's and H. Man's paper proposes two novel unsupervised learning methods for analyzing functional magnetic resonance imaging (fMRI) data based on hidden Markov model (HMM). The HMM approach is focused on capturing the first-order statistical evolution among the samples of a voxel time series and not like conventional approaches to model the blood oxygen level-dependent (BOLD) response of a voxel as a function of time.

5. Applications in Mammography

F. Steinbruecker et al.'s paper "*Selection of spatiotemporal features in breast MRI (Magnetic Resonance Imaging) to differentiate between malignant and benign small lesions using computer-aided diagnosis*" addresses the important aspect of motion compensation in breast MRI (Magnetic Resonance Imaging) and provides a new nonrigid technique for it. F. Steinbruecker et al.'s paper "*Evaluation of a nano-rigid motion compensation technique based on spatiotemporal features for small lesion detection in breast MRI (Magnetic Resonance Imaging)*" extracts spatiotemporal features for small lesions and evaluates their discriminative power in connection with motion compensation.

6. Sleep Staging

M. Langkvist et al.'s paper addresses the important aspect of sleep stage classification. He proposes deep belief nets (DBN), an unsupervised learning paradigm, and shows how to apply it to sleep data in order to eliminate the use of hand-made features.

7. Hemodialysis

The paper by T.-C. Lu et al. presents an entropy function to identify key features related to hemodialysis and based on these determined features a decision is made whether a patient needs hemodialysis.

In closing, many novel algorithmic techniques emerged for unsupervised learning and much more applications in biosciences and medicine are identified. Artificial neural network research has matured over the past decades and the power of unsupervised learning techniques is increasingly recognized. This special issue is brought to the reader with the hope to inform about the wealth of applications of unsupervised learning techniques and their indispensable role in biology and medicine.

Acknowledgments

We would like to thank all authors who have contributed to the special issue as well as all reviewers for their time and effort.

Anke Meyer-Baese
Sylvain Lespinats
Juan Manuel Gorriz Saez
Olivier Bastien

Hindawi Publishing Corporation
Advances in Artificial Neural Systems
Volume 2012, Article ID 713581, 16 pages
doi:10.1155/2012/713581

Research Article

A Radial Basis Function Spike Model for Indirect Learning via Integrate-and-Fire Sampling and Reconstruction Techniques

X. Zhang,[1] G. Foderaro,[1] C. Henriquez,[2] A. M. J. VanDongen,[3] and S. Ferrari[1]

[1] *Laboratory for Intelligent Systems and Control (LISC), Department of Mechanical Engineering and Materials Science, Duke University, Durham, NC 27708, USA*
[2] *Department of Biomedical Engineering and Department of Computer Science, Duke University Durham, NC 27708, USA*
[3] *Program in Neuroscience & Behavioral Disorders, Duke-NUS Graduate Medical School, Singapore, Singapore*

Correspondence should be addressed to S. Ferrari, sferrari@duke.edu

Received 17 February 2012; Accepted 21 May 2012

Academic Editor: Olivier Bastien

This paper presents a deterministic and adaptive spike model derived from radial basis functions and a leaky integrate-and-fire sampler developed for training spiking neural networks without direct weight manipulation. Several algorithms have been proposed for training spiking neural networks through biologically-plausible learning mechanisms, such as spike-timing-dependent synaptic plasticity and Hebbian plasticity. These algorithms typically rely on the ability to update the synaptic strengths, or weights, directly, through a weight update rule in which the weight increment can be decided and implemented based on the training equations. However, in several potential applications of adaptive spiking neural networks, including neuroprosthetic devices and CMOS/memristor nanoscale neuromorphic chips, the weights cannot be manipulated directly and, instead, tend to change over time by virtue of the pre- and postsynaptic neural activity. This paper presents an indirect learning method that induces changes in the synaptic weights by modulating spike-timing-dependent plasticity by means of controlled input spike trains. In place of the weights, the algorithm manipulates the input spike trains used to stimulate the input neurons by determining a sequence of spike timings that minimize a desired objective function and, indirectly, induce the desired synaptic plasticity in the network.

1. Introduction

This paper presents a deterministic and adaptive spike model obtained from radial basis functions (RBFs) and a leaky integrate-and-fire (LIF) sampler for the purpose of training spiking neural networks (SNNs), without directly manipulating the synaptic weights. Spiking neural networks are computational models of biological neurons comprised of systems of differential equations that can reproduce some of the spike patterns and dynamics observed in real neuronal networks [1, 2]. Recently, SNNs have also been shown capable of simulating sigmoidal artificial neural networks (ANNs) and of solving small-dimensional nonlinear function approximation problems through reinforcement learning [3–5]. Like all ANN learning techniques, existing SNN training algorithms rely on the direct manipulation of the synaptic weights [4–9]. In other words, the learning algorithms typically include a weight-update rule by which the synaptic weights are updated over several iterations, based on the reinforcement signal or network performance.

In many potential SNN applications, including neuroprosthetic devices, light-sensitive neuronal networks grown *in vitro*, and CMOS/memristor nanoscale neuromorphic chips [10], the synaptic weights cannot be updated directly by the learning algorithm. In several of these applications, the objective is to stimulate a network of biological or artificial spiking neurons to perform a complex function, such as processing an auditory signal or restoring a cognitive function. In neuroprosthetic medical implants, for example, the artificial device may consist of a microelectrode array or integrated circuit that stimulates biological neurons via spike trains. Therefore, the device is not capable of directly modifying the synaptic efficacies of the biological neurons, as do existing SNN training algorithms, but it is capable of

stimulating a subset of neurons through controlled pulses of electrical current.

As another example, light-sensitive neuronal networks grown *in-vitro* can be similarly stimulated through controlled light patterns that cause selected neurons to fire at precise moments in time, in an attempt to induce plasticity, while their output is being recorded in real time using a multi-electrode array (MEA) [11]. In this case, a digital computer can be used to determine the desired stimulation patterns for an *in-vitro* neuronal network with random connectivity, produced by culturing dissociated cortical neurons derived from embryonic day E18 rat brain [11, 12]. As a result, the cultures may be used to verify biophysical models of the mechanisms by which biological neuronal networks execute the control and storage of information via temporal coding and learn to solve complex tasks over time. In these networks, the actual connectivity and synaptic plasticities are typically unknown and cannot be manipulated directly as required by existing SNN learning algorithms.

This paper presents a novel indirect learning approach and algorithm that assume synaptic weights cannot be updated or manipulated at any time. The learning algorithm induces changes in the SNN weights by modulating spike-timing dependent plasticity (STDP) through controlled input spike trains. The algorithm adapts the input spike trains that are used to stimulate the input neurons of the SNN and, thus, are realizable through controlled pulses of electric voltage or controlled pulses of blue light. The main difficulty to be overcome in indirect learning is that the algorithm aims at adapting pulse signals, such as square waves, in place of continuous-valued weights. While available in closed analytic form, these signals typically are represented by piece-wise continuous, multi-valued (or many-to-one), and nondifferentiable functions that are difficult to adapt or update using optimization or reinforcement-learning algorithms. Furthermore, stimulation patterns typically are generated by spike models that are stochastic, such as the Poisson spike model [5, 13–15]. Thus, even when the spike model is optimized, it does not allow for precise timing of pre- and post-synaptic firings, and as a result, may induce undesirable changes in the synaptic weights.

In this paper, a deterministic spike model that allows for precise timing of neuron firings is obtained using adaptive RBFs to model the characteristics of the spike pattern through a continuous and infinitely differentiable function that also is one to one. The RBF model is combined with an LIF sampling technique originally developed in [16, 17] for the approximate reconstruction of bandlimited functions. It is shown that, by this approach, the spike trains generated by the LIF sampler display the precise characteristics specified by the RBF model. Furthermore, this deterministic spike model can be optimized to modulate STDP through controlled input spike trains that bring about the desired SNN weight change without direct manipulation. The indirect learning approach presented in this paper is applicable both in supervised and unsupervised settings. In fact, when the desired SNN output is unknown, it can be replaced by a reinforcement signal produced by a critic SNN, as shown by the adaptive critic method reviewed in Section 3.

The paper is organized as follows. The model of spiking neural network used to derive and demonstrate the training equations is presented in Section 2. In Section 3, an adaptive critic approach is described to illustrate how the proposed indirect training methodology can be applied using reinforcement learning, for example, to model or control a dynamical plant. The novel spike model and indirect training methodology are presented in Section 4 and demonstrated on a benchmark problem involving a two-node spiking neural network. The generalized form of gradient equations for indirect training is presented in Section 5 and demonstrated through a three-node spiking neural network. These gradient equations show how the methodology can be generalized to any spiking neural network with the characteristics described in the next section.

2. Spiking Neural Network Model

2.1. Models of Neuron and Synapse. Various models of SNNs have been proposed in recent years, motivated by biological studies that have shown complex spike patterns and dynamics to be an essential component of information processing and learning in the brain. The two crucial considerations involved in determining a suitable SNN model are the range of neurocomputational behaviors the model can reproduce, and its computational efficiency [18]. As can be expected, the implementation efficiency typically increases with the number of features and behaviors that can be accurately reproduced [18], such that each model offers a tradeoff between these competing objectives. One of the computational neuron models that is most biophysically accurate is the well-known Hodgkin-Huxley (HH) model [2]. Due to its extremely low computational efficiency, however, using the HH model to simulate large networks of neurons can be computationally prohibitive [18]. Recently, bifurcation studies have been used to reduce the HH dynamics from four to two differential equations, referred to as the Izhikevich model, which are capable of reproducing a wide range of spiking patterns and behaviors with much higher efficiency than the HH model [19].

In [15], the authors proposed an indirect training method based on a Poison spike model and demonstrated it on a network of Izhikevich neurons. In this work it was found that the adaptive critic architecture described in Section 3 could be implemented without modeling the neuron response in closed form. However, the effectiveness of the approach in [15] was limited in that, due to the use of a stochastic Poison model, the Izhikevich SNN could not converge to the optimal control law. Therefore, in this paper, a new deterministic spike model is proposed and implemented by deriving the training equations in closed form, using a leaky integrate-and-fire (LIF) SNN. The LIF is the simplest model of spiking neuron. It has the advantages that it displays the highest computational efficiency and is amenable to mathematical analysis [13, 14].

The LIF membrane potential, $v(t)$, is governed by

$$C_m \frac{dv(t)}{dt} = I_{\text{leak}}(t) + I_s(t) + I_{\text{inj}}(t), \tag{1}$$

where C_m is the membrane capacitance, $I_{\text{leak}}(t)$ is the current due to the leak of the membrane, $I_s(t)$ is the synaptic input to the neuron, and $I_{\text{inj}}(t)$ is the current injected to the neuron [20]. The leak current is defined as follow:

$$I_{\text{leak}}(t) = -\frac{C_m}{\tau_m}[v(t) - V_0], \tag{2}$$

where V_0 is the resting potential and τ_m is the passive-membrane time constant [20]. τ_m is related to the capacitance, C_m, and the membrane resistance, R_m, of the membrane potential by $\tau_m = R_m C_m$.

The response of the membrane potential is obtained by solving the differential equation (1) analytically for $v(t)$, such that

$$v(t) = V_0 + e^{-t/\tau_m} \int_{t_0}^{t} \frac{I_{\text{inj}}(\rho)}{C_m} e^{\rho/\tau_m} d\rho, \tag{3}$$

where ρ is a dummy variable used for integration and t_0 is the time at which the membrane potential equals V_0 [20]. Whenever the membrane potential, $v(t)$, reaches a prescribed threshold value, V_{th}, the neuron will fire. At this point, we have considered an isolated neuron that is stimulated by an external current, $I_{\text{inj}}(t)$. When the LIF neuron (1) is part of a larger network, the input current, referred to as the *synaptic current*, is generated by the activity of presynaptic neurons.

Let the synaptic current, denoted by $I_s(t)$, be modeled as the sum of the currents of *excitatory* presynaptic neurons and of *inhibitory* presynaptic neurons,

$$I_s(t) = C_m \sum_{k=1}^{N_E} a_{E,k} S_{E,k}(t) + C_m \sum_{k=1}^{N_I} a_{I,k} S_{I,k}(t), \tag{4}$$

where the subscript E denotes inputs from *excitatory* neurons, while the subscript I denotes inputs from *inhibitory* neurons. The amplitudes, $a_{E,k} > 0$ and $a_{I,k} < 0$, represent the change in potential due to a single synaptic event and depend on the weight of the synapse. N_E and N_I are the numbers of excitatory and inhibitory current synapses, respectively. $S_{E,k}$ and $S_{I,k}$ describe the excitatory and inhibitory synaptic inputs as a series of input spikes to each synapse. The synaptic inputs are modeled as a sum of instantaneous impulse functions,

$$S_{E,k}(t) = \sum_{t_{E,k}} \delta(t - t_{E,k}), \tag{5}$$

$$S_{I,k}(t) = \sum_{t_{I,k}} \delta(t - t_{I,k}), \tag{6}$$

where $t_{E,k}$ and $t_{I,k}$ are the firing times of the presynaptic neurons and δ represents the Dirac delta function [21].

In addition to the above deterministic properties, there are prevalent stochastic qualities in biological neural networks caused by effects such as thermal noise and random variations in neurotransmitters. For simplicity, these effects are assumed negligible in this paper. However, the reader is referred to [15] for a technique that can be used to incorporate these effects in the above SNN model.

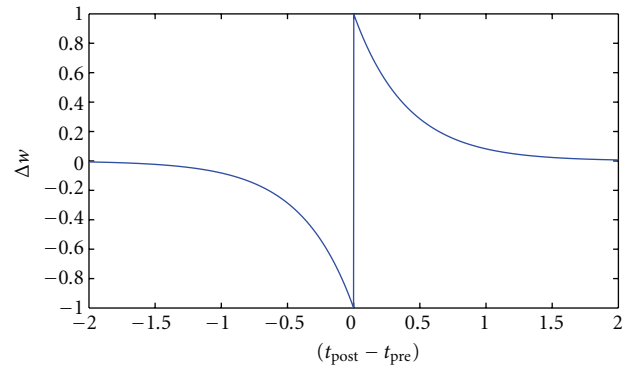


Figure 1: STDP term as a function of the time delay between the last spike of postsynaptic neuron and presynaptic neuron.

2.2. Model of Spike-Timing-Dependent Plasticity (STDP). A persistent learning mechanism known as spike-timing-dependent plasticity, recently observed in biological neuronal networks, is used in this paper to model synaptic plasticity in the LIF SNN. Synaptic plasticity refers to the mechanism by which the synaptic efficacies or *strengths* between neurons are modified over time, typically as a result of the neuronal activity. These changes are known to be driven in part by the correlated activity of adjacently connected neurons. The directions and magnitudes of the changes are dependent on the relative timings of the presynaptic spike arrivals and postsynaptic firings. For simplicity, in this paper, all changes in synaptic strengths are assumed to occur solely as a result of the spike-timing-dependent plasticity mechanism. The approach presented in [5] can be used to also incorporate a model of the Hebbian plasticity.

Spike-timing-dependent plasticity (STDP) is known to modify the synaptic strengths according to the relative timing of the output and input action potentials, or spikes, of a particular neuron. If the presynaptic neuron fires shortly before the postsynaptic neuron, the strength of the connection will be increased. In contrast, if the presynaptic neuron fires after the postsynaptic neuron, the strength of the connection will be decreased, as illustrated in Figure 1. Two constants τ_+ and τ_- determine the ranges of the presynaptic to postsynaptic interspike intervals over which synaptic strengthening and weakening occur. Let the synaptic efficacy or strength be referred to as *weight* and denoted by w. A is the maximum amplitude of the change of weight due to a pair of spikes. t_{pre} and t_{post} denote firing times of the presynaptic neuron and the postsynaptic neuron, respectively. Then, for each set of neighboring spikes, the weight adjustment is given by,

$$\Delta w = A e^{[(t_{\text{pre}} - t_{\text{post}})\tau]}, \tag{7}$$

such that the connection weight increases when the postsynaptic spike follows the presynaptic spike and the weight decreases when the opposite occurs. The amplitude of the adjustment Δw lessens as the time between the spikes becomes larger, as is illustrated in Figure 1.

Different methods can be used to identify the spikes that give rise to the STDP mechanism. In this paper, the *nearest-spike STDP model* [22] is adopted, by which, for every spike of the presynaptic neuron, its nearest postsynaptic spike is used to calculate the timing difference in (7), regardless of whether it takes place before or after the presynaptic firing. Then, from (7), the weight change due to one of the spikes of the ith neuron is modeled by the rule

$$\Delta w_i(\Delta t_i) = \begin{cases} A_+ e^{\Delta t_i/\tau_+} & \text{if } \Delta t_i \le 0 \\ A_- e^{-\Delta t_i/\tau_-} & \text{if } \Delta t_i > 0, \end{cases} \tag{8}$$

where

$$\Delta t_i = t_{1,i} - \underset{t_{2,k}}{\operatorname{argmin}} \left| t_{1,i} - t_{2,k} \right|. \tag{9}$$

Δt_i is the firing time difference between the two neurons, $t_{1,i}$ is the firing time of the presynaptic neuron, and $t_{2,i}$ is the firing time of the postsynaptic neuron. The constants used in this paper are $\tau_+ = \tau_- = 20\,\text{ms}$, $A_+ = 0.006$, and $A_- = 1.5A_+$, where A_+ and A_- determine the maximum amounts of synaptic modification that occur when Δt is approximately zero [21].

From (8), the firing time of the postsynaptic neuron is chosen such that the absolute firing time difference between the presynaptic neuron and the postsynaptic neuron, $|t_{1,i} - t_{2,k}|$, is minimized. Therefore, the postsynaptic spike $t_{2,k}$ that is nearest to the presynaptic firing time $t_{1,i}$ is used to calculate the firing time difference Δt_i. The value of $t_{2,k}$ that minimizes $|t_{1,i} - t_{2,k}|$ can be found by comparing the firing time difference of the two neurons. In order to obtain an indirect learning method that does not rely on the direct manipulation of the synaptic weights, in this paper, it is assumed that the weights can only be modified according to the STDP rule (8). Also, the training algorithm cannot specify any of the terms in (8) directly but can only induce the firing times in (8) by stimulating the input neurons, for example, through controlled pulses of electric voltage or blue light.

3. Adaptive Critic Architecture for Indirect Reinforcement Learning

Many approaches have been proposed to train SNNs by modifying the synaptic weights by means of reward signals or by update rules inspired by reward-driven Hebbian learning and modulation of STDP [5, 8]. However, these methods are not applicable to approaches involving biological neuronal networks (e.g., neuronal cultures grown *in vitro*), because biological synaptic efficacies cannot be manipulated directly by the training algorithm. Similarly, in nanoscale neuromorphic chips, the CMOS/memristor synaptic weights cannot be manipulated directly but may be modified via controlled programming voltages that induce a mechanism analogous to STDP, as demonstrated in [10]. This paper presents an approach for training an SNN through controlled input pulse signals (e.g., voltages or blue light) that are generated by a new deterministic and adaptive spike model presented in Section 4.1.

The derivation of the training equations used to adapt the proposed spike model and subsequent pulse signals are demonstrated in Section 4.2 using a two-node LIF SNN. It is shown that the synaptic weight can be updated by the proposed indirect training method and driven precisely to a desired value, by minimizing a function of the synaptic weight error. It follows that, using a simple chain rule, the same training equations can be used to minimize a function of the decoded SNN output error, as is typically required by supervised training. In this section, we show how the indirect training method can be used for reinforcement learning, using a critic SNN to modulate the STDP mechanisms in an action SNN with synaptic weights that cannot be manipulated directly (e.g., a biological neuronal network).

Let NN_a represent an action SNN of LIF neurons, exhibiting STDP and modeled as described in Section 2. It is assumed that the weights of the action SNN cannot be manipulated directly, and the only controls available to the learning algorithm are the training signals (Figure 2), comprised of programming voltages that can be delivered using a square wave or the Rademacher function (Section 4.1). Now, let NN_c denote a critic SNN of feedforward fully connected HH neurons. In this architecture, schematized in Figure 2, NN_a is treated as a biological network and NN_c is treated as an artificial network implemented on a computer or integrated circuit. Thus, the synaptic weights of NN_c, defined as w_{ij}, are directly adjustable, while the synaptic efficacies of NN_a can only be modified through a simulated STDP mechanism which is modulated by the input spikes from NN_c. The algorithm presented in this section trains the network by changing the values of w_{ij} inside NN_c, which are assumed to be bounded by a positive constant $w_{\max}$ such that $-w_{\max} \le w_{ij} \le w_{\max}$, for all i, j.

In this paper, spike frequencies are used to code continuous signals, and the leaky integrator

$$\hat{u} = \alpha \sum_{t_k \in S_i(T)} e^{\beta(t_k - t)} H(t - t_k) - \gamma \tag{10}$$

is used to decode spike trains and convert them into continuous signals as required according to the architecture in Figure 2 Where, α, β, and γ are user-specified constants, $H(\cdot)$ is the Heaviside function, and $S_i(T)$ is the set of spiking times of the output neuron. One advantage of the leaky integrator decoder is its effectiveness of filtering inevitable noise during the flight control without influencing the speed of matching the continuous value with the target function.

3.1. Adaptive Critic Recurrence Relations. Typically, adaptive critic algorithms are used to update the actor and critic weights by computing a synaptic weight increment through an optimization-based algorithm, such as backpropagation ([23], page 359). Therefore a new approach is required in order to apply adaptive critics to SNNs in which changes in the synaptic weights can occur only as a result of pre- and postsynaptic neuronal activity. The training approach presented in Section 4 can be combined with the policy and value-iteration procedures described in this section to supervise stimulation patterns in the action SNN such

that the synaptic strengths, and subsequently their dynamic mappings, are adapted according to the STDP rule in (8).

Suppose the action SNN is being trained to control or model a dynamical system which obeys a nonlinear differential equation of the form

$$\dot{x}(t) = f[x(t), u(t), t], \tag{11}$$

where $x \in X \subset \mathbb{R}^n$ and $u \in U \subset \mathbb{R}^m$ denote the dynamical system state and control inputs, respectively, and $\dot{x}$ denotes the derivative of x with respect to time. The model of the dynamical system (11) may be unknown or imperfect and may be improved upon over time by a system identification (ID) algorithm. The macroscopic behavioral goals of the plant can be expressed by the value function or *cost-to-go*

$$V^{\pi}[x(t_k)] = \sum_{t_k=t}^{t_f-1} \mathcal{L}[x(t_k), u(t_k), x(t_k+1)], \tag{12}$$

where $u(t_k) = \pi[x(t_k)]$ is the unknown control law or dynamic mapping to be learned by the action SNN, NN_a. Then, the cost-to-go in (12) can be used to represent the future performance of the action network and dynamical system, as is accrued from the present, t_k, up to the final time, t_f, if subject to the present control law $\pi[\cdot]$. The Lagrangian $\mathcal{L}[\cdot]$ represents instantaneous behavioral goals as a function of x and u.

Since the dynamic mapping $\pi[\cdot]$ must be adapted over time through the learning algorithm and an accurate plant model may not be available, the cost-to-go typically is unknown and is learned by the critic, NN_c. Then, by Bellman's principle of optimality [24], at any time t_k the cost-to-go can be minimized online with respect to the present control law, based on the known value of $x(t_k)$ and predictions of $x(t_k + 1)$. The value of $x(t_k)$ is assumed to be fully observable from the dynamical system at any present time t_k, and $x(t_k + 1)$ is predicted or estimated from the approximation of the system's dynamics (11), based on the present values of the state and the control. In [25], Howard showed that if iterative approximations of the control law and optimal cost-to-go, denoted by π_ℓ and V_ℓ, respectively, are modified by a policy-improvement routine and a value-determination operation, respectively, they eventually converge to their optimal counterparts π^* and V^*.

The policy-improvement routine states that, given a cost-to-go function $V(\cdot, \pi_\ell)$ corresponding to a control law π_ℓ, an improved control law $\pi_{\ell+1}$ can be obtained as follows:

$$\pi_{\ell+1}[x(t_k)] = \arg\min_{u \in U} \{\mathcal{L}[x(t_k), u(t_k), x(t_k+1)] + V[x(t_{k+1}, \pi_\ell)]\}, \tag{13}$$

such that $V[x(t_k), \pi_{\ell+1}] \leq V[x(t_k), \pi_\ell]$, for any $x(t_k)$. Furthermore, the sequence of functions $C = \{\pi_\ell \mid \ell = 0, 1, 2, \ldots\}$ converges to the optimal control law π^*. The value-determination operation states that given a control law $\pi(\cdot)$, the cost-to-go can be updated according to the following rule:

$$V_{\ell+1}[x(t_{k+1}), \pi] = \mathcal{L}[x(t_k), u(t_k), x(t_k+1)] + V_{\text{ell}}[x(t_{k+1}, \pi)], \tag{14}$$

such that the sequence of functions $V = \{V_\ell \mid \ell = 0, 1, 2, \ldots\}$ converges to V^*.

Then, at every iteration cycle ℓ of the adaptive critic algorithm, the above policy and value-iteration procedures can be used to in tandem to supervise training of the actor and critic SNN; that is,

$$u(t_k) \longleftarrow NN_a[\ell, x(t_k)] \approx \pi_\ell[x(t_k)], \tag{15}$$

$$V^{\pi}[x(t_k)] \longleftarrow NN_c[\ell, x(t_k), \pi] \approx V_\ell[x(t_k), \pi], \tag{16}$$

respectively where $\ell = Mk$ and M is a positive constant chosen based on the desired frequency of the updates. It can be seen that, at $\ell + 1$, the desired mappings for NN_a and NN_c, provided by (13) and (14), respectively, are based on the actor and critic outputs at ℓ, and on $x(t_{k+1})$, which can be estimated from an available dynamical system model (11). To accomplish (15) and (16), two error metrics defined as a function of the actual (decoded) actor and critic outputs and of the desired actor and critic outputs (13) and (14) must be minimized by the chosen training algorithm, as explained in the following subsection.

3.2. Action and Critic Network Training Algorithm. The action and critic networks implemented in Figure 2 differ in that while the critic network can be trained by a conventional algorithm that manipulates the synaptic weights directly (e.g., see [5, 15]), the action network must be trained by inducing weight changes indirectly through STDP (8), such that its (decoded) output $\hat{u}$ will closely match u in (15), for all $x \in X$. Since the weights within NN_a cannot be adjusted directly, they are manipulated with training signals from NN_c. Connections are made between q pairs of action/critic neurons which serve as outputs in NN_c and inputs for NN_a (Figure 2). To provide feedback signals to the critic, connections are also created between r pairs of neurons from NN_a to NN_c. The information flow through the networks is illustrated in Figure 2. Since it is not known what training signals will produce the desired results in NN_a, the critic must also be trained to match V^{π} in (16) for all $x \in X$ and $u \in U$.

Both updates (15) and (16) can be formulated as follows. Let SNN$[\cdot]$ denote an action or critic network comprised of multiple spiking neurons and STDP synapses. Then, as shown in (15) and (16), SNN$[\cdot]$ must represent a desired mapping between two vectors $\xi \in \mathbb{R}^p$ and $y \in \mathbb{R}^r$:

$$y(t_k) = g_\ell[\xi(t_k)] \longleftarrow \text{SNN}[\xi(t_k), W_\ell(t_k)] \triangleq \hat{y}(t_k). \tag{17}$$

The desired mapping $g_\ell : \mathbb{R}^p \to \mathbb{R}^r$ can be assumed known and stationary during every iteration cycle ℓ and is updated by the policy/value-iteration routine. Let $W_\ell(t_k) = \{w_{ij}(t_k)\}$ denote a matrix containing the SNN synaptic weights at time t_k. Then, for proper values of W_ℓ, if the SNN is provided

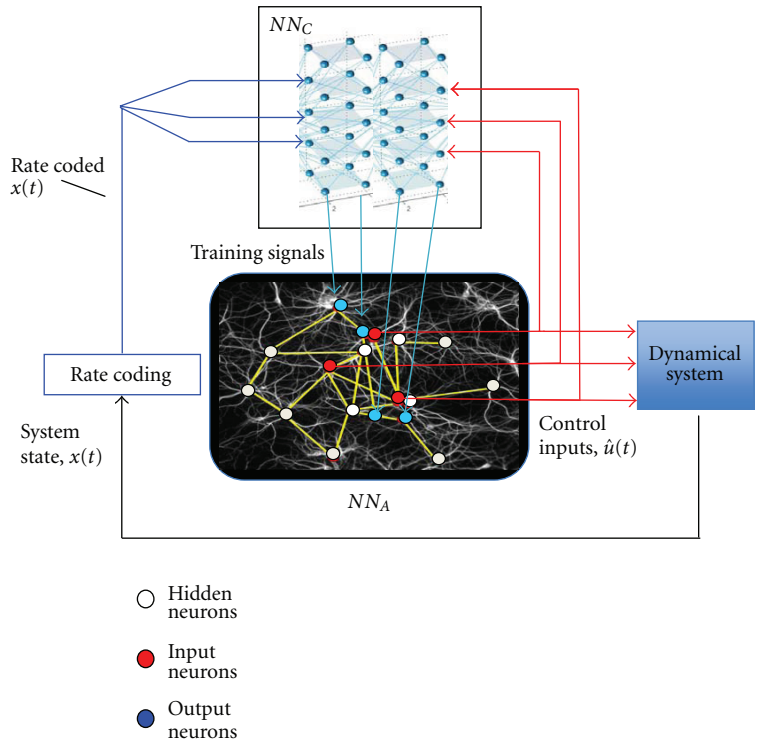


Figure 2: Adaptive critic architecture comprised of a critic network, NN_c, and, an action network, NN_a, to control a dynamical system.

with an observation of the input, ξ^l, encoded as n spike train sequences $X_i^l = \{\hat{t}_{i,\kappa} : \kappa = 1,\ldots,N_i^l\}$, $i = 1,\ldots,n$, over a time interval $[t_k, t_{k+\tau}]$ with the firing times at $\hat{t}_{i,\kappa}$, the decoded output of the SNN must match the output $y^l = g_\ell[\xi^l]$, where since $\tau \ll (t_{k+1} - t_k)$, X_i^l can be used to encode instantaneous values of ξ^l at any t_k. Then, the chosen SNN training algorithm must modify the synaptic weights such that a figure of merit representing distance is optimized at the output space of the mapping in (17).

For the critic network, NN_c, the synaptic weights are adjusted manually, using the reward-modulated Hebbian approach presented in [5, 15]. Because the target function y is known for all values of ξ, the SNN output error $(y - \hat{y})$ is also known and can be used as a feedback to the critic in the form of an imitated chemical reward, $r(t)$, that decays over time with time constant τ_c. The critic reward is modeled as

$$r(t) = \frac{[b(\hat{y}, y) + r(t - \Delta t)]e^{-(t-\hat{t}_i)}}{\tau_c}, \tag{18}$$

where for the critic $y \in \mathbb{R}$. Therefore, the error function can be defined as $b(\hat{y}, y) = \text{sgn}(y - \hat{y})$, where $\text{sgn}(\cdot)$ is the signum function. Thus, the value of $b(\cdot)$ is positive when the critic's output is too low, and it is negative when the critic's output is too high.

For the action network, NN_a, the synaptic weights cannot be manipulated directly; therefore the distance between y and the (decoded) SNN's output $\hat{y}$ is minimized with respect to the parameters of the RBF spike model described in the next section. From (17), we identify (tag) two sets of neurons referred to as input and output neurons (Figure 2), where each neuron in the set provides the response for one element of ξ and y, respectively, in the form of a spike train. It is assumed that the set of input neurons can be induced to fire on command with very high precision over a training time interval $[t_k, t_{k+\tau}]$, with $\tau < T \ll (t_{k+1} - t_k)$. The input neurons' firings could be implemented in practice by using local programming voltages with controlled pulse width and height that are easily realizable both in CMOS/memristor chips [10] and in neuronal networks grown *in vitro* [26]. In order to induce STDP in a manner that will improve the SNN representation of the mapping in (17), the programming voltages are delivered based on an optimized spiking sequence $S_i^\iota = \{\hat{t}_{i,\kappa} : \kappa = 1,\ldots,N_i^\iota\}$, $i = 1,\ldots,q$, during the ιth time interval of the training algorithm, $[t_\iota, t_{\iota+1}]$.

Existing spike models [13, 14] cannot be used to generate S_i^ι because they are stochastic and, as such, they do not allow for precise timing of pre- and postsynaptic firings. As a result, they may induce undesirable changes in the synaptic weights by virtue of the STDP rule (7), illustrated in Figure 1, which shows that a small difference in the arrival time of a pre- and postsynaptic spike can obtain the opposite effect in terms of the weight change Δw. Beside allowing for precise timing of neuron firings, the new deterministic spike model presented in the next section can be updated by the training algorithm, by optimizing a corresponding continuous signal modeled by a superposition of Gaussian radial basis functions (RBFs), which is characterized by a set of adjustable parameters $P = \{w_k, c_k, \beta_k \mid k = 1,\ldots,N\}$, where w_k is the height the RBF

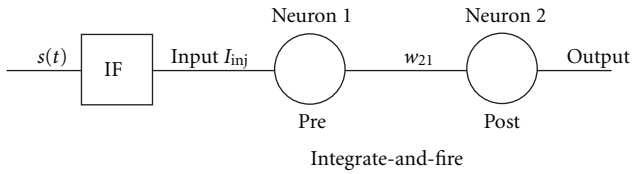


Figure 3: Model of two-node LIF spiking neural network.

pulse, c_k is the center of RBF pulse, β_k is the width of the RBF pulse, and N is the number of RBF pulses. As shown in the next section, the continuous RBF signal can be integrated against a suitable averaging function in a leaky integrator and then compared to a positive threshold, by means of a leaky integrate-and-fire (LIF) sampler. Subsequently, a precise pulse function with desired widths and intensity can be generated at a sequence of time instants, $\hat{t}_{i,\kappa}$, during the interval $[t_\iota, t_{\iota+1}]$, that correspond to the centers of the RBF signal specified by P. Thus, a set of optimal RBF parameters P^* used to generate S_i^ι can be determined by minimizing a measure of the distance between the (decoded) SNN's output $\hat{y}$ and the desired output y, computed by the policy-improvement routine (13).

In the next section, the derivation of gradient equations that can be used to minimize the SNN's output error $(y - \hat{y})$ with respect to P is illustrated by means of a two-node LIF neural network, with STDP modeled as shown in Section 2. Let $P = \{p_{lk}\}$ denote a matrix of RBF parameters to be adjusted by the training algorithm. It can be easily shown that the minimization of an error function $E(y - \hat{y})$ with respect to the elements of P when y is a known constant can be accomplished using the gradients $\partial\hat{y}/\partial p_{lk}$, or some function thereof, based on the chosen unconstrained minimization algorithm [27]. As shown in (17), for a given input ξ^l the only SNN variables to be optimized are the synaptic weights w_{ij}. From the STDP rule (7), the weights are a function of the parameters p_{lk}, which determine the input spike sequence (or training signal) to the SNN (Figure 3). By virtue of the chain rule of differentiation, it follows that

$$\frac{\partial\hat{y}}{\partial p_{lk}} = \frac{\partial\hat{y}}{\partial w_{ij}}\frac{\partial w_{ij}}{\partial p_{lk}}. \tag{19}$$

Since $\partial\hat{y}/\partial w_{ij}$ can be computed from the SNN model (Section 2), it follows that if a training algorithm can successfully modify the synaptic weights w_{ij} using the proposed spike model, it also can successfully modify $\hat{y}$, simply by redefining the error function. In other words, if a training algorithm can successfully train the synaptic weights of an SNN using the gradient $\partial w_{ij}/\partial p_{lk}$, it follows that it can also train the SNN using the gradient $\partial\hat{y}/\partial p_{lk}$, since the component $\partial\hat{y}/\partial w_{ij}$ is known and given by the SNN equations (1)–(5). Based on this property, the novel spike model and indirect training algorithm are illustrated by training the synaptic weight of a two-node LIF SNN to meet a desired value w^*, without direct manipulation.

4. Indirect Training Methodology

Consider the two-node LIF SNN schematized in Figure 3, modeled using the approach described in Section 2. $s(t)$ represents the input given to the LIF sampler, and $S(t)$ denotes the corresponding LIF sampler's output. Based on the approach presented in Section 3, the SNN synaptic strength w_{21}, representing the synaptic efficacy for a presynaptic neuron (labeled by $i = 1$) and a postsynaptic neuron (labeled by $i = 2$) cannot be modified directly by the training algorithm but can only change as a result of the STDP mechanism described in Section 2. In place of controlling w_{21}, the goal of the indirect training algorithm is to determine a spike train that can be used to stimulate the input neuron ($i = 1$) using I_{inj}, thereby inducing it to spike, such that the synaptic weight w_{21} changes from an initial (random) value to a desired value w^*.

For this purpose, we introduce a continuous spike model comprised of a superposition of Gaussian radial basis functions (RBFs):

$$s(t) = \sum_{k=1}^{N} w_k \exp\left[-\beta_k(\|t - c_k\|)^2\right], \tag{20}$$

where w_k determines the height of the kth RBF, which will decide the constant current input I_{inj}. β_k determines the width of the kth RBF and c_k determines the center of the kth RBF, where $k = 1, \ldots, N$ and N is the number of RBFs. Thus, the set of RBF adjustable parameters is $P = \{w_k, c_k, \beta_k \mid k = 1, \ldots, N\}$. A spike train can be obtained from the continuous spike model (20) by processing s using a suitable LIF sampler that outputs a square pulse function $S(t)$ with the same heights, widths, and centers, as the RBF spike model (20).

In this two-node SNN model, there is no synaptic current sent to the first neuron because there are no presynaptic neurons connected to it. Rather, the input for the first neuron is the square pulse function provided by the output of the LIF sampler. Therefore, during one square pulse, the input, $I_{inj}(t)$, to neuron $i = 1$ can be viewed as a constant, which for our case is h, the height of the RBF. For a fixed threshold and a constant input, the time it takes for the first neuron to fire (or a spike to be generated) is

$$T = -\tau_m \ln\left[1 - \frac{\theta}{hR_m}\right] \quad (hR_m > \theta), \tag{21}$$

where θ is the potential difference between the spike generating threshold, V_{th}, and the resting potential, V_0; that is, $\theta = V_{th} - V_0$. h is the height of input square pulse, R_m is the resistance of the membrane, and τ_m is the passive-membrane time constant [20]. Thus, from (21), the value of T can be controlled by adjusting h. We allow the first neuron to fire near the end of a square pulse by inputting a spike sequence generated by an RBF model with a suitable width and height. Then, the firing times of the first neuron can be easily adjusted by altering the centers of the RBF.

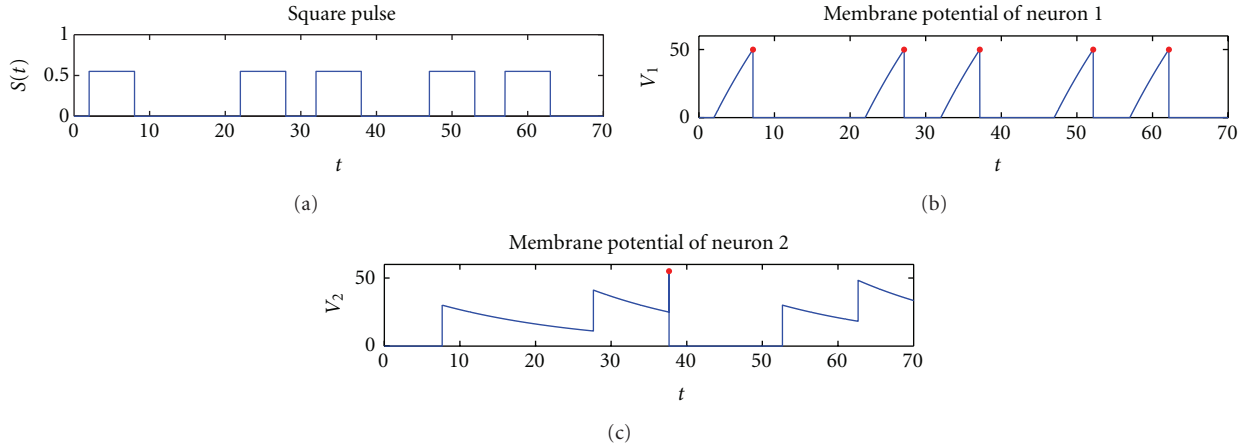


FIGURE 4: Membrane potential of the two neurons in the SNN in Figure 3.

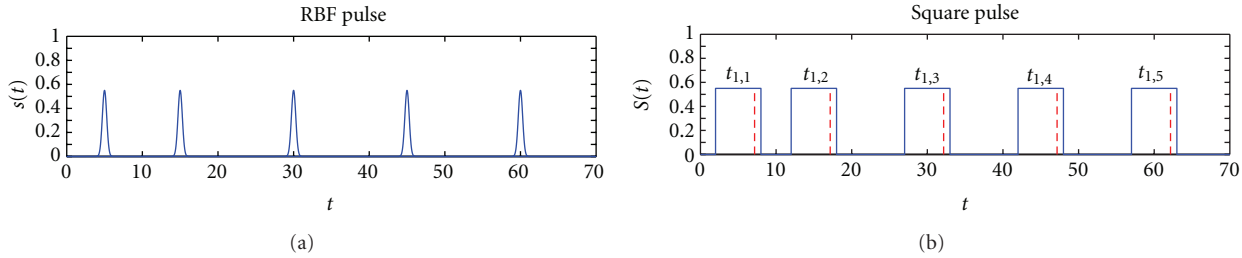


FIGURE 5: Deterministic spike model signals.

In this simple example, it is assumed that the synapse is of the excitatory type. Therefore, (4) can be written as,

$$I_s(t) = C_m a_E \sum_{k=k_0}^{k_t} \delta(t - t_{1,k} - \tau_d), \tag{22}$$

where τ_d is a known constant that represents the conduction delay of the synaptic current from neuron $i = 1$ and $t_{1,k}$ are the firing times of the first neuron. k_0 is the index of the spike of the first neuron, which occurs after the previous spike of the second neuron, and k_t is the index of the spike of the first neuron, which provokes the firing time of the second neuron. The only input to the second neuron is the synaptic current defined in (22). Therefore, substituting (22) and (2) into (1) results in the following equation for the membrane potential,

$$C_m \frac{dv(t)}{dt} = -\frac{C_m}{\tau_m}[v(t) - V_0] + C_m a_E \sum_{k=k_0}^{k_t} \delta(t - t_{1,k} - \tau_d). \tag{23}$$

Solving (23) for the membrane potential of the second neuron provides the response of neuron $i = 2$,

$$v(t) = V_0 + \sum_{k=k_0}^{k_t} a_E \exp\left[-\frac{t - t_{1,k} - \tau_d}{\tau_m}\right] H(t - t_{1,k} - \tau_d), \tag{24}$$

where $H(\cdot)$ is the Heaviside function. Whenever the membrane potential in (24) exceeds the threshold V_{th}, the second neuron fires, and the membrane potential $v(t)$ is then set instantly equal to V_0. It follows that the firing times of the second neuron $t_{2,j}$ can be written as a function of the firing times of the first neuron,

$$t_{2,j} = (t_{1,k_t} + \tau_d) H\left\{\sum_{k=k_0}^{k_t} a_E \exp\left[-\frac{t_{2,j} - t_{1,k} - \tau_d}{\tau_m}\right] \times H\left(t_{2,j} - t_{1,k} - \tau_d\right) - (V_{\text{th}} - V_0)\right\}, \tag{25}$$

where j is the index of the firing times. In addition, the membrane potentials of the first neuron and the second neuron can be calculated using (24) and (21).

An example of the membrane potential time history for the two neurons is plotted in Figure 4, where the square pulse function S used to stimulate the first neuron is plotted along with the neurons' membrane potentials v_1 and v_2. The red dots denote the firing times and potentials for the two neurons. It can be seen from Figure 4 that, with this input, the first neuron fires five times and the second neuron fires one time. In this SNN, the input to the second neuron is given only by the synaptic current caused by the firing of the first neuron. It can be seen that only when the second and third firing times of the first neuron are very close, the membrane potential of the second neuron increases over the threshold, causing it to fire, whereas the fourth and fifth

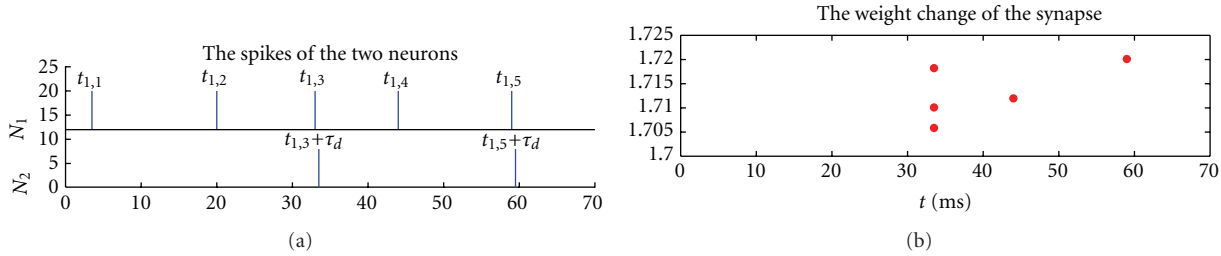


Figure 6: Optimized input spike train and indirect weight changes brought about by STDP.

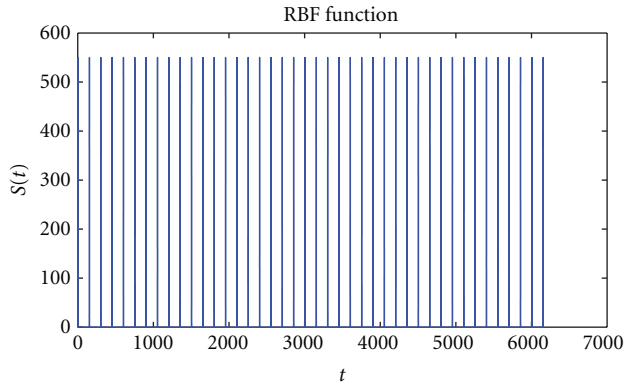


Figure 7: Optimized RBF spike model for the SNN in Figure 3.

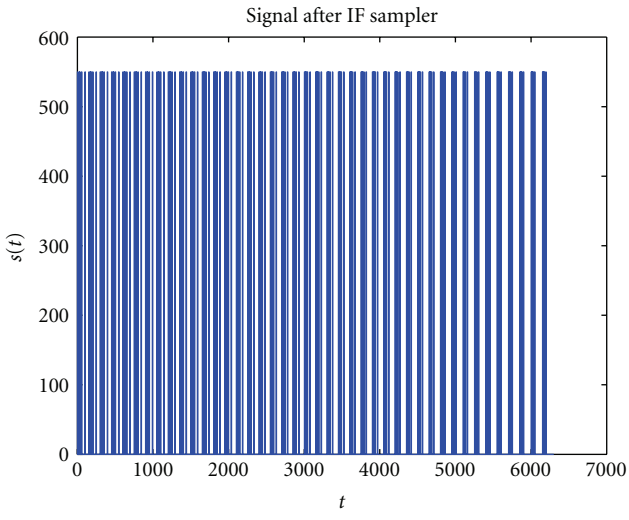


Figure 8: Square wave obtained by the LIF sampler, for the RBF spike model in Figure 7.

firing times of the first neuron are too far apart to cause firing of the second neuron.

4.1. Deterministic Spike Model. The deterministic spike model consists of a continuous RBF model in the form (20) combined with an LIF sampler that converts the RBF into a square wave function. The LIF sampler developed in [28] for the approximate reconstruction of bandlimited functions is adopted, which converts any continuous signal $f(t)$ into a square wave function by integrating it against an averaging function $u_{k,z}(t)$. The integrated result is compared to a positive threshold and a negative threshold, such that when either one of the two thresholds is reached, a pulse is generated at time $t_k = z$. The value of the integrator is then reset and the process repeats. In this paper, the averaging function $u_{k,z}(t)$ is chosen to be the exponential $e^{\alpha(t-z)}X_{[t_k,z]}$, where X_I is the characteristic function of I and $\alpha > 0$ is a constant that models the leakage of the integrator, as due to practical implementations [28]. Then, the LIF sampler firing condition that generates the square wave (or sequence of pulses) is

$$\pm\theta = \int_{t_k}^{t_{k+1}} f(t)e^{\alpha(t-t_{k+1})}dt \triangleq \langle f, u_k\rangle, \tag{26}$$

where t_k is the time instant of the sample, t_{k+1} is the next time instant of the sample, u_k is the averaging function, θ is the threshold value, and α is the leakage of the integrator. The output of the LIF sampler can be expressed in terms of the time instants at which the integral reaches the threshold, $\{t_0, \ldots, t_n\}$, and by the samples $q_1, \ldots, q_n$, defined as

$$q_j \triangleq \int_{t_j}^{t_{j-1}} f(x)e^{\alpha(x-t_j)}dx, \quad \text{for } 1 \le j \le n. \tag{27}$$

From (26), it follows that $|q_j| = \theta$. By adjusting the parameters of the LIF sampler, it is possible to convert the RBF signal in (20) to a square wave comprised of pulses with the same width, height, and centers as the RBFs in (20). Thus, by using the RBF spike model in (26) with suitable widths and heights, it is possible to induce the first neuron to fire shortly before the end of each pulse of the square-pulse function (also considering the refractory period of the neuron). For simplicity, in this example, it is assumed that the heights w_k and widths β_k are known positive constants of equal magnitudes for all k. Then, the centers of the RBF comprise the set of adjustable parameters, $P = \{c_k \mid k = 1, \ldots, N\}$, to be optimized. The same approach can be easily extended to a case where all of the RBF parameters are adapted.

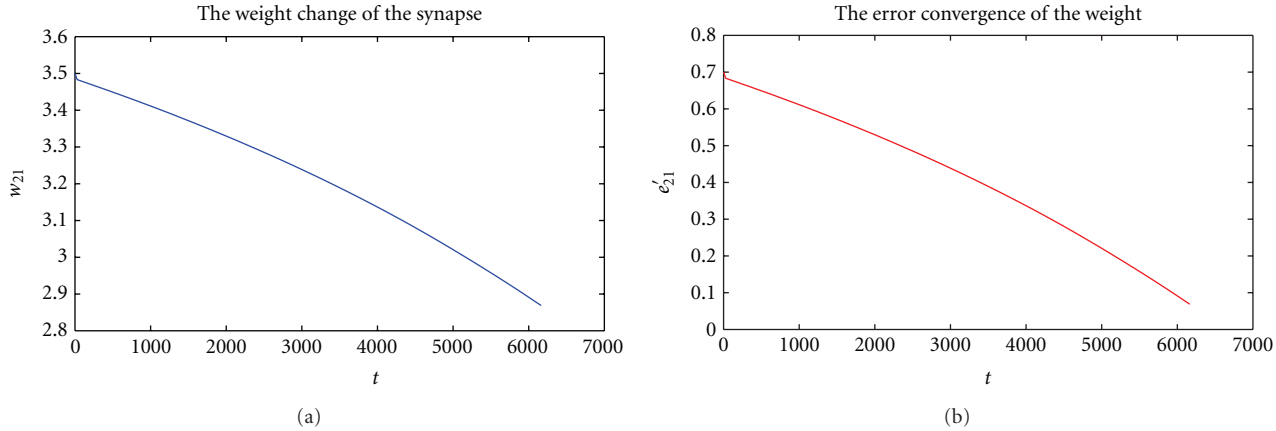


Figure 9: Indirect weight changes brought about by STDP and corresponding deviation from $w^* = 2.8$, for the SNN in Figure 3 stimulated using the input spike train in Figure 7.

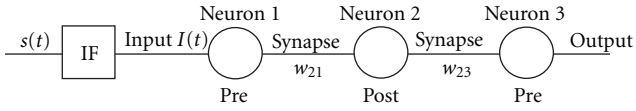


Figure 10: Model of three-node LIF spiking neural network.

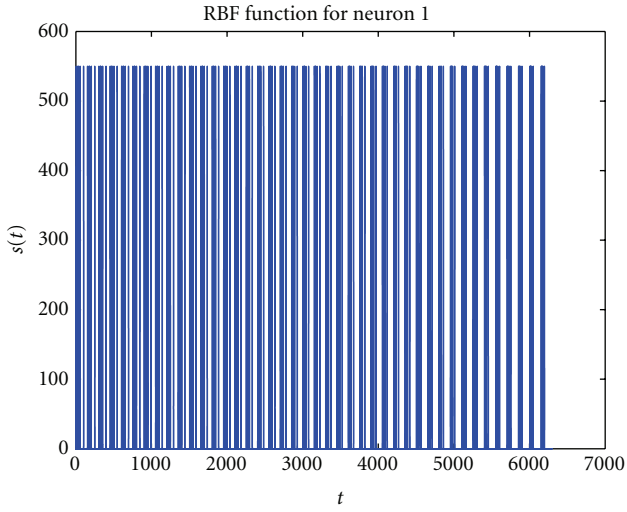


Figure 11: Optimized RBF spike model for the SNN in Figure 10.

It can be shown using the LIF SNN model in Section 2 that, under the aforementioned assumptions, the firing times of the first neuron satisfy the constraints,

$$t_{1,k} \leq c_k + \frac{\beta}{2},$$
$$t_{1,k} + \Delta^{\text{abs}} \geq c_k + \frac{\beta}{2}, \tag{28}$$

where Δ^{abs} is an absolute refractory time of the neuron. Then, the firing times of the first neuron can be written as a function of the RBF centers, c_k. By design, the first neuron fires after the same time interval, relative to each square pulse

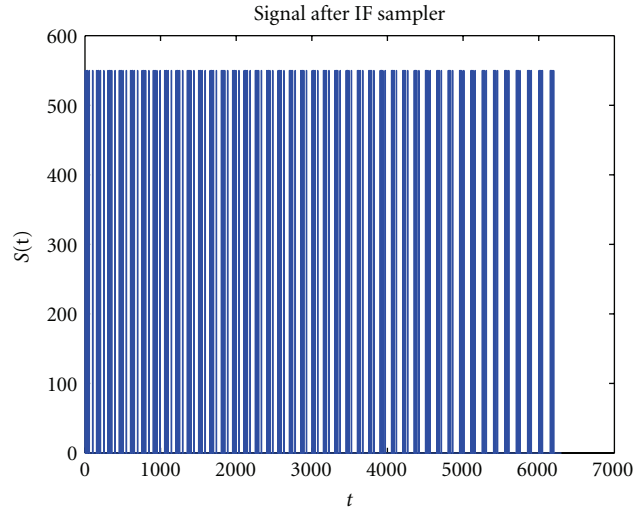


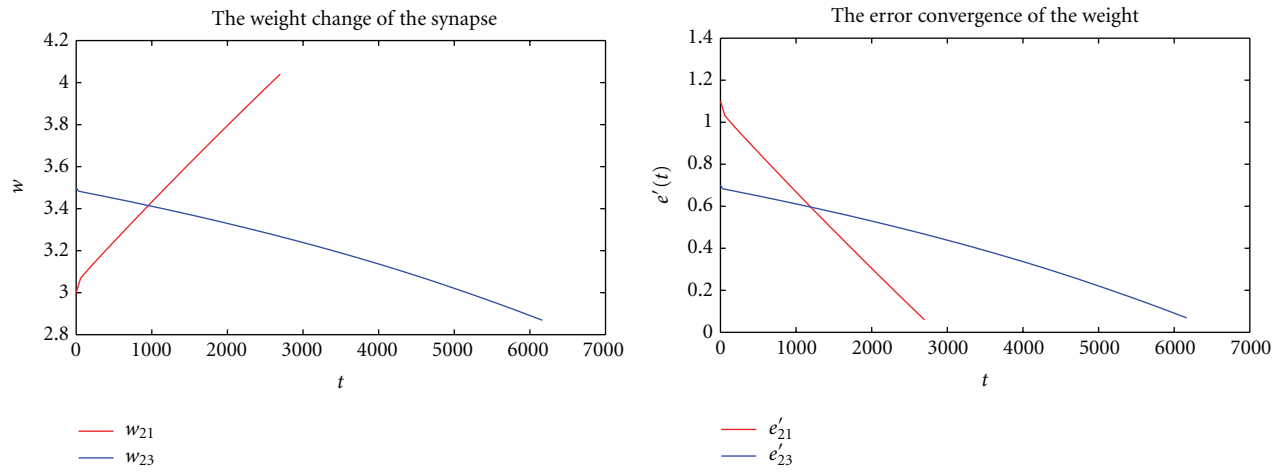
Figure 12: Square wave obtained from the LIF sampler, using the optimized RBF spike model in Figure 11.

(Figure 4), due to the chosen RBF heights and widths. Then, the firing times of the first neuron can be written as,

$$t_{1,k} = c_k - \frac{\beta}{2} + T, \tag{29}$$

where $t_{1,k}$ denotes the firing times of the first neuron, c_k are the centers of the RBF, β is the constant width of the RBF, and T is given by (21).

The RBF spike model is demonstrated in Figure 5, where an example of RBF output obtained from (20) is plotted and, after being fed to the LIF sampler, produces a corresponding square wave which can be used as controlled pulse. It can be seen that the centers of the RBFs precisely determine the times at which a pulse occurs in the square wave and that the square wave maintains the desired constant width and magnitude for all k. The next subsection illustrates how, by adapting the continuous RBF spike model in (20), it is



(a) (b)

FIGURE 13: Indirect weight changes brought about by STDP and corresponding error functions, for the SNN in Figure 10 stimulated using the input spike train in Figure 12.

possible to minimize a desired error function and train the synaptic weight of the SNN without directly manipulating it.

4.2. Indirect Training Equations. As explained in Section 2.2, it is assumed that the synaptic weight w_{21} can only be modified by controlling the activity of the SNN input neuron(s), with $i = 1$, and that it obeys the nearest-spike STDP model in (8). Over time, the synaptic weight can change repeatedly, and therefore its final value can be written as the sum of all incremental changes that have occurred over the time interval $[t_l, t_{l+1}]$,

$$w_{21}(t) = \prod_{i=1}^{M} w_0(1 + \Delta w_i), \tag{30}$$

where, for the two-neuron SNN in Figure 3, $\Delta w_1, \ldots, \Delta w_M$ are due to M pairs of pre- and postsynaptic spikes that occur at any time $t \in [t_l, t_{l+1}]$. Every weight increment Δw_i is induced via STDP and, thus, obeys equation (8). Since the firing times in (8) depend on the centers of the RBF input through (21)–(29), it follows that every weight increment Δw_i is a function of the RBF centers. In this example, the constraints (28) can be written as,

$$\frac{\beta}{2} < c_1, \qquad c_N + \frac{\beta}{2} < t_f, \qquad c_k + \beta < c_{k+1}, \quad \text{for } k = 1, \ldots, N, \tag{31}$$

where, from Figure 5, $N = 5$.

A training-error function is defined in terms of the desired synaptic weight w^* and the actual value of the synaptic weight $w_{21}(t)$. While different forms of error function may be used, the chosen form determines the complexity of the derivation of the training gradients. It was found that the most convenient form of training-error function can be derived from the ratio of the actual weight over the desired weight,

$$e(t) = \frac{w_{21}(t)}{w^*}, \tag{32}$$

where $w_{21}(t)$ is the weight value at time $t \in [t_l, t_{l+1}]$, obtained from all previous spikes. As explained in Section 3, w^* can be viewed as the weight that leads to desired output $\hat{y}$ for a given input ξ. It can be seen that when w_{21} is equal to w^*, e is equal to one. Plugging (30) into (36), the weight ratio can be rewritten as,

$$e(t) = \frac{1}{w^*} \prod_{i=1}^{M} w_0(1 + \Delta w_i), \tag{33}$$

and the error between w_{21} and w^* can be minimized by minimizing the natural logarithm of the ratio (33):

$$E(t) \triangleq \ln[e(t)] = \ln(w_0) + \ln(1 + \Delta w_1) + \cdots + \ln(1 + \Delta w_M) - \ln(w^*). \tag{34}$$

As is typical of all optimization problems, minimizing a quadratic form presents several advantages [29]. Therefore, at any time $t \in [t_l, t_{l+1}]$, the indirect training algorithm seeks to minimize the quadratic training-error function:

$$J(t) \triangleq E(t)^2 = \{\ln[e(t)]\}^2 = \{\ln(w_0) + \ln(1 + \Delta w_1) + \cdots + \ln(1 + \Delta w_M) - \ln(w^*)\}^2. \tag{35}$$

Then, indirect training can be formulated as an unconstrained optimization problem in which J is to be minimized with respect to the RBF centers, or $P = \{c_k : c_k \in [t_l, t_{l+1}]\}$. Any gradient-based numerical optimization algorithm can be utilized for this purpose. The analytical form of the gradient $\partial J/\partial c_k$ depends on the form of the spike patterns. The following subsection derives this gradient analytically for one example of spike patterns and demonstrates that, using this gradient, the synaptic weight can be trained indirectly to meet the desired value w^* exactly. The same results were derived and demonstrated numerically for all other possible

spike patterns, but they are omitted here for brevity. Another representation of error $e'(t)$ is defined below to make the convergence of error more understandable in figures,

$$e'(t) = |w_{21} - w^*|. \tag{36}$$

where, $|\cdot|$ denotes the absolute value.

4.3. Derivation of Gradient Equations for Indirect Training. Consider the case in which $M = 5$, $w^* = 1.72$, and $2a_E > V_{\text{th}} - V_0 > a_E$, which results in a two-node SNN (Figure 3) in which neuron $i = 1$ must fire at least two times in order for neuron $i = 2$ to fire. An example of such a spike pattern is shown in Figure 6, where N_1 and N_2 denote spike trains of neuron $i = 1$ and $i = 2$, respectively. In this case, the first neuron fires five times, and the second neuron fires two times. The firing time of the first neuron is controlled by the RBF spike model described in Section 4.1. In Figure 6, $t_{i,k}$ denotes the kth firing time of the ith neuron with $k \in \mathcal{I}_i$, where $\mathcal{I}_i = \{k = 1, \ldots, 5\}$ is an index set for the firing times and i is the index labeling the neurons. In this example, the second neuron fires after $t_{1,3}$ and $t_{1,5}$, because $t_{1,2}$, $t_{1,3}$ and $t_{1,4}$, $t_{1,5}$ are close enough to cause the membrane potential of the second neuron, v_2, to exceed the threshold.

Initially, the synaptic weight is equal to 1.7. The weight change is discontinuous due to the discrete property of spikes. As shown in Figure 6, the weight increases three times at $t_{1,3} + \tau_d$, decreases at $t_{1,4}$, and increases again at $t_{1,5} + \tau_d$. For this example, the synaptic weight changes in five increments:

$$\begin{aligned}
\Delta w_1 &= A_+ \exp\left(\frac{t_{1,1}}{\tau_+}\right) \exp\left(\frac{-t_{1,3}}{\tau_+}\right) \exp\left(\frac{-\tau_d}{\tau_+}\right) \\
\Delta w_2 &= A_+ \exp\left(\frac{t_{1,2}}{\tau_+}\right) \exp\left(\frac{-t_{1,3}}{\tau_+}\right) \exp\left(\frac{-\tau_d}{\tau_+}\right) \\
\Delta w_3 &= A_+ \exp\left(\frac{-\tau_d}{\tau_+}\right) \\
\Delta w_4 &= -A_- \exp\left(\frac{t_{1,3}}{\tau_-}\right) \exp\left(\frac{-t_{1,4}}{\tau_-}\right) \exp\left(\frac{\tau_d}{\tau_-}\right) \\
\Delta w_5 &= A_+ \exp\left(\frac{-\tau_d}{\tau_+}\right).
\end{aligned} \tag{37}$$

When the equations above are substituted in (35), the training-error function can be written as

$$\begin{aligned}
J = \Bigg\{ &\ln(w_0) + \ln\left[1 + A_+ \exp\left(\frac{t_{1,1}}{\tau_+}\right) \exp\left(\frac{-t_{1,3}}{\tau_+}\right) \exp\left(\frac{-\tau_d}{\tau_+}\right)\right] \\
&+ \ln\left[1 + A_+ \exp\left(\frac{t_{1,2}}{\tau_+}\right) \exp\left(\frac{-t_{1,3}}{\tau_+}\right) \exp\left(\frac{-\tau_d}{\tau_+}\right)\right] \\
&+ \ln\left[1 + A_+ \exp\left(\frac{-\tau_d}{\tau_+}\right)\right] \\
&+ \ln\left[1 - A_- \exp\left(\frac{t_{1,3}}{\tau_-}\right) \exp\left(\frac{-t_{1,4}}{\tau_-}\right) \exp\left(\frac{\tau_d}{\tau_-}\right)\right] \\
&+ \ln\left[1 + A_+ \exp\left(\frac{-\tau_d}{\tau_+}\right)\right] - \ln(w^*)\Bigg\}^2.
\end{aligned} \tag{38}$$

Then, the gradients of J with respect to the RBF centers are given by

$$\begin{aligned}
\frac{\partial J}{\partial c_1} &= \frac{\partial E^2}{\partial c_1} = \frac{\partial E^2}{\partial t_{1,1}} = 2E\left[\frac{\Delta w_1}{\tau_+(1 + \Delta w_1)}\right] \\
\frac{\partial J}{\partial c_2} &= \frac{\partial (E^2)}{\partial c_2} = \frac{\partial E^2}{\partial t_{1,2}} = 2E\left[\frac{\Delta w_2}{\tau_+(1 + \Delta w_2)}\right] \\
\frac{\partial J}{\partial c_3} &= \frac{\partial E^2}{\partial c_3} = \frac{\partial E^2}{\partial t_{1,3}} \\
&= 2E\left[\frac{-\Delta w_1}{\tau_+(1 + \Delta w_1)} + \frac{-\Delta w_2}{\tau_+(1 + \Delta w_2)} + \frac{\Delta w_4}{\tau_-(1 + \Delta w_4)}\right] \\
\frac{\partial J}{\partial c_4} &= \frac{\partial E^2}{\partial c_4} = \frac{\partial E^2}{\partial t_{1,4}} = 2E\left[\frac{-\Delta w_4}{\tau_-(1 + \Delta w_4)}\right] \\
\frac{\partial J}{\partial c_5} &= \frac{\partial E^2}{\partial c_5} = \frac{\partial E^2}{\partial t_{1,5}} = 0.
\end{aligned} \tag{39}$$

Using the above gradients, the optimal values of $c_1, \ldots, c_5$ can be obtained by minimizing J using a gradient-based numerical algorithm such as Newton's method.

An example of indirect learning algorithm implementation is shown in Figure 6, where the RBF-adjustable parameters (which, in this case, coincide with the firing times of neuron $i = 1$) are optimized to induce a change in the synaptic weight w_{21} from an initial value of 1.704, to a desired value $w^* = 1.72$. Another example, for which the gradient equations are omitted for brevity, is shown in Figures 7–9. In this case, the desired weight value $w^* = 2.8$ is far from the initial weight $w_{21}(t_t) = 3.5$, and thus $N = 43$ spikes are required to train the SNN. The optimal RBF input and corresponding controlled pulse used to stimulate neuron $i = 1$ are shown in Figures 7 and 8, respectively. The time history of the weight w_{21} is plotted in Figure 9(a), along with the corresponding deviation from w^* plotted in Figure 9(b).

5. Generalized Form of Gradient Equations for Indirect Training

A more general form of the gradient equations can be found by rewriting the response of the membrane potential for neuron $i = 2$ in (24), as follows,

$$v(t) = V_0 + \sum_{k=a}^{b} a_E \exp\left[-\frac{t - t_{1,k} - \tau_d}{\tau_m}\right] H(t - t_{1,k} - \tau_d), \tag{40}$$

where $t_{1,k}$ denotes the kth firing time of neuron $i = 1$. Then, the gradients of the training objective function (35) with respect to the centers of the RBF spike model for $M = 5$ are given by the equations in the Appendix, which are obtained in terms of the two functions,

$$\begin{aligned}
g_+\left(t_{1,i}, t_{1,j}\right) &= A_+ \exp\left(\frac{t_{1,i}}{\tau_+}\right) \exp\left(\frac{-t_{1,j}}{\tau_+}\right) \exp\left(\frac{-\tau_d}{\tau_+}\right), \\
g_-\left(t_{1,i}, t_{1,j}\right) &= -A_- \exp\left(\frac{t_{1,i}}{\tau_-}\right) \exp\left(\frac{-t_{1,j}}{\tau_-}\right) \exp\left(\frac{\tau_d}{\tau_-}\right),
\end{aligned} \tag{41}$$

where $t_{1,i}$, $t_{1,j}$ are two distinct firing times of neuron $i = 1$.

The methodology presented in Section 4 can also be extended to larger SNNs, although in this case it may be more convenient to compute the gradients of the objective function numerically. As an example, consider the three-neuron SNN in Figure 10, modeled by the approach in Section 2 and with two synaptic weights w_{21} and w_{23} that each obey the STDP mechanism in (8). Suppose that the desired values of the synaptic weights are $w_{21}^* = 4.1$ and $w_{23}^* = 2.8$. Using the indirect learning method presented in this paper, and the gradient provided in the Appendix, a training-error function formulated in terms of the deviations of $w_{21}(t)$ and $w_{23}(t)$ from w_{21}^* and w_{23}^*, respectively, can be minimized with respect to the parameters (centers) P of the RBF spike model (20).

Once the optimal RBF spike model, plotted in Figure 11, is fed to the LIF sampler, the controlled pulse plotted in Figure 12 is obtained and implemented via I_{inj}. The controlled pulse is thus used to stimulate neuron $i = 1$ at precise instants in time that corresponds to centers of the optimal RBF spike model. By this approach, $w_{21}(t)$ can be made to converge to the desired value w_{21}^*. The same method is implemented to stimulate neuron $i = 2$ to make the value of $w_{23}(t)$ converge to w_{23}^*. The time histories of the weight values $w_{21}(t)$ and $w_{23}(t)$, obtained by the indirect training algorithm are plotted in Figure 13(a). As is also shown by the corresponding training errors, plotted in Figure 13(b), the SNN weights over time converge to the desired values, that is $w_{21}^* = 4.1$ and $w_{23}^* = 2.8$. These results demonstrate that, even for larger SNNs, the indirect training method presented in this paper is capable of modifying synaptic weights until they meet their desired values, without direct manipulation. Since this indirect training algorithm only relies on modulating the activity of the input neurons via controlled input spike trains, it also has the potential of being realizable *in vitro* and *in silico* to train biological neuronal networks and CMOS/memristor nanoscale chips, respectively.

6. Summary and Conclusion

Recently, several algorithms have been proposed for training spiking neural networks through biologically plausible learning mechanisms, such as spike-timing-dependent synaptic plasticity. These algorithms, however, rely on being able to modify the synaptic weights directly or STDP. In other words, they minimize a desired objective function with respect to weight increments that are assumed to be controllable and, thus, are decided by a weight update rule, and then are implemented directly by the training algorithm. In several potential applications of spiking neural networks, synaptic weights cannot be manipulated directly but change over time by virtue of pre- and postsynaptic neural activity. In these applications, the activity of selected input neurons can be controlled via programming voltages or pulses of blue light that induce precise spiking of the input neurons, at precise moments in time.

This paper presents an indirect learning method that induces changes in the synaptic weights by modulating spike-timing-dependent plasticity using controlled input spike trains, in lieu of the weight increments. The key difficulty to be overcome is that indirect learning seeks to adapt a pulse signal, such as a square wave or Rademacher function, in place of continuous-valued weights. Pulse signals that can be delivered to stimulate input neurons and cause them to spike, such as blue light patterns or programming voltages, are represented by piece-wise continuous, multi-valued (or many-to-one), and nondifferentiable functions that are not well suited to numerical optimization. Furthermore, stimulation patterns typically are generated by spike models that are stochastic, such as the Poisson spike model. Therefore, even when the spike model is optimized, it does not allow for precise timing of pre- and post-synaptic firings, and as a result, may induce undesirable changes in the synaptic weights. This paper presents a deterministic and adaptive spike model derived from radial basis functions and a leaky integrate-and-fire sampler. This spike model can be easily optimized to determine the sequence of spike timings that minimizes a desired objective function and, then, used to stimulate input neurons. The results demonstrate that this methodology is capable of inducing the desired synaptic plasticity in the network and modify synaptic weights to meet their desired values only by virtue of controlled input spike trains that are realizable both in biological neuronal networks and in CMOS/memristor nanoscale chips.

Appendix

The gradients of the training objective function can be derived as follows. The membrane potential of neuron 2 is denoted by $v_{i.j}(t)$, where the membrane potential can only be affected by presynaptic spikes that occur within the time interval $[t_{1.i}, t_{1,j}]$.

$$\frac{\partial E^2}{\partial c_1} = \begin{cases} 2E\left[\dfrac{g_+(t_{1,1}, t_{1,3})}{\tau_+(1 + g_+(t_{1,1}, t_{1,3}))}\right] & \text{if } v(t_{1,3} + \tau_d)_{1,3} > V_{\text{th}} \\ 2E\left[\dfrac{g_+(t_{1,1}, t_{1,2})}{\tau_+(1 + g_+(t_{1,1}, t_{1,2}))}\right] & \text{if } v(t_{1,2} + \tau_d)_{1,2} > V_{\text{th}} \\ 2E\left[\dfrac{g_+(t_{1,1}, t_{1,4})}{\tau_+(1 + g_+(t_{1,1}, t_{1,4}))}\right] & \text{if } v(t_{1,4} + \tau_d)_{1,4} > V_{\text{th}} \\ 2E\left[\dfrac{g_+(t_{1,1}, t_{1,5})}{\tau_+(1 + g_+(t_{1,1}, t_{1,5}))}\right] & \text{if } v(t_{1,5} + \tau_d)_{1,5} > V_{\text{th}}. \end{cases}$$

$$\frac{\partial E^2}{\partial c_2} = \begin{cases} 2E\left[\dfrac{g_+(t_{1,2},t_{1,3})}{\tau_+(1+g_+(t_{1,2},t_{1,3}))}\right] & \text{if } v(t_{1,3}+\tau_d)_{1,3} > V_{\text{th}} \\ 2E\left[\dfrac{-g_+(t_{1,1},t_{1,2})}{\tau_+(1+g_+(t_{1,1},t_{1,2}))}\right] & \text{if } v(t_{1,2}+\tau_d)_{1,2} > V_{\text{th}},\ v(t_{1,5}+\tau_d)_{3,5} > V_{\text{th}} \\ & c_3 > \dfrac{c_5+c_2+2\tau_d}{2} \\ 2E\left[\dfrac{-g_+(t_{1,1},t_{1,2})}{\tau_+(1+g_+(t_{1,1},t_{1,2}))} + \dfrac{g_-(t_{1,2},t_{1,3})}{\tau_-(1+g_-(t_{1,2},t_{1,3}))}\right] & \text{if } v(t_{1,2}+\tau_d)_{1,2} > V_{\text{th}},\ v(t_{1,4}+\tau_d)_{3,4} > V_{\text{th}}, \\ & c_3 < \dfrac{c_4+c_2+2\tau_d}{2} \text{ or } v(t_{1,2}+\tau_d)_{1,2} > V_{\text{th}}, \\ & v(t_{1,5}+\tau_d)_{3,5} > V_{\text{th}},\ c_3 < \dfrac{c_5+c_2+2\tau_d}{2} \\ 2E\left[\dfrac{-g_+(t_{1,1},t_{1,2})}{\tau_+(1+g_+(t_{1,1},t_{1,2}))} + \dfrac{g_-(t_{1,2},t_{1,3})}{\tau_-(1+g_-(t_{1,2},t_{1,3}))}\right. & \\ \left. + \dfrac{g_-(t_{1,2},t_{1,4})}{\tau_-(1+g_-(t_{1,2},t_{1,4}))} + \dfrac{g_-(t_{1,2},t_{1,5})}{\tau_-(1+g_-(t_{1,2},t_{1,5}))}\right] & \text{if } v(t_{1,2}+\tau_d)_{1,2} > V_{\text{th}},\ v(t_{1,5}+\tau_d)_{3,5} < V_{\text{th}} \\ 2E\left[\dfrac{g_+(t_{1,2},t_{1,4})}{\tau_+(1+g_+(t_{1,2},t_{1,4}))}\right] & \text{if } v(t_{1,4}+\tau_d)_{1,4} > V_{\text{th}}. \end{cases}$$

$$\frac{\partial E^2}{\partial c_3} = \begin{cases} 2E\left[\dfrac{-g_+(t_{1,1},t_{1,3})}{\tau_+(1+g_+(t_{1,1},t_{1,3}))} + \dfrac{-g_+(t_{1,2},t_{1,3})}{\tau_+(1+g_+(t_{1,2},t_{1,3}\ ))}\right. & \\ \left. + \dfrac{g_-(t_{1,3},t_{1,4})}{\tau_-(1+g_-(t_{1,3},t_{1,4}\ ))}\right] & \text{if } v(t_{1,3}+\tau_d)_{1,3} > V_{\text{th}}, \\ & v(t_{1,5}+\tau_d)_{4,5} > V_{\text{th}},\ c_3 > 2c_4 - c_5 - 2\tau_d \\ 2E\left[\dfrac{-g_+(t_{1,1},t_{1,3})}{\tau_+(1+g_+(t_{1,1},t_{1,3}))} + \dfrac{-g_+(t_{1,2},t_{1,3})}{\tau_+(1+g_+(t_{1,2},t_{1,3}))}\right] & \text{if } v(t_{1,3}+\tau_d)_{1,3} > V_{\text{th}},\ v(t_{1,5}+\tau_d)_{4,5} > V_{\text{th}}, \\ & c_3 < 2c_4 - c_5 - 2\tau_d \\ 2E\left[\dfrac{g_+(t_{1,3},t_{1,5})}{\tau_+(1+g_+(t_{1,3},t_{1,5}))}\right] & \text{if } v(t_{1,2}+\tau_d)_{1,2} > V_{\text{th}},\ v(t_{1,5}+\tau_d)_{3,5} > V_{\text{th}}, \\ & c_3 > \dfrac{c_5+c_2+2\tau_d}{2} \quad \text{or} \\ & v(t_{1,5}+\tau_d)_{1,5} > V_{\text{th}} \\ 2E\left[\dfrac{-g_-(t_{1,2},t_{1,3})}{\tau_-(1+g_-(t_{1,2},t_{1,3}))}\right] & \text{if } v(t_{1,2}+\tau_d)_{1,2} > V_{\text{th}},\ v(t_{1,5}+\tau_d)_{3,5} > V_{\text{th}}, \\ & c_3 < \dfrac{c_5+c_2+2\tau_d}{2} \quad \text{or} \\ & v(t_{1,2}+\tau_d)_{1,2} > V_{\text{th}},\ v(t_{1,4}+\tau_d)_{3,4} > V_{\text{th}}, \\ & c_3 < \dfrac{c_4+c_2+2\tau_d}{2} \text{ or} \\ & v(t_{1,2}+\tau_d)_{1,2} > V_{\text{th}},\ v(t_{1,5}+\tau_d)_{3,5} < V_{\text{th}} \\ 2E\left[\dfrac{g_+(t_{1,3},t_{1,4})}{\tau_+(1+g_+(t_{1,3},t_{1,4}))}\right] & \text{if } v(t_{1,2}+\tau_d)_{1,2} > V_{\text{th}},\ v(t_{1,4}+\tau_d)_{3,4} > V_{\text{th}}, \\ & c_3 > \dfrac{c_4+c_2+2\tau_d}{2} \text{ or} \\ & v(t_{1,4}+\tau_d)_{1,4} > V_{\text{th}} \\ 2E\left[\dfrac{-g_+(t_{1,1},t_{1,3})}{\tau_+(1+g_+(t_{1,1},t_{1,3}))} + \dfrac{-g_+(t_{1,2},t_{1,3})}{\tau_+(1+g_+(t_{1,2},t_{1,3}))}\right. & \\ \left. + \dfrac{g_-(t_{1,3},t_{1,4})}{\tau_-(1+g_-(t_{1,3},t_{1,4}))} + \dfrac{g_-(t_{1,3},t_{1,5})}{\tau_-(1+g_-(t_{1,3},t_{1,5}))}\right] & \text{if } v(t_{1,3}+\tau_d)_{1,3} > V_{\text{th}},\ v(t_{1,5}+\tau_d)_{4,5} < V_{\text{th}}. \end{cases}$$

$$\frac{\partial E^2}{\partial c_4} = \begin{cases} 2E\left[\dfrac{-g_-(t_{1,3},t_{1,4})}{\tau_-(1+g_-(t_{1,3},t_{1,4}))}\right] & \text{if } \nu(t_{1,3}+\tau_d)_{1,3} > V_{\text{th}},\ \nu(t_{1,5}+\tau_d)_{4,5} > V_{\text{th}}, \\ & c_4 < \dfrac{c_3+c_5+2\tau_d}{2} \text{ or} \\ & \nu(t_{1,3}+\tau_d)_{1,3} > V_{\text{th}},\ \nu(t_{1,5}+\tau_d)_{4,5} < V_{\text{th}} \\ 2E\left[\dfrac{g_+(t_{1,4},t_{1,5})}{\tau_+(1+g_+(t_{1,4},t_{1,5}))}\right] & \text{if } \nu(t_{1,3}+\tau_d)_{1,3} > V_{\text{th}},\ \nu(t_{1,5}+\tau_d)_{4,5} > V_{\text{th}}, \\ & c_4 > \dfrac{c_3+c_5+2\tau_d}{2} \text{ or} \\ & \nu(t_{1,2}+\tau_d)_{1,2} > V_{\text{th}},\ \nu(t_{1,5}+\tau_d)_{3,5} > V_{\text{th}}, \\ & c_3 > \dfrac{c_2+c_5+2\tau_d}{2} \text{ or} \\ & \nu(t_{1,2}+\tau_d)_{1,2} > V_{\text{th}},\ \nu(t_{1,5}+\tau_d)_{3,5} > V_{\text{th}}, \\ & c_3 < \dfrac{c_2+c_5+2\tau_d}{2} \\ 2E\left[\dfrac{-g_+(t_{1,3},t_{1,4})}{\tau_+(1+g_+(t_{1,3},t_{1,4}))} + \dfrac{g_-(t_{1,4},t_{1,5})}{\tau_-(1+g_-(t_{1,4},t_{1,5}))}\right] & \text{if } \nu(t_{1,2}+\tau_d)_{1,2} > V_{\text{th}}, \\ & \nu(t_{1,4}+\tau_d)_{3,4} > V_{\text{th}},\ c_4 < 2c_3 - c_2 - 2\tau_d \\ 2E\left[\dfrac{g_-(t_{1,4},t_{1,5})}{\tau_-(1+g_-(t_{1,4},t_{1,5}))}\right] & \text{if } \nu(t_{1,2}+\tau_d)_{1,2} > V_{\text{th}},\ \nu(t_{1,4}+\tau_d)_{3,4} > V_{\text{th}}, \\ & c_4 > 2c_3 - c_2 - 2\tau_d \\ 2E\left[\dfrac{-g_-(t_{1,2},t_{1,4})}{\tau_-(1+g_-(t_{1,2},t_{1,4}))}\right] & \text{if } \nu(t_{1,2}+\tau_d)_{1,2} > V_{\text{th}},\ \nu(t_{1,5}+\tau_d)_{3,5} < V_{\text{th}} \\ 2E\left[\dfrac{-g_+(t_{1,1},t_{1,4})}{\tau_+(1+g_+(t_{1,1},t_{1,4}))} + \dfrac{-g_+(t_{1,2},t_{1,4})}{\tau_+(1+g_+(t_{1,2},t_{1,4}))}\right. & \\ \quad \left. + \dfrac{-g_+(t_{1,3},t_{1,4})}{\tau_+(1+g_+(t_{1,3},t_{1,4}))} + \dfrac{g_-(t_{1,4},t_{1,5})}{\tau_-(1+g_-(t_{1,4},t_{1,5}))}\right] & \text{if } \nu(t_{1,4}+\tau_d,1,4) > V_{\text{th}} \\ 2E\left[\dfrac{g_+(t_{1,4},t_{1,5})}{\tau_+(1+g_+(t_{1,4},t_{1,5}))}\right] & \text{if } \nu(t_{1,5}+\tau_d,1,5) > V_{\text{th}}. \end{cases}$$

$$\frac{\partial E^2}{\partial c_5} = \begin{cases} 0 & \text{if } \nu(t_{1,3}+\tau_d)_{1,3} > V_{\text{th}}, \\ & \nu(t_{1,5}+\tau_d)_{4,5} > V_{\text{th}},\ c_5 > 2c_4 - c_3 - 2\tau_d \\ 2E\left[\dfrac{-g_+(t_{1,4},t_{1,5})}{\tau_+(1+g_+(t_{1,4},t_{1,5}))}\right] & \text{if } \nu(t_{1,3}+\tau_d)_{1,3} > V_{\text{th}}, \\ & \nu(t_{1,5}+\tau_d)_{4,5} > V_{\text{th}},\ c_5 < 2c_4 - c_3 - 2\tau_d \\ 2E\left[\dfrac{-g_+(t_{1,3},t_{1,5})}{\tau_+(1+g_+(t_{1,3},t_{1,5}))} + \dfrac{-g_+(t_{1,4},t_{1,5})}{\tau_+(1+g_+(t_{1,4},t_{1,5}))}\right] & \text{if } \nu(t_{1,2}+\tau_d)_{1,2} > V_{\text{th}}, \\ & \nu(t_{1,5}+\tau_d)_{3,5} > V_{\text{th}},\ c_5 < 2c_3 - c_2 - 2\tau_d \\ 2E\left[\dfrac{-g_+(t_{1,4},t_{1,5})}{\tau_+(1+g_+(t_{1,4},t_{1,5}))}\right] & \text{if } \nu(t_{1,2}+\tau_d)_{1,2} > V_{\text{th}}, \\ & \nu(t_{1,5}+\tau_d)_{3,5} > V_{\text{th}}, c_5 > 2c_3 - c_2 - 2\tau_d \\ 2E\left[\dfrac{-g_-(t_{1,4},t_{1,5})}{\tau_-(1+g_-(t_{1,4},t_{1,5}))}\right] & \text{if } \nu(t_{1,2}+\tau_d)_{1,2} > V_{\text{th}},\ \nu(t_{1,4}+\tau_d)_{3,4} > V_{\text{th}} \\ 2E\left[\dfrac{-g_-(t_{1,2},t_{1,5})}{\tau_-(1+g_-(t_{1,2},t_{1,5}))}\right] & \text{if } \nu(t_{1,2}+\tau_d)_{1,2} > V_{\text{th}},\ \nu(t_{1,5}+\tau_d)_{3,5} < V_{\text{th}} \\ 2E\left[\dfrac{-g_-(t_{1,3},t_{1,5})}{\tau_-(1+g_-(t_{1,3},t_{1,5}))}\right] & \text{if } \nu(t_{1,3}+\tau_d)_{1,3} > V_{\text{th}},\ \nu(t_{1,5}+\tau_d)_{4,5} < V_{\text{th}} \\ 2E\left[\dfrac{-g_-(t_{1,4},t_{1,5})}{\tau_-(1+g_-(t_{1,4},t_{1,5}))}\right] & \text{if } \nu(t_{1,4}+\tau_d)_{1,4} > V_{\text{th}} \\ 2E\left[\dfrac{-g_+(t_{1,1},t_{1,5})}{\tau_+(1+g_+(t_{1,1},t_{1,5}))} + \dfrac{-g_+(t_{1,2},t_{1,5})}{\tau_+(1+g_+(t_{1,2},t_{1,5}))}\right. & \\ \quad \left. + \dfrac{-g_+(t_{1,3},t_{1,5})}{\tau_+(1+g_+(t_{1,3},t_{1,5}))} + \dfrac{-g_+(t_{1,4},t_{1,5})}{\tau_+(1+g_+(t_{1,4},t_{1,5}))}\right] & \text{if } \nu(t_{1,5}+\tau_d)_{1,5} > V_{\text{th}}. \end{cases}$$

(A.1)

Acknowledgment

This work was supported by the National Science Foundation, under ECCS Grant 0925407.

References

[1] J. J. B. Jack, D. Nobel, and R. Tsien, *Electric Current Flow in Excitable Cells*, Oxford University Press, Oxford, UK, 1st edition, 1975.

[2] A. L. Hodgkin and A. F. Huxley, "A quantitative description of membrane current and its application to conduction and excitation in nerve," *The Journal of Physiology*, vol. 117, no. 4, pp. 500–544, 1952.

[3] W. Maass, "Noisy spiking neurons with temporal coding have more computational power than sigmoidal neurons," *Advances in Neural Information Processing Systems*, vol. 9, pp. 211–217, 1997.

[4] C. M. A. Pennartz, "Reinforcement learning by Hebbian synapses with adaptive thresholds," *Neuroscience*, vol. 81, no. 2, pp. 303–319, 1997.

[5] S. Ferrari, B. Mehta, G. Di Muro, A. M. J. VanDongen, and C. Henriquez, "Biologically realizable reward-modulated hebbian training for spiking neural networks," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '08)*, pp. 1780–1786, Hong Kong, June 2008.

[6] R. Legenstein, C. Naeger, and W. Maass, "What can a neuron learn with spike-timing-dependent plasticity?" *Neural Computation*, vol. 17, no. 11, pp. 2337–2382, 2005.

[7] J. P. Pfister, T. Toyoizumi, D. Barber, and W. Gerstner, "Optimal spike-timing-dependent plasticity for precise action potential firing in supervised learning," *Neural Computation*, vol. 18, no. 6, pp. 1318–1348, 2006.

[8] R. V. Florian, "Reinforcement learning through modulation of spike-timing-dependent synaptic plasticity," *Neural Computation*, vol. 19, no. 6, pp. 1468–1502, 2007.

[9] S. G. Wysoski, L. Benuskova, and N. Kasabov, "Adaptive learning procedure for a network of spiking neurons and visual pattern recognition," in *Proceedings of the 8th International Conference on Advanced Concepts for Intelligent Vision Systems (ACIVS '06)*, vol. 4179 of *Lecture Notes in Computer Science*, pp. 1133–1142, Antwerp, Belgium, September 2006.

[10] S. H. Jo, T. Chang, I. Ebong, B. B. Bhadviya, P. Mazumder, and W. Lu, "Nanoscale memristor device as synapse in neuromorphic systems," *Nano Letters*, vol. 10, no. 4, pp. 1297–1301, 2010.

[11] A. M. VanDongen, "Vandongen laboratory," http://www.vandongen-lab.com/.

[12] T. J. Van De Ven, H. M. A. VanDongen, and A. M. J. VanDongen, "The nonkinase phorbol ester receptor α1-chimerin binds the NMDA receptor NR2A subunit and regulates dendritic spine density," *Journal of Neuroscience*, vol. 25, no. 41, pp. 9488–9496, 2005.

[13] P. Dayan and L. F. Abbott, *Theoretical Neuroscience: Computational and Mathematical Modeling of Neural Systems*, MIT Press, Cambridge, Mass, USA, 2001.

[14] W. Gerstner and W. Kistler, *Spiking Neuron Models: Single Neurons, Populations, Plasticity*, Cambridge University Press, Cambridge, UK, 2006.

[15] G. Foderaro, C. Henriquez, and S. Ferrari, "Indirect training of a spiking neural network for flight control via spike-timing-dependent synaptic plasticity," in *Proceedings of the 49th IEEE Conference on Decision and Control (CDC '10)*, pp. 911–917, Atlanta, Ga, USA, December 2010.

[16] A. Aldroubi and K. Gröchenig, "Nonuniform sampling and reconstruction in shift-invariant spaces," *SIAM Review*, vol. 43, no. 4, pp. 585–620, 2001.

[17] A. A. Lazar and L. T. Toth, "A toeplitz formulation of a real-time algorithm for time decoding machines," in *Proceedings of the Telecommunication Systems, Modeling and Analysis Conference*, 2003.

[18] E. M. Izhikevich, "Which model to use for cortical spiking neurons?" *IEEE Transactions on Neural Networks*, vol. 15, no. 5, pp. 1063–1070, 2004.

[19] E. M. Izhikevich, "Simple model of spiking neurons," *IEEE Transactions on Neural Networks*, vol. 14, no. 6, pp. 1569–1572, 2003.

[20] A. N. Burkitt, "A review of the integrate-and-fire neuron model: I. Homogeneous synaptic input," *Biological Cybernetics*, vol. 95, no. 1, pp. 1–19, 2006.

[21] S. Song, K. D. Miller, and L. F. Abbott, "Competitive Hebbian learning through spike-timing-dependent synaptic plasticity," *Nature Neuroscience*, vol. 3, no. 9, pp. 919–926, 2000.

[22] P. Sjostrom, G. Turrigiano, and S. Nelson, "Rate, timing, and cooperativity jointly determine cortical synaptic plasticity," *Neuron*, vol. 32, no. 6, pp. 1149–1164, 2001.

[23] S. Ferrari and R. Stengel, "Model-based adaptive critic designs," in *Learning and Approximate Dynamic Programming*, J. Si, A. Barto, and W. Powell, Eds., John Wiley & Sons, 2004.

[24] R. E. Bellman, *Dynamic Programming*, Princeton University Press, Princeton, NJ, USA, 1957.

[25] R. Howard, *Dynamic Programming and Markov Processes*, MIT Press, Cambridge, Mass, USA, 1960.

[26] A. M. J. VanDongen, J. Codina, J. Olate et al., "Newly identified brain potassium channels gated by the guanine nucleotide binding protein G(o)," *Science*, vol. 242, no. 4884, pp. 1433–1437, 1988.

[27] J. E. Dennis and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, SIAM, Englewood Cliffs, NJ, USA, 1996.

[28] H. G. Feichtinger, J. C. Príncipe, J. L. Romero, A. Singh Alvarado, and G. A. Velasco, "Approximate reconstruction of bandlimited functions for the integrate and fire sampler," *Advances in Computational Mathematics*, vol. 36, no. 1, pp. 67–78, 2012.

[29] R. F. Stengel, *Optimal Control and Estimation*, Dover Publications, Inc., 1986.

Hindawi Publishing Corporation
Advances in Artificial Neural Systems
Volume 2012, Article ID 808602, 10 pages
doi:10.1155/2012/808602

Research Article

Evaluation of a Nonrigid Motion Compensation Technique Based on Spatiotemporal Features for Small Lesion Detection in Breast MRI

F. Steinbruecker,[1] A. Meyer-Baese,[2] T. Schlossbauer,[3] and D. Cremers[1]

[1] *Department of Computer Science, Technical University of Munich, 85748 Garching, Germany*
[2] *Department of Scientific Computing, Florida State University, Tallahassee, FL 32306, USA*
[3] *Institute for Clinical Radiology, University of Munich, 81679 Munich, Germany*

Correspondence should be addressed to A. Meyer-Baese, ameyerbaese@fsu.edu

Received 17 March 2012; Accepted 21 May 2012

Academic Editor: Olivier Bastien

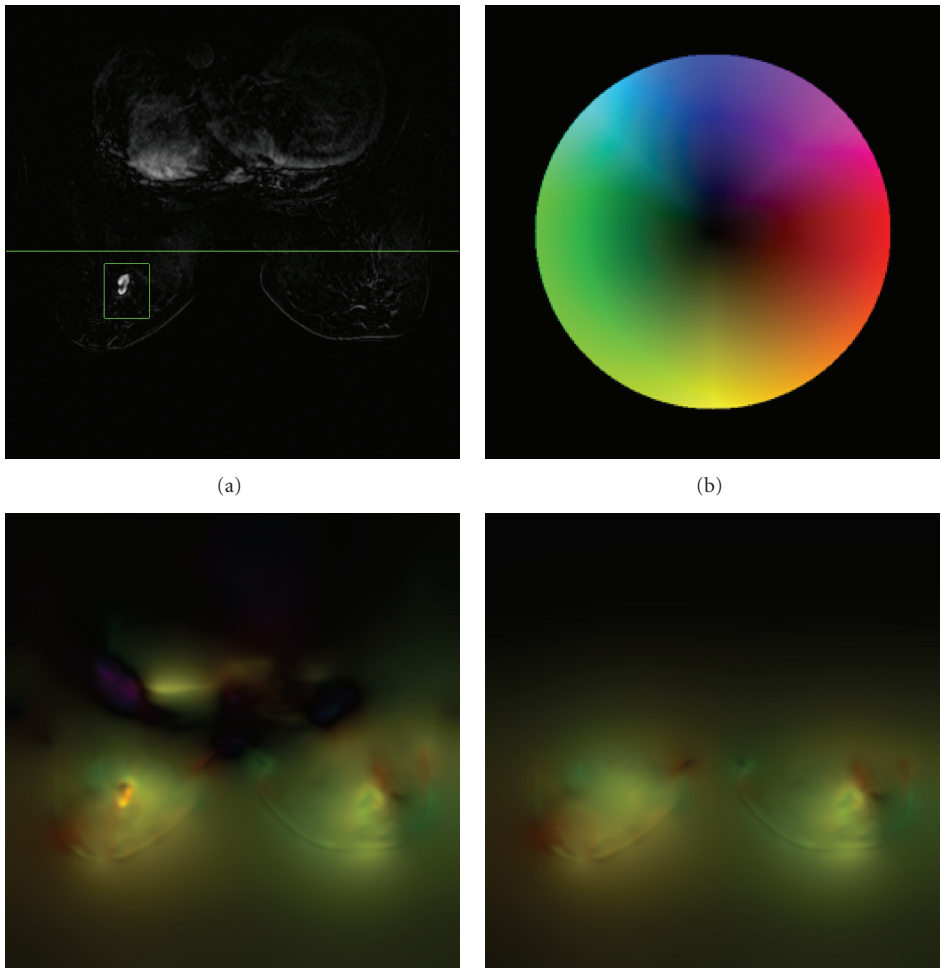
Motion-induced artifacts represent a major problem in detection and diagnosis of breast cancer in dynamic contrast-enhanced magnetic resonance imaging. The goal of this paper is to evaluate the performance of a new nonrigid motion correction algorithm based on the optical flow method. For each of the small lesions, we extracted morphological and dynamical features describing both global and local shape, and kinetics behavior. In this paper, we compare the performance of each extracted feature set under consideration of several 2D or 3D motion compensation parameters for the differential diagnosis of enhancing lesions in breast MRI. Based on several simulation results, we determined the optimal motion compensation parameters. Our results have shown that motion compensation can improve the classification results. The results suggest that the computerized analysis system based on the non-rigid motion compensation technique and spatiotemporal features has the potential to increase the diagnostic accuracy of MRI mammography for small lesions and can be used as a basis for computer-aided diagnosis of breast cancer with MR mammography.

1. Introduction

Breast cancer is one of the leading death cases among women in the US. MR technology has advanced tremendously and became a highly sensitive method for the detection of invasive breast cancer [1]. While only dynamic signal intensity characteristics are integrated in today's CAD systems leaving morphological features to the interpretation of the radiologist, an automated diagnosis based on a combination of both should be strived for. Clinical studies have shown that combinations of different dynamic and morphologic characteristics [2] achieve diagnostic sensitivities up to 97% and specificities up to 76.5%.

However, most of these techniques have not been applied to small enhancing foci having a size of less than 1 cm. These diagnostically challenging cases found unclear ultrasound or mammography indications can be adequately visualized in magnetic resonance imaging (MRI) [3] with MRI providing an accurate estimation of invasive breast cancer tumor size [4].

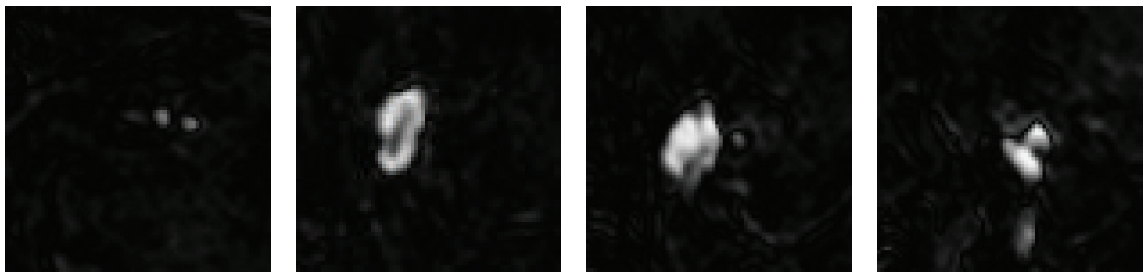
Automatic motion correction represents an important prerequisite to a correct automated small lesion evaluation [5]. Motion artifacts are caused either by the relaxation of the pectoral muscle or involuntary patient motion and invalidate the assumption of same spatial location within the breast of the corresponding voxels in the acquired volumes for assessing lesion enhancement. Especially for small lesions, the assumption of correct spatial alignment often leads to misinterpretation of the diagnostic significance of enhancing lesions [6]. Therefore, spatial registration has to be performed before enhancement curve analysis. Due to the elasticity and heterogeneity of breast tissue, only nonrigid image registration methods are suitable. Although there has been a significant amount of research on nonrigid motion



(a) (b)

(c) (d)

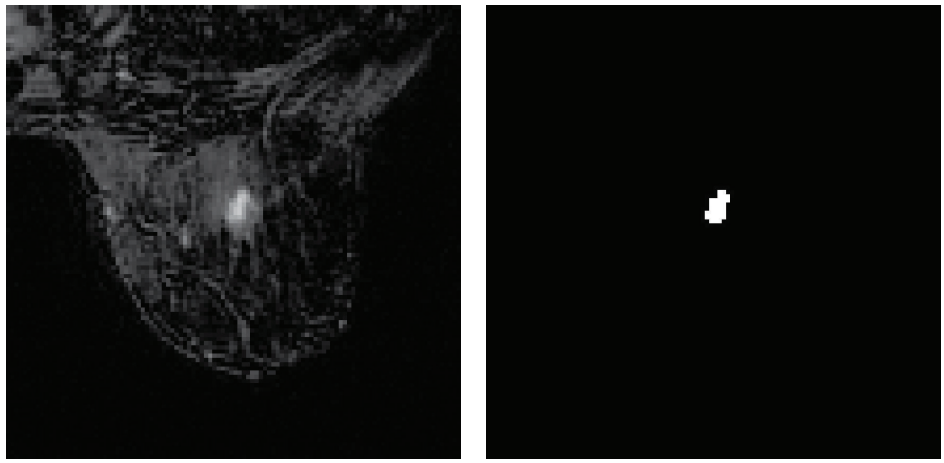
Figure 1: Motion detection on a transverse image. (a) Masking the data term: the green lines separate the boundary between masked and unmasked areas. (b) Color code describing motion from the interior of the image. (c) Motion in two directions determined without mask and (d) based on the mask from (a). The values for the standard deviation of the Gaussian presmoothing kernel and for the smoothness term are $\sigma = 3$ and $\alpha = 500$.



(a) (b) (c) (d)

Figure 2: Tumor in adjacent transverse slices of a $512 \times 512 \times 32$ image.

compensation techniques in brain imaging, few methods have been so far proposed for breast MRI. Most proposed techniques employ physically motivated deformation models [6, 7], transformations based on the deformation of B-splines, [8, 9], elastic transformations [10], and more recently adaptive grid generation algorithms [11]. We present in this paper a novel elastic image registration method based on the variational optical flow computation and will study its impact on the shape of the enhancement curves for small lesions.



(a) (b)

Figure 3: Example of an MR images segmentation showing the transverse image of a tumor in the right breast (a) and the binary segmentation of the tumor (b).

Table 1: Motion compensation parameters for two or three directions. σ represents the standard deviation for presmoothing and α the regularization parameter.

01	No motion compensation
02	3 directions, $\sigma = 1$, $\alpha = 100$
03	3 directions, $\sigma = 1$, $\alpha = 500$
04	3 directions, $\sigma = 3$, $\alpha = 100$
05	3 directions, $\sigma = 13$, $\alpha = 500$
06	2 directions, $\sigma = 1$, $\alpha = 100$
07	2 directions, $\sigma = 1$, $\alpha = 500$
08	2 directions, $\sigma = 3$, $\alpha = 100$
09	2 directions, $\sigma = 13$, $\alpha = 500$

The automated evaluation is a multistep system which includes a non-rigid motion compensation technique based on the optical flow, global and local feature extraction methods as shape descriptors, dynamical features, and spatiotemporal features combining both morphology and dynamics aspects to determine the optimal motion compensation parameters.

The paper is organized as follows. Section 2 describes the materials and methods, Section 3 the motion compensation technique, Section 4 the feature extraction techniques, and Section 5 the results. In Section 3, we present the non-rigid motion compensation technique, the feature extraction based on geometrical features, morphological features, dynamical features, and scaling index method for simultaneous time and space descriptions, and as classification technique the Fisher's discriminant analysis.

2. Material and Methods

2.1. Patients. A total of 40 patients, all female and age range 42–73, with indeterminate small mammographic breast lesions were examined. All patients were consecutively selected after clinical examinations, mammography in standard projections (craniocaudal and oblique mediolateral projections) and ultrasound. Only lesions BIRADS 3 and 4 were selected where at least one of the following criteria was present: nonpalpable lesion, previous surgery with intense scarring, or location difficult for biopsy (close to chest wall). All patients had histopathologically confirmed diagnosis from needle aspiration/excision biopsy and surgical removal. Breast cancer was diagnosed in 17 out of the total 31 cases. The average size of both benign and malignant tumors was less than 1.1 cm.

2.2. MR Imaging. MRI was performed with a 1.5 T system (Magnetom Vision, Siemens, Erlangen, Germany) with two different protocols equipped with a dedicated surface coil to enable simultaneous imaging of both breasts. The patients were placed in a prone position. First, transversal images were acquired with a STIR (short TI inversion recovery) sequence (TR = 5600 ms, TE = 60 ms, FA = 90°, IT = 150 ms, matrix size 256 × 256 pixels, slice thickness 4 mm). Then a dynamic T-weighted gradient echo sequence (3D fast low angle shot sequence) was performed (TR = 11 ms and TR = 9 ms, TE = 5 ms, FA = 25°) in transversal slice orientation with a matrix size of 256 × 256 pixels and an effective slice thickness of 4 mm or 2 mm.

The dynamic study consisted of 6 measurements with an interval of 83 s. The first frame was acquired before injection of paramagnetic contrast agent (gadopentetate dimeglumine, 0.1 mmol/kg body weight, Magnevist, Schering, Berlin, Germany) immediately followed by the 5 other measurements. The initial localization of suspicious breast lesions was performed by computing difference images, that is subtracting the image data of the first from the fourth acquisition. As a preprocessing step to clustering, each raw gray level time-series $S(\tau), \tau \in \{1, \ldots, 6\}$ was transformed into a signal time-series of relative signal enhancement $x(\tau)$

Table 2: Areas under the ROC curves for contour features using FLDA. The rows represent the motion compensation as given in Table 1. Numbers in boldface show the best results.

Feature type	Area under the ROC curve (%)								
	01	02	03	04	05	06	07	08	09
Radius min.	70.2	**85.1**	79.0	72.9	63.9	80.3	66.8	74.4	71.2
Radius max.	**83.4**	76.3	81.7	84.0	84.2	80.7	80.0	79.6	83.4
Radius mean	**83.2**	80.0	83.0	83.4	79.8	81.1	76.7	82.6	84.2
Radius st. dev.	82.4	70.4	76.1	80.3	80.7	76.9	79.2	75.0	76.9
Radius entropy	83.0	76.7	79.6	84.0	74.8	79.6	75.0	80.7	80.9
Azimuth entropy	80.7	81.9	77.7	79.6	77.5	81.9	79.0	78.4	79.6
Compactness	69.5	**77.1**	73.7	75.6	67.9	68.9	68.7	71.4	65.8

Table 3: Areas under the ROC curves for morphological features (geometric moments I_1 to I_6) using FLDA. The rows represent the motion compensation as given in Table 1. Numbers in boldface show the best results.

Feature type	Area under the ROC curve (%)								
	01	02	03	04	05	06	07	08	09
I_1	**64.5**	54.4	64.7	60.3	63.2	**67.6**	65.8	64.1	66.2
I_2	56.3	58.0	56.7	54.2	69.5	60.7	63.7	66.8	59.7
I_3	56.3	**78.8**	65.5	58.0	59.0	67.2	68.1	61.6	62.2
I_4	56.3	55.9	61.8	63.4	56.3	58.2	63.9	59.2	54.0
I_5	56.3	52.9	58.8	66.2	64.5	64.5	64.5	61.8	59.2
I_6	56.3	56.3	63.7	59.9	60.1	67.2	66.0	64.3	66.4

for each voxel, the precontrast scan at $\tau = 1$ serving as reference. Thus, we ensure that the proposed method is less sensitive to changing between different MR scanners and/or protocols.

3. Motion Compensation Technique

The employed motion compensation algorithm is based on the Horn and Schunck method [12] and represents a variational method for computing the displacement field, the so-called optical flow, in an image sequence. It is based on two typical assumptions for variational optical flow methods, the brightness constancy, and smoothness assumption.

In this context, the MR image sequence f_0 is a differentiable function of brightness values on a four-dimensional spatiotemporal image domain Ω:

$$f_0 : \underbrace{\Omega}_{\subset \mathbb{R}^3} \times \mathbb{R}_+ \longrightarrow \mathbb{R}_+. \tag{1}$$

From this image sequence, we want to compute a dense vector field $\mathbf{u} = (u_1, u_2, u_3)^\top : \Omega \to \mathbb{R}^{\{2,3\}}$ that describes the motion between the precontrast image at time point t and a postcontrast image at time point $t + k$, either for all three dimensions ($\mathbb{R}^3$) or only in one transversal slice ($\mathbb{R}^2$).

The initial image sequence f_0 is preprocessed and convolved with a Gaussian K_σ of a standard deviation σ to remove noise in the image:

$$f = K_\sigma * f_0. \tag{2}$$

The brightness constancy assumption dictates that under the motion $\mathbf{u}$, the image brightness values of the precontrast image at time t and the postcontrast image at time $t + k$ remain constant in every pixel:

$$f(\mathbf{x} + \mathbf{u}(\mathbf{x}), t + k) = f(\mathbf{x}, t), \quad \forall \mathbf{x} \in \Omega. \tag{3}$$

Naturally, this condition by itself is not sufficient to describe the motion field $\mathbf{u}$ properly, since for a brightness value in an image voxel in the precontrast image, there are generally many voxels in the postcontrast image with the same brightness value, or, in the presence of noise, possibly even none at all. Therefore, we include the smoothness assumption, dictating that neighboring voxels should move in the same direction, which is expressed as the gradient magnitude of the flow field components is supposed 0 to be as follows:

$$\left| \nabla u_{\{1,2,3\}}(\mathbf{x}) \right| = 0, \quad \forall \mathbf{x} \in \Omega. \tag{4}$$

This constraint by itself would force the motion field to be a rigid translation, which is not the case in MR images. However, if we use both the brightness constancy assumption and the smoothness assumption as weak constraints in an energy formulation, the motion field $\mathbf{u}$ that minimizes this energy matches the postcontrast image to the precontrast image and is spatially smooth.

The variational method is based on the minimization of the continuous energy functional which penalizes all deviations from model assumptions:

$$E(\mathbf{u}) = \int_\Omega \underbrace{\left(f(\mathbf{x} + \mathbf{u}(\mathbf{x}), t + k) - f(\mathbf{x}, t)\right)^2}_{\text{Data term}} + \underbrace{\alpha\left(|\nabla u_1(\mathbf{x})|^2 + |\nabla u_2(\mathbf{x})|^2 + |\nabla u_3(\mathbf{x})|^2\right)}_{\text{Smoothness term}} \mathrm{d}\mathbf{x}. \tag{5}$$

Table 4: Areas under the ROC curves for dynamic features (slope) using FLDA. The rows represent the motion compensation as given by Table 1.

Feature type	Area under the ROC curve (%)								
	01	02	03	04	05	06	07	08	09
Slope	70.4	75.6	74.2	75.0	75.0	73.9	75.8	72.7	75.4

The weight term $\alpha > 0$ represents the regularization parameter where larger values correspond to smoother flow fields. This technique is a global method where the filling-in-effect yields dense flow fields, and no subsequent interpolation is necessary as with the technique proposed in [13]. This method works within a single variational framework.

Given the computed motion from the precontrast image to a postcontrast image, the postcontrast image is being registered backwards before its difference image with the precontrast image is being computed for tumor classification:

$$f_{\text{post-registered}}(\mathbf{x}) = f_{\text{post}}(\mathbf{x} + \mathbf{u}(\mathbf{x})). \tag{6}$$

Since this registration explicitly yields the brightness constancy assumption, if the motion has been estimated correctly, the registered postcontrast image is equal to the precontrast image.

Breast MR images are mostly characterized by brightness, since bright regions are created by fatty tissue or contrast agent enhancing tumor tissue, while dark regions describe glandular tissue and background. Tumors are mainly located in the glandular tissue and proliferate either into the fatty tissue (invasive) or along the boundary inside the glandular tissue (noninvasive).

Two major concerns have to be addressed when applying the optical flow approach to breast MRI: (1) The constancy assumption does not hold for objects appearing from one image to the next such as lesions the contrast agent enhancement of which is much stronger than in the surrounding tissue and (2) The lack of constant grid size in all directions since voxel size is smaller than the slice thickness. The first concern is alleviated by masking suspicious areas by a radiologist and by detecting the sharp gradients in the motion field in the unmasked image. Figure 1 shows the masking of the entire upper image and visualizes the motion in the inner slice by a color code. This color code describes the motion direction based on the hue and the motion magnitude based on the brightness and thus identifies suspicious regions by detecting the sharp gradients. The mask $m : \Omega \to \{0, 1\}$ can be easily incorporated in the energy formulation and it forces the data term to disappear in suspicious regions:

$$\begin{aligned} E(\mathbf{u}) = & \int_\Omega m(\mathbf{x})\big(f(\mathbf{x} + \mathbf{u}(\mathbf{x}), t + k) - f(\mathbf{x}, t)\big)^2 \\ & + \alpha\Big(|\nabla u_1(\mathbf{x})|^2 + |\nabla u_2(\mathbf{x})|^2 + |\nabla u_3(\mathbf{x})|^2\Big)\mathrm{d}\mathbf{x}. \end{aligned} \tag{7}$$

In addition, there can be gaps between the slices where no nuclei are being excited in order to avoid overlapping of the slices. Figure 2 shows an example of a tumor considerably shifting its position on adjacent transversal slices.

To overcome the second concern, it is important to decide whether the motion in transverse direction is having a significant impact. The present research has shown that there is no significant difference in visual quality if motion is computed in two or three directions. In the following notation, we will consider the motion in three directions, the one in two directions is analogous.

Based on the work of Horn and Schunck, a vast variety of research in optical flow estimation has been conducted, most of it on more robust, edge preserving regularity terms, for example [14]. In our work however, we favor the original quadratic formulation, since we explicitly need the filling-in effect of a non-robust regularizer to fill in the information in masked regions. To overcome the problem of having a non-convex energy in (5), we use the coarse-to-fine warping scheme detailed in [14], which linearizes the data term as in [12] and computes incremental solutions on different image scales.

For this, we approximate the image f at the distorted point $(\mathbf{x} + \mathbf{u}(\mathbf{x}), t + k)$ by a first-order Taylor approximation:

$$\begin{aligned} & f(\mathbf{x} + \mathbf{u}(\mathbf{x}), t + k) - f(\mathbf{x}, t) \\ & \quad \approx f(\mathbf{x}, t + k) + \nabla f(\mathbf{x}, t + k)^\top \mathbf{u}(\mathbf{x}) - f(\mathbf{x}, t). \end{aligned} \tag{8}$$

Alternatively, one can develop the Taylor series at time t, getting

$$\begin{aligned} & f(\mathbf{x} + \mathbf{u}(\mathbf{x}), t + k) - f(\mathbf{x}, t) \\ & \quad \approx f(\mathbf{x}, t) + \nabla f(\mathbf{x}, t)^\top \mathbf{u}(\mathbf{x}) + \frac{\partial}{\partial t} f(\mathbf{x}, t)k - f(\mathbf{x}, t). \end{aligned} \tag{9}$$

Since the term $f(\mathbf{x}, t)$ cancels and the temporal derivative is again approximated by the difference between the two images at point $\mathbf{x}$, the only difference between (8) and (9) is the time at which the spatial derivative ∇f is being computed. In optical flow computation, one usually uses the arithmetic mean $(1/2)(\nabla f(\mathbf{x}, t + k) + \nabla f(\mathbf{x}, t))$. For the purpose of registration, the scalar factor k can be neglected since it is arbitrary whether one computes a motion for a $k \neq 1$ and then scales the motion later on with k when registering the image, or simply sets k to 1. Incorporating this into the energy formulation and leaving out the indices for better readability, the linearized energy functional then reads as

$$\begin{aligned} E_{\text{lin}}(\mathbf{u}) = & \int_\Omega m\Big(\frac{\partial}{\partial t} f + \nabla f^\top \mathbf{u}\Big)^2 \\ & + \alpha\Big(|\nabla u_1|^2 + |\nabla u_2|^2 + |\nabla u_3|^2\Big)d\mathbf{x}. \end{aligned} \tag{10}$$

This functional is convex in $\mathbf{u}$, and a minimizer can be found by solving its Euler-Lagrange equations:

$$0 = m\left(\frac{\partial f}{\partial x}\frac{\partial f}{\partial x}u_1 + \frac{\partial f}{\partial x}\frac{\partial f}{\partial y}u_2 + \frac{\partial f}{\partial x}\frac{\partial f}{\partial z}u_3 + \frac{\partial f}{\partial x}\frac{\partial f}{\partial t}\right) - \alpha\Delta u_1 \quad \forall \mathbf{x} \in \Omega,$$

$$0 = m\left(\frac{\partial f}{\partial x}\frac{\partial f}{\partial y}u_1 + \frac{\partial f}{\partial y}\frac{\partial f}{\partial y}u_2 + \frac{\partial f}{\partial y}\frac{\partial f}{\partial z}u_3 + \frac{\partial f}{\partial y}\frac{\partial f}{\partial t}\right) - \alpha\Delta u_2 \quad \forall \mathbf{x} \in \Omega,$$

$$0 = m\left(\frac{\partial f}{\partial x}\frac{\partial f}{\partial z}u_1 + \frac{\partial f}{\partial y}\frac{\partial f}{\partial z}u_2 + \frac{\partial f}{\partial z}\frac{\partial f}{\partial z}u_3 + \frac{\partial f}{\partial z}\frac{\partial f}{\partial t}\right) - \alpha\Delta u_3 \quad \forall \mathbf{x} \in \Omega. \tag{11}$$

Since the linearization in (8) or (9) is only valid for small motions in the subpixel range, a typical strategy to overcome the problem of large motions is to downsample the MR images to a coarse resolution, compute an approximate motion on the coarse resolution, interpolate this motion to the next finer resolution, register the second image with the approximate motion, compute the incremental motion from the first image to the registered second image, add the incremental motion to the approximate motion, and repeat this iteration up to the original resolution.

4. Segmentation

Before we proceed with feature extraction, each MR image has to be segmented into two regions, the region of interest (ROI), that is, the voxels belonging to the tumor, and the background. We are using an interactive region growing algorithm, creating a binary mask the tumor voxels of which are true, and all other voxels are false. The image used for the region growing algorithm was the difference image of the second postcontrast image and the native precontrast image. The center of the lesion was interactively marked on one slice of the subtraction images and then a region growing algorithm included all adjacent contrast-enhancing voxels also those from neighboring slices. Thus a 3D form of the lesion was determined. An interactive ROI was necessary whenever the lesion was connected with diffuse contrast enhancement, as it is the case in mastopathic tissue.

Figure 3 shows a transverse image of a tumor in the right breast and its binary segmentation, created with region growing.

5. Feature Extraction

The complexity of the spatio-temporal tumor representation requires specific morphology and/or kinetic descriptors. We analyzed geometric and dynamical features as shape descriptors, provided a temporal enhancement modeling for kinetic feature extraction and the scaling index method for the simultaneous morphological and dynamics representation.

5.1. Contour Features. To represent the shape of a tumor, we compute the following features: the maximal, the minimum, the mean, and the standard deviation of a radius as well as the azimuth, entropy, and compactness.

First, we determine the center of mass of the tumor as

$$\overline{v} := \frac{1}{n}\sum_{i=1}^{n} v_i, \tag{12}$$

where n is the number of voxels belonging to the tumor, and v_i is location of the ith voxel.

We define for each contour voxel c_i, the radius r_i and the azimuth ω_i (i.e, the angle between the vector from the center of mass to the voxel c_i and the sagittal plane) and give in the following the definitions of the geometrical features to be used in context with the motion compensation technique.

From the set of the values $r_1, \ldots, r_m$ representing the radii, we determine the minimum value $r_{\min}$ and the maximum value $r_{\max}$ as well as

the radius:

$$r_i := \|c_i - \overline{v}\|_2, \tag{13}$$

the azimuth:

$$\omega_i := \operatorname{arcsine}\left(\frac{c_{ix} - \overline{v}_x}{(c_{ix} - \overline{v}_x)^2 + \left(c_{iy} - \overline{v}_y\right)^2}\right), \tag{14}$$

the mean value:

$$\overline{r} := \frac{1}{m}\sum_{i=1}^{m} r_i \tag{15}$$

the standard deviation:

$$\sigma_r := \sqrt{\frac{1}{m}\sum_{i=1}^{m}(r_i - \overline{r})^2}, \tag{16}$$

and the entropy:

$$h_r := -\sum_{i=1}^{100} p_i \cdot \log_2(p_i). \tag{17}$$

The subscripts x and y denote the position of the voxel in sagittal and coronal direction respectively. ω_i was also extended to the range from $-\pi$ to π by taking into account the sign of $(c_{iy} - \overline{v}_y)$.

The entropy h_r was computed from the normalized distribution of the values into 100 "buckets," where p_i is defined as follows:

For $0 \le \mathrm{i} \le 99$:

$$p_i := \frac{\left|\left\{r_j \mid i \le \left(r_j - r_{\min}\right)/(r_{\max} - r_{\min}) \cdot 100 < i + 1\right\}\right|}{m}. \tag{18}$$

From the radius, $r_{\min}$, $r_{\max}$, $\overline{r}$, σ_r, and h_r were used as morphological features of the tumor. From the azimuth, only the entropy h_ω (computed for ω as in (17) and (18)) was used

as a feature, since the values $\omega_{\min}$ and $\omega_{\max}$ are always around π and $-\pi$, and the values $\overline{\omega}$, and σ_ω are not invariant under rotation of the tumor image.

An additional measurement describing the compactness of the tumor, which was also used as a feature, is the number of contour voxels, divided by the number of all voxels belonging to the tumor.

5.2. Geometric Moments. The spatial and morphological variations of a tumor can be easily captured by shape descriptors. Here, we will employ geometric moments [15] as shape descriptors for our lesions. For each lesion, we determine 6 three-dimensional moments that are invariant under rotation, translation, and scaling and are able to completely characterize the shape of the tumor.

Geometric moments are defined in the discrete three-dimensional case as

$$m_{\mathrm{pqr}} = \sum_x \sum_y x^p y^q z^r f(x,y,z), \tag{19}$$

with $f(x,y,z)$ being the image intensity function (gray values). Finally, we compute three-dimensional moment invariants [16] that are not computational intensive and provide results stable to noise and distortion.

5.3. Dynamical Features. For each lesion, the time/intensity curve is computed based on the average signal intensity of the tumor before contrast agent administration (SI) and after contrast agent administration (SI_C), the relative enhancement as

$$\Delta\mathrm{SI} := \frac{\mathrm{SI}_C - \mathrm{SI}}{\mathrm{SI}} \cdot 100\%. \tag{20}$$

To capture the slope of the curve of relative signal intensity enhancement (RSIE) versus time in the late postcontrast time, we compute the linear approximation of the RSIE curve for the last three time scans. The slope of this curve represents an important parameter for lesion classification.

5.4. Simultaneous Morphology and Dynamics Representations. The scaling index method [17] describes mathematically an estimation of the local scaling properties of a point represented in an n-dimensional embedding space. For every single point in the distribution, the number N of points is determined, which are included in an n-dimensional sphere of a radius r centered at $\mathbf{x_i}$

$$N(\mathbf{x_i}, r) = \sum_j \Theta\left(r - \left|\mathbf{x_j} - \mathbf{x_i}\right|\right), \tag{21}$$

where Θ represents the Heaviside function. The function $N(\mathbf{x_i}, r)$, is usually determined within a specified range $r \in [r_1, r_2]$, the so-called scaling range.

If the scaling region is large, then $N(\mathbf{x_i}, r)$ becomes $N(\mathbf{x_i}, r) \propto \mathbf{r}^\alpha$ where α is the scaling index. A first-order approximation yields

$$\alpha_i = \frac{(\log N(\mathbf{x_i}, r_2) - \log N(\mathbf{x_i}, r_1))}{(\log r_2 - \log r_1)} \tag{22}$$

with r_1 and r_2 being the lower and upper limit of the scaling range.

This technique can easily distinguish between point-like ($\alpha = 0$), string-like ($\alpha = 1$), and sheet-like ($\alpha = 2$) structures, in images and be thus employed for texture extraction. In the context of breast MRI, such a point consists of the sagittal, coronal, and transverse positions of a tumor voxel and its third-time scan gray value, and the scaling index serves as an approximation of the dimension of local point distributions.

The scaling index serves as a descriptor of the simultaneous enhancement kinetics and morphology. In particular, it can capture the blooming and the hook sign [18] and identify cancers with benign-like kinetics, discriminate normal tissue showing cancer-like enhancement curves, and thus improve the performance of computer-assisted detection of small enhancing lesions.

5.5. Morphologic Blooming. Another feature relying on morphology as well as on dynamics is the so-called *blooming*: the tumor contour becoming blurred and proliferating over the time of contrast agent enhancement is an indication for malignancy [19]. Beside blooming, [PTL+ 06] also attributes "centripetal" enhancement (strong, contrast agent enhancement near the tumor contour, weak enhancement in the tumor center with advancing time), and inhomogeneous contrast agent enhancement in general to a high probability of malignancy of the tumor.

To capture the blooming of a tumor, the difference of the relative signal intensity enhancement (RSIE) of every contour voxel and the average RSIE of its nontumor neighbor voxels was computed, and their mean, standard deviation and entropy from the 3rd to the 5th postcontrast image were used as features. Unlike for computation of the radial transform, for these features not only voxels in the contour of each transverse slice were considered contour voxels, but also voxels of the tumor with a non-tumor neighbor voxel on an adjacent transverse slice. The minimum and maximum values were neglected as features, since the gray values in the images are sensitive to noise.

The general irregularity of contrast agent enhancement in the tumor interior was computed as the standard deviation and entropy of the RSIE $s_{i,j}$ described above for the time scans 1, 3, and 5 ($i \in \{1,3,5\}$), and for all tumor voxels j. For each of the three time scans, the standard deviation and entropy were used as a feature.

In order to compensate general contrast and brightness differences between the images, for all features described in this section, the images were normalized to have the same mean as the precontrast image, and the standard deviation was divided by the standard deviation of the precontrast image.

6. Results

In the following, we will explore the applicability of the previously described motion compensation algorithm under different motion compensation parameters and features sets. The results will elucidate the applicability of motion

Table 5: Areas under the ROC curves for scaling index (SI) method using FLDA. The rows represent the motion compensation as given by Table 1. Numbers in boldface show the best results.

Feature type	Area under the ROC curve (%)								
	01	02	03	04	05	06	07	08	09
SI max.	74.6	66.6	64.3	69.3	62.8	**80.3**	69.1	60.9	62.8
SI mean	**80.3**	79.8	79.6	80.0	78.6	80.5	77.1	79.6	77.5
SI st. dev.	52.7	**81.5**	73.3	70.8	74.4	61.1	67.4	79.2	72.5
SI entropy	70.8	71.4	75.0	72.7	71.6	68.9	65.1	75.0	**76.5**

Table 6: Areas under the ROC curves for spatio-temporal features using FLDA. The rows represent the motion compensation as given in Table 1.

Feature type	Area under the ROC curve (%)								
	01	02	03	04	05	06	07	08	09
RSIE st. dev. (3)	56.7	52.5	55.0	52.3	48.1	52.5	55.5	58.4	55.3
RSIE st. dev. (4)	64.9	65.8	68.5	62.8	63.7	66.6	66.8	69.5	64.5
RSIE st. dev. (5)	64.7	70.6	70.0	64.9	68.3	69.7	65.8	66.8	69.5
RSIE entropy (3)	**80.9**	84.2	**85.3**	83.0	79.4	84.5	81.3	81.5	83.4
RSIE entropy (4)	77.9	87.4	81.9	80.7	76.7	83.8	78.8	77.3	78.4
RSIE entropy (5)	**74.6**	79.8	81.7	81.7	73.3	**81.5**	76.3	76.9	73.5
Contour RSIE mean (3)	54.4	51.7	52.7	52.3	54.0	55.7	52.7	52.7	55.5
Contour RSIE mean (4)	63.7	56.9	62.8	59.2	61.1	59.5	59.5	59.0	59.9
Contour RSIE mean (5)	62.2	64.9	60.3	68.5	68.7	62.6	63.9	62.6	66.6
Contour RSIE st. dev. (3)	55.3	57.4	58.0	52.1	55.7	58.0	55.0	57.4	57.8
Contour RSIE st. dev. (4)	58.4	59.9	57.1	56.7	61.3	60.7	58.6	56.9	62.4
Contour RSIE st. dev. (5)	63.7	63.2	67.6	60.5	65.1	58.6	58.2	66.2	64.3
Contour RSIE entropy (3)	77.5	81.3	77.1	74.2	76.9	77.1	74.8	75.4	74.4
Contour RSIE entropy (4)	83.2	84.5	82.8	79.8	77.9	84.9	77.5	81.3	79.6
Contour RSIE entropy (5)	80.5	81.3	78.2	80.7	72.9	79.6	78.2	77.5	78.4

compensation as an artefact correction method and also the descriptive power of several tumor features with and without motion correction.

Table 1 describes the motion compensation parameters used in the subsequent evaluations.

The effect of the motion compensation based on motion compensation parameters such as the amount of presmoothing and the regularization parameter was analyzed based on different combinations of feature groups.

We analyzed the performance of each proposed feature set and combinations of those for tumor classification based on receiver-operating characteristic (ROC) and compare the area-under-the-curve (AUC) values.

As a classification method to evaluate the effect of motion compensation on breast MR images, we chose the Fisher's linear discriminant analysis. Discriminant analysis represents an important area of multivariate statistics and finds a wide application in medical imaging problems. The most known approaches are linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), and Fisher's canonical discriminant analysis.

Let us assume that $\mathbf{x}$ describes a K-dimensional feature vector, that there are J classes and there are N_j samples available in group j. The mean in group j is given by μ_j, and the covariance matrix is given by Σ_j.

The underlying idea of Fisher's linear discriminant analysis (FLDA) is to determine the directions in the multivariate space that allow the best discrimination between the sample classes. FLDA is based on a common covariance estimate and finds the most dominant direction and afterwards searches for "orthogonal" directions with the same property. The technique can extract at most $J - 1$ components, and for visualization purposes scores of the first component are plotted against that of the second one, thus yielding the optimal separation among the classes.

This technique identifies the first discriminating component based on finding the vector $\mathbf{a}$ that maximizes the discrimination index given as

$$\frac{\mathbf{a}^{\mathbf{T}}\mathbf{B}\mathbf{a}}{\mathbf{a}^{\mathbf{T}}\mathbf{W}\mathbf{a}}, \tag{23}$$

with $\mathbf{B}$ denoting the interclass sum-of-squares matrix and $\mathbf{W}$ the intraclass sum-of-squares matrix.

6.1. Effectiveness of Contour Features. The contour features described in Section 5.1 show for almost all motion compensation parameters high ROC values as shown in Table 2. Without motion compensation, the entropy as well as radius mean and maximum yield the best results. The radius minimum followed by the compactness show a significant improvement for motion compensation in 3D directions.

Table 7: Areas under the ROC curves for all combined features and for the optimal number of PCs using FLDA. The rows represent the motion compensation as given by Table 1.

	Area under the ROC curve (%)								
	01	02	03	04	05	06	07	08	09
FLDA classification	85.1	80.0	77.1	78.6	84.2	76.3	83.8	84.7	84.0
Number of PC	14	20	19	17	20	5	12	11	21

6.2. Effectiveness of Geometric Moments. The geometric moments describe a representation of average shape parameters. Table 3 shows that motion compensations in two directions yields better results than in three directions for almost all geometric moments. However, the geometric moments seen not to be an adequate mean to extract features for small lesions.

6.3. Effectiveness of Kinetic Features. The slope is derived from the first-order approximation of relative signal intensity enhancement from the last three scans. Table 4 shows that both 2D as well as 3D motion compensation yield almost equally good results.

6.4. Effectiveness of Spatiotemporal Features. For the two radii r_1 and r_2, we used radii in the size of tumor structures, $r_1 = 3\,\text{mm}$ and $r_2 = 6\,\text{mm}$. The maximum, mean, and standard deviation and entropy of the set of scaling-indices, computed from tumor points as in (15)–(17), were used as features of the tumor. The minimum was neglected, since for almost every tumor it was 0, due to isolated points.

Table 5 shows the ROC values for the scaling index method for different motion compensation parameters shown in Table 1. The scaling index mean value yields the highest results without motion compensation and for both 2D and 3D motion compensation.

The performance results for the spatio-temporal features related to both contour and tumor relative signal intensity enhancement are shown in Table 6 for the third, fourth, and fifth scans. For both the tumor and contour, the entropy showed the best results in the ROC analysis: the third scan for the tumor entropy and the fourth for the contour entropy for both uncompensated and compensated motion.

6.5. Effectiveness of All Combined Features. In a final step, we considered all determined features as a vector, reduced its dimension with PCA and used FLDA as a classification method. The AUCvalues for the optimal number of PCs and depending on the motion compensation parameters are shown in Table 7. We observe again that motion compensation in 2D yields promising results for a CAD for small lesions.

7. Conclusion

Breast MRI reading is often impeded by motion artifacts of different grades. Motion correction algorithms become a necessary correction tool in order to improve the diagnostic value of these mammographic images. We applied a new motion compensation algorithm based on the Horn and Schunck method for motion correction and determined the optimal parameters for lesion classification. Several novel lesion descriptors such as morphologic, kinetic, and spatiotemporal are applied and evaluated in context with benign and malignant lesion discrimination. The optimal motion correction results were achieved for motion compensation in two directions for mostly small standard deviations of the Gaussian kernel and smoothing parameter. Consistent with the only study known for evaluating the effect of motion correction algorithms [20], the proposed motion compensation technique achieved good results for weak motion artifacts.

The performed ROC-analysis shows that an integrated motion compensation step in a CAD system represents a valuable tool for supporting radiological diagnosis in dynamic breast MR imaging.

Acknowledgments

The research was supported by NIH Grant 5K25CA106799-04.

References

[1] S. G. Orel, M. D. Schnall, C. M. Powell et al., "Staging of suspected breast cancer: effect of MR imaging and MR-guided biopsy," *Radiology*, vol. 196, no. 1, pp. 115–122, 1995.

[2] B. Szabó, P. Aspelin, M. Wiberg, and B. Bone, "Dynamic mr imaging of the breast—analysis of kinetic and morphologic diagnsotic criteria," *Acta Radiologica*, vol. 44, no. 4, pp. 379–386, 2003.

[3] A. P. Schouten van der Velden, C. Boetes, P. Bult, and T. Wobbes, "Variability in the description of morphologic and contrast enhancement characteristics of breast lesions on magnetic resonance imaging," *The American Journal of Surgery*, vol. 192, no. 2, pp. 172–178, 2006.

[4] G. M. Grimsby, R. Gray, A. Dueck et al., "Is there concordance of invasive breast cancer pathologic tumor size with magnetic resonance imaging?" *The American Journal of Surgery*, vol. 198, no. 4, pp. 500–504, 2009.

[5] S. Behrens, H. Laue, T. Boehler, B. Kuemmerlen, H. Hahn, and H. O. Peitgen, "Computer assistance for MR based diagnosis of breast cancer: present and future challenges," *Computerized Medical Imaging and Graphics*, vol. 31, no. 4-5, pp. 236–247, 2007.

[6] A. Hill, A. Mehnert, S. Crozier, and K. McMahon, "Evaluating the accuracy and impact of registration in dynamic contrast-enhanced breast MRI," *Concepts in Magnetic Resonance B*, vol. 35, no. 2, pp. 106–120, 2009.

[7] W. R. Crum, C. Tanner, and D. J. Hawkes, "Anisotropic multi-scale fluid registration: evaluation in magnetic resonance

breast imaging," *Physics in Medicine and Biology*, vol. 50, no. 21, pp. 5153–5174, 2005.

[8] T. Rohlfing, C. R. Maurer, D. A. Bluemke, and M. A. Jacobs, "Volume-preserving nonrigid registration of MR breast images using free-form deformation with an incompressibility constraint," *IEEE Transactions on Medical Imaging*, vol. 22, no. 6, pp. 730–741, 2003.

[9] D. Rueckert, L. Sonoda, C. Hayes, D. Hill, M. Leach, and D. Hawkes, "Nonrigid registration using free-form deformations: application to breast mr images," *IEEE Transactions on Medical Imaging*, vol. 18, no. 8, pp. 712–721, 1999.

[10] R. Lucht, S. Delorme, J. Heiss et al., "Classification of signal-time curves obtained by dynamic magnetic resonance mammography: statistical comparison of quantitative methods," *Investigative Radiology*, vol. 40, no. 7, pp. 442–447, 2005.

[11] M. Y. Chu, H. M. Chen, C. Y. Hsieh et al., "Adaptive grid generation based non-rigid image registration using mutual information for breast MRI," *Journal of Signal Processing Systems*, vol. 54, no. 1–3, pp. 45–63, 2009.

[12] B. K. P. Horn and B. G. Schunck, "Determining optical flow," *Artificial Intelligence*, vol. 17, no. 1–3, pp. 185–203, 1981.

[13] K. H. Herrmann, S. Wurdinger, D. R. Fischer et al., "Application and assessment of a robust elastic motion correction algorithm to dynamic MRI," *European Radiology*, vol. 17, no. 1, pp. 259–264, 2007.

[14] N. Papenberg, A. Bruhn, T. Brox, S. Didas, and J. Weickert, "Highly accurate optic flow computation with theoretically justified warping," *International Journal of Computer Vision*, vol. 67, no. 2, pp. 141–158, 2006.

[15] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Academic Press, 1998.

[16] D. Xu and H. Li, "Geometric moment invariants," *Pattern Recognition*, vol. 41, no. 1, pp. 240–249, 2008.

[17] F. Jamitzky, R. W. Stark, W. Bunk et al., "Scaling-index method as an image processing tool in scanning-probe microscopy," *Ultramicroscopy*, vol. 86, no. 1-2, pp. 241–246, 2001.

[18] W. A. Kaiser, *Signs in MR Mammography*, Springer, 2008.

[19] A. Penn, S. Thompson, C. Lehman et al., "Morphologic blooming in breast mri as a characterization of margin for discriminating benign from malignant lesions," *Academic Radiology*, vol. 13, no. 11, pp. 1344–1354, 2006.

[20] U. Schwarz-Boeger, M. Mueller, G. Schimpfle et al., "Moco—comparison of two different algorithms for motion correction in breast mri," *Onkologie*, vol. 31, no. 2, pp. 141–158, 2008.

Hindawi Publishing Corporation
Advances in Artificial Neural Systems
Volume 2012, Article ID 190359, 12 pages
doi:10.1155/2012/190359

Research Article

Activation Detection on fMRI Time Series Using Hidden Markov Model

Rong Duan[1] and Hong Man[2]

[1] *AT&T Labs, Florham Park, NJ 07932, USA*
[2] *Department of Electrical and Computer Engineering, Stevens Institute of Technology, Hoboken, NJ 07030, USA*

Correspondence should be addressed to Rong Duan, rongduan@research.att.com

Received 16 March 2012; Revised 23 June 2012; Accepted 23 June 2012

Academic Editor: Anke Meyer-Baese

This paper introduces two unsupervised learning methods for analyzing functional magnetic resonance imaging (fMRI) data based on hidden Markov model (HMM). HMM approach is focused on capturing the first-order statistical evolution among the samples of a voxel time series, and it can provide a complimentary perspective of the BOLD signals. Two-state HMM is created for each voxel, and the model parameters are estimated from the voxel time series and the stimulus paradigm. Two different activation detection methods are presented in this paper. The first method is based on the likelihood and likelihood-ratio test, in which an additional Gaussian model is used to enhance the contrast of the HMM likelihood map. The second method is based on certain distance measures between the two state distributions, in which the most likely HMM state sequence is estimated through the Viterbi algorithm. The distance between the on-state and off-state distributions is measured either through a t-test, or using the Kullback-Leibler distance (KLD). Experimental results on both normal subject and brain tumor subject are presented. HMM approach appears to be more robust in detecting the supplemental active voxels comparing with SPM, especially for brain tumor subject.

1. Introduction

Functional magnetic resonance imaging (fMRI) is a well-established technique to monitor brain activities in the field of cognitive neuroscience. The temporal behavior of each fMRI voxel reflects the variations in the concentration of oxyhemoglobin and deoxyhemoglobin, measured through blood oxygen level-dependent (BOLD) contrast. BOLD signal is generally considered as an indirect indicator for brain activities, because neural activations may increase blood flow in certain regions of the brain.

1.1. Characteristics of fMRI Data. fMRI data are collected as a time series of 3D images. Each point in the 3D image volume is called a voxel. fMRI data have four important characteristics: (1) large data volume; (2) relatively low SNR; (3) hemodynamic delay and dispersion; (4) fractal properties. Typically, one fMRI data set includes over 100-K voxels from a whole brain scan and therefore has 100-K time series. The observed time sequences are combinations of different types of signals, such as task-related, function-related, and transiently task-related (different kinds of transiently task-related signals coming from different regions of brain). These are the signals that convey brain activation information. There are also many types of noises, which can be physiology-related, motion-related, and scanning-related. The signal to noise ratio (SNR) in typical fMRI time series can be quite low, for example, around 0.2 to 0.5. For different regions and different trails, the SNR level also varies significantly. Such noise nature causes major difficulty in signal analysis. Hemodynamic delay and dispersion further increase the complexity of fMRI signal structure. Special efforts have been made to construct flexible hemodynamic response function (HRF) which can model the hemodynamic delay and dispersion through various regions and different subjects [1, 2]. fMRI data also have fractal properties, which means that a class of objects may have certain interesting properties in common. In other words, fMRI data are approximately scale invariant or scale free.

1.2. Methodology of Analyzing fMRI Data. Two areas of fMRI-based neural systems study have attracted lots of attention over the past two decades: functional activity detection and functional connectivity detection. Functional activity detection aims to locate the spatial areas that are associated with certain psychological tasks, commonly specified by a predefined paradigm. Functional connectivity detection focuses on finding spatially separated areas that have high temporal correlations [3]. Generally, functional connectivity detection is conducted under the resting-state condition. The differences in biophysical motivations and experiment designs of these two studies are reflected in their methodologies. Functional activity detection compares the temporal series of each voxel with the excitation paradigm and functional connectivity detection compares the voxel timeseries with other series in a predefined spatial region, that is, Region of Interesting (ROI), or "seed" region. Functional activity detection is more on the temporal correlation and functional connectivity detection is more on spatial correlation. Even though the two areas have some differences, they share many common statistical modelling strategies. Both of them attempt to build models to abstract the spatial and temporal relations from the observed fMRI data. To choose a model that can capture the properties of the time series accurately and efficiently is essential in both study areas.

A large number of methods have been proposed to analyze fMRI data. Most of them can be characterized into one of these two categories: modelling based approach and data driven approach. Reference [4] provided a detailed review. Even though the authors claimed the methodologies as functional connectivity detection, certain amount of the reviewed methods were commonly used in functional activity detection study too. This paper focuses on constructing a model to extract voxel temporal characteristics and test the voxel activities using functional activity detection as example. All the models referred following are for this task if no further clarification.

An established software called statistical parametric map (SPM) is a typical functional activity detection modelling package based on general linear model (GLM) [5], general linear model transforms a voxel time series into a space spanned by a set of basis vectors defined in the design matrix. These basis vectors include a set of paradigm waveforms convolved with hemodynamic response function (HRF), as well as several low-frequency DCT bases. The residual errors of this linear transform is modelled as Gaussian pdf. The key component of this method is how to constitute the design-matrix which can accurately model the brain activation effects and separate noises. Using GLM to analyze fMRI data has following intrinsic assumptions:

(i) the activation patterns are spatially distributed in the same way for all subjects,

(ii) the response between input stimulation and brain response is linear,

(iii) the HRF function is the same for every voxel,

(iv) the time series observation has a known Gaussian distribution,

(v) the variance and covariance between repeated measurements are invariant,

(vi) the time courses of different factors affecting the variance of fMRI signals can be reliably estimated in advance,

(vii) the signals at different voxels are independent, and

(viii) the intensity distribution of background (nonactive areas) is known whereas the distribution of active areas is not known.

Reference [6] substituted paradigm with the average temporal series of ROI as seed and applied SPM on functional connectivity detection. And the most recent release of SPM incorporates dynamic causal modelling to infer the interregional coupling, but it is designed more on EEG and MEG data [7].

There are some other methods which can be considered as special cases of GLM. For example, direct subtraction method subtracts the average of "off" period from the average of "on" period. Voxels with significant difference will then be identified as active. Students' t-test can be used to measure the difference in the means by the standard deviations in "off" and "on" periods. The larger the t-value is, the larger the "on-off" difference is, and the more active the voxel is. Correlation coefficient is another special case of GLM. It measures the correlation coefficient between a reference function waveform and each voxel temporal signal waveform. Voxels with large correlation coefficient are considered to be connected. If the reference function waveform is defined by paradigm [8], it is functional activity detection, and if the reference function waveform is defined by some seed time course, it is functional connectivity detection [9].

There exist some problems in GLM-based methods. For example, GLM assumes one HRF function for all voxels. The BOLD signal is only an indirect indicator of neural activity, and many nonneural changes in the body could also influence the BOLD signal. Different brain areas may have different hemodynamic responses, which would not be accurately reflected by the general linear model. Also, GLM method typically requires grouping or averaging data over several task/control blocks, which reduces sensitivity for detecting transient task-related changes, and make it insensitive to significant changes not consistently time-synched to the task block design. Low SNR makes it possible for nontask-relevant components overshadow task-relevant components and further reduces the sensitivity and specificity. GLM only considers the time series and ignores relationships between voxels, hindering the detection of brain regions acting as functional units during the experiment. Another problem is that the GLM method does not extract the intrinsic structure of the data, which may significantly weaken its effectiveness when the a priori of fMRI signal in response to the experimental events is not known or may not be constant across all voxels.

Besides GLM-based methods, a few methods have been proposed to improve the accuracy of modelling fMRI temporal signals. Reference [10] introduced a Bayesian modelling method which used a two-state HMM to infer an optimal state sequence through Markov Chain Monte Carlo sampling. The method assumed the observation is a linear combination of a two-state HMM, which infer hidden psychological states, plus constant and trend. The model is designed as the combination of offset, linear trend and a set of two-state HMMs state sequences start at different time points. MCMC is used to estimate the optimal state sequence for each voxel. This method is good in interpreting the dynamics in each voxel, but the computing complexity is big concern as mentioned by the authors, and also the totally paradigm-free approach might introduce noise that irrelevant with the experiment design. Reference [11] applied state-space model and Kalman filter to model the baseline and stimulus effect without any parametric constrain. Reference [12] employed multiple reference functions with 100 ms shift to find the highest correlation coefficient reference function for specific voxels, which avoids the common practice of using a single-reference function for all voxels. Reference [13] proposed Gaussian mixture models to describe the mutually exclusive fMRI time sequence. Reference [14] introduced an unsupervised learning method based on hidden semi-Markov event sequence models (HSMESMs) method which had the advantage of explicitly modelling the state occupancy duration. The method decomposed an observation into true positive events, false positive events, and missing observations. The "off-on" paradigm transitions were modelled as left to right HMM true positive states, and the other periods were considered as semi-Markov false positive states. The likelihood of HSMESM was calculated iteratively to detect activity base on predefined threshold. Reference [15] used first-order Markov chain to estimate the time series, t-test, and mutual information to detect the actions. All these methods are essentially two-stage approach. The temporal property is modelled at each voxel independently and then spatial modelling is performed based on the summarized statistics from temporal analysis. Fully Bayesian spatiotemporal modelling [16] considered spatial and temporal information together. This method decomposed the observation data into spatiotemporal signal and noise, and space-time simultaneously specified autoregressive model (STSAR) was employed to construct noise model. Half-cosine HRF model and activation height model were used to construct fMRI signal models.

Granger causality analysis [17] and dynamic causal modelling [18] are two popular methods in functional connectivity detection in recent years. A series comments and controversies [19–22] have been dedicated to comparing these two methods on model selection, causality, and deconvolution from biophysiological view. Reference [23] criticized dynamic causal modelling from computation complexity and model validation from mathematical perspective. The advantage of these two methods are that they consider the spatial and temporal information at the same time, but the disadvantage is the model complexity.

All the modelling-based approaches mentioned above are either too simple to capture the temporal or spatial dynamics for different voxels and subjects, or too complicated to estimate parameters accurately and inference easily.

In addition to these modelling-based methods, there are also data driven methods in analyzing fMRI data. One popular example of this approach is independent component analysis (ICA). ICA decomposes a 4D fMRI data volume (3D spatial and 1D temporal) into a set of maximum temporal or spatial independent components by minimizing the mutual information between these components. ICA does not require the knowledge of stimulus or paradigm in data decomposition, and similar voxel activation patterns will usually appear in the same component. ICA is also called blind source separation, because it does not need prior knowledge, and it is able to identify "transient task related" components that could not be easily identified by the paradigm. The first application of ICA in fMRI data was the spatial ICA (sICA) [24]. Temporal ICA (tICA) was introduced later by [25]. Reference [26] compared the sICA with tICA and reported that the beneficial of each method depends on the independence of the underlying spatial or temporal signal. sICA maximizes independence spatially and the corresponding temporal information might be highly correlated, and vice versa for tICA. To consider the mutual independence between space and time simultaneously, [27] proposed a spatiotemporal independent component analysis (stICA). Extended from entropy-based one-dimension ICA decomposition introduced in [28], the authors embedded the spatial and temporal components at the same time and also incorporated the spatial skewed probability density function to replace the kurtosis and symmetric probability density function in decomposing the independent signals. As pointed out in [29], the disadvantages of the Infomax and entropy-based stICA algorithms used in [27] are that the number of parameters needed to be estimated is large, the local minima and sensitive to noise characteristics of the gradient descent optimization methods. To improve the stability, robustness, and simplify the computing complexity, [29] adopted the generalized eigenvalue decomposition and joint diagonalization on both spatial and temporal autocorrelation to achieve spatial and temporal independent signals simultaneously. ICA methods have showed promising results in fMRI analysis, but similar as all other data-driven methods, it is hard to interpret the output and it usually requires special knowledge and human intervention. Also, ICA does not specify which component, among many output components, is the activation component, and there is no statistical confidence level of each components extracted.

Clustering is another well-developed data driven approach in brain activity and connectivity detection. It has been used to identify regions with similar patterns of activations. Common clustering algorithms include hierarchical clustering, crisp clustering, K-means, self-organizing maps (SOM), and fussy clustering. The major drawback of most clustering methods is that they make assumptions about cluster shapes and sizes, which may deviate in observed data structures. The optimization techniques used in clustering may also result in local maxima

and instable results. In addition, the number of clusters is frequently determined heuristically and randomly initialized, which makes the output inconsistent with each trial.

In this paper, we propose a simple dynamic state space model, which attempts to model the voxel time series as a random process driven by the experimental paradigm or some ROI area seed series. For a given voxel, its behavior is described by a two-state hidden Markov model with certain state distributions and state transitions. The HMM parameters are estimated from the prior statistics of the paradigm as well as from the testing time series. Two methods are introduced to detect the voxel activation based on the estimated HMM. The first method calculates the likelihood of each time series, given its HMM, and forms a likelihood map for all the voxels reside in a fMRI slice. A simple Gaussian model is also used to improve the contrast of this likelihood map. The second method uses the t-test or the Kullback-Leibler distance (KLD) to measure the distance between the on-state distribution and the off-state distribution. These distributions are estimated based on the most likely HMM state sequence, which is calculated through a Viterbi algorithm. The contribution of the method is that it unifies the robustness, stability, and reliability under the same framework in estimating paradigm driven fMRI study. First, it incorporates the dynamic characteristics of fMRI time series by adapting 2-state HMM model, which is robust in detecting active voxels with different delay and dispersion behaviors. Second, the proposed method utilizes paradigm prior knowledge in parameter estimation, which is not only to simplify the computing compared with the approach in [10], but also to improve the stability and reliability of the output due to the stability of paradigm.

The rest of this paper is organized as follows. In Section 2, we introduce the two-state hidden Markov model approach for fMRI data. In Section 3, we discuss activation detection methods base on the estimated HMM. In Section 4, we present the experimental results on two sets of fMRI data, one is normal subject and the other is brain tumor subject, and compare the results with GLM-based statistical parametric mapping package (SPM) [5].

2. Hidden Markov Model for fMRI Time Series

2.1. Hidden Markov Model. HMM is a very efficient stochastic method in modelling sequential data of which the distribution patterns tend to cluster and alternate among different clusters [30]. A hidden Markov model consists of a finite set of states. In a traditional Markov chain, the state is directly visible to the observer, and the state transition probabilities are the only parameters. In an HMM, only the observations influenced by the state are visible. Each of the hidden state is associated with a probability distribution. Transitions among the states are measured by transition probabilities. The most common first-order HMM implies that the state at any given time depends only on the state at the previous time step.

HMM is well developed in temporal pattern recognition applications. It was first applied to speech recognition [31]. Now it is widely used in multimedia [32], bioinformatics [33, 34], informational retrieval [35], and so forth.

An HMM can be described by the following elements [31]: (1) a set of observations $O\{T\}$, where T is the number of time samples; (2) a set of states $Q\{N\}$, where N is the number of states; (3) a state-transition probability distribution $A = \{a_{ij}\}$, where $a_{ij} = P[q_{t+1} = S_j \mid q_t = S_i]$, $1 \le i, j \le N$; (4) observation probability distribution for each state $B = \{b_j(x)\}$, where $b_j(x) = P[o_t = x \mid q_t = S_j]$, $1 \le j \le N$, $x \in \Re$ is a possible observation value; (5) an initial state distribution $\pi = \{\pi_j\}$, where $\pi_j = P[q_1 = S_j]$, $1 \le j \le N$.

An HMM is therefore denoted by $\lambda = \{A, B, \pi\}$. We further model each state distribution as a Gaussian pdf:

$$P\Big[O_t = x \mid q_t = S_j\Big] = \frac{1}{\sigma_j\sqrt{2\pi}} e^{-(x-\mu_j)^2/(2\sigma_j{}^2)}, \quad j \in \{0, 1\}. \tag{1}$$

Let $Q = q_1, q_2, \ldots, q_T$ be a possible state sequence and assume that the observation samples are independent, the likelihood of an observed sequence given this HMM can be calculated as:

$$\begin{aligned} P(O \mid \lambda) &= \sum_{Q} P(O \mid Q, \lambda) P(Q \mid \lambda) \\ &= \sum_{Q} \pi_{q_1} b_{q_1}(o_1) a_{q_1 q_2} b_{q_2}(o_2) \cdots a_{q_{T-1} q_T} b_{q_T}(o_T). \end{aligned} \tag{2}$$

Given the observation O and the HMM, the most likely state sequence $Q = \{q_1, q_2, \ldots, q_T\}$, which maximizes the likelihood $P(Q \mid O, \lambda)$, can be calculated through the Viterbi algorithm [36]. The Viterbi path score function is defined as:

$$\delta_t(i) = \max_{q_1, q_2, \ldots, q_{t-1}} P[q_1 q_2 \cdots q_t = i, o_1 o_2 \cdots o_t \mid \lambda], \tag{3}$$

where $\delta_t(i)$ is the highest probable path ending in state i at time t. The induction can be expressed as:

$$\delta_{t+1}(j) = \max_{1 \le i \le N} \Big[\delta_t(i) a_{ij}\Big] b_j(o_{t+1}). \tag{4}$$

In an application of HMM, multiple HMMs are trained by different groups of labeled data. The HMM parameters are estimated based on these training data. The test data will be assigned to the one which has the maximum likelihood.

2.2. Brain Activation Detection

2.2.1. HMM Likelihood Methods. In our unsupervised learning methods, HMM parameters are estimated directly from the experimental paradigm or the voxel time series under examination. This is different from conventional HMM applications where HMM parameters are usually estimated from some training data. The attempt of avoiding training process is motivated by the fact that the true activation behavior varies from voxel to voxel and from patient to patient. Therefore, it is not advisable to use the parameters from certain set of voxels to characterize other voxels.

Since the simple block paradigm has only two levels, "on, off," in this work we let the number of state $N = 2$, that is, on-state S_1 and off-state S_0.

Because of the first-order Markov assumption, that is, $P(q_t = j \mid q_{t-1} = i, q_{t-2} = k, \ldots) = P(q_t = j \mid q_{t-1} = i)$, the distribution of a state duration is exponential, and the expected value of a state duration can be expressed as:

$$\overline{d}_i = \frac{1}{1 - a_{ii}}. \tag{5}$$

Given an experimental paradigm, let the length (i.e., time samples) of the ON period be L_{on}, and the length of off period be L_{off}, the transition matrix A can be estimated as $a_{00} = 1/(1 - L_{\text{off}})$, $a_{01} = 1 - a_{00}$, $a_{11} = 1/(1 - L_{\text{on}})$, $a_{10} = 1 - a_{11}$.

The parameters in B can be estimated from the voxel time series O. Assuming that the time samples are normalized, let p_{on} denote the paradigm ON periods, and p_{off} denote the paradigm off periods, the off-state S_0 Gaussian parameters are

$$\mu_0 = \frac{1}{|p_{\text{off}}|} \sum_{t \in p_{\text{off}}} o_t, \qquad \sigma_0 = \sqrt{\frac{1}{|p_{\text{off}}|} \sum_{t \in p_{\text{off}}} (o_t - \mu_0)^2}, \tag{6}$$

and the on-state S_1 Gaussian parameters are

$$\mu_1 = \frac{1}{|p_{\text{on}}|} \sum_{t \in p_{\text{on}}} o_t, \qquad \sigma_1 = \sqrt{\frac{1}{|p_{\text{on}}|} \sum_{t \in p_{\text{on}}} (o_t - \mu_1)^2}, \tag{7}$$

where $|p_{\text{off}}|$ is the total number of time samples in the off periods, and $|p_{\text{on}}|$ is the total number of time samples in the ON periods.

Because the paradigm always starts at the off state, the parameters in π are set as $\pi_0 = 1$ and $\pi_1 = 0$.

Given a 2-state HMM as specified, if an observation sequence does have two distinguishable states in consistence with the paradigm states, the resulting $\{\mu_0, \sigma_0\}$ will be clearly different from $\{\mu_1, \sigma_1\}$, and the likelihood of such sequence given this model will be relatively high. If an observation sequence does not have such clear 2-state characteristic, the corresponding state transition will be somehow random and will not fit well with the specified A matrix. In such situation, the likelihood of this sequence will be relatively low. Therefore, the value of voxel sequence likelihood can provide an indication about the activation of this voxel. A likelihood test on an fMRI slice will be able to produce a likelihood map with each point representing the likelihood of a voxel on this slice.

To enhance the contrast of this likelihood map, we introduce a simple Gaussian model for the p_{off} samples. This model is consistent with the S_0 state distribution in the 2-state HMM. The likelihood of the entire sequence is calculated based on this model. The expectation is that if a voxel is non-active, its distribution in p_{off} periods and p_{on} periods should be similar, and therefore the likelihood to this model should be relatively high; on the other hand, if the voxel is active, its distribution in p_{on} periods will be quite different from the distribution in p_{off} periods, and therefore the likelihood of the whole sequence on this model will be relatively low. The substraction of the HMM log likelihood map and the Gaussian log likelihood map is equivalent to a general likelihood ratio test, and it provides an activation map with enhanced contrast.

2.2.2. State Distribution Distance Methods. If a voxel is active, its fMRI time series can be partitioned into segments associated with two states, and each state can be described by a distribution. The assumption is that if the on-state distribution is significantly different from the off-state distribution, we have high confidence to declare a voxel as active, and vice versa. Therefore, the second method we are investigating attempts to measure the distance between the presumed on-state and off-state distributions.

There are many techniques available for measuring the distance of two distributions. We study two of such measures in this work, one is the t-test, and the other is the Kullback-Leibler divergence. Both on-state and off-state distributions are models as simple Gaussian pdfs.

Given the Gaussian parameters, $\{\mu_0, \sigma_0\}$ and $\{\mu_1, \sigma_1\}$, the t-test calculates the difference of two mean values corrected by their variance values

$$t = \frac{\mu_1 - \mu_0}{\sqrt{\left(\sigma_{\text{on}}^2 / |p_{\text{on}}|\right) + \left(\sigma_{\text{off}}^2 / |p_{\text{off}}|\right)}}. \tag{8}$$

A t-map is produced after the t-test is applied to all the voxles on an fMRI slice. High t values in the map usually indicate active voxles.

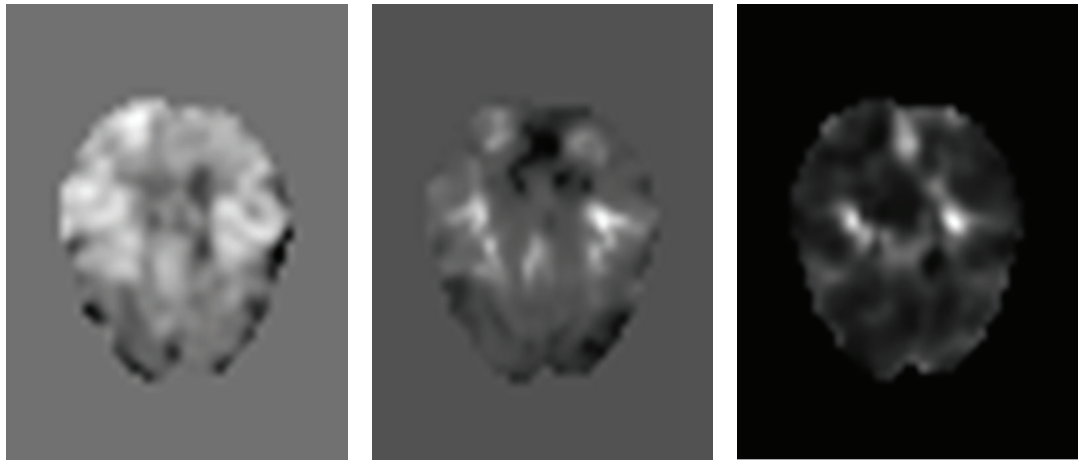
The Kullback-Leibler divergence [37] is frequently used as a distance measure for two probability densities, although in theory it is not a true distance measure because it is not symmetric. In general it is defined in the form of "relative entropy,"

$$D\left(p_i(x) \| p_j(x)\right) = -\int p_i(x) \log \frac{p_j(x)}{p_i(x)} dx. \tag{9}$$

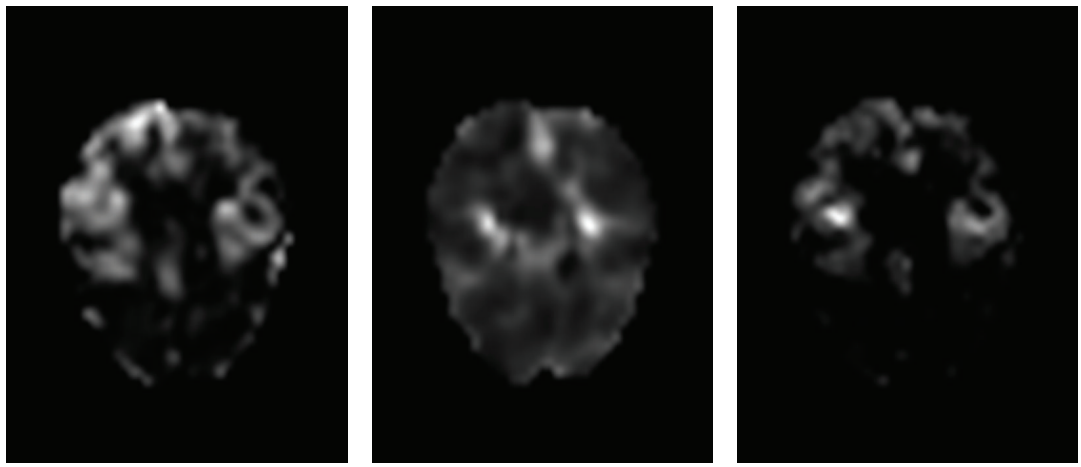
For two Gaussian pdfs, a close form expression for KLD is available:

$$D\left(p_i(\cdot; \mu_i, \sigma_i) \| p_j\left(\cdot; \mu_j, \sigma_j\right)\right) = \frac{1}{2}\left[\log\left(\frac{\sigma_j^2}{\sigma_i^2}\right) - 1 + \frac{\sigma_i^2}{\sigma_j^2} + \frac{\left(\mu_i - \mu_j\right)^2}{\sigma_j^2}\right]. \tag{10}$$

These are well-established methods. However a critical issue in fMRI analysis is how to estimate the correct on-state and off-state distributions. A simple assumption is to let all time samples in the paradigm ON periods be the on-state samples and let all samples in the paradigm off periods be the off-state samples. We refer to this approach as the "paradigm state" approach. The SPM takes a similar approach, except that the block paradigm is convolved with an HRF, which is normally a low-pass filter characterizing



(a) Paradigm state t-test (b) SPM t-test (c) 2-state HMM Viterbi path t-test

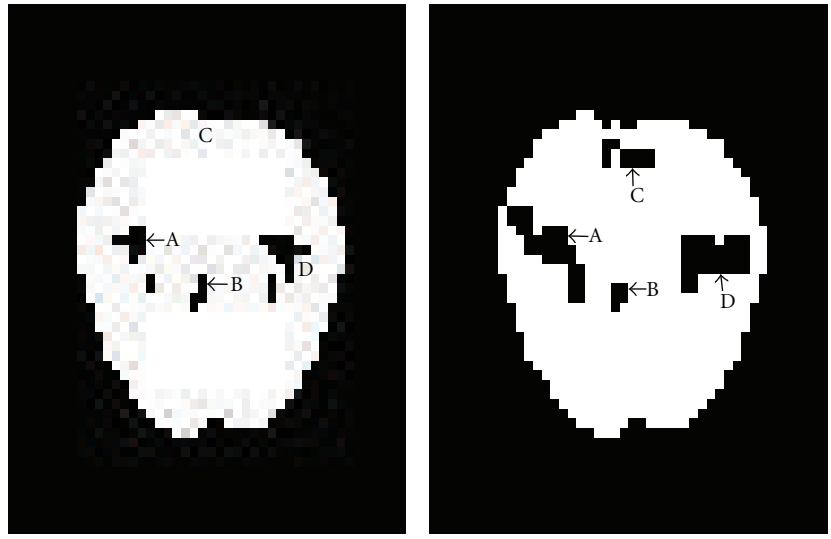


(d) Paradigm state KLD map (e) SPM F-test map (f) 2-state HMM Viterbi path KLD

Figure 1: (a), (b), and (c) are t-test map for paradigm-defined state, SPM method, and 2-state HMM Viterbi path map respectively; (d), (e), and (f) are KLD map for paradigm defined state, SPM F-test, and 2-state HMM Viterbi path KLD map, respectively. The HMM Viterbi path methods (c) and (f) produce more compact and clearly highlighted regions than paradigm state estimation (a) and (d); HMM Viterbi path methods perform similarly to SPM with some minor differences.

(a) HMM log-likelihood map (b) Gaussian log-likelihood map (c) HMM likelihood ratio test map

Figure 2: (a) HMM log-likelihood map—2-state HMM model is applied to all voxels. The active voxels with obvious 2-state on/off patterns will have relative high likelihood given this model; (b) Gaussian log-likelihood map—"off"-state Gaussian model is applied to all voxels. The nonactive voxels with obvious 2-state on/off patterns will have relative high likelihood given this model; (c) HMM likelihood ratio test map—subtraction of HMM log-likelihood map and Gaussian log-likelihood map equivalent to general likelihood ration test, and it enhances the contrast for activation map.



(a) SPM t-test map with 4 voxel locations (b) HMM LLR map with 4 voxel locations

Figure 3: Glass map for SPM t-test and HMM likelihood ratio and 4 voxel time series. "A" and "B" are detected by both SPM and HMM methods, while "C" and "D" are only detected by the HMM log-likelihood ratio method.

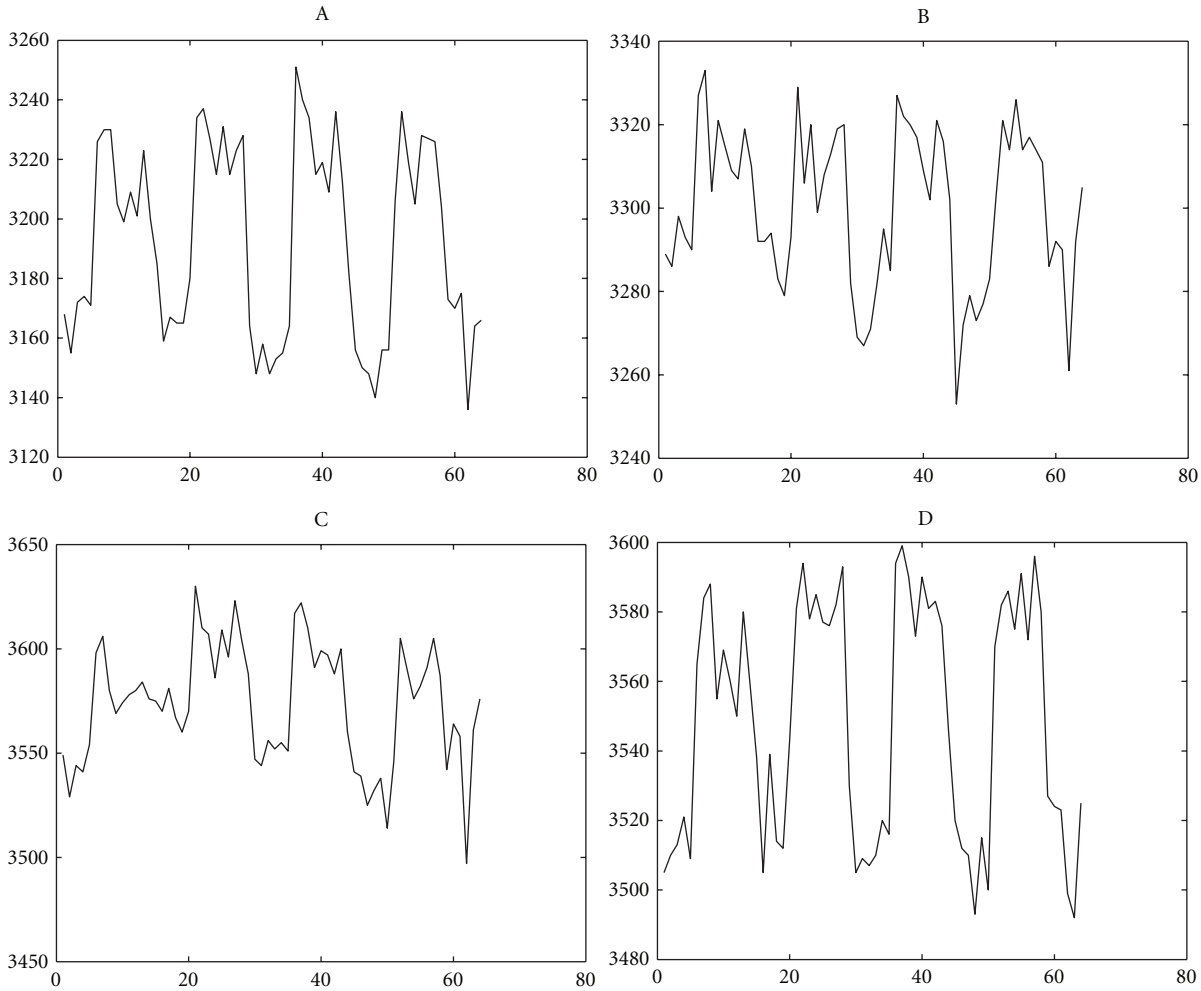
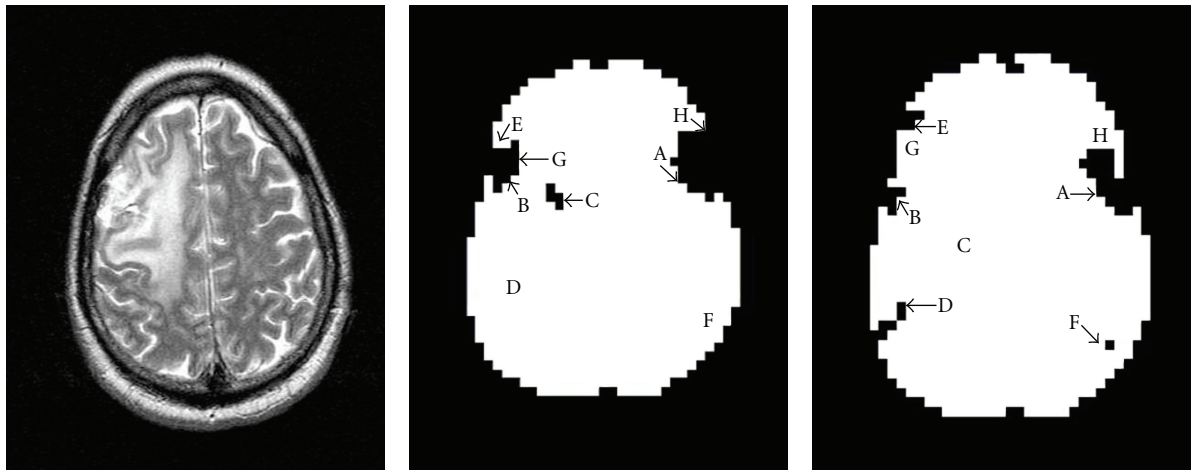


Figure 4: The time series of four active voxels. "A" and "B" are detected by both SPM and HMM methods, while "C" and "D" are only detected by the HMM log-likelihood ratio method.



(a) Tumor patient MRI brain image (b) SPM t-test map (c) HMM LLR map

FIGURE 5: Tumor patient high-resolution image and glass maps for SPM test and 2-state HMM likelihood ratio test. The patient has a tumor on his left frontal lobe from (a). No supplemental voxels from SPM t-test (b). "D" and "F" are supplemental voxels detected from HMM LLR map in (c).

the nature voxel response to a stimulus. The μ_0 and μ_1 are obtained by projecting the time series to the HRF convolved paradigm waveform, and the σ_0 and σ_1 are set to be the same to model the residual error between the voxel time series and the weighted paradigm waveform.

We take a different approach by applying the 2-state HMM on each voxel series and calculate the most likely state sequence using the Viterbi algorithm. We refer to this approach as the "Viterbi path" approach. Then the on-state and off-state statistics are calculated according to the optimal state assignment for each time sample. The $\{\mu_0, \sigma_0\}$ are obtained from all samples belonging to the off-state, and $\{\mu_1, \sigma_1\}$ are obtained from all samples belonging to the on-state.

3. Experimental Results

3.1. Normal Subject. The data set is collected from a test with self-paced bilateral sequential thumb-to-digits opposition task. The task paradigm consists of a 32 sec baseline followed by 4 cycles of 30 sec ON and 30 sec OFF. The time series is sampled at 0.25 Hz, which produces 68 time samples for each voxel. The first four samples are ignored during analysis because of initial unstable measurement. The BOLD image is acquired in a 1.5 T GE echo speed horizon scanner with the following parameters: TR/TE = 4000/60, FOV = 24 cm, 64 × 64 matrix, slice thickness 5 mm without gap, and 28 slices to cover the entire brain. Following acquisition of the functional data (with a resolution of $3.75 \times 3.75 \times 5\,\text{mm}^3$), a set of 3 mm slice thickness, high resolution (256×256 matrix size), gadolinium-enhanced images are also obtained according to clinical imaging protocol. The data is aligned to remove the limited motion between data sets then smoothed with a Gaussian kernel before further processing [5]. We further normalize each time series with the mean and variance of its paradigm off period. In order to compensate DC drifting in many voxels, each time series is partitioned into four equal-length segments, and normalization is performed separately on each of these segments. In the reported results, only one fMRI transverse slice is shown.

We first compare three methods based on two different distribution distance measures. These include the SPM with a t-test or an f-test, and our HMM Viterbi path method with a t-test or a KLD measure. The results are shown in Figure 1. From these results we can see that the primary motor and secondary motor areas are effectively highlighted by all these methods. We also have the following observations: (1) the HMM Viterbi path methods produce more compact and clearly highlighted regions, which indicates that Viterbi path estimation is more accurate than paradigm state estimation; (2) the HMM Viterbi path t-test method performs similarly to SPM t-test with some minor differences, mostly along the outer frontal regions; (3) the KLD methods have resemblance to SPM F-test in the sense that their results are pure positive, while t-test results are signed.

To test the effectiveness of our HMM likelihood ratio method, we compare its result with an SPM t-test result. In Figure 2, (a) shows the two-state HMM log-likelihood map; (b) shows the Gaussian log-likelihood map of the same slice; (c) shows the log-likelihood ratio test map. It can be seen that HMM log-likelihood map is almost the reverse of Gaussian log-likelihood map, which validates our expectation in Section 2.2.1. (c) is similar to (a), yet with enhanced contrast. This result resembles the SPM t-test result, although their magnitude scales are quite different.

We examine several active voxels detected by SPM and by HMM likelihood ratio test. In Figure 3, the SPM t-map and the HMM log likelihood ratio map are thresholded at certain level to yield similar number of active voxels. The corresponding voxel time series marked with "A," "B," "C," and "D" are shown in Figure 4. The voxels "A" and "B" can be detected by both SPM and HMM likelihood ratio test.

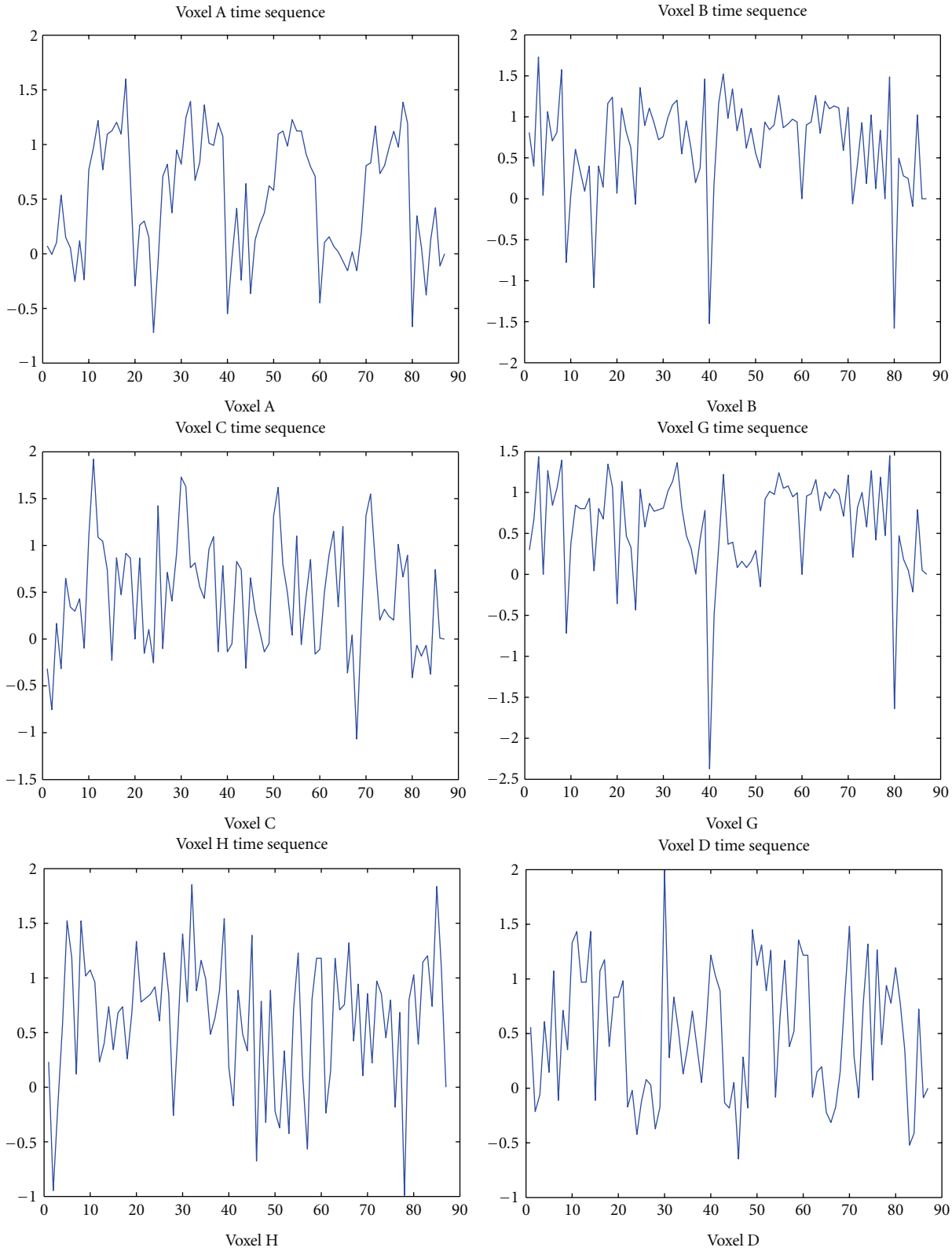


Figure 6: Continued.

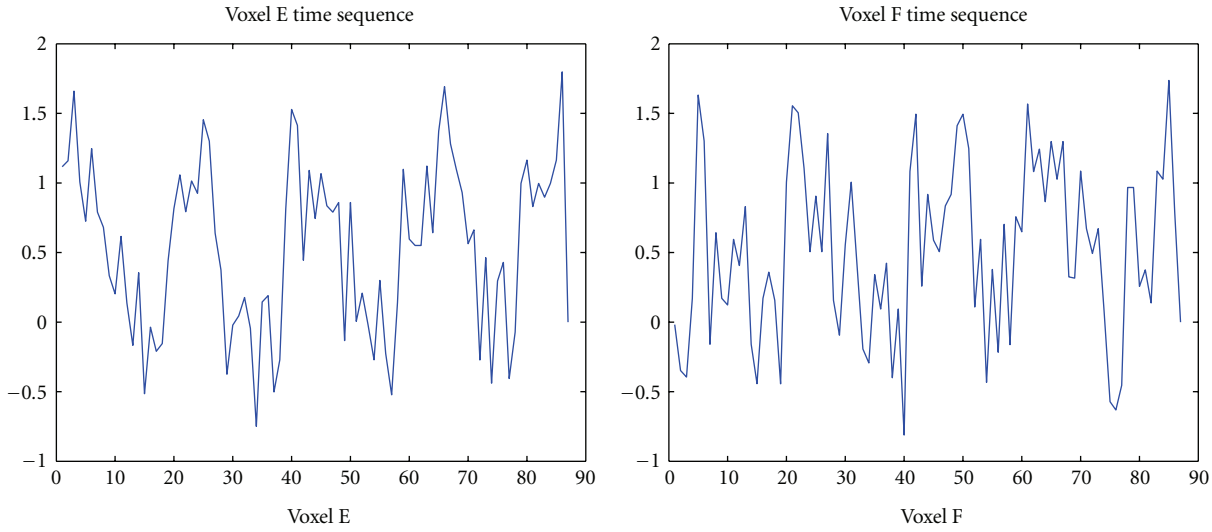


Figure 6: Voxel time series, voxels "A" and "B" are detected by both SPM *t*-test and 2-state HMM likelihood ratio test; voxels "C," "G," and "H" are detected by SPM *t*-test only. "D," "E," and "F" are detected by 2-state HMM likelihood ratio test only.

The voxels "C" and "D" are only highlighted by the HMM likelihood ratio test.

3.2. Brain Tumor Subject. Functional MRI is not only used for normal brain function mapping, it is also widely used for neurosurgical planning and neurologic risk assessment in the treatment of brain tumors. The growth of a tumor can cause functional areas to shift from their original locations. Large tumors can cause these critical regions to shift dramatically. Localizing the motor strip and coregistering the results to a surgical scan prior to a neurosurgical intervention can help guide the direct cortical stimulation during an awake craniotomy and possibly shorten operation time. In some cases, using fMRI to confirm the expected location of the motor strip may avoid awake neurosurgery altogether.

The HRF for brain tumor patient is more complicated than that for normal healthy subjects. We compare our unsupervised 2-state HMM model with GLM-based SPM on brain tumor patient and found 2-state HMM model is more robust to HRF and it is more sensitive in detecting supplemental motor activation.

The machine specification and the functional data acquisition for a tumor patient is the same as for the normal subject described above. The experiment design is different. The test is self-paced bilateral sequential thumb-to-digits opposition task. The task paradigm consists of a 20 sec baseline followed by 4 cycles of 20 sec ON and 20 sec off. Each point is 2 sec for a total of 3 min. The time series is totally 90 time samples for each voxel. The first 3 samples are ignored during analysis because of initial unstable measurement. The patient has a tumor on his left frontal lobe. As seen in the high resolution fMRI image in Figure 5(a).

Figure 5(b) is a thresholded SPM *t*-test map, both left and right motor areas are detected by SPM and there is no supplemental voxels. SPM *t*-test shows that the tumor does not impact the patient's motor area. Thresholded HMM likelihood ratio test result shown in Figure 5(c) indicates some weak motor activation in the left side motor area. In addition, there are some supplemental motor activation detected on surrounding areas. We further study several active voxels from Figures 5(b) and 5(c). The locations of selected active voxels are marked as "A," "B," "C," "D," "E." "F," "G," and "H" in each figures, respectively. The corresponding voxel time series are shown from Figure 6. From these results, we can see that there are three types of voxels. The voxels "A" and "B" are detected by both SPM and HMM likelihood ratio test, which exhibit strong activation patterns. Voxels "C", "G," and "H" are detected only by SPM, and in fact they are either very weak activations or false positives. Voxels "D," "E," and "F" are detected only in HMM likelihood ratio test and their time series have strong activation patterns related to the paradigm but with different delay and dispersion. These results reaffirmed our understanding that SPM has difficulty in locating active voxels with unexpected delay and dispersion behaviors.

4. Conclusion Remarks and Future Works

In this paper we have presented HMM-based method to detect active voxels in fMRI data. A 2-state HMM model is built based on paradigm on/off periods, and a 1-state HMM model is built based on paradigm off period. A log-likelihood ratio map is generated using the two log-likelihoods. Viterbi path is obtained for the 2-state HMM model. According to the Viterbi path, *t*-test map and KLD map are generated. From experiments we see that HMM methods are as effective as SPM method, and sometime HMM methods can detect supplemental active voxels that SPM may miss, especially in complicated cases with such as tumor patients. Overall we consider that the HMM methods are complementary to the SPM method, because SPM focuses on capturing fMRI signal waveform characteristics while HMM method attempts to

describe fMRI signal stochastic behaviors. In other words, the HMM methods can provide a second opinion to the SPM test results, which can be very helpful in practical situations.

References

[1] M. W. Woolrich, T. E. J. Behrens, and S. M. Smith, "Constrained linear basis sets for HRF modelling using Variational Bayes," *NeuroImage*, vol. 21, no. 4, pp. 1748–1761, 2004.

[2] P. Ciuciu, J. B. Poline, G. Marrelec, J. Idier, C. Pallier, and H. Benali, "Unsupervised robust nonparametric estimation of the hemodynamic response function for any fMRI experiment," *IEEE Transactions on Medical Imaging*, vol. 22, no. 10, pp. 1235–1251, 2003.

[3] L. Lee, L. M. Harrison, and A. Mechelli, "A report of the functional connectivity workshop, Dusseldorf 2002," *NeuroImage*, vol. 19, no. 2, pp. 457–465, 2003.

[4] K. Li, L. Guo, J. Nie, G. Li, and T. Liu, "Review of methods for functional brain connectivity detection using fMRI," *Computerized Medical Imaging and Graphics*, vol. 33, no. 2, pp. 131–139, 2009.

[5] K. J. Friston, A. P. Holmes, K. J. Worsley, J. P. Poline, C. D. Frith, and R. S. J. Frackowiak, "Statistical parametric maps in functional imaging: a general linear approach," *Human Brain Mapping*, vol. 2, no. 4, pp. 189–210, 1994.

[6] M. D. Greicius, B. Krasnow, A. L. Reiss, and V. Menon, "Functional connectivity in the resting brain: a network analysis of the default mode hypothesis," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 100, no. 1, pp. 253–258, 2003.

[7] J. Ashburner, "Spm:a history," *NeuroImage*, vol. 62, no. 2, pp. 791–800, 2012.

[8] G. K. Wood, "Visualization of subtle contrast-related intensity changes using temporal correlation," *Magnetic Resonance Imaging*, vol. 12, no. 7, pp. 1013–1020, 1994.

[9] J. Cao and K. Worsley, "The geometry of correlation fields with an application to functional connectivity of the brain," *Annals of Applied Probability*, vol. 9, no. 4, pp. 1021–1057, 1999.

[10] P. Hojen-Sorensen, L. Hansen, and C. Rasmussen, "Baysian modelling of fmri time series," in *Proceedings of the 13th Annual Conference on Advances in Neural Information Processing Systems (NIPS '99)*, pp. 754–760, 2000.

[11] C. Gossl, D. Auer, and L. Fahtmeir, "Dynamic models in fmri," *Magnetic Resonance in Medicine*, vol. 43, no. 1, pp. 72–81, 2000.

[12] M. Singh, W. Sungkarat, J. W. Jeong, and Y. Zhou, "Extraction of temporal information in functional MRI," *IEEE Transactions on Nuclear Science*, vol. 49, no. 5, pp. 2284–2290, 2002.

[13] V. Sanguineti, C. Parodi, S. Perissinotto et al., "Analysis of fMRI time series with mixtures of Gaussians," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '2000)*, vol. 1, pp. 331–335, July 2000.

[14] S. Faisan, L. Thoraval, J. P. Armspach, and F. Heitz, "Unsupervised learning and mapping of brain fMRI signals based on hidden semi-Markov event sequence models," in *Proceedings of the 6th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI '03)*, pp. 75–82, November 2003.

[15] B. Thirion and O. Faugeras, "Revisiting non-parametric activation detection on fMRI time series," in *Proceedings of the IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*, vol. 1, pp. 121–128, December 2001.

[16] M. W. Woolrich, M. Jenkinson, J. M. Brady, and S. M. Smith, "Fully bayesian spatio-temporal modeling of FMRI data," *IEEE Transactions on Medical Imaging*, vol. 23, no. 2, pp. 213–231, 2004.

[17] A. Roebroeck, E. Formisano, and R. Goebel, "Mapping directed influence over the brain using Granger causality and fMRI," *NeuroImage*, vol. 25, no. 1, pp. 230–242, 2005.

[18] K. J. Friston, L. Harrison, and W. Penny, "Dynamic causal modelling," *NeuroImage*, vol. 19, no. 4, pp. 1273–1302, 2003.

[19] K. Friston, "Dynamic causal modeling and Granger causality comments on: the identification of interacting networks in the brain using fMRI: model selection, causality and deconvolution," *NeuroImage*, vol. 58, pp. 303–305, 2011.

[20] O. David, "fMRI connectivity, meaning and empiricism. Comments on: Roebroeck et al. The identification of interacting networks in the brain using fMRI: model selection, causality and deconvolution," *NeuroImage*, vol. 58, pp. 306–309, 2011.

[21] A. Roebroeck, E. Formisano, and R. Goebel, "Reply to Friston and David. After comments on: the identification of interacting networks in the brain using fMRI: model selection, causality and deconvolution," *NeuroImage*, vol. 58, pp. 296–302, 2011.

[22] A. Roebroeck, E. Formisano, and R. Goebel, "Reply to Friston and David. After comments on: the identification of interacting networks in the brain using fMRI: model selection, causality and deconvolution," *NeuroImage*, vol. 58, pp. 310–311, 2011.

[23] G. Lohmann, K. Erfurth, K. Mller, and R. Turner, "Critical comments on dynamic causal modelling," *NeuroImage*, vol. 59, pp. 2322–2329, 2012.

[24] M. Mckeown, S. Makeig, G. Brown et al., "Analysis of fmri data by blind seperation into independent spatial components," *Human Brain Mapping*, vol. 6, pp. 160–188, 1998.

[25] B. B. Biswal and J. L. Ulmer, "Blind source separation of multiple signal sources of fMRI data sets using independent component analysis," *Journal of Computer Assisted Tomography*, vol. 23, no. 2, pp. 265–271, 1999.

[26] V. D. Calhoun, T. Adali, G. D. Pearlson, and J. J. Pekar, "Spatial and temporal independent component analysis of functional MRI data containing a pair of task-related waveforms," *Human Brain Mapping*, vol. 13, no. 1, pp. 43–53, 2001.

[27] J. V. Stone, J. Porrill, N. R. Porter, and I. D. Wilkinson, "Spatiotemporal independent component analysis of event-related fMRI data using skewed probability density functions," *NeuroImage*, vol. 15, no. 2, pp. 407–421, 2002.

[28] A. J. Bell and T. J. Sejnowski, "An information-maximization approach to blind separation and blind deconvolution," *Neural Computation*, vol. 7, no. 6, pp. 1129–1159, 1995.

[29] F. J. Theis, P. Gruber, I. R. Keck, and E. W. Lang, "A robust model for spatiotemporal dependencies," *Neurocomputing*, vol. 71, no. 10–12, pp. 2209–2216, 2008.

[30] G. McLachlan and D. Peel, *Finite Mixture Models*, John Wiley & Sons, New York, NY, USA, 2000.

[31] L. R. Rabiner, "Tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[32] G. Xu, Y. F. Ma, H. J. Zhang, and S. Q. Yang, "An HMM-based framework for video semantic analysis," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 15, no. 11, pp. 1422–1433, 2005.

[33] A. Krogh, M. Brown, I. S. Mian, K. Sjolander, and D. Haussler, "Hidden Markov Models in computational biology applications to protein modeling," *Journal of Molecular Biology*, vol. 235, no. 5, pp. 1501–1531, 1994.

[34] R. Durbin, S. Eddy, A. Krogh, and G. Mitchison, *Biological Sequence Analysis: Probabilistic Models of Proteins and Nucleic Acids*, Cambridge University Press, 1999.

[35] D. Miller, T. Leek, and R. Schwartz, "A hidden markov model information retrieval system," in *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 214–221, ACM, 1999.

[36] G. D. Forney, "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973.

[37] S. Kullback and R. Leibler, "On information and sufficiency," *Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

Hindawi Publishing Corporation
Advances in Artificial Neural Systems
Volume 2012, Article ID 835903, 11 pages
doi:10.1155/2012/835903

Research Article

Hemodialysis Key Features Mining and Patients Clustering Technologies

Tzu-Chuen Lu and Chun-Ya Tseng

Department of Information Management, Chaoyang University of Technology, Wufeng District, Taichung 41349, Taiwan

Correspondence should be addressed to Tzu-Chuen Lu, tclu@cyut.edu.tw

Received 3 March 2012; Revised 4 June 2012; Accepted 8 June 2012

Academic Editor: Anke Meyer-Baese

The kidneys are very vital organs. Failing kidneys lose their ability to filter out waste products, resulting in kidney disease. To extend or save the lives of patients with impaired kidney function, kidney replacement is typically utilized, such as hemodialysis. This work uses an entropy function to identify key features related to hemodialysis. By identifying these key features, one can determine whether a patient requires hemodialysis. This work uses these key features as dimensions in cluster analysis. The key features can effectively determine whether a patient requires hemodialysis. The proposed data mining scheme finds association rules of each cluster. Hidden rules for causing any kidney disease can therefore be identified. The contributions and key points of this paper are as follows. (1) This paper finds some key features that can be used to predict the patient who may has high probability to perform hemodialysis. (2) The proposed scheme applies k-means clustering algorithm with the key features to category the patients. (3) A data mining technique is used to find the association rules from each cluster. (4) The mined rules can be used to determine whether a patient requires hemodialysis.

1. Introduction

The human kidney is located on the posterior abdominal wall on both sides of the spinal column. The main functions of the kidney include metabolism control, waste and toxin excretion, regulation of blood pressure, and maintaining the body's fluid balance. All blood in the body passes through the kidney 20 times per hour. When renal function is impaired, the body's waste cannot be metabolized, which can result in back pain, edema, uremia, high blood pressure, inflammation of the urethra, lethargy, insomnia, tinnitus, hair loss, blurred vision, slow reaction time, depression, fear, mental disorders, and other adverse consequences. Furthermore, an impaired kidney will produce and secrete erythropoietin. When secretion of red blood cells is insufficient, patients will have the anemia. The kidney also helps maintain the calcium and phosphate balance in blood, such that a patient with renal failure may develop bone lesions.

When renal function is abnormal, toxins can be produced, damaging organs and possibly leading to death. To extend or save the lives of patients with impaired kidney function, kidney replacement is typically utilized, including kidney transplantation, hemodialysis (HD), and peritoneal dialysis (PD). Although kidney transplantation is the most clinically effective method, few donor kidneys are available and transplantation can be limited by the physical conditions of patients. Notably, HD can extend the lives of kidney patients.

Although medical technology is mature, factors causing diseases are changing due to changing environments. Any factor may potentially lead to disease. When the detection index of a patient exceeds the standard and kidney disease has been diagnosed, patients must go the hospital for kidney replacement therapy. For instance, a doctor may recommend that high-risk patients adjust their habits by, say, stopping smoking, controlling blood pressure, maintaining normal

urination, controlling urinary protein levels, maintaining normal sleeping patterns, controlling blood sugar levels, reducing the use of medications, avoiding reductions in the body's resistance, maintaining low body fat levels, and reducing the burden on the kidneys.

However, improving one's physical condition and diet are insufficient. To control one's physical condition, periodic health examinations at a hospital have become a common disease-prevention strategy. Doctors may offer advice to patients based on health examination results to reduce disease risk.

Many scholars have applied data mining techniques for disease prediction. These techniques include clustering, association rules, and time-series analysis. Different analyses may require different mining techniques. Selection of an appropriate mining technique is the key to obtaining valuable data. However, choosing a data mining technique is very difficult for general hospitals, especially when dealing with different forms of original data. Therefore, to help medical professionals identify hidden factors that cause kidney diseases, this work applies a novel hemodialysis system (HD system). The HD system may identify factors not previously known.

General medical staff may perform routine examinations for particular factors associated with a particular disease and ignore other factors that may be associated with other diseases, such as kidney diseases. For example, staff may only assess blood urea nitrogen (BUN) and creatinine (CRE) levels and CRE clearance (CC). However, increasing amounts of data indicate that some hidden rules and relationships may exist. Therefore, this work uses an entropy function to identify key features related to HD. By identifying these key features, one can determine whether a patient requires HD. This work uses these key features as dimensions in cluster analysis. When patients requiring HD are classified into the same group, and the other patients are classified into the other group, the key features can effectively determine whether a patient requires HD. The proposed data mining scheme finds association rules of each cluster. Hidden rules for causing any kidney disease can therefore be identified.

2. Literature Review

2.1. Hemodialysis. Hemodialysis is also called dialysis. An artificial kidney discharges uremic toxins and water to eliminate uremic symptoms. In an HD system, a semipermeable membrane separates the blood and dialysate. The human blood continues passing through on one side of an artificial kidney and the dialysate carries away uremic toxins on the other side. Finally, the cleaned blood will back into the body. This continuous cycle eventually purifies blood.

A doctor may recommend that patient undergo dialysis according to the difference between acute and chronic. If kidney failure is acute, the doctor will recommend that the patient undergo dialysis before the occurrence of uremic toxins accumulate. For chronic kidney failure, medical treatment is first utilized and HD may be initiated after uremia occurs. Additionally, a doctor may assess according to the causes of kidney failure, kidney size, anemic state, degradation of kidney function, and recovery. Moreover, each examination indicator will be assessed. The most commonly used indicators are BUN concentration, CRE concentration, CC, urine-specific gravity, and osmotic pressure [1, 2].

2.1.1. Blood Urea Nitrogen (BUN). Blood urea nitrogen is the metabolite of proteins and amino acids excreted by the kidneys. The BUN concentration in blood can be used to determine whether kidney function is normal. The normal BUN range is 10–20 mg/dL. If the BUN concentration exceeds 20 mg/dL, this is called high azotemia. However, the BUN concentration may increase temporarily because of dehydration, eating large amounts of high-protein foods, upper gastrointestinal bleeding, severe liver disease, infection, steroid use, and impaired kidney blood flow. When the BUN concentration is high and the CRE concentration is normal, kidney function is normal. Although the BUN concentration can be used as an indicator of kidney function, it is not as accurate as the CRE concentration and CC.

2.1.2. Creatinine (CRE). Creatinine is mainly a metabolite of muscle activity and daily production is excreted through the kidneys. Daily CRE production cannot be fully excreted and the CRE concentration increases when TRY kidney function is impaired. As the CRE concentration increases, kidney function decreases. Because CRE is a waste generated by muscle metabolism, the CRE concentration is associated with the total amount of muscle or weight but is not related to diet or water intake. The CRE concentration may reflect kidney function more accurately than the BUN concentration. When the CRE concentration is in the normal range, it does mean that kidney function is normal; that is, CC is a better tool when assessing kidney function. The compensatory capacity of the kidney is large. For example, although the CRE concentration may increase from 1.4 mg/dL to 1.5 mg/dL, kidney function may have declined by more than 50%.

2.1.3. Creatinine Clearance (CC). Creatinine clearance is widely used and is an accurate estimation of kidney function. Creatinine Clearance is the amount of CRE cleared per minute. The CC for a healthy person is 80–120 mL/min; the average is 100 mL/min. Kidney failure is minor when the CC is 50–70 mL/min and moderate when CC is only 30–50 mL/min. If CC is <30 mL/min, kidney failure is severe and uremic symptoms will develop gradually. When CC is <10 gradually, a patient must start dialysis. By collecting all the urine produced within 24 hours, CC can be determined easily. Notably, CC is derived as follows:

$$\mathrm{CC} = \frac{\text{Urine CRE}_{\text{concentration}}\ (\text{mg\%}) \times 24\,\text{hours urine volume (c.c.)}}{\text{Blood CRE}_{\text{concentration}}\ (\text{mg\%}) \times 1440\,(\text{minutes})}. \quad (1)$$

2.1.4. Urine-Specific Gravity and Osmotic Pressure. Urine-specific gravity and osmotic pressure reflects the ability of the kidney to concentrate urine. If the specific gravity of urine is ≤1.018 or each urine-specific gravity gap is ≤0.008, the ability of the kidney to concentrate urine is impaired. Moreover, the ratio of osmolality to blood osmotic pressure must exceed 1.0; otherwise, the ability of the kidney to concentrate urine is impaired. If the ratio of urine to blood osmotic pressure is ≤3 after water fasting for 12 hours, the ability of the kidney to concentrate urine is impaired. Abnormal urine concentration function usually occurs in patients with analgesic nephropathy.

Doctors recommend patients undergo dialysis when their BUN concentration exceeds 90 mg/dL, the CRE concentration exceeds 9 mg/dL, and CC is <0.17 mL/sec, or the CRE concentration exceeds 707.2 mg/dL. However, when the BUN concentration begins increasing, the kidney is very fragile. That is, the kidney that has been damaged exceeds 1/3 when HD is required [3]. Thus, indexes such as the albumin globulin ratio (A/G ratio) of kidney function (Table 1), red blood cell (RBC) count in blood tests (Table 2), or white blood cell (WBC) count by urinalysis (Table 3) are related to kidney function [1]. This work proposes an effective scheme that identifies unknown key features to predict HD. This work uses the entropy function to identify key features that are strongly related to HD and applies the k-means clustering algorithm to these key features to group patients.

Hung proposed an association rule mining with multiple minimum supports for predicting hospitalization of HD patients [4]. Hung used this association rule to analyze factors that may lead to HD to reduce the number of patients hospitalized for kidney impairment.

Hung relied on routinely examined HD indexes for patients per month, including BUN, CRE, uric acid (UA), natrium (Na), potassium (K), calcium (Ca), phosphate (IP), and alkaline phosphatase levels and analyzed 667 derived variables, such as protein ratio, to determine whether monocytes infected or a patient was undernourished. Hung obtained 9 rules from 5,793 records. For instance, diabetic patients with high cholesterol levels were hospitalized most. Inadequate dialysis was a high risk factor for hospitalization. If patient is female, aged 40–49, infected with monocytes, and had a recent hemoglobin (Hb/Ht) test value that was too low, the frequency of hospitalization was high. If hematocrit (Ht) was abnormal twice in the last three months, average platelet volume (MPV) was abnormal twice, and total protein (TP) was abnormal once, the probability of hospitalization was 93%. If TP, glutamic oxaloacetic transaminase (GOT), and glutamic pyruvic transaminase (GPT) of patients were abnormal twice in the last three months and uric acid was also abnormal, hospitalization risk was 100%.

Huang analyzed risk of mortality for patients on long-term HD in 2009 [5]. Huang used the Classification and Regression Tree, Mann-Whitney *U* Test, Chi-square Test, Pearson Correlation, and the Nomogram to analyze 992 patients on long-term HD. Albumin level and age were the factors most strongly related to mortality. Huang clustered and analyzed patients. If a patient had good nutrition and was young, mortality of diabetic patients was 5.45 times that of nondiabetic patients. However, if a patient was malnourished and older, albumin and CRE levels were the factors most strongly related to mortality. Thus, albumin

Table 1: Kidney function test features.

Kidney function test items		Reference	Units
Blood urea nitrogen	BUN	5–25	mg/dL
Creatinine	CRE	0.3–1.4	mg/dL
Uric acid	UA	2.5–7.0	mg/dL
Albumin-globulin in ratio	A/G ratio	1.0–1.8	
Creatinine clearance/24 hrs urine	CC	M: 71–135 F: 78–116	mL/min
Renin	Penin	0.15–3.95	pg/mL/hr
Creatinine urine	Creatinine urine	60–250	mg/dL
Natrium	Na	135–145	meq/L
Potassium	K	3.4–4.5	meq/L
Calcium	Ca	8.4–10.6	mg/dL
Phosphorus	IP	2.1–4.7	mg/dL
Alkaline phosphatase	ALP	27–110	U/L

Table 2: Blood test features.

Blood test items		Reference	Units
Hemoglobin	Hb	M: 14–18 F: 12–16	g/dL
Red blood cell	RBC	M: 450–600 F: 400–550	mil/mm^3
White blood cell	WBC	5000–10000	mm^3
Hematocrit	Hct	M: 40–55 F: 37–50	%
Platelets	PLT	15–40.0	10^3/uL
Mean corpuscular volume	MCV	83–100	u^3
Mean corpuscular hemoglobin	MCH	27–32.5	uug
Mean corpuscular hemoglobin concentration	MCHC	32–36	%
Reticulocyte	Reticulocyte	0.5–2.0	%
Malaria	Malaria	(−)	
Erythrocyte sedimentation Rate.	ESR	M: 1–15 F: 1–20	mm/hr
Differential count	DC		
Band	Band	0–2	%
Neutrophils	Neutrophils	50–70	%
Lymphocytes	Lymphocytes	20–40	%
Monocytes	Monocytes	2–6	%
Eosinophils	Eosinophils	1–4	%
Basophils	Basophils	0–1	%
Bleeding times	BT	0–3	Minute
Coagulation times	CT	2–6	Minute
Blood type	Blood type		
Rhesus factor	Rh Factor	(+)	
Blood pressure	BP		mm/Hg
Height	Height		cm
Weight	Weight		kg

Table 3: Urine test features.

Urine test items		Reference	Units
Color/appearance	Color/appearance		
Reaction pH	Reaction PH	5.5–8.5	
Protein	Protein	<(+)	mg/mL
Sugar	Sugar	(−)	g/dL
Bilirubin	BIL	(−)	
Urobilinogen	URO	≤1; 4	umol/L
Urine red blood cells	RBC	0–3	/HPF
Urine white blood cells	WBC	0–5	/HPF
Pus cell	Pus cell	0-1	/HPF
Epith cell	Epith cell	M: 0–3 F: 0–15	/HPF
Casts	Casts	Not found	/LPF
Ketones	Ketones	(−)	mmol/L
Crystals	Crystals	− ~ (±)	/LPF
Bacteria and other	Bacteria and other	−	/HPF

level, age, diabetes status, and CRE level can help predict risk of mortality.

Yeh et al. used a data mining technique to predict hospitalization of HD patients in 2011 [6]. The availability of medical resources and dialysis quality may decline when too many patients are admitted to a hospital. Therefore, Yeh et al. used analysis of the C4.5 decision tree and the multiple minimum support (MS) association rule mining technology for analysis. The C4.5 decision tree was used to eliminate null values and association rule mining was used to identify hospitalization of HD patients. According to the records of hospitalized patients, hospitalized patients seldom have a chronic disease or may not have a chronic disease, but doctors only determine whether a patient should be hospitalized during an examination.

Lin used hospital records of patients combined with the association rule and the time-series analysis to establish a health-management information system for chronic diseases [7]. Lin found that occluded cerebral arteries may lead to cerebral thrombosis and a cerebral embolism. After examination by a doctor, the rule is effective in avoiding a second stroke. Additionally, ill-defined heart diseases still require improvement. Lin used data mining to provide the chronic disease patients' family members and medical staffs for controlling their disease.

These scholars usually used well-known blood tests as mining rules. This work uses an effective and novel scheme to identify some previously unknown features to predict HD. The entropy function is applied to identify features that are strongly related to HD, and the k-means clustering algorithm is applied with these key features to group patients.

2.2. Entropy Function. Information gain, proposed by Quinlan in 1979 [8], is a basis of the decision tree constructed by Interactive Dichotomiser 3 (ID3). Information gain can also be utilized to determine differences in feature attributes and other classification attributes. Further, it is usually used to select the split point of ID3.

We assume a classification problem that includes N data records, m feature dimensions, and k clusters. The measurement of a single feature's information gain must be determined based on two correlated values, called entropy; the difference between two correlated values is called information entropy

$$\text{Entropy}(N) = \sum P_t \times \log\left(\frac{1}{P_t}\right) = -\sum p_t \times \log(p_t), \quad (2)$$

$$\text{Entropy}\left(D_j\right) = \sum_{v=1}^{|D_j|} \frac{D_{jv}}{N} \times \text{Entropy}\left(D_{jv}\right), \quad (3)$$

$$\text{Gain}\left(D_j\right) = \text{Entropy}(N) - \text{Entropy}\left(D_j\right). \quad (4)$$

In (2), Entropy (N) is the total information content of whole problems, and this total information content is taken as a basis of single feature information gain, in which P_t is the probability of occurrence of t classification in N dataset.

In (3), Entropy (D_{jv}) is the information content of the j feature dimension, the v value, and classification and information quantity, D_{jv} is the j feature dimension, including v kinds of values, and the j feature dimension has $|D_j|$ values.

In (4), Gain (D_j) is a classification problem, the information gain received by the j feature dimension. Through (2)–(4), the information gain of each feature for a classification problem is found. This work then evaluates all threshold settings and collects the features with the greatest information gain to form a feature set for classification. Entropy is used to identify key features and cluster HD patients to determine the accuracy of key features.

2.3. Clustering Algorithm. Although many clustering techniques have been proposed, the k-means algorithm is the most representative and widely applied [9]. The k-means algorithm is also called the generalized Lloyd algorithm (GLA) [10]. The k-means algorithm transforms each data record into a data point and random numbers are utilized to generate the initial cluster center to determine which data point belongs to which cluster point. The divided data points are used to calculate the distance between a data point and the cluster center, such that a data point will belong to one cluster center when the data point is closer to one cluster center than another cluster center. The newly recomputed cluster center is the average among all data points in a cluster, and the new cluster center is taken as a basis for the next iteration. This process is repeated until no change occurs. The steps of the k-means algorithm are as follows.

(1) Use random numbers to generate the initial cluster centers $C_i = \{1, 2, \ldots, k\}$.

(2) Calculate the Euclidean distance $d(X, C_i)$ for each data point $X = \{x_1, x_2, \ldots, x_m\}$ and each cluster center C_i. The point with the shortest distance is classified in to C_i, and the distance formula is as follows:

$$d(X, C_i) = \sqrt{\sum_{j=1}^{m} \left(x_j - c_{ij}\right)^2}. \quad (5)$$

(3) Recompute the new cluster center C_i. If the movement of all data points in a cluster stop moving, all clustering work stops; otherwise, steps (1) and (2) are repeated for clustering.

2.4. Association Rule. An association rule is a widely used technique. It progressively scans a database to identify rules for the relationships between items. For instance, the probability that people will buy bread after buying milk is milk → bread (support = 50% and confidence = 100%); support means that the probability of a consumer buying both milk and bread is 50%, and confidence means that the probability of a consumer buying bread after buying milk is 100%.

Agrawal et al. developed the Apriori algorithm in 1994 [11]. The Apriori algorithm is one of the most popular data mining methods, where I is all itemsets, each data record is $X = \{x_1, x_2, \ldots, x_m\}$, and $X \subseteq I$. The expression of the association rule is $x_1 \rightarrow x_2$ (support, confidence), where $x_1 \subseteq I$, $x_2 \subseteq I$, and $x_1 \cap x_2 = \varphi$. Support and confidence affect mining results most. Support is the occupied percentage for N data records and the probability of occurrence of both x_1 and x_2 is $(x_1 \cup x_2)/N$. Confidence is the probability of x_1 and x_2 and is called a strong association rule.

First, set the threshold of minimum support and minimum confidence to generate frequently occurring items, where L_b represents frequently occurring b-itemsets, and all generated L_b frequent itemsets are combined to generate candidate itemsets. Only the support and confidence values that are greater than the minimum support and minimum confidence thresholds are retained. This process is repeated until all L_b frequent itemsets are identified.

3. Proposed Algorithms

This work applies a novel and effective scheme to find key features that predict HD. This work uses the entropy function to find the key features that are strongly related to HD and applies the k-means clustering algorithm with these key features to group patients. Furthermore, the proposed scheme applies the data mining technique to identify association rules from each cluster. These rules can be used to warn patients who may require HD. Figure 1 shows the system architecture, which is divided into four procedures.

These procedures are as follows.

(1) The input procedure, which should be handled very carefully, can determine the disease target and input various sources and formats into a database. This procedure has a marked impact on the subsequent procedure.

(2) The preprocess procedure is divided into two subprocedures. For quantitative processing, one subprocedure, data are converted into an appropriate analytical form; for example, a string form is converted into a numeric form, or a numeric form is converted into a similar spacing. For selecting features, the other subprocedure, this work uses the entropy function to find the key features that are strongly related to diseases.

(3) The mining procedure is also divided in two subprocedures. For clustering analysis, one subprocedure, the clustering algorithm is applied to these key features to group patients. For the association rule, the other subprocedure, the Apriori algorithm is applied to find the association rule in each cluster.

(4) The output procedure may express the entire mining result, and a medical professional will explain the mining result, and find any factor that may cause a disease.

3.1. Input Procedure. Examination information is from many sources, such as a hospital information system (HIS), laboratory information system (LIS), or Excel report. These different systems may have different data storage formats. For example, in the A database, gender is 1 for male and 2 for female, but in the B database, M is for male and F is for female. Thus, an error may occur while collecting data. Therefore, one should apply the preprocess process to ensure that information is correct, complete, and sufficient. The preprocess process is divided into five steps.

(1) Unified data storage format: to simplify mining, all information must be in the same format.

(2) Irrelevant data: if one does not specify the mining topic, mining efficiency and even accuracy will be adversely affected.

(3) Incorrect data: incorrect data may be caused by a source error or login error; thus, one should modify or remove.

(4) Formats do not match: to smooth information mining, information must be converted into an appropriate format when necessary.

(5) Incomplete data: incomplete data is a common problem; for example, some information may be lost, lacking for a certain period.

3.2. Preprocess Procedure. Data are standardized to improve analytical accuracy. A standard value may be applied to an item such as triglycerides (TG). If the TG level is ≥201 mg/dL, it exceeds and the standard is 100; if TG is normal it is in the range of 20–200 and the standard is 50; if TG is smaller than <19 mg/dL, it is lower than the standard and the standard is 0. If data are consecutive, a packing normalization method is used; its formula is as follows:

$$v'_j = \left\lfloor \left(\frac{v_j - \min_j}{\max_j - \min_j} \right) \times Q_j \right\rfloor, \quad (6)$$

where v_j represents raw data, $\min_j$ is the minimum value of j, $\max_j$ is the maximum value of j, v'_j is the packing normalized value, and Q_j is quantified distance. Table 4 shows example data after quantization.

Table 4 is a normalized form used to derive information gain and in association rule analysis, and it can effectively

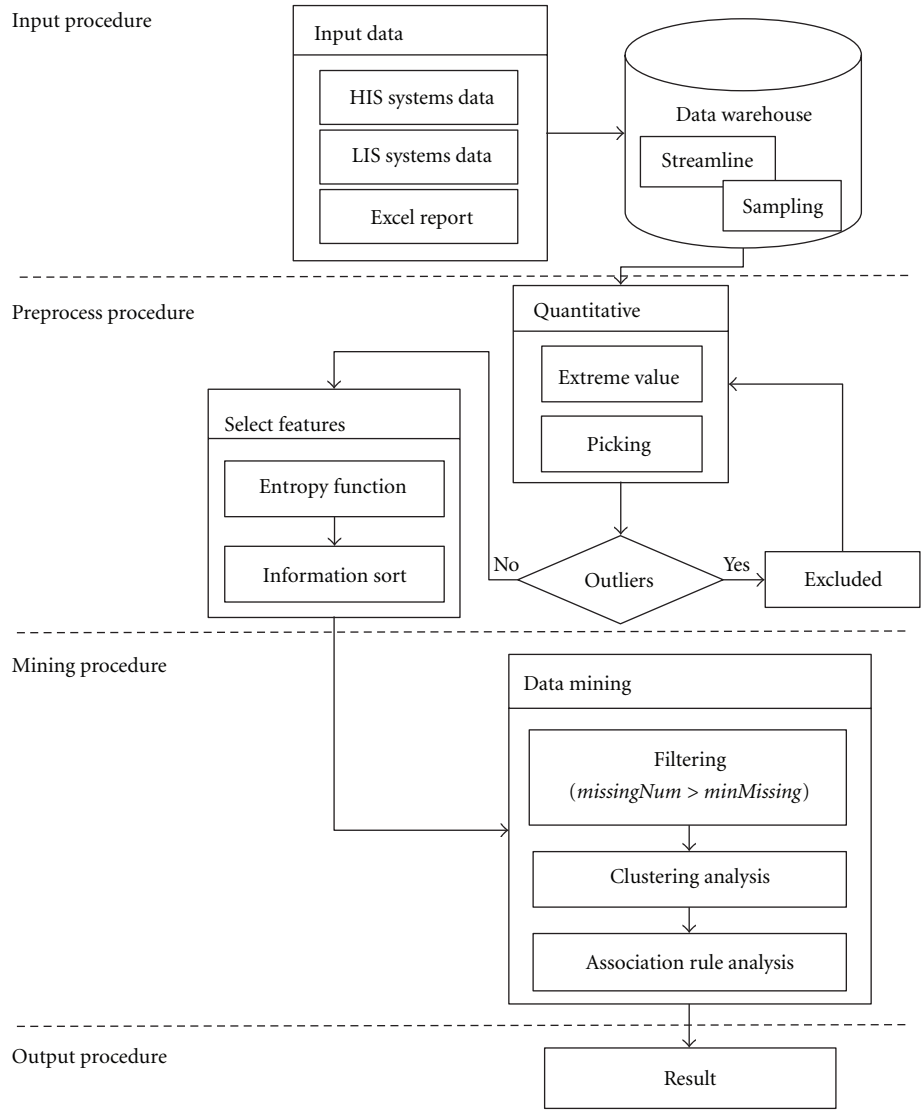


Figure 1: The system architecture.

differentiate between patients. This work simultaneously uses extreme value normalization; its formula is

$$v_j'' = \frac{v_j - \min_j}{\max_j - \min_j} \times 100, \tag{7}$$

where v_j represents raw data, $\min_j$ is the minimum value of j, $\max_j$ is the maximum value of j, and v_j'' is the packing normalized value. For instance, if the WBC value is 1, max = 10.7, and min = 3.5, then $v'' = [(1 - 3.5)/(10.7 - 3.5)] \times 100 = 38.89\%$ can be derived by applying (7).

In the entire database, the maximum and minimum values of each item markedly affect the quantification result, and the values are called outliers. If outliers exist, anomalies will also exist; for example, suppose that Q of CRE is 80, and CRE values are generally 0.37–2.99; however, a polarization datum may occur when a record is 6990. After quantization, values in the range of 0.37–2.99 will be quantified as 1, and the value recorded as 6990 will be assigned 80. Therefore, this work creates a mechanism to remove outliers. To avoid the influence of outlier values, this work sets a minNum threshold for each record. For example, assume minNum = 3 is the threshold. The total number of hemoglobin (HB), which is quantified as 2 (HB = 2), is 9; however, that of HB, which is quantified as 0 (HB = 0), is 1. This means that most data are assigned to HB = 2, and only 1 datum is assigned to HB = 0. The total number of quantified values that are smaller than minNum is the extreme value. This scheme replaces the extreme value with the average value.

3.3. Information Gain Analysis. This work uses dialysis item to identify information gain. For example, 6 patients are on dialysis (Dialysis = 1) (Table 4), the occurrence probability is $P_1 = 6/15$, and information gain is $P_1 \times \log(1/P_1) = (6/15) \times \log(6/15) = 0.528771$. When 9 patients are nondialysis (Dialysis = 0), occurrence probability is $P_0 = 9/15$, information gain is $P_0 \times \log(1/P_0) = (9/15) \times \log(9/15) = 0.442179$, and total information gain of P_0 and P_1 is 0.970951.

Table 4: Packing method normalized data.

Q_j	Sex	Age	WBC	RBC	HB	BUN	CRE	UA	GOT	GPT	TP	ALB	GLO	A/G	TG	Dialysis
	2	5	4	3	3	4	2	2	4	5	2	2	2	2	3	2
1	1	3	1	1	1	3	1	0	0	0	0	0	0	0	1	1
2	0	1	3	1	0	0	0	0	0	1	1	0	1	0	1	1
3	0	1	1	0	0	1	0	0	1	1	0	0	0	0	1	1
4	0	2	0	1	0	0	0	0	2	0	0	0	0	0	2	0
5	1	3	1	1	1	2	1	0	3	4	1	1	1	0	1	0
6	1	1	1	1	2	1	1	1	0	0	0	0	0	0	1	1
7	0	4	2	0	0	1	1	0	0	0	1	0	1	0	1	0
8	1	1	1	1	2	3	1	0	2	4	0	0	0	1	2	1
9	1	2	0	1	2	2	1	1	1	2	0	0	0	1	1	0
10	0	2	3	1	0	2	0	1	2	1	0	0	1	0	2	0
11	1	0	1	2	2	1	0	0	1	1	1	1	1	0	1	0
12	0	0	1	1	0	3	0	0	0	0	0	0	0	0	1	0
13	0	2	1	2	0	0	0	0	0	1	1	0	1	0	1	1
14	1	2	2	0	1	1	1	1	1	0	0	1	0	1	1	0
15	1	2	3	1	2	3	1	1	1	1	0	1	0	1	1	0

Table 5: Calculation information gain of sex relative to dialysis.

Sex j	Dialysis	Count (D_{jv})	$P_{D_{jv}}$	$P_{D_{jv}} \times \log(1/P_{D_{jv}})$	Entropy (D_{jv})	Entropy (D_j)
0	0	4	4/7	0.46	0.99	0.459773
	1	3	3/7	0.52		
1	0	5	5/8	0.42	0.95	0.509031
	1	3	3/8	0.53		
			Sum			0.968804

Next, this work calculates the information gain of each item relative to dialysis item. Take Sex (Table 5) as an example. The Sex of 7 women is 0 (Sex = 0) and only 4 records with non-dialysis (Dialysis = 0), the probability is $P_{D_{jv}} = 4/7$ of Sex = 0 and Dialysis = 0, and information gain is 0.46. Three records have Sex = 0 and Dialysis = 1; thus, the probability $P_{D_{jv}} = 3/7$, and information gain is 0.52. Total information gain of 0.46 and 0.52 is 0.99. Information gain of the women is $0.99 \times (7/16) = 0.459773$ because the probability of Sex = 0 is 7/16. After summing the information gain of the women (Sex = 0) and men (Sex = 1), total information gain is 0.968804, where $0.968804 = 0.459773 + 0.509031$. Next, via (3), which is Entropy (N) − Entropy (D_j), Gain $(D_j) = 0.970951 - 0.968804 = 0.002147$.

The information gain of each item related to dialysis can be obtained and ranked, and the association rule can be mined using the top few items as key features. Take Table 6 as an example. Assume that the top three items are chosen. Thus, Age, WBC, and BUN are taken as key features.

3.4. Data Mining Procedure

3.4.1. Missing Values. Some patients may have missing values. If their records are removed directly, some import information may be lost. Thus, this work applies a second filter before data mining analysis. This research sets minMissing as the threshold and takes missingNum as a null value of each record. If missingNum > minMissing, then the record is removed. Otherwise, missingNum $\leqq$ minMissing, the record will be retained and the missing null values will be replaced by the mean value. For instance, Age, WBC, and BUN are the top three key features when records are missing records. Assume minMissing is 1. When a record for which missingNum > 1, the record is removed; otherwise, the record is retained and the missing null values are replaced by the mean value.

3.4.2. Clustering. This work uses key features for clustering, where $x_1, x_2, \ldots, x_m$ as m key features, $X = \{x_1, x_2, \ldots, x_m\}$ are patient records, x_j is a key feature in X, $1 \leq j \leq m$, and k is the cluster number. The k-means process is as follows.

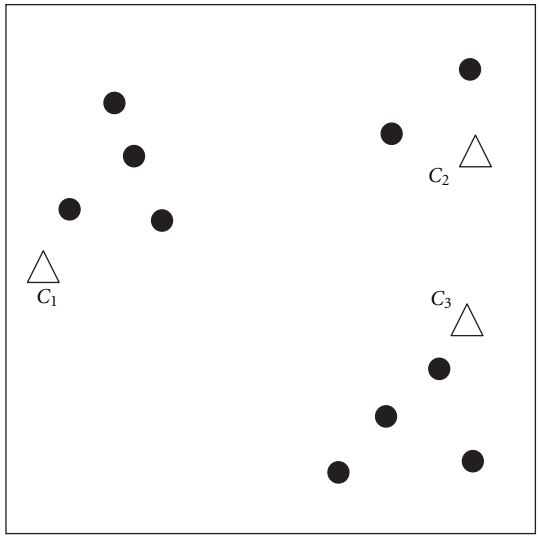
(1) First, randomly generate k initial cluster centers $C_i = \{c_1, c_2, \ldots, c_m\}$. Figure 2(a) has ten solid circles, $N = 10$, which are the locations of each record, and three triangles, $k = 3$, which are the locations of cluster centers C_i.

(2) Apply (5), $d(X, C_i) = \sqrt{\sum_{j=1}^{m} (x_j - c_{ij})^2}$, to calculate the distance between each patient's data point X and the cluster center C_i. When some X distance d_1 is less than d_i, X will be classified to C_1.

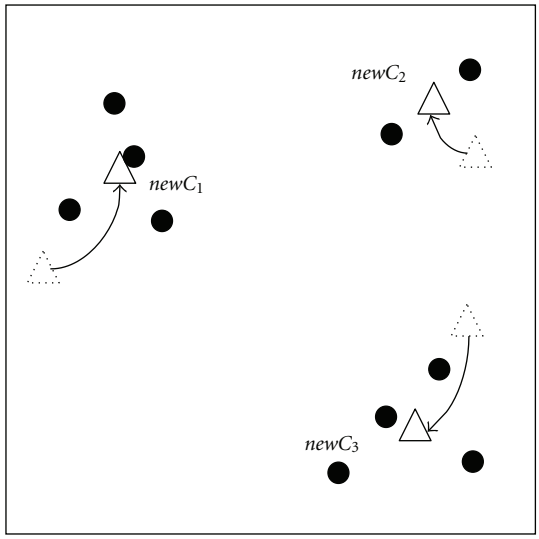
(3) Let $\widehat{C}_i = \{X_{c_i1}, X_{c_i2}, \ldots, X_{c_iS}\}$ be a cluster center membership, where S is the total number of members in C_i, and X_{c_iu} is u patient's data point in C_i. Thus,

Table 6: Information gain of each item.

Items	Sex	Age	WBC	RBC	HB	BUN	CRE	UA	GOT	GPT	TP	ALB	GLO	A/G	TG
Gain	0.002	0.577	0.329	0.14	0.06	0.28	0.05	0.09	0.18	0.2	0.05	0.24	0.02	0.06	0.03



(a) Initial dataset and cluster center (Before)



(b) Center displacement (After)

Figure 2: The diagram of clustering algorithm.

newC$_i$ will be added to the sum of X_{c_iS} in each $\widehat{C}_i$, and newC$_i$ = $\{(\sum_{u=1}^{s} x_{u1}/S), (\sum_{u=2}^{s} x_{u2}/S), \ldots, (\sum_{u=1}^{s} x_{uj}/S)\}$ can then be obtained. This function can also be taken as a new cluster center.

(4) Repeat steps (2) and (3) until each C_i remains the same.

3.4.3. Association Rule. Next, the proposed scheme finds each clustering characteristic rule using Apriori association rule analysis. We assume that the total number of records in cluster C_i is S, and each cluster membership is $\widehat{C}_i = \{X_{c_i1}, X_{c_i2}, \ldots, X_{c_iS}\}$; thus, the u patient's data point $X_{c,u}$ is in the $\widehat{C}_i$, and the j key features x_{uj} are in $X_{c_iu} = \{x_{u1}, x_{u2}, \ldots, x_{um}\}$. Next, the association rule is used to analyze each cluster C_i.

(1) First, set the values of minimum support minSup and minimum confidence minConf.

(2) Convert the normalization table into an extreme values table.

(3) Find the candidate set. We assume $\alpha = \text{item}_{jp}^{i}$, where item_{jp}^{i} is the p quantified value of the j key feature in $\widehat{C}_i$, $1 \leqq p \leqq Q$, and Sup(α) denotes the occurrence probability of item_{jp}^{i} in $\widehat{C}_i$. If Sup(α) $\geqq$ minSup, then item_{jp}^{i} becomes a candidate itemset L_Z and proceed to the next step.

(4) Through candidate set $L_z = \{\alpha_1, \alpha_2, \ldots, \alpha_y\}$, generate a set of two items, $\widehat{L}_y = \{\alpha_1 \cup \alpha_2, \alpha_1 \cup \alpha_3, \ldots, \alpha_1 \cup \alpha_y, \alpha_2 \cup \alpha_3, \ldots, \alpha_{y-1} \cup \alpha_y\}$; however, α_A and α_B cannot be the same item. Calculate the occurrence probability of each group, Sup($\alpha_A \cup \alpha_B$). If Sup($\alpha_A \cup \alpha_B$) > minSup, it becomes a member of frequent itemset L_{Z+1}.

(5) Take L_Z as a candidate set and repeat step (4) until the candidate set is null.

(6) Generate the association rule of the frequent itemset. If the confidence of the rule exceeds minConf, the rule is set up and the process is as follows.

(i) Let α^* be one of the frequent itemsets L_f, $\alpha^* = (R_A \cup R_B)$.

(ii) Generate rules $R_A \rightarrow R_B$ and $R_B \rightarrow R_A$.

In the case of A clustering C_A, where minSup = 2 and minConf = 0.5, the key features are item$_1$ = Age, item$_2$ = WBC, and item$_3$ = BUN, and $S = 7$ is the total number of records in C_A. Thus, this work finds the frequent itemsets L_1 using the minSup and minConf thresholds. The proposed scheme merges two items by L_1 as a candidate set, where j = Age, $p = 3$ in α_1, and j = WBC and $p = 1$ in α_3, and then calculates Sup($\alpha_1 \cup \alpha_3$). If Sup($\alpha_1 \cup \alpha_3$) $\geqq$ minSup, then let α_1 and α_3 be the two frequent itemsets until no more frequent itemsests are found.

Next, the quantified values are converted back into their original values if all rules are found; the formula is

$$v_j = \frac{v'_j}{Q_j} \times \left(\max_j - \min_j\right) + \min_j, \tag{8}$$

where v'_j is a quantified value, $\min_j$ is the minimum value of j, $\max_j$ is the maximum value of j, v_j is the original value,

and Q_j is a quantified interval. Take WBC = 1 → Age = 3 as an example rule. If the $\max_j$ of WBC is 10.7, the $\min_j$ value is 3.5, and Q_j is 4; then the original value of WBC = 1 is WBC $= 1/4 \times (10.7 - 3.5) + 3.5 = 5.3$. If the $\max_j$ value of Age is 68, the $\min_j$ value is 30, and Q_j is 5; then the original value of Age = 3 is Age $= 3/5 \times (68 - 30) + 30 = 52.8$. Through (8), the association rule WBC = 1 → Age = 3 can be transformed into WBC = 5.3 → Age = 52.8.

4. Experimental Results

This experiment uses health examination records provided by hospitals. The data are mainly for outpatient dialysis and general outpatients. The hospital has 105 records with many values missing. This is because each patient does not undergo all examinations. Therefore, data must first be filtered to eliminate records with missing values. This work adopts BUN and CRE, which are related to kidney function, as the first filter. If any null value occurs in BUN or CRE, the record is removed. In total, 18,166 records are retained after the first filtering.

The purpose of quantification in the preprocess procedure is to convert values into a continuity value or significant difference value from a finite interval. This work sets interval Q_j for each item based on recommendations by medical staff. Table 7 shows the intervals.

4.1. Choose Key Features. The mining result does not make sense when too many items are used. The proposed scheme uses the Entropy function to identify the top 4 key features between each item and dialysis; these features are are UA, AST (GOT), TG, and K (Blood).

4.2. Mining Procedure

4.2.1. Clustering Analysis. Based upon the above clustering algorithm, this work applies the k-means clustering algorithm with these key features to group patients. Before the experiment, records with many missing values were filtered out, leaving 7118 records. Table 8 shows the cluster grouping result. For example, 1169 patients are classified into the first group. The average indicator values are UA = 6.54, AST (GOT) = 24.48, TG = 119.79, and K (Blood) = 5.10, and the average density of the first group is 13.26. The average difference among all groups is 27.02, which is the best result of 100 random trial runs.

4.2.2. Association Rule Analysis. This work identifies the top four items related to dialysis as TG, AST (GOT), UA, and K (Blood); AST (GOT) is the main indicator of liver function. These four items are adopted as key features and the association rule technique is applied to analyze each group rule after clustering, where minSup = 35% and minConf = 65%. The association rules of the four clusters are shown in Table 9.

4.3. Summary. This work uses the clustering algorithm and the association rule algorithm to identify some previously unknown features of HD patients and possible association rules. This work then evaluates all threshold settings and collects the features with the greatest information gained to form a feature set for classification. Entropy is used to identify key features and cluster HD patients to determine the accuracy of key features. During the clustering process, the clustering algorithm is applied on these key features to group patients, and the entropy function can effectively determine clustering analysis with the key features. Furthermore, this work applies the apriori algorithm to find the association rules of each cluster. Hidden rules for causing any kidney disease can therefore be identified.

Table 7: Each interval of item.

ID	Item	Interval
1	TG	50
2	AST (GOT)	20
3	Ch	50
4	ALT (GPT)	20
5	UA	2
6	K (Boold)	2
7	BUN	5
8	Amylase (B)	50
...	...	...

Table 8: Clustering results.

Cluster	UA	AST (GOT)	TG	K (Blood)	Density
Cluster-1	6.54	24.48	119.72	5.10	14.14
Cluster-2	6.16	30.12	138.92	3.92	11.59
Cluster-3	4.47	24.72	112.33	4.07	11.22
Cluster-4	8.40	28.03	228.72	4.20	20.91

This experiment adopts the health examination records provided by one general hospital of Taiwan. During the experiment process, the experimental results will be discussed with medical staffs. From the experimental results, we can find that if BUN is in the range of 58.5–61.5 (60 ± 1.5) and Na (Blood) is in the range of 137.5–140.25 (140 ± 2.5), patients have a high risk of receiving a dialysis. The BUN is reported to be a reliable indicator of high risk, but the Na (Blood) is not clearly defined. Therefore, the Na (Blood) needs for further analysis and clarification. Conversely, if UA is in the range of 6.25–6.75 (6.5 ± 0.25), TG is in the range of 134.75–184.75 (159.75 ± 25), and K (Blood) is in the range of 3.89–4.39 (4.14 ± 0.25), or AC-GLU is in the range of 111–161 (136 ± 25), patients have a low risk of receiving a dialysis.

The medical staffs express that the UA, TG, and AC-GLU will definitely affect the possibility of patients to receive a dialysis, but K (Blood) is not clearly defined to create an influence on patients. The factor should be further analysis. At last, there is one more special feature, AST (GOT) because it appears both in the groups of high risk and low risk. The medical staffs express, actually AST (GOT) is not directly related to HD. Thus, AST (GOT) is not a key factor to determine whether a patient requires HD.

Table 9: Association rule of each cluster-k.

α^*	Sup (α^*)	Conf.
Cluster-1 ($k = 1$)		
BUN = 60 ± 1.5 → Dialysis = Yes	487	91%
Dialysis = Yes → AST (GOT) = 24.5 ± 10	708	74%
AST (GOT) = 24.5 ± 10 → Dialysis = Yes	523	73%
Na (Blood) =140 ± 2.5 → Dialysis = Yes	455	70%
Dialysis = Yes → BUN = 60 ± 1.5	487	69%
Na (Blood) = 140 ± 2.5 → AST (GOT) = 24.5 ± 10	434	66%
Cluster-2 ($k = 2$)		
CRE = 0.85 ± 0.15 → Dialysis = No	487	91%
UA = 6.5 ± 0.25 TG = 159.75 ± 25 → Dialysis = No	1341	97%
AC-GLU = 136 ± 25 → Dialysis = No	1265	94%
TG = 159.75 ± 25 → Dialysis = No	1696	93%
UA = 6.5 ± 0.25 → Dialysis = No	1920	93%
AST (GOT) = 45 ±10 → Dialysis = No	1479	92%
K (Boold) = 4.14 ± 0.25 → Dialysis = No	1938	91%
TG = 159.75 ± 25 Dialysis = No → UA = 6.5 ± 0.25	1341	79%
TG = 159.75 ± 25 → UA = 6.5 ± 0.25	1378	76%
TG =159.75 ± 25 → UA = 6.5 ± 0.25 Dialysis = No	1341	74%
UA = 6.5 ± 0.25 Dialysis = No → TG = 159.75 ± 25	1341	70%
UA = 6.5 ± 0.25 → TG = 159.75 ± 25	1378	67%
UA = 6.5 ± 0.25 → TG = 159.75 ± 25 Dialysis = No	1341	65%
CRE = 0.85 ± 0.15 →Dialysis = No	487	91%
UA = 6.5 ± 0.25 TG = 159.75 ± 25 → Dialysis = No	1341	97%
Cluster-3 ($k = 3$)		
CRE = 0.85 ± 0.15 → Dialysis = No	732	100%
CRE = 0.85 ± 0.15 K (Boold) = 5 ± 0.25 → Dialysis = No	560	100%
K (Boold) = 4.14 ± 0.25 → Dialysis = No	910	95%
AST (GOT) = 24.5 ± 10 K (Boold) = 4.14 ± 0.25 → Dialysis = No	507	94%
AST (GOT) = 24.5 ± 10 → Dialysis = No	505	92%
AST (GOT) = 24.5 ± 10 → Dialysis = No	679	86%
CRE = 0.85 ± 0.15 → K (Boold) = 4.14 ± 0.25	560	77%
CRE = 0.85 ± 0.15 Dialysis = No → K (Boold) = 4.14 ± 0.25	560	77%
CRE = 0.85 ± 0.15 → K (Boold) = 4.14 ± 0.25 Dialysis = No	560	77%
AST (GOT) = 24.5 ± 10 Dialysis = No → K (Boold) = 4.14 ± 0.25	507	75%
Dialysis = No → K (Boold) = 4.14 ± 0.25	910	74%
AST (GOT) = 24.5 ± 10 → K (Boold) = 4.14 ± 0.25	539	68%
Cluster-4 ($k = 4$)		
AST (GOT) = 45 ± 10 K (Boold) = 4.14 ± 0.25 → Dialysis = No	364	98%
K (Boold) = 4.14 ± 0.25 → Dialysis = No	503	91%
AST (GOT) = 45 ± 10 → Dialysis = No	537	90%
K (Boold) = 4.14 ± 0.25 Dialysis = No → AST (GOT) = 45 ± 10	364	72%
Dialysis = No → AST (GOT) = 45 ± 10	537	71%
AST (GOT) = 45 ± 10 Dialysis = No → K (Boold) = 4.14 ± 0.25	364	68%
K (Boold) = 4.14 ± 0.25 → AST (GOT) = 45 ± 10	372	68%
Dialysis = No → K (Boold) = 4.14 ± 0.25	503	67%
K (Boold) = 4.14 ± 0.25 → AST (GOT) = 45 ± 10 Dialysis = No	364	66%

5. Conclusion

Medical staffs try to find some information from patient's health examination records to reduce the occurrence of disease. However, some hidden information may be ignored because of the human observation or the restriction of book. Although there are many data mining techniques that have been proposed, most of them are focused on some known items. Seldom techniques in regard with searching for hidden key features are proposed. The reason is because the examination items are too many but incomplete. It is hard to find out the association rule by using system.

This research will help medical staffs to find some unknown key features to predict the hemodialysis. We apply k-means clustering algorithm with these key features to group the patients. Furthermore, the proposed scheme applies data mining technique to find the association rule from each cluster. The rules can help the patients to detect any occurrence possibility of disease.

Acknowledgment

The authors would like to thank the National Science Council of the Republic of China, Taiwan, for financially supporting this paper under Contract no. NSC 99-2622-E-324-006-CC3.

References

[1] DrKao, "Normal Test Values," 2010, http://www.drkao.com/1st_site/health_wap/normal_main.htm.

[2] Green Cross, "How to Detect Renal Function," 2010, http://www.greencross.org.tw/kidney/symptom_sign/kid_func.html.

[3] Shin Kong Wu Ho-Su Memorial Hospital, 2010, http://www.skh.org.tw/mnews/178/4-2.htm.

[4] K. C. Hung, *Multiple minimum support association rule mining for hospitalization prediction of hemodialysis patients [M.S. thesis]*, Computer Science and Information Engineering, 2004.

[5] S. Y. Huang, *The evaluation & analysis of the risk of mortality for patients receiving long-term hemodialysis proposal [M.S. thesis]*, Graduate Institute of Biomedical Informatics, 2009.

[6] J. Y. Yeh, T. H. Wu, and C. W. Tsao, "Using data mining techniques to predict hospitalization of hemodialysis patients," *Decision Support Systems*, vol. 50, no. 2, pp. 439–448, 2011.

[7] Y. J. Lin, *Applying data mining in health management information system for chronic desease [M.S. thesis]*, Department of Computer Science and Information Management, 2008.

[8] J. R. Quinlan, "Induction of decision trees," *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986.

[9] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, "An efficient k-means clustering algorithms: analysis and implementation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 881–892, 2002.

[10] J. Z. C. Lai, T. J. Huang, and Y. C. Liaw, "A fast k-means clustering algorithm using cluster center displacement," *Pattern Recognition*, vol. 42, no. 11, pp. 2551–2556, 2009.

[11] R. Agrawal, R. Srikant, H. Mannila et al., "Fast discovery of association rules," in *Advances in Knowledge Discovery and Data Mining*, pp. 307–328, 1996.

Hindawi Publishing Corporation
Advances in Artificial Neural Systems
Volume 2012, Article ID 107046, 9 pages
doi:10.1155/2012/107046

Research Article

Sleep Stage Classification Using Unsupervised Feature Learning

Martin Längkvist, Lars Karlsson, and Amy Loutfi

Center for Applied Autonomous Sensor Systems, Örebro University, 701 82 Örebro, Sweden

Correspondence should be addressed to Amy Loutfi, amy.loutfi@oru.se

Received 17 February 2012; Revised 5 May 2012; Accepted 6 May 2012

Academic Editor: Juan Manuel Gorriz Saez

Most attempts at training computers for the difficult and time-consuming task of sleep stage classification involve a feature extraction step. Due to the complexity of multimodal sleep data, the size of the feature space can grow to the extent that it is also necessary to include a feature selection step. In this paper, we propose the use of an unsupervised feature learning architecture called *deep belief nets* (DBNs) and show how to apply it to sleep data in order to eliminate the use of handmade features. Using a postprocessing step of hidden Markov model (HMM) to accurately capture sleep stage switching, we compare our results to a feature-based approach. A study of anomaly detection with the application to home environment data collection is also presented. The results using raw data with a deep architecture, such as the DBN, were comparable to a feature-based approach when validated on clinical datasets.

1. Introduction

One of the main challenges in sleep stage classification is to isolate features in multivariate time-series data which can be used to correctly identify and thereby automate the annotation process to generate sleep hypnograms. In the current absence of a set of universally applicable features, typically a two-stage process is required before training a sleep stage algorithm, namely, feature extraction and feature selection [1–9]. In other domains which share similar challenges, an alternative to using hand-tailored feature representations derived from expert knowledge is to apply unsupervised feature learning techniques, where the feature representations are learned from unlabeled data. This not only enables the discovery of new useful feature representations that a human expert might not be aware of, which in turn could lead to a better understanding of the sleep process and present a way of exploiting massive amounts of unlabeled data.

Unsupervised feature learning and in particular deep learning [10–15] propose ways for training the weight matrices in each layer in an unsupervised fashion as a preprocessing step before training the whole network. This has proven to give good results in other areas such as vision tasks [10], object recognition [16], motion capture data [17], speech recognition [18], and bacteria identification [19].

This work presents a new approach to the automatic sleep staging problem. The main focus is to learn meaningful feature representations from unlabeled sleep data. A dataset of 25 subjects consisting of electroencephalography (EEG) of brain activity, electrooculography (EOG) of eye movements, and electromyography (EMG) of skeletal muscle activity is segmented and used to train a deep belief network (DBN), using no prior knowledge. Validation of the learned representations is done by integrating a hidden Markov model (HMM) and compare classification accuracy with a feature-based approach that uses prior knowledge. The inclusion of an HMM serves the purpose of improving upon capturing a more realistic sleep stage switching, for example, hinders excessive or unlikely sleep stage transitions. It is in this manner that the knowledge from the human experts is infused into the system. Even though the classifier is trained using labeled data, the feature representations are learned from unlabeled data. The architecture of the DBN follows previous work with unsupervised feature learning for electroencephalography (EEG) event detection [20].

A secondary contribution of the proposed method leverages the information from the DBN in order to perform

anomaly detection. Particularly, in light of an increasing trend to streamline sleep diagnosis and reduce the burden on health care centers by using at home sleep monitoring technologies, anomaly detection is important in order to rapidly assess the quality of the polysomnograph data and determine if the patient requires another additional night's collection at home. In this paper, we illustrate how the DBN once trained on datasets for sleep stage classification in the lab can still be applied to data which has been collected at home to find particular anomalies such as a loose electrode.

Finally, inconsistencies between sleep labs (equipment, electrode placement), experimental setups (number of signals and categories, subject variations), and interscorer variability (80% conformance for healthy patients and even less for patients with sleep disorder [9]) make it challenging to compare sleep stage classification accuracy to previous works. Results in [2] report a best result accuracy of around 61% for classification of 5 stages from a single EEG channel using GOHMM and AR coefficients as features. Works by [8] achieved 83.7% accuracy using conditional random fields with six power spectra density features for one EEG signal on four human subjects during a 24-hour recording session and considering six stages. Works by [7] achieved 85.6% accuracy on artifact-free, two expert agreement sleep data from 47 mostly healthy subjects using 33 features with SFS feature selection and four separately trained neural networks as classifiers.

The goal of this work is not to replicate the R&K system or improve current state-of-the-art sleep stage classification but rather to explore the advantages of deep learning and the feasibility of using unsupervised feature learning applied to sleep data. Therefore, the main method of evaluation is a comparison with a feature-based shallow model. Matlab code used in this paper is available at http://aass.oru.se/~mlt.

2. Deep Belief Networks

DBN is a probabilistic generative model with deep architecture that searches the parameter space by unsupervised greedy layerwise training. Each layer consists of a restricted Boltzmann machine (RBM) with visible units, **v**, and hidden units, **h**. There are no visible-visible connections and no hidden-hidden connections. The visible and hidden units have a bias vector, **c** and **b**, respectively. The visible and hidden units are connected by a weight matrix, **W**, see Figure 1(a). A DBN is formed by stacking a user-defined number of RBMs on top of each other where the output from a lower-level RBM is the input to a higher-level RBM, see Figure 1(b). The main difference between a DBN and a multilayer perceptron is the inclusion of a bias vector for the visible units, which is used to reconstruct the input signal, which plays an important role in the way DBNs are trained.

A reconstruction of the input can be obtained from the unsupervised pretrained DBN by encoding the input to the top RBM and then decoding the state of the top RBM back to the lowest level. For a Bernoulli (visible)-Bernoulli (hidden) RBM, the probability that hidden unit h_j is activated given

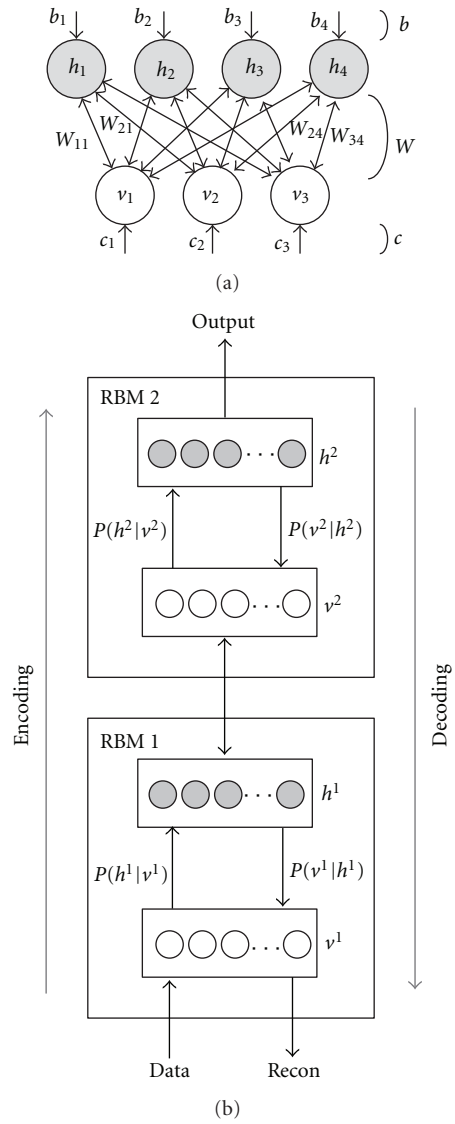


Figure 1: Graphical depiction of (a) RBM and (b) DBN.

visible vector, **v**, and the probability that visible unit v_i is activated given hidden vector, **h**, are given by

$$P\left(h_j \mid \mathbf{v}\right) = \frac{1}{1+\exp^{b_j+\sum_i W_{ij}v_i}}$$
$$P(v_i \mid \mathbf{h}) = \frac{1}{1+\exp^{c_i+\sum_j W_{ij}h_j}}. \quad (1)$$

The energy function and the joint distribution for a given visible and hidden vector are

$$E(\mathbf{v},\mathbf{h}) = \mathbf{h}^T\mathbf{W}\mathbf{v} + \mathbf{b}^T\mathbf{h} + \mathbf{c}^T\mathbf{v}$$
$$P(\mathbf{v},\mathbf{h}) = \frac{1}{z}\exp^{E(\mathbf{v},\mathbf{h})}. \quad (2)$$

The parameters **W**, **b**, and **v** are trained to minimize the reconstruction error. An approximation of the gradient of

the log likelihood of $\mathbf{v}$ using contrastive divergence [21] gives the learning rule for RBM:

$$\frac{\partial \log P(\mathbf{v})}{\partial W_{ij}} \approx \left\langle v_i h_j \right\rangle_{\text{data}} - \left\langle v_i h_j \right\rangle_{\text{recon}}, \tag{3}$$

where $\langle \cdot \rangle$ is the average value over all training samples. In this work, training is performed in three steps: (1) unsupervised pretraining of each layer, (2) unsupervised fine-tuning of all layers with backpropagation, and (3) supervised fine-tuning of all layers with backpropagation.

3. Experimental Setup

3.1. Automatic Sleep Stager. The five sleep stages that are at focus are awake, stage 1 (S1), stage 2 (S2), slow wave sleep (SWS), and rapid eye-movement sleep (REM). These stages come from a unified method for classifying an 8 h sleep recording introduced by Rechtschaffen and Kales (R&K) [22]. A graph that shows these five stages over an entire night is called a hypnogram, and each epoch according to the R&K system is either 20 s or 30 s. While the R&K system brings consensus on terminology, among other advantages [23], it has been criticized for a number of issues [24]. Even though the goal in this work is not to replicate the R&K system, its terminology will be used for evaluation of our architecture. Each channel of the data is divided into segments of 1 second with zero overlap, which is a much higher temporal resolution than the one practiced by the R&K system.

We compare the performance of three experimental setups as shown in Figure 2.

3.1.1. Feat-GOHMM. A Gaussian observation hidden Markov model (GOHMM) is used on 28 handmade features; see the appendix for a description of the features used. Feature selection is done by sequential backward selection (SBS), which starts with the full set of features and greedily removes a feature after each iteration step. A principal component analysis (PCA) with five principal components is used after feature selection, followed by a Gaussian mixture model (GMM) with five components. The purpose of the PCA is to reduce dimensionality, and the choice of five components was made since it captured most of the variance in the data, while still being tractable for the GMM step. Initial mean and covariance values for each GMM component are set to the mean and covariance of annotated data for each sleep stage. Finally, the output from the GMM is used as input to a hidden Markov model (HMM) [25].

3.1.2. Feat-DBN. A 2-layer DBN with 200 hidden units in both layers and a softmax classifier attached on top is used on 28 handmade features. Both layers are pretrained for 300 epochs, and the top layer is fine-tuned for 50 epochs. Initial biases of hidden units are set empirically to -4 to encouraged sparsity [26], which prevents learning trivial or uninteresting feature representations. Scaling to values between 0 and 1 is done by subtracting the mean, divided by the standard deviation, and finally adding 0.5.

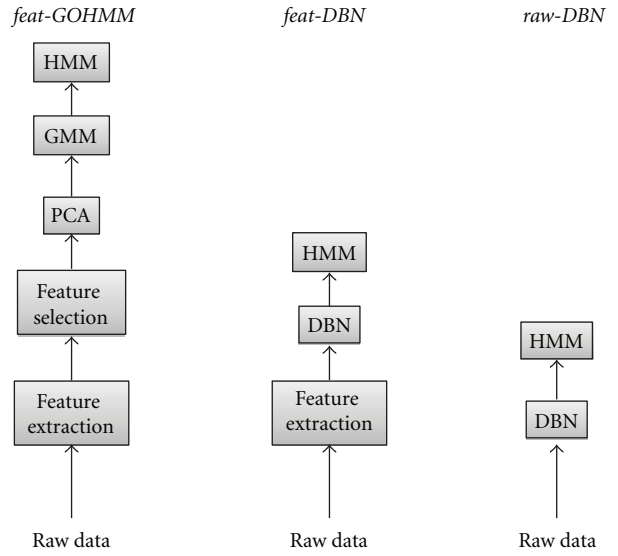


Figure 2: Overview of three setups for an automatic sleep stager used in this work. The first method, *feat-GOHMM*, is a shallow method that uses prior knowledge. The second method, *feat-DBN*, is a deep architecture that also uses prior knowledge. And, lastly, the third method, *raw-DBN*, is a deep architecture that does not use any prior knowledge. See text for more details.

3.1.3. Raw-DBN. A DBN with the same parameters as *feat-DBN* is used on preprocessed raw data. Scaling is done by saturating the signal at a saturation constant, $\text{sat}_{\text{channel}}$, then divide by $2 * \text{sat}_{\text{channel}}$, and finally adding 0.5. The saturation constant was set to $\text{sat}_{\text{EEG}} = \text{sat}_{\text{EOG}} = \pm 60\,\mu\text{V}$ and $\text{sat}_{\text{EMG}} = \pm 40\,\mu\text{V}$. Input consisted of the concatenation of EEG, EOG1, EOG2, and EMG. With window width, w, the visible layer becomes

$$v = \begin{bmatrix} \text{EEG}_1^{64} & \text{EOG1}_1^{64} & \text{EOG2}_1^{64} & \text{EMG}_1^{64} \\ \text{EEG}_{1+w}^{64+w} & \text{EOG1}_{1+w}^{64+w} & \text{EOG2}_{1+w}^{64+w} & \text{EMG}_{1+w}^{64+w} \\ \vdots & \vdots & \vdots & \vdots \end{bmatrix}. \tag{4}$$

With four signals, 1 second window, and 64 samples per second, the input dimension is 256.

3.2. Anomaly Detection for Home Sleep Data. In this work, anomaly detection is evaluated by training a DBN and calculating the root mean square error (RMSE) from the reconstructed signal from the DBN and the original signal. A faulty signal in one channel often affects other channels for sleep data, such as movement artifacts, blink artifacts, and loose reference or ground electrode. Therefore, a detected fault in one channel should label all channels at that time as faulty.

Figure 3 shows data that has been collected at a healthy patient's home during sleep. All signals, except EEG2, are nonfaulty prior to a movement artifact at $t = 7$ s. This movement affected the reference electrode or the ground electrode, resulting in disturbances in all signals for the rest of the night, thereby rendering the signals unusable by a clinician. A poorly attached electrode was the cause for

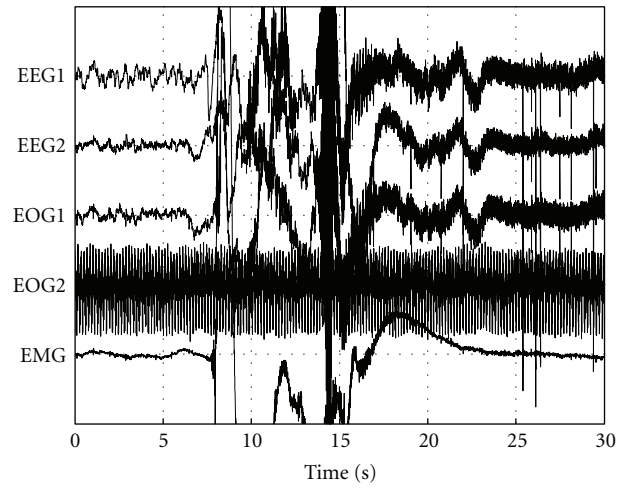


Figure 3: PSG data collected in a home environment. A movement occurs at $t = 7$ s resulting in one of the electrodes to be misplaced affecting EOG1 and both EEG channels. EOG2 is not properly attached resulting in a faulty signal for the entire night.

the noise in signal EEG2. Previous approaches to artifact rejection in EEG analysis range from simple thresholding on abnormal amplitude and/or frequency to more complex strategies in order to detect individual artefacts [27, 28].

4. Experimental Datasets

Two datasets are used in this work. The first consists of 25 acquisitions and is used to train and test the automatic sleep stager. The second consists of 5 acquisitions and is used to validate anomaly detection on sleep data collected at home.

4.1. Benchmark Dataset. This dataset has kindly been provided by St. Vincent's University Hospital and University College Dublin, which can be downloaded from PhysioNet [29]. The dataset consists of 25 acquisitions (21 males 4 females with average age 50, average weight 95 kg, and average height 173 cm) from subjects with suspected sleep-disordered breathing. Each acquisition consists of 2 EEG channels (C3-A2 and C4-A1), 2 EOG channels, and 1 EMG channel using 10–20 electrode placements system. Only one of the EEG channel (C3-A2) is used in this work. Sample rate is 128 Hz for EEG and 64 Hz for EOG and EMG. Average recording time is 6.9 hours. Sleep stages are divided into S1: 16.7%, S2: 33.3%, SWS: 12.7%, REM: 14.5%, awake: 22.7%, and indeterminate: 0.1%. Scoring was performed by one sleep expert.

All signals are preprocessed by notch filtering at 50 Hz in order to cancel out power line disturbances and down-sampled to 64 Hz after being prefiltered with a band-pass filter of 0.3 to 32 Hz for EEG and EOG, and 10 to 32 Hz for EMG. Each epoch before and after a sleep stage switch is removed from the training set to avoid possible subsections of mislabeled data within one epoch. This resulted in 20.7% of total training samples to be removed.

A 25 leave-one-out cross-validation is performed. Training samples are randomly picked from 24 acquisitions in order to compensate for any class imbalance. A total of approximately 250000 training samples and 50000 training validation samples are used for each validation.

4.2. Home Sleep Dataset. PSG data of approximately 60 hours (5 nights) was collected at a healthy patient's home using a Embla Titanium PSG. A total of 8 electrodes were used: EEG C3, EEG C4, EOG left, EOG right, 2 electrodes for the EMG channel, reference electrode, and ground electrode. Data was collected with a sampling rate of 256 Hz, which was downsampled to match the sampling rate of the training data. The signals are preprocessed using the same method as the benchmark dataset.

5. Results

5.1. Automatic Sleep Stager. A full leave-one-out cross-validation of the 25 acquisitions is performed for the three experimental setups. The classification accuracy and confusion matrices for each setup and sleep stage are presented in Tables 1, 2, 3, and 4. Here, the performance of using a DBN based approach, either with features or using the raw data, is comparable to the *feat-GOHMM*. While the best accuracy was achieved with *feat-DBN*, followed by *raw-DBN* and lastly, *feat-GOHMM*, it is important to examine the performances individually. Figure 4 shows classification accuracy for each subject. The *raw-DBN* setup gives best, or second best, performance in the majority of the sets, with the exception of subjects 9 and 22. An examination of the performance when comparing the F_1-score for individual sleep stages indicates that S1 is the most difficult stage to classify and awake and slow wave sleep is the easiest.

For the *raw-DBN*, it is also possible to analyze the learned features. In Figure 6, the learned features for the first layer are given. Here, it can clearly be seen that both low and high frequency features for the EEG and high and low amplitude features for the EMG are included, which to some degree correspond to the features which are typically selected in handmade feature selection methods.

Some conclusions from analyzing the selected features from the SBS algorithm used in *feat-GOHMM* can be made. Fractal exponent for EEG and entropy for EOG were selected for all 25 subjects and thus proven to be valuable features. Correlation between both EOG signals was also among the top selected features, as well as delta, theta, and alpha frequencies for EEG. Frequency features for EOG and EMG were excluded early, which is in accordance to the fact that these signals do not exhibit valuable information in the frequency domain [30]. The kurtosis feature was selected more frequently when it was applied to EMG and less frequently when it was applied to EEG or EOG. Features of spectral mean for all signals, median for EMG, and standard deviation for EOG were not frequently selected. See Figure 5 for errors bars for each feature at each sleep stage.

It is worth noting that variations in the number of layers and hidden units were attempted, and it was found

Table 1: Classification accuracy and F_1-score for the three experimental setups.

	Accuracy (mean ± std)	F_1-score				
		Awake	S1	S2	SWS	REM
feat-GOHMM	63.9 ± 10.8	0.71	0.31	0.72	0.82	0.47
feat-DBN	72.2 ± 9.7	0.78	0.37	0.76	0.84	0.78
raw-DBN	67.4 ± 12.9	0.69	0.36	0.78	0.83	0.58

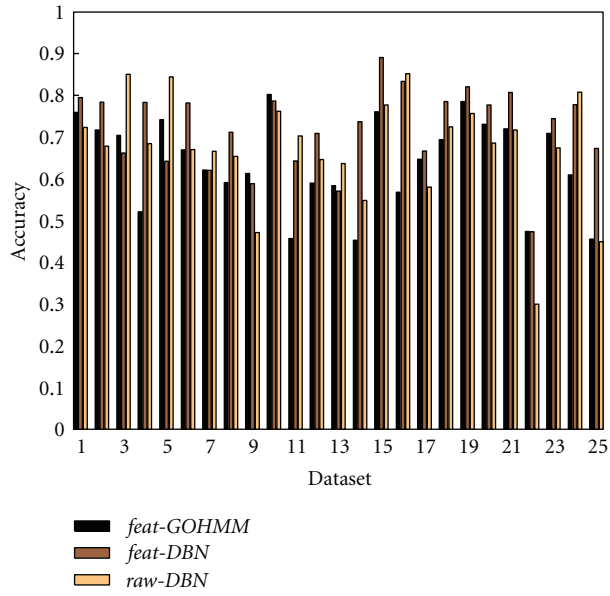


Figure 4: Classification accuracy for 25 testing sets for three setups.

Table 2: Confusion matrix for *feat-GOHMM*.

%	Classified				
	Awake	S1	S2	SWS	REM
Awake	72.5	16.8	3.0	2.5	5.2
S1	29.4	31.1	25.6	1.5	12.4
S2	2.0	8.4	71.9	7.1	10.6
SWS	1.1	1.3	9.6	87.8	0.2
REM	11.7	13.3	29.4	2.7	42.9

Table 3: Confusion matrix for *feat-DBN*.

%	Classified				
	Awake	S1	S2	SWS	REM
Awake	75.8	18.2	1.8	0.2	4.1
S1	26.0	37.6	25.1	0.7	10.6
S2	1.0	9.8	73.1	7.2	9.0
SWS	0.4	0.1	13.9	85.5	0.1
REM	1.9	4.0	10.3	0.1	83.7

Table 4: Confusion matrix for *raw-DBN*.

%	Classified				
	Awake	S1	S2	SWS	REM
Awake	68.4	13.4	2.5	0.7	15.1
S1	20.3	33.1	24.8	1.6	20.2
S2	1.0	6.3	76.5	9.1	7.1
SWS	0.1	0.0	11.1	88.8	0.0
REM	21.1	6.9	11.1	0.8	60.1

that an increase did not significantly improve classification accuracy. Rather, an increase in either the number of layers or hidden units often resulted in a significant increase in simulation time, and therefore to maintain a reasonable training time, the layers and hidden units were kept to a minimum. With the configuration of the three experimental setups described above and simulations performed on a Windows 7, 64-bit machine with quad-core Intel i5 3.1 GHz CPU with use of a nVIDIA GeForce GTX 470 GPU using GPUmat, simulation time for *feat-GOHMM*, *feat-DBN*, and *raw-DBN* were approximately 10 minutes, 1 hour, and 3 hours per dataset, respectively.

5.2. Anomaly Detection on Home Sleep Data. A total of five acquisitions were recorded at a patient's home during sleep and manually labeled into faulty or nonfaulty signals. A DBN with the *raw-DBN* setup was trained using the benchmark dataset. The root mean square error (RMSE) between the home sleep data and the reconstructed signal from the trained DBN for the five night runs and a close-up for night

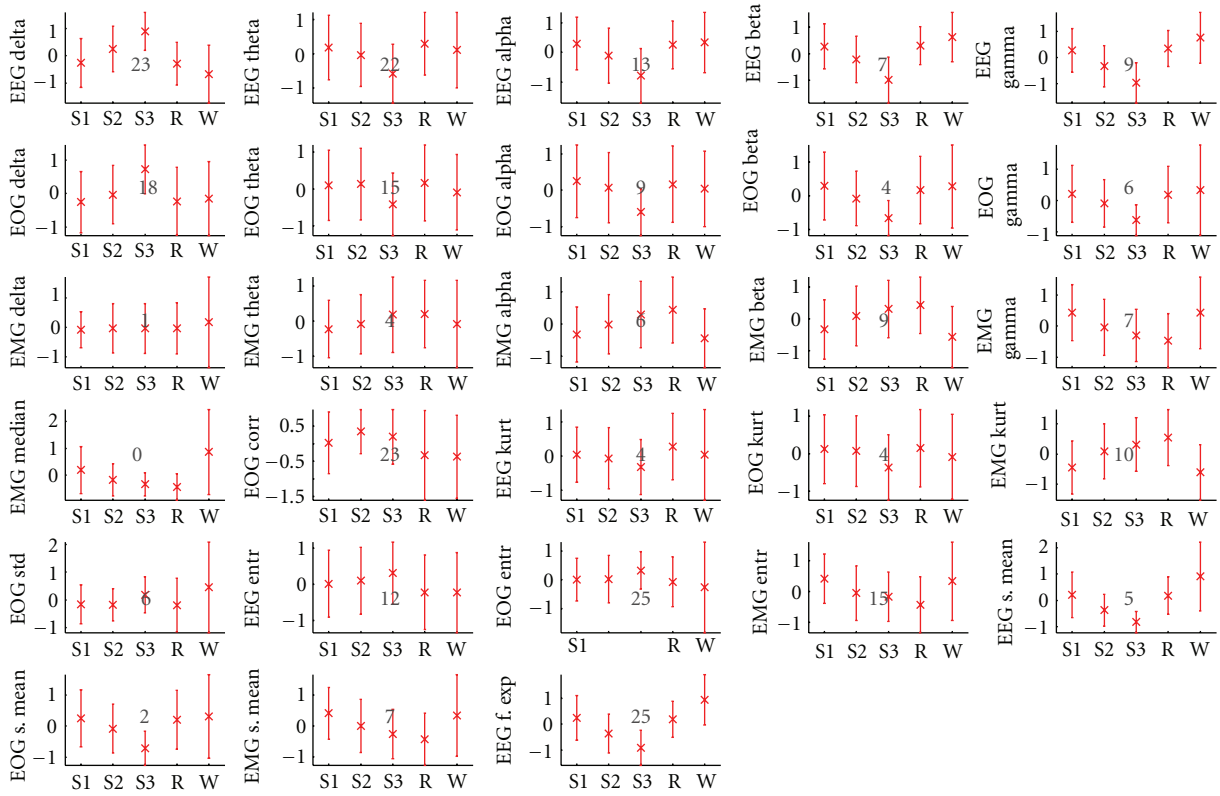


Figure 5: Error bar of the 28 features. Gray number in background represents how many times that feature was part of best subset from SBS algorithm (maximum is 25).

2 where an electrode falls off after around 380 minutes can be seen in Figure 7.

Interestingly, attempts on using the *feat-GOHMM* for sleep stage classification on the home sleep dataset resulted in faulty data to be misclassified as awake. This could be explained by the fact that faulty data mostly resembles signals in awake state.

6. Discussion

In this work, we have shown that an automatic sleep stager can be applied to multimodal sleep data without using any handmade features. We also compared the reconstructed signal from a trained DBN on data collected in a home environment and saw that the RMSE was large where an obvious error had occurred.

Regarding the DBN parameter selection, it was noticed that setting initial biases for the hidden units to -4 was an important parameter for achieving good accuracy. A better way of encourage sparsity is to include a sparsity penalty term in the cost objective function [31] instead of making a crude estimation of initial biases for the hidden units. For the *raw-DBN* setup, it was also crucial to train each layer with a large number of epochs and in particular the fine tuning step.

We also noticed a lower performance if sleep stages were not set to equal sizes in the training set. There was also a high variation in the accuracy between patients, even if they came from the same dataset. Since the DBN will find a generalization that best fits all training examples, a testing set that deviates from the average training set might give poor results. Since data might differs greatly between patients, a single DBN trained on general sleep data is not specialized enough. The need for a more dynamic system, especially one including the transition and emission matrices for the HMM, is made clear when comparing the hypnograms of a healthy patient and a patient with sleep disordered breathing. Further, although the HMM provides a simple solution that captures temporal properties of sleep data, it makes two critical assumptions [13]. The first one is that the next hidden state can be approximated by a state depending only on the previous state, and the second one is that observations at different time steps are conditionally independent given a state sequence. Replacing HMM with conditional random fields (CRFs) could improve accuracy but is still a simplistic temporal model that does not exploit the power of DBNs [32].

While a clear advantage of using DBN is the natural way in which it deals with anomalous data, there are some limitations to the DBN. One limitation is that correlations between signals in the input data are not well captured. This gives a feature-based approach an advantage where, for example, the correlation between both EOG channels can easily be represented with a feature. This could be solved by either representing the correlation in the input or extending the DBN to handle such correlations, such as a cRBM [33].

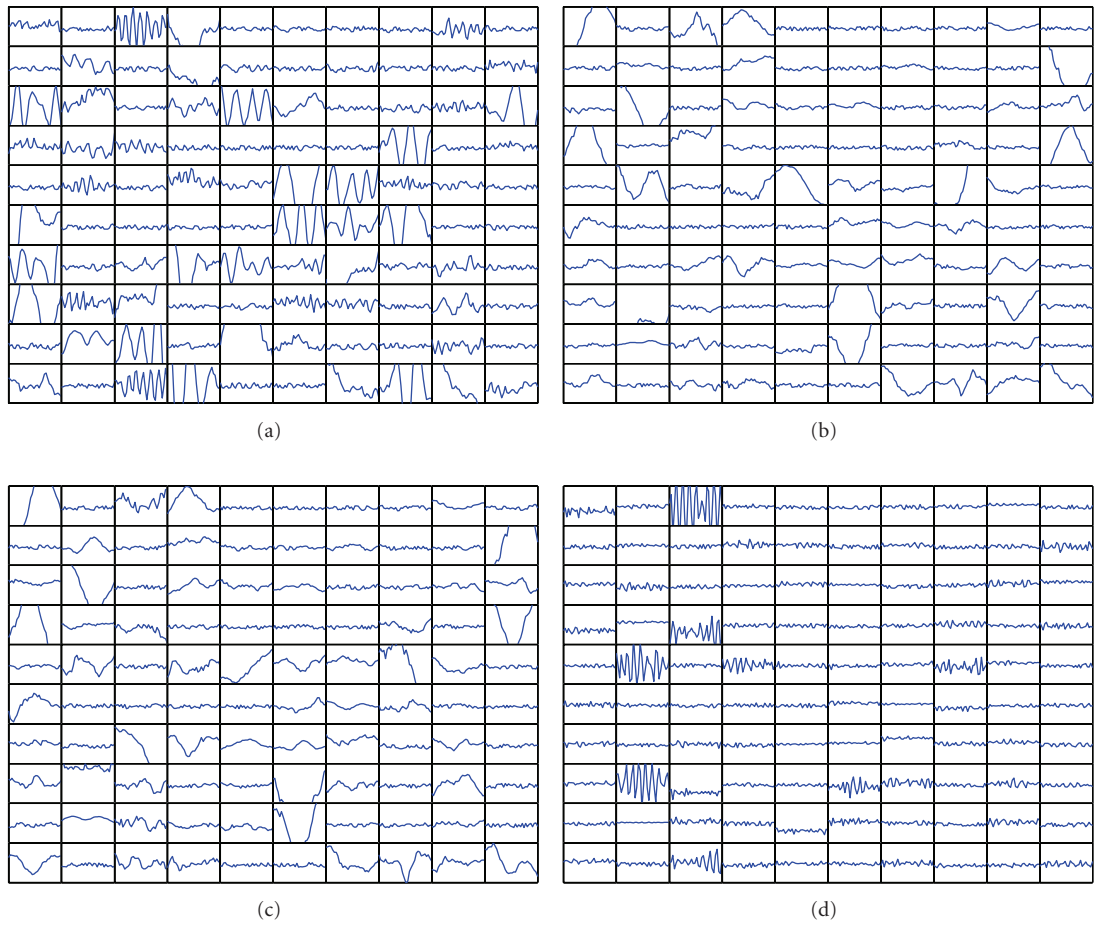


Figure 6: Learned features of layer 1 for (a) EEG, (b) EOG1, (c) EOG2, and (d) EMG. It can be observed that the learned features are of various amplitudes and frequencies and some resemble known sleep events such as a K-complex or blink artifacts. Only the first 100 of the 200 features are shown here.

Regarding the implemented *feat-GOHMM*, we have tried our best to get as high accuracy with the setup as possible. It is almost certain that another set of features, different feature selection algorithm, and/or another classifier could outperform our *feat-GOHMM*. However, we hope that this work illustrates the advantages of unsupervised feature learning, which not only removes the need for domain specific expert knowledge, but inherently provides tools for anomaly detection and noise redundancy.

It has been suggested for multimodal signals to train a separate DBN for each signal first and then train a top DBN with concatenated data [34]. This not only could improve classification accuracy, but also provide the ability to single out which signal contains the anomalous signal. Further, this work has explored clinical data sets in close cooperation with physicians, and future work will concentrate on the application for at home monitoring as sleep data is an area where unsupervised feature learning is a highly promising method for sleep stage classification as data is abundant and labels are costly to obtain.

Appendix

A. Features

A total of 28 features are used in this work.

Relative power for signal y in frequency band f is calculated as

$$y_{P_{\text{rel}}}(f) = \frac{y_P(f)}{\sum_{f=f_1}^{f_5} y_P(f)}, \tag{A.1}$$

where $y(f)_P$ is the sum of the absolute power in frequency band f for signal y. The five frequency bands used are delta (0.5–4 Hz), theta (4–8 Hz), alpha (8–13 Hz), beta (13–20 Hz), and gamma (20–64 Hz).

The median of the absolute value for EMG is calculated as

$$\text{EMG}_{\text{median}} = \text{median}\left(\sum_{k=1}^{N} |\text{EMG}(k)|\right). \tag{A.2}$$

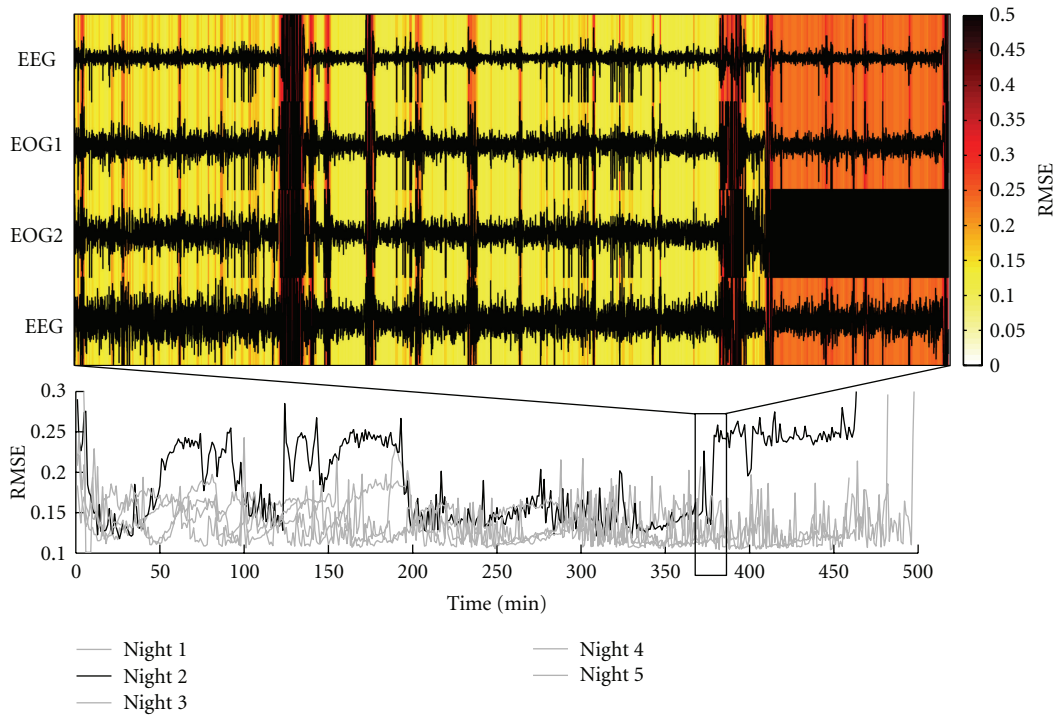


Figure 7: RMSE for five night runs recorded at home (bottom). Color-code of RMSE for night run 2 where the redder areas more anomalous areas of the signal. EOG2 falls off at around 380 minutes (top).

The eye correlation coefficient for the EOG is calculated as

$$\mathrm{EOG}_{\mathrm{corr}} = \frac{E\left[\left(y_1 - \mu_{y_1}\right)\left(y_2 - \mu_{y_2}\right)\right]}{\sigma_{y_1}\sigma_{y_2}}, \tag{A.3}$$

where $y_1 = \mathrm{EOG}_{\mathrm{left}}$ and $y_2 = \mathrm{EOG}_{\mathrm{right}}$.

The entropy for a signal y is calculated as

$$y_{\mathrm{entropy}} = -\sum_{k=1}^{8} \frac{n_k}{N} \ln \frac{n_k}{N}, \tag{A.4}$$

where N is the number of samples in signal y, and n_k is the number of samples from y that belongs to the kth bin from a histogram of y.

The kurtosis for a signal y is calculated as

$$y_{\mathrm{kurtosis}} = \frac{E[y-\mu]^4}{\sigma^4}, \tag{A.5}$$

where μ and σ are the mean and standard deviation, respectively, for signal y.

The spectral mean for signal y is calculated as

$$y_{\mathrm{spectralmean}} = \frac{1}{F}\sum_{f=f_1}^{f_5} y(f)_{P_{\mathrm{rel}}} \cdot f, \tag{A.6}$$

where F is the sum of the lengths of the 5 frequency bands.

Fractal exponent [35, 36] for the EEG is calculated as the negative slope of the linear fit of spectral density in the double logarithmic graph.

Normalization is performed for some features according to [37] and [30]. The absolute median for EMG is normalized by dividing with the absolute median for the whole EMG signal.

Acknowledgments

The authors are grateful to Professor Walter T. McNicholas of St. Vincents University Hospital, Ireland, and Professor Conor Heneghan of University College Dublin, Ireland, for providing the sleep training data for this study. They would also like to thank senior physician Lena Leissner and sleep technician Meeri Sandelin at the sleep unit of the neuroclinic at Örebro University Hospital for their continuous support and expertise. Finally, special thanks to D F Wulsin for writing and sharing the open-source implementation of DBN for Matlab that was used in this work [20]. This work was funded by NovaMedTech.

References

[1] K. Šušmákováemail and A. Krakovská, "Discrimination ability of individual measures used in sleep stages classification," *Artificial Intelligence in Medicine*, vol. 44, no. 3, pp. 261–277, 2008.

[2] A. Flexer, G. Gruber, and G. Dorffner, "A reliable probabilistic sleep stager based on a single EEG signal," *Artificial Intelligence in Medicine*, vol. 33, no. 3, pp. 199–207, 2005.

[3] L. Johnson, A. Lubin, P. Naitoh, C. Nute, and M. Austin, "Spectral analysis of the EEG of dominant and non-dominant

alpha subjects during waking and sleeping," *Electroencephalography and Clinical Neurophysiology*, vol. 26, no. 4, pp. 361–370, 1969.

[4] J. Pardey, S. Roberts, L. Tarassenko, and J. Stradling, "A new approach to the analysis of the human sleep/wakefulness continuum," *Journal of Sleep Research*, vol. 5, no. 4, pp. 201–210, 1996.

[5] N. Schaltenbrand, R. Lengelle, M. Toussaint et al., "Sleep stage scoring using the neural network model: comparison between visual and automatic analysis in normal subjects and patients," *Sleep*, vol. 19, no. 1, pp. 26–35, 1996.

[6] H. G. Jo, J. Y. Park, C. K. Lee, S. K. An, and S. K. Yoo, "Genetic fuzzy classifier for sleep stage identification," *Computers in Biology and Medicine*, vol. 40, no. 7, pp. 629–634, 2010.

[7] L. Zoubek, S. Charbonnier, S. Lesecq, A. Buguet, and F. Chapotot, "A two-steps sleep/wake stages classifier taking into account artefacts in the polysomnographic signa," in *Proceedings of the 17th World Congress, International Federation of Automatic Control (IFAC '08)*, July 2008.

[8] G. Luo and W. Min, "Subject-adaptive real-time sleep stage classification based on conditional random field," in *Proceedings of the American Medical Informatics Association Annual Symposium (AMIA '07)*, pp. 488–492, 2007.

[9] T. Penzel, K. Kesper, V. Gross, H. F. Becker, and C. Vogelmeier, "Problems in automatic sleep scoring applied to sleep apnea," in *Proceedings of the 25th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (IEEE EMBS '03)*, pp. 358–361, September 2003.

[10] G. E. Hinton, S. Osindero, and Y. W. Teh, "A fast learning algorithm for deep belief nets," *Neural Computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[11] Y. Bengio, P. Lamblin, D. Popovici, and H. Larochelle, "Greedy layer-wise training of deep networks," in *Advances in Neural Information Processing Systems (NIPS '06)*, vol. 19, pp. 153–160, 2006.

[12] M. Ranzato, C. Poultney, S. Chopra, and Y. LeCun, "Efficient learning of sparse representations with an energy-based model," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS '06)*, J. Platt, T. Hoffman, and B. Schölkopf, Eds., MIT Press, 2006.

[13] Y. Bengio and Y. LeCun, "Scaling learning algorithms towards AI," in *Large-Scale Kernel Machines*, L. Bottou, O. Chapelle, D. DeCoste, and J. Weston, Eds., MIT Press, 2007.

[14] Y. Bengio, "Learning deep architectures for AI," Tech. Rep. 1312, Department of IRO, Universite de Montreal, 2007.

[15] I. Arel, D. Rose, and T. Karnowski, "Deep machine learning—a new frontier in artificial intelligence research," *IEEE Computational Intelligence Magazine*, vol. 14, pp. 12–18, 2010.

[16] V. Nair and G. E. Hinton, "3-d object recognition with deep belief nets," in *Proceedings of the Advances in Neural Information Processing Systems (NIPS '06)*, 2006.

[17] G. Taylor, G. E. Hinton, and S. Roweis, "Modeling human motion using binary latent variables," in *Proceedings of the Advances in Neural Information Processing Systems*, 2007.

[18] N. Jaitly and G. E. Hinton, "Learning a better representation of speech sound waves using restricted boltzmann machines," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '11)*, 2011.

[19] M. Längkvist and A. Loutfi, "Unsupervised feature learning for electronic nose data applied to bacteria identification in blood," in *NIPS Workshop on Deep Learning and Unsupervised Feature Learning*, 2011.

[20] D. F. Wulsin, J. R. Gupta, R. Mani, J. A. Blanco, and B. Litt, "Modeling electroencephalography waveforms with semi-supervised deep belief nets: fast classification and anomaly measurement," *Journal of Neural Engineering*, vol. 8, no. 3, Article ID 036015, 2011.

[21] G. E. Hinton, "Training products of experts by minimizing contrastive divergence," *Neural Computation*, vol. 14, no. 8, pp. 1771–1800, 2002.

[22] A. Rechtschaffen and A. Kales, *A Manual of Standardized Terminology, Techniques and Scoring System for Sleep Stages of Human Subjects*, U.S. Government Printing Office, Washington DC, USA, 1968.

[23] M. Hirshkowitz, "Standing on the shoulders of giants: the Standardized Sleep Manual after 30 years," *Sleep Medicine Reviews*, vol. 4, no. 2, pp. 169–179, 2000.

[24] S. L. Himanen and J. Hasan, "Limitations of Rechtschaffen and Kales," *Sleep Medicine Reviews*, vol. 4, no. 2, pp. 149–167, 2000.

[25] L. R. Rabiner and B. H. Juang, "An introduction to hidden markov models," *IEEE ASSP Magazine*, vol. 3, no. 1, pp. 4–16, 1986.

[26] G. E. Hinton, *A Practical Guide to Training Restricted Boltzmann Machines*, 2010.

[27] S. Charbonnier, L. Zoubek, S. Lesecq, and F. Chapotot, "Self-evaluated automatic classifier as a decision-support tool for sleep/wake staging," *Computers in Biology and Medicine*, vol. 41, no. 6, pp. 380–389, 2011.

[28] A. Schlögl, C. Keinrath, D. Zimmermann, R. Scherer, R. Leeb, and G. Pfurtscheller, "A fully automated correction method of EOG artifacts in EEG recordings," *Clinical Neurophysiology*, vol. 118, no. 1, pp. 98–104, 2007.

[29] A. L. Goldberger, L. A. Amaral, L. Glass et al., "PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals," *Circulation*, vol. 101, no. 23, pp. E215–220, 2000.

[30] L. Zoubek, S. Charbonnier, S. Lesecq, A. Buguet, and F. Chapotot, "Feature selection for sleep/wake stages classification using data driven methods," *Biomedical Signal Processing and Control*, vol. 2, no. 3, pp. 171–179, 2007.

[31] G. Huang, H. Lee, and E. Learned-Miller, "Learning hierarchical representations for face verification with convolutional deep belief networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR '12)*, 2012.

[32] D. Yu, L. Deng, I. Jang, P. Kudumakis, M. Sandler, and K. Kang, "Deep learning and its applications to signal and information processing," *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 145–154, 2011.

[33] M. Ranzato, A. Krizhevsky, and G. E. Hinton, "Factored 3-way restricted boltzmann machines for modeling natural images," in *Proceedings of the 30th International Conference on Artificial Intelligence and Statistics*, 2010.

[34] J. Ngiam, A. Khosla, M. Kim, H. Lee, and A. Y. Ng, "Multimodal deep learning," in *Proceedings of the 28th International Conference on Machine Learning*, 2011.

[35] A. R. Osborne and A. Provenzale, "Finite correlation dimension for stochastic systems with power-law spectra," *Physica D*, vol. 35, no. 3, pp. 357–381, 1989.

[36] E. Pereda, A. Gamundi, R. Rial, and J. González, "Non-linear behaviour of human EEG: fractal exponent versus correlation dimension in awake and sleep stages," *Neuroscience Letters*, vol. 250, no. 2, pp. 91–94, 1998.

[37] T. Gasser, P. Baecher, and J. Moecks, "Transformations towards the normal distribution of broad band spectral parameters of the EEG," *Electroencephalography and Clinical Neurophysiology*, vol. 53, no. 1, pp. 119–124, 1982.

Hindawi Publishing Corporation
Advances in Artificial Neural Systems
Volume 2012, Article ID 919281, 8 pages
doi:10.1155/2012/919281

Research Article

Selection of Spatiotemporal Features in Breast MRI to Differentiate between Malignant and Benign Small Lesions Using Computer-Aided Diagnosis

F. Steinbruecker,[1] A. Meyer-Baese,[2] C. Plant,[2] T. Schlossbauer,[3] and U. Meyer-Baese[4]

[1] *Department of Computer Science, Technical University of Munich, 8574 Garching, Germany*
[2] *Department of Scientific Computing, Florida State University, Tallahassee, FL 32306-4120, USA*
[3] *Institute for Clinical Radiology, University of Munich, 81377 Munich, Germany*
[4] *Department of Electrical and Computer Engineering, FAMU/FSU College of Engineering, Tallahassee, FL 32310-6046, USA*

Correspondence should be addressed to A. Meyer-Baese, ameyerbaese@fsu.edu

Received 29 February 2012; Accepted 14 May 2012

Academic Editor: Juan Manuel Gorriz Saez

Automated detection and diagnosis of small lesions in breast MRI represents a challenge for the traditional computer-aided diagnosis (CAD) systems. The goal of the present research was to compare and determine the optimal feature sets describing the morphology and the enhancement kinetic features for a set of small lesions and to determine their diagnostic performance. For each of the small lesions, we extracted morphological and dynamical features describing both global and local shape, and kinetics behavior. In this paper, we compare the performance of each extracted feature set for the differential diagnosis of enhancing lesions in breast MRI. Based on several simulation results, we determined the optimal feature number and tested different classification techniques. The results suggest that the computerized analysis system based on spatiotemporal features has the potential to increase the diagnostic accuracy of MRI mammography for small lesions and can be used as a basis for computer-aided diagnosis of breast cancer with MR mammography.

1. Introduction

Breast cancer is one of the most common cancers among women. Contrast-enhanced MR imaging of the breast was reported to be a highly sensitive method for the detection of invasive breast cancer [1]. Different investigators described that certain dynamic signal intensity (SI) characteristics (rapid and intense contrast enhancement followed by a wash out phase) obtained in dynamic studies are a strong indicator for malignancy [2]. Morphologic criteria have also been identified as valuable diagnostic tools [3]. Recently, combinations of different dynamic and morphologic characteristics have been reported [4] that can reach diagnostic sensitivities up to 97% and specificities up to 76.5%.

As an important aspect remains the fact that many of these techniques were applied on a database of predominantly tumors of a size larger than 2 cm. In these cases, MRI reaches a very high sensitivity in the detection of invasive breast cancer due to both morphological criteria as well as characteristic time-signal intensity curves. However, the value of dynamic MRI and of automatic identification and classification of characteristic kinetic curves is not well established in small lesions when clinical findings, mammography, and ultrasound are unclear. Recent clinical research has shown that DCIS with small invasive carcinoma can be adequately visualized in MRI [5] and that MRI provides an accurate estimation of invasive breast cancer tumor size, especially in tumors of 2 cm or smaller [6].

Visual assessment of morphological properties is highly interobserver variable [7], while automated computation of features leads to more reproducible indices and thus to a more standardized and objective diagnosis. In this sense, we present novel mathematical descriptors for both morphology and dynamics and will compare their performance regarding

small lesion classification based on novel feature selection algorithms.

More than 40% of the false-negative MR diagnosis are associated with pure ductal carcinoma in situ (DCIS) and with small lesion size and thus indicating a lower sensitivity of MRI for these cases. It has been shown that double reading achieves a higher sensitivity but is time-consuming and as an alternative a computer-assisted system was suggested [8]. The success of CAD in conventional X-ray mammography [9–13] motivates further the research of similar automated diagnosis techniques in breast MRI.

In the present study, we design and evaluate a computerized analysis system for the diagnosis of small breast masses with an average diameter of <1 cm.

The automated evaluation is a multistep system which includes global and local features such as shape descriptors, dynamical features, and spatiotemporal features combining both morphology and dynamics aspects. Different classification techniques are employed to test the performance of the complete system. Summarizing, in the present paper, a multifactorial protocol, including image registration, and morphologic and dynamic criteria are evaluated in predominantly small lesions of 1.0 cm or less as shown in Figure 1.

2. Material and Methods

2.1. Patients. A total of 40 patients, all females having an age range 42–73, with indeterminate small mammographic breast lesions were examined. All patients were consecutively selected after clinical examinations, mammography in standard projections (craniocaudal and oblique mediolateral projections) and ultrasound. Only lesions BIRADS 3 and 4 were selected where at least one of the following criteria was present: nonpalpable lesion, previous surgery with intense scarring, or location difficult for biopsy (close to chest wall). All patients had histopathologically confirmed diagnosis from needle aspiration/excision biopsy and surgical removal. Breast cancer was diagnosed in 17 out of the total 31 cases. The average size of both benign and malignant tumors was less than 1.1 cm.

2.2. MR Imaging. MRI was performed with a 1.5 T system (Magnetom Vision, Siemens, Erlangen, Germany) with two different protocols equipped with a dedicated surface coil to enable simultaneous imaging of both breasts. The patients were placed in a prone position. First, transversal images were acquired with a STIR (short TI inversion recovery) sequence (TR = 5600 ms, TE = 60 ms, FA = 90°, IT = 150 ms, matrix size 256 × 256 pixels, slice thickness 4 mm). Then a dynamic T1 weighted gradient echo sequence (3D fast low angle shot sequence) was performed (TR = 12 ms, TE = 5 ms, FA = 25°) in transversal slice orientation with a matrix size of 256 × 256 pixels and an effective slice thickness of 4 mm or 2 mm.

The dynamic study consisted of 6 measurements with an interval of 83 s. The first frame was acquired before injection of paramagnetic contrast agent (gadopentetate dimeglumine, 0.1 mmol/kg body weight, Magnevist, Schering, Berlin, Germany) immediately followed by the 5 other measurements. The initial localization of suspicious breast lesions was performed by computing difference images, that is, subtracting the image data of the first from the fourth acquisition. As a preprocessing step to clustering, each raw gray level time-series $S(\tau)$, $\tau \in \{1, \ldots, 6\}$ was transformed into a signal time-series of relative signal enhancement $x(\tau)$ for each voxel, the precontrast scan at $\tau = 1$ serving as reference, in other words $x(\tau) = (x(\tau) - x(1))/x(1)$. Thus, we ensure that the proposed method is less sensitive to changing between different MR scanners and/or protocols.

Automatic motion correction represents an important prerequisite to a correct automated small lesion evaluation [14]. Especially for small lesions, the assumption of correct spatial alignment often leads to misinterpretation of the diagnostic significance of enhancing lesions [15]. Therefore, we performed an elastic image registration method based on the optical flow method [16]. The employed motion compensation algorithm is based on the Horn and Schunck method [17] and represents a variational method for computing the displacement field, the so-called optical flow, in an image sequence. In contrast to optical flow, we do not want to compute the displacement field in a projected image of our data, but the actual displacement in 3D space. In our work, however, we favor the original quadratic formulation, since we explicitly need the filling-in effect of a nonrobust regularizer to fill in the information in masked regions. To overcome the problem of having a nonconvex energy in the energy functional, we use the coarse-to-fine warping scheme detailed in [16], which linearizes the data term as in [17] and computes incremental solutions on different image scales.

We tested motion compensation for two and three directions and found the optimal motion compensation results in two directions [18]. Segmentation of the tumor is semiautomatic and we define an ROI including all voxels of a lesion with an initial contrast enhancement of ≥50%. The center of the lesion was interactively marked on one slice of the subtraction images and then a region growing algorithm included all adjacent contrast-enhancing voxels and also those from neighboring slices. Thus a 3-D form of the lesion was determined. An interactive ROI was necessary whenever the lesion was connected with diffuse contrast enhancement, as it is the case in mastopathic tissue.

3. Computer-Aided Diagnosis (CAD) System

The small lesion evaluation is based on a multi-step system that includes a reduction of motion artifacts based on a novel nonrigid registration method, an extraction of morphologic features, dynamic enhancement patterns as well as mixed features for diagnostic feature selection and performance of lesion evaluation. Figure 1 visualizes the proposed automated system for small lesion detection.

3.1. Feature Extraction. The complexity of the spatio-temporal tumor representation requires specific morphology and/or kinetic descriptors. We analyzed geometric and

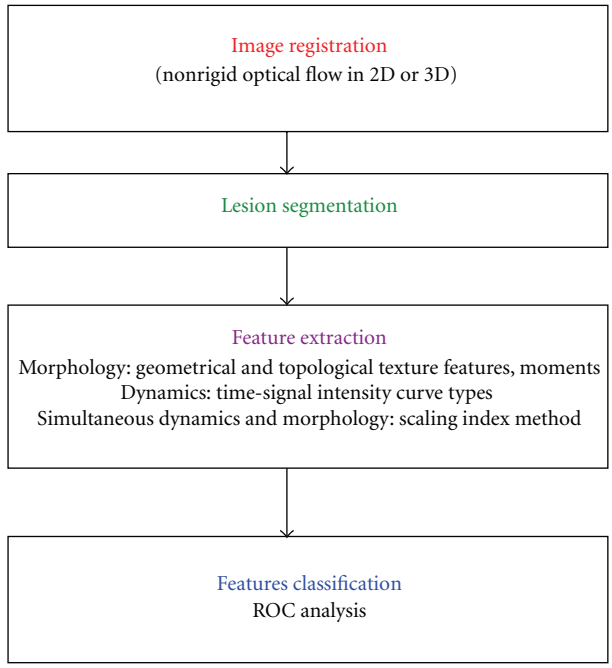


Figure 1: Diagram of a computer-assisted system for the evaluation of small contrast enhancing lesions.

Krawtchouk moments and geometrical features as shape descriptors, provided a temporal enhancement modeling for kinetic feature extraction and the scaling index method for the simultaneous morphological and dynamics representation.

3.1.1. Contour Features. To represent the shape of the tumor contour, the tumor voxels having nontumor voxel as a neighbor were extracted to represent the contour of the tumor. In this context, neighbor voxels include diagonally adjacent voxels, but not voxels from a different transverse slice. Due to the different grid sizes in the three directions of the MR images and possible gaps between transverse slices, the tumor contour in one transverse slice does not necessarily continue smoothly into the next transverse slice. Considering tumor contours between transverse slices therefore introduces contour voxels that are completely in the tumor interior in one slice. This is illustrated in Figure 2: the dark voxels are contour voxels and the arrows indicate the computed contour chain. If voxels in the tumor having at least one non-tumor voxel as a neighbor on an adjacent transverse slice were considered part of the contour, in this example, the crossed-out voxels would belong to the contour.

Figure 3 shows an example for a tumor where the contour shifts considerably from one transverse slice to another.

The contour in each slice was stored as an 1D chain of the 3D position of each contour voxel, constituting a "walk" along the contour. The chains of several slices were spliced together end to end to form a chain of 3D vectors representing the contour of the tumor.

Next, the center of mass of the tumor was computed as

$$\overline{v} := \frac{1}{n}\sum_{i=1}^{n} v_i, \tag{1}$$

where n is the number of voxels belonging to the tumor, and v_i is location of the ith tumor voxel. Since the center of mass was computed from the binary image of the tumor, irregularities in the voxel gray values of the tumor were not taken into account.

Knowing the center of mass, for each contour voxel c_i, the radius r_i and the azimuth ω_i (i.e., the angle between the vector from the center of mass to the voxel c_i and the sagittal plane) were computed the following way:

$$\begin{aligned} r_i &:= \|c_i - \overline{v}\|_2, \\ \omega_i &:= \arcsin\left(\frac{c_{ix} - \overline{v}_x}{(c_{ix} - \overline{v}_x)^2 + \left(c_{iy} - \overline{v}_y\right)^2}\right), \end{aligned} \tag{2}$$

where the subscripts x and y denote the position of the voxel in sagittal and coronal direction, respectively. ω_i was also extended to the range from $-\pi$ to π by taking into account the sign of $(c_{iy} - \overline{v}_y)$.

From the chain of floating point values $r_1, \ldots, r_m$, the minimum value $r_{\min}$ and the maximum value $r_{\max}$ were computed, as well as

$$\text{the mean value } \overline{r} := \frac{1}{m}\sum_{i=1}^{m} r_i, \tag{3}$$

$$\text{the standard deviation } \sigma_r := \sqrt{\frac{1}{m}\sum_{i=1}^{m}(r_i - \overline{r})^2}, \tag{4}$$

$$\text{the entropy } h_r := -\sum_{i=1}^{100} p_i \cdot \log_2(p_i). \tag{5}$$

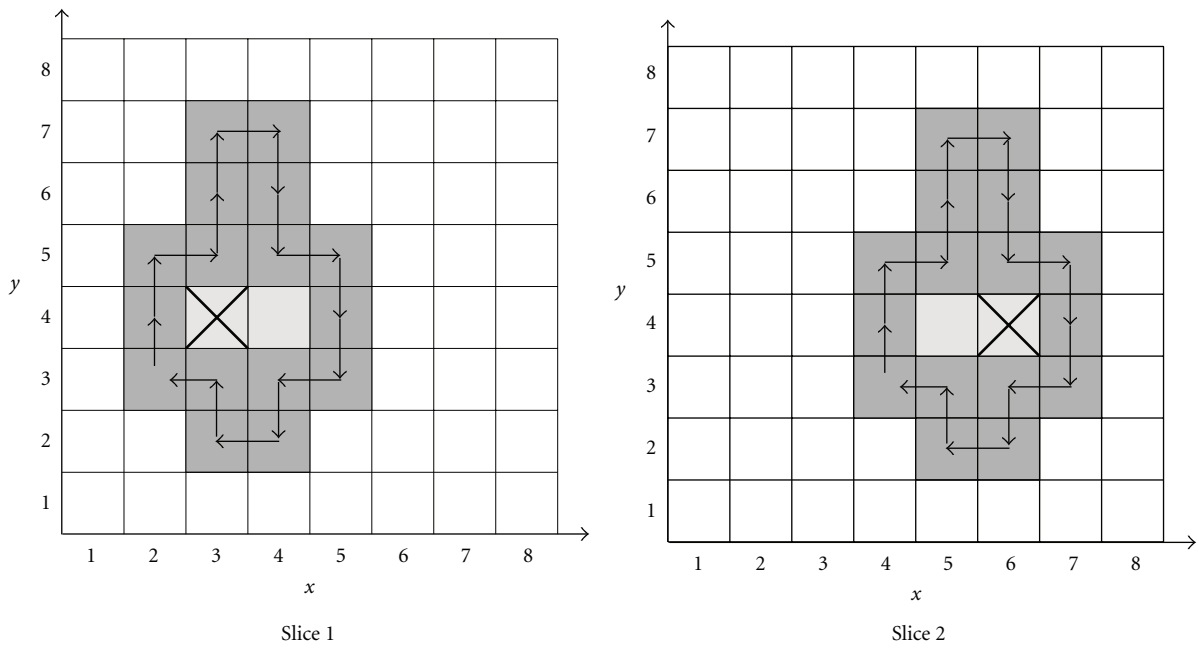


Figure 2: Example of contour computation.

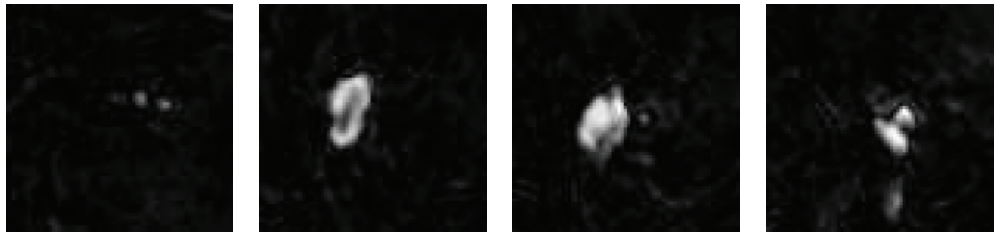


Figure 3: Left to right: tumor in adjacent transverse slices of a $512 \times 512 \times 32$ voxel MR image.

The entropy h_r was computed from the normalized distribution of the values into 100 "buckets", where p_i is defined as follows:

For $0 \leq i \leq 99$:

$$p_i := \frac{\left|\left\{r_j \mid i \leq \left(r_j - r_{\min}\right)/\left(r_{\max} - r_{\min}\right) \cdot 100 < i + 1\right\}\right|}{m}. \tag{6}$$

From the radius, $r_{\min}$, $r_{\max}$, $\bar{r}$, σ_r, and h_r were used as morphological features of the tumor. From the azimuth, only the entropy h_ω (computed for ω as in (5) and (6)) was used as a feature, since the values $\omega_{\min}$ and $\omega_{\max}$ are always around π and $-\pi$, respectively, and the value σ_ω is not invariant under rotation of the tumor image.

An additional measurement describing the compactness of the tumor, which was also used as a feature, is the number of contour voxels, divided by the number of all voxels belonging to the tumor.

3.1.2. Morphological Features. The spatial and morphological variations of a tumor can be easily captured by shape descriptors. We analyze two modalities as shape descriptors based on moments: the geometric and Krawtchouk moments.

Geometrical Moments. We will employ low-order three-dimensional geometrical moment invariants as described in [19] because they have a low computation time and the results are stable to noise and distortion. We will utilize the 6 low-order finite-term three-dimensional moment invariants as described in [19]. There are one second-order and fourth-order, two third-order and three fourth-order moment invariants.

Krawtchouk Moments. Global and local shape description represents an important field in 3D medical image analysis. For breast lesion classification, there is a stringent need to describe properly the huge data volumes stemming from 3D images by a small set of parameters which captures the morphology (shape) well. However, very few techniques have been proposed for both global and local shape description. We employed Krawtchouk moments [20] as shape descriptors for both malignant and benign lesions. Weighted 3-D

Krawtchouk moments have several advantages compared to other known methods: (1) they are defined in the discrete field and thus do not introduce any discretization error like Spherical Harmonics defined in a continuous field and (2) low-order moments can capture abrupt changes in the shape of an object. The weighted 3D Krawtchouk moments [20] form a very compact descriptor of a tumor, achieved in a very short computational time. Every tumor can be represented by Krawtchouk moments since it is expressed as a function $f(x, y, z)$ in a discrete grid $[0 \cdots N-1] \times [0 \cdots M-1] \times [0 \cdots L-1]$.

Krawtchouk moments represent a set of orthonormal polynomials associated with the binomial distribution [21]. The nth order Krawtchouk classical polynomials can be expressed as a hypergeometric function:

$$K_n(x; p, N) = \sum_{k=0}^{N} a_{k,n,p} x^k = {}_2F_1\left(-n, -x; -N; \frac{1}{p}\right) \tag{7}$$

with $x, n = 0, 1 \cdots N$; $N > 0$; $p \in (0, 1)$ and the hypergeometric function ${}_2F_1$ is defined as

$${}_2F_1(a, b; c; z) = \sum_{k=0}^{\infty} \frac{(a)_k (b)_k}{(c)_k} \frac{z^k}{k!}, \tag{8}$$

and with $(a)_k$ being the Pochhammer symbol

$$(a)_k = a(a+1) \cdots (a+k-1) = \frac{\Gamma(a+k)}{\Gamma(a)}. \tag{9}$$

The set of the Krawtchouk polynomials $S = \{K_n(x; p, N), n = 0 \cdots N\}$ has $N+1$ elements. This corresponds to a set of discrete basis functions with the weight function

$$\begin{aligned} w(x; p, N) &= \binom{N}{x} p^x (1-p)^{N-x}, \\ \rho(n; p, N) &= (-1)^n \left(\frac{1-p}{p}\right)^n \frac{n!}{(-N)_n}. \end{aligned} \tag{10}$$

We assume that $f(x, y, z)$ is a 3-dimensional function defined in a discrete field $A = \{(x, y, z) : x, y, z \in N, x = [0 \cdots N-1], y = [0 \cdots M-1], z = [0 \cdots L-1]\}$. The weighted three-dimensional moments of order $(n+m+l)$ of f are given as

$$\begin{aligned} \widetilde{Q}_{mnl} = \sum_{x=0}^{N-1} \sum_{y=0}^{M-1} \sum_{z=0}^{L-1} & \overline{K}_n(x; p_x, N-1) \\ & \cdot \overline{K}_m\left(y; p_y, M-1\right) \overline{K}_l(z; p_z, L-1) \\ & \cdot f(x, y, z), \end{aligned} \tag{11}$$

with $p_x, p_y, p_z \in (0, 1)$. Local features can be extracted by the appropriate selection of low-order Krawtchouk moments. $\overline{K}_n(x; p, N)$ is given as

$$\overline{K}_n(x; p, N) = K_n(x; p, N) \sqrt{\frac{w(x; p, N)}{\rho(n; p, N)}}. \tag{12}$$

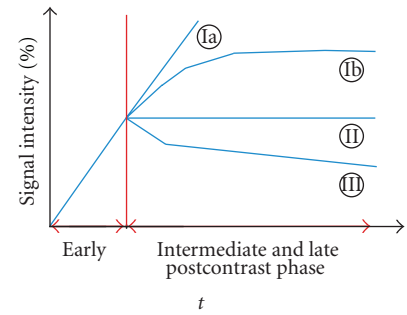


Figure 4: Schematic drawing of the time-signal intensity (SI) curve types [2]. Type I corresponds to a straight (Ia) or curved (Ib) line; enhancement continues over the entire dynamic study. Type II is a plateau curve with a sharp bend after the initial upstroke. Type III is a washout time course. In breast cancer, plateau or washout-time courses (type II or III) prevail. Steadily progressive signal intensity time courses (type I) are exhibited by benign enhancing lesions.

$\overline{K}_m(y; p_y, M-1)$ and $\overline{K}_l(z; p_z, L-1)$ are defined correspondingly. Thus, every 3-dimensional function $f(x, y, z)$ in a 3-dimensional field can be decomposed into weighted 3-dimensional Krawtchouk moments $\widetilde{Q}_{nml}$.

The tumor can be represented by Krawtchouk moments since it is expressed as a function $f(x, y, z)$ in a discrete space $[0 \cdots N-1] \times [0 \cdots M-1] \times [0 \cdots L-1]$.

3.1.3. Dynamical Features. Lesion differential diagnosis in dynamic protocols is based on the assumption that benign and malignant lesions exhibit different enhancement kinetics. In [2], it was shown that the shape of the time-signal intensity curve represents an important criterion in differentiating benign and malignant enhancing lesions in dynamic breast MR imaging. The results indicate that the enhancement kinetics, as represented by the time-signal intensity curves visualized in Figure 4, differ significantly for benign and malignant enhancing lesions and thus represent a basis for differential diagnosis. In breast cancer, plateau or washout-time courses (type II or III) prevail. Steadily progressive signal intensity time courses (type I) are exhibited by benign enhancing lesions. Also, these enhancement kinetics are not only present in benign tumors but also in fybrocystic changes [2].

Computing the average signal intensity of the tumor before contrast agent administration (SI) and after contrast agent administration (SI_C), the relative enhancement can be computed as

$$\Delta \mathrm{SI}(t) := \frac{\mathrm{SI}_C(t) - \mathrm{SI}(t)}{\mathrm{SI}(t)} \cdot 100\%. \tag{13}$$

To capture the slope of the curve of relative signal intensity enhancement (RSIE) versus time in the late postcontrast time, we computed the line $(s = a \cdot t + b)$ that approximates the curve of the RSIE for the last three time scans. The values a and b are the least square solutions of the overdetermined system of equations $a \cdot t_i + b = s_{i,j}$ for the three last points in

time ($i \in \{3,4,5\}$), as well as for all tumor voxels j, with $s_{i,j}$ being the RSIE in voxel j at time scan i.

The solutions to these equations are given by

$$a = \frac{3n\sum t_i s_{i,j} - \sum t_i \sum s_{i,j}}{3n \cdot \sum t_i^2 - \left(\sum t_i\right)^2},$$
$$b = \frac{\sum s_{i,j} \sum t_i^2 - \sum t_i \sum t_i s_{i,j}}{3n \cdot \sum t_i^2 - \left(\sum t_i\right)^2}, \tag{14}$$

where n is the number of voxels in the tumor, and $\sum$ is an abbreviation for $\sum_{j=1}^{3}\sum_{i=1}^{n}$. The slope a was used as a feature to describe the dynamics.

3.1.4. Simultaneous Morphology and Dynamics Representations. The scaling index method [22] is a technique that is based on both morphology and kinetics. It represents the local structure around a given point. In the context of breast MRI, such a point consists of the sagittal, coronal, and transverse position of a tumor voxel and its third time scan gray value, and the scaling index serves as an approximation of the dimension of local point distributions.

Mathematically, the scaling index represents the 2-D image as a set of points in a three-dimensional state space defined by the coordinates x, y, z and the gray value $f(x, y, z)$. For every point P_i with coordinates (x_i, y_i, z_i) the number of points in a sphere with radius r_1 and a sphere with radius r_2 is determined and the scaling index α_i is computed based on the following equation:

$$\alpha_i = \frac{\left(\log N(P_i, r_2) - \log N(P_i, r_1)\right)}{\left(\log r_2 - \log r_1\right)}, \tag{15}$$

where $N(P_i, r)$ is the number of points located within an n-dimensional sphere of radius r centered at P_i. As radii, we choose the bounds of the tumor shape. Thus, the obtained scaling-index is a measure for the local dimensionality of the tumor and thus quantifies its morphological and dynamical features. There is a correlation between the scaling index and the structural nature: $\alpha = 0$ for clumpy structures, $\alpha = 1$ for points embedded in straight lines, and $\alpha = 2$ for points in a flat distribution.

For each of the three time scans ($i \in \{1,3,5\}$), the standard deviation and entropy were determined and used as a feature to capture the heterogeneous behavior of the enhancement in a tumor.

3.2. Classification Techniques. The following section gives a description of classification methods applied to evaluate the effect of spatiotemporal features in breast MR images.

Discriminant analysis represents an important area of multivariate statistics and finds a wide application in medical imaging problems. The most known approaches are linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), and Fisher's canonical discriminant analysis.

Let us assume that $\mathbf{x}$ describes a K-dimensional feature vector that is, there are J classes and there are N_j samples available in group j. The mean in group j is given by μ_j and the covariance matrix is given by Σ_j.

3.2.1. Bayes Classification Based on LDA and QDA. The Bayes classification [23] is based on estimating the prior probabilities π_i for each class which describe the prior estimates about how probable a class is.

This classification method assigns each new sample to the group with the highest a posterior probability. Thus, the classification rule becomes

$$C_j = \left(\mathbf{x}_i - \mu_j\right)^T \left(\Sigma_j\right)^{-1} \left(\mathbf{x}_i - \mu_j\right) + \log\left|\Sigma_j\right| - 2\log\pi_j, \tag{16}$$

where μ_j represent the means of the classes and Σ_j the corresponding covariance matrix. The assignment to a certain class j for a certain pattern is made based on the smallest determined value of C_j.

There are two cases to be distinguished regarding the covariance matrices: if the covariance matrices are different for each class, then we have a QDA (quadratic discriminant analysis) classifier, while if they are identical for the different classes, it becomes an LDA (linear discriminant analysis) classifier.

3.2.2. Fisher's Linear Discriminant Analysis. The underlying idea of Fisher's linear discriminant analysis (FLDA) is to determine the directions in the multivariate space which allow the best discrimination between the sample classes. FLDA is based on a common covariance estimate and finds the most dominant direction and afterwards searches for "orthogonal" directions with the same property. The technique can extract at most $J - 1$ components.

This technique identifies the first discriminating component based on finding the vector $\mathbf{a}$ that maximizes the discrimination index given as

$$\frac{\mathbf{a}^T\mathbf{B}\mathbf{a}}{\mathbf{a}^T\mathbf{W}\mathbf{a}} \tag{17}$$

with $\mathbf{B}$ denoting the interclass sum-of-squares matrix and $\mathbf{W}$ the intraclass sum-of-squares matrix.

4. Results

In the following, we will explore the results of the previously described features' sets from different classification techniques. The results will elucidate the descriptive power of several tumor features for small lesion detection and diagnosis.

4.1. Effectiveness of Krawtchouk Moments. The Krawtchouk moments describe a representation of local shape parameters and can thus describe the differences in morphology between benign and malignant tumors. Since the obtained number of Krawtchouk moments is very high (>200), we reduced their dimension based on principal component analysis (PCA). Table 1 shows the results for the Krawtchouk moments for different classifiers and number of principal components. In general, the quadratic discriminant analysis shows the best results and for PC >10 they tend to deteriorate.

Table 1: Classification results based on Krawtchouk moments for different principal components (PC). Abbreviations: linear discriminant analysis (LDA), naive Bayes linear discriminant analysis (N.B.LDA), quadratic discriminant analysis (QDA), naive Bayes quadratic discriminant analysis (N.B.QDA), and Fisher's linear discriminant (FLDA).

PC	Correctly classified (%)				
	LDA	N.B.LDA	QDA	N.B.QDA	FLDA
1	71.0	71.0	64.5	64.5	71.0
2	71.0	74.2	58.1	64.5	74.2
3	64.5	67.7	67.7	58.1	67.7
4	64.5	64.5	74.2	71.0	64.5
5	64.5	64.5	74.2	64.5	64.5
6	61.3	64.5	**77.4**	64.5	61.3
7	67.7	74.2	74.2	67.7	71.0
8	71.0	74.2	74.2	67.7	71.0
9	61.3	74.2	71.0	67.7	61.3
10	61.3	74.2	74.2	67.7	61.3
11	58.1	67.7	71.0	67.7	58.1

Table 2: Combined classification of the feature groups and different classification methods. Abbreviations: linear discriminant analysis (LDA), naive Bayes linear discriminant analysis (N.B.LDA), quadratic discriminant analysis (QDA), naive Bayes quadratic discriminant analysis (N.B.QDA), and Fisher's linear discriminant (FLDA).

Features	Correctly classified (%)				
	LDA	N.B.LDA	QDA	N.B.QDA	FLDA
Contour features	64.5	74.2	67.7	77.4	64.5
Scaling index features	67.7	71.0	61.3	51.6	67.7
Tumor RSIE features	64.5	74.2	74.2	77.4	64.5
Contour RSIE features	64.5	74.2	54.8	54.8	64.5
Geometric moments	51.6	54.8	51.6	64.5	51.6
Krawtchouk moments	71.0	74.2	**77.4**	71.0	74.2

4.2. Effectiveness of Combined Feature Groups. We now examine not anymore every single feature but group the features together in specific classes that contain the features described in the previous sections. Table 2 shows the results for five distinct classifiers assuming motion compensation in 2 directions. The Krawtchouk moments (reduced to a six-dimensional vector by PCA) yield the best results since they capture both local and global shape properties.

We perform receiver operating characteristic (ROC) analysis to determine the sensitivity, specificity, and area under the curve (AUC) of the CAD system. The results of the sensitivity and specificity for the current data set based on specific features selected based on their discrimination capability and also in combination are shown in Table 3. The scaling index entropy yields the highest sensitivity and the 5th geometric moment the highest specificity. This finding is not surprising since the scaling index is a spatio-temporal feature while the geometric moment is averaging over the tumor's shape. Since benign lesions tend to have smoother surfaces than malignant, this feature can be used as a first-step discriminator between those lesions. The inclusion of geometric moments in the feature set increases the sensitivity but leaves the specificity unchanged.

Table 3: Sensitivity and specificity for specific features alone and in combination based on linear naive Bayes classification.

Features	True positive (%)	True negative (%)
Contour feature (radius standard deviation)	70.5	85.7
Scaling index features (entropy)	**82.3**	57.1
Tumor RSIE features (entropy (time scan 4))	76.4	71.4
Contour RSIE features (entropy (time scan 4))	76.4	71.4
Slope of RSIE	76.4	64.3
Geometric 5th moment	71.6	**100**
Bayes classification without geometric moments	76.4	78.5
Bayes classification with geometric moments	88.2	78.5

Table 4: AUC values for selected single features and all features combined based on an FLDA classification.

Features	AUC (%)
Contour feature (radius mean)	**82.6**
Scaling index features (mean)	79.6
Tumor RSIE features (entropy (time scan 3))	81.5
Contour RSIE features (entropy (time scan 4))	81.3
Slope of RSIE	72.7
FLDA classification with all features	**84.7**

The best AUC-values for single features as well as for all features combined can be found in Table 4.

The AUC-values demonstrate that the contour features are very powerful descriptors and are able to capture the spatio-temporal behavior of small lesions.

5. Conclusion

The goal of the presented study was the introduction of new techniques for the automatic evaluation of dynamic MR mammography in *small lesions* and is motivated to increase specificity in MRI and thus improve the quality of breast MRI postprocessing, reduce the number of missed or misinterpreted cases leading to false-negative diagnosis.

Several novel lesion descriptors such as morphological, kinetic and spatio-temporal are applied and evaluated in context with benign and malignant lesion discrimination. Different classification techniques were applied to the classification of the lesions. A surprisingly low number of eight features proved to contain relevant information and achieved for both Fisher's LDA and LDA good classification results. Krawtchouk moments proved to capture both the local and global shape features and represent thus in term

of classification the best shape descriptors. In terms of spatio-temporal features, the scaling index entropy yields the highest sensitivity demonstrating that the enhancement pattern in small lesions has to be analyzed both in terms of spatial and temporal information. The benign characteristics are best described by geometric moments. The AUC-values demonstrate that the contour features can capture very well the spatio-temporal behavior of these small lesions.

The results suggest that quantitative diagnostic features can be employed for developing automated CAD for small lesions to achieve a high detection and diagnosis performance. The performed ROC-analysis shows the potential of increasing the diagnostic accuracy of MR mammography by improving the sensitivity without reduction of specificity for the data sets examined.

Acknowledgments

The research was supported by NIH Grant 5K25CA106799-05 and by an Alexander von Humboldt Fellowship.

References

[1] S. G. Orel, M. D. Schnall, C. M. Powell et al., "Staging of suspected breast cancer: effect of MR imaging and MR-guided biopsy," *Radiology*, vol. 196, no. 1, pp. 115–122, 1995.

[2] C. K. Kuhl, P. Mielcareck, S. Klaschik et al., "Dynamic breast MR imaging: are signal intensity time course data useful for differential diagnosis of enhancing lesions?" *Radiology*, vol. 211, no. 1, pp. 101–110, 1999.

[3] M. D. Schnall, S. Rosten, S. Englander, S. G. Orel, and L. W. Nunes, "A combined architectural and kinetic interpretation model for breast MR images," *Academic Radiology*, vol. 8, no. 7, pp. 591–597, 2001.

[4] B. K. Szabó, P. Aspelin, M. Wiberg, and B. Bone, "Dynamic MR imaging of the breast. Analysis of kinetic and morphologic diagnostic criteria," *Acta Radiologica*, vol. 44, no. 4, pp. 379–386, 2003.

[5] A. P. Schouten van der Velden, C. Boetes, P. Bult, and T. Wobbes, "The value of magnetic resonance imaging in diagnosis and size assessment of in situ and small invasive breast carcinoma," *American Journal of Surgery*, vol. 192, no. 2, pp. 172–178, 2006.

[6] G. M. Grimsby, R. Gray, A. Dueck et al., "Is there concordance of invasive breast cancer pathologic tumor size with magnetic resonance imaging?" *The American Journal of Surgery*, vol. 198, no. 4, pp. 500–504, 2009.

[7] M. J. Stoutjesdijk, J. J. Fütterer, C. Boetes, L. E. Van Die, G. Jager, and J. O. Barentsz, "Variability in the description of morphologic and contrast enhancement characteristics of breast lesions on magnetic resonance imaging," *Investigative Radiology*, vol. 40, no. 6, pp. 355–362, 2005.

[8] I. M. A. Obdeijn, C. E. Loo, A. J. Rijnsburger et al., "Assessment of false-negative cases of breast MR imaging in women with a familial or genetic predisposition," *Breast Cancer Research and Treatment*, vol. 119, no. 2, pp. 399–407, 2010.

[9] G. D. Tourassi, R. Vargas-Voracek, D. M. Catarious, and C. E. Floyd, "Computer-assisted detection of mammographic masses: a template matching scheme based on mutual information," *Medical Physics*, vol. 30, no. 8, pp. 2123–2130, 2003.

[10] G. D. Tourassi, B. Harrawood, S. Singh, and J. Y. Lo, "Information-theoretic CAD system in mammography: entropy-based indexing for computational efficiency and robust performance," *Medical Physics*, vol. 34, no. 8, pp. 3193–3204, 2007.

[11] G. D. Tourassi, R. Ike, S. Singh, and B. Harrawood, "Evaluating the effect of image preprocessing on an information-theoretic CAD system in mammography," *Academic Radiology*, vol. 15, no. 5, pp. 626–634, 2008.

[12] L. Hadjiiski, B. Sahiner, and H. Chan, "Evaluating the effect of image preprocessing on an information-theoretic cad system in mammography.," *Current Opinion in Obstetrics and Gynecology*, vol. 18, no. 7, pp. 64–70, 2006.

[13] M. A. Kupinski and M. L. Giger, "Automated seeded lesion segmentation on digital mammograms," *IEEE Transactions on Medical Imaging*, vol. 17, no. 4, pp. 510–517, 1998.

[14] S. Behrens, H. Laue, M. Althaus et al., "Computer assistance for MR based diagnosis of breast cancer: present and future challenges," *Computerized Medical Imaging and Graphics*, vol. 31, no. 4-5, pp. 236–247, 2007.

[15] A. Hill, A. Mehnert, S. Crozier, and K. McMahon, "Evaluating the accuracy and impact of registration in dynamic contrast-enhanced breast MRI," *Concepts in Magnetic Resonance B*, vol. 35, no. 2, pp. 106–120, 2009.

[16] N. Papenberg, A. Bruhn, T. Brox, S. Didas, and J. Weickert, "Highly accurate optic flow computation with theoretically justified warping," *International Journal of Computer Vision*, vol. 67, no. 2, pp. 141–158, 2006.

[17] B. Horn and B. Schunck, *Determining optical flow*, 1981.

[18] F. Steinbruecker, A. Meyer-Baese, A. Wismueller, and T. Schlossbauer, "Application and evaluation of motion compensation technique to breast mri," in *Proceedings of the Evolutionary and Bio-Inspired Computation: Theory and Applications III*, vol. 7347 of *Proceedings of SPIE*, pp. 73470J–73470J-8, 2009.

[19] D. Xu and H. Li, "Geometric moment invariants," *Pattern Recognition*, vol. 41, no. 1, pp. 240–249, 2008.

[20] A. Mademlis, A. Axenopoulos, P. Daras, D. Tzovaras, and M. G. Strintzis, "3D content-based search based on 3D Krawtchouk moments," in *Proceedings of the 3rd International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT '06)*, pp. 743–749, June 2006.

[21] P. T. Yap, R. Paramesran, and S. H. Ong, "Image analysis by Krawtchouk moments," *IEEE Transactions on Image Processing*, vol. 12, no. 11, pp. 1367–1377, 2003.

[22] F. Jamitzky, R. W. Stark, W. Bunk et al., "Scaling-index method as an image processing tool in scanning-probe microscopy," *Ultramicroscopy*, vol. 86, no. 1-2, pp. 241–246, 2001.

[23] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*, Academic Press, 1998.

Hindawi Publishing Corporation
Advances in Artificial Neural Systems
Volume 2012, Article ID 703878, 20 pages
doi:10.1155/2012/703878

Research Article

Modelling Biological Systems with Competitive Coherence

Vic Norris, Maurice Engel, and Maurice Demarty

Theoretical Biology Unit, EA 3829, Department of Biology, University of Rouen, 76821 Mont-Saint-Aignan, France

Correspondence should be addressed to Vic Norris, victor.norris@univ-rouen.fr

Received 8 February 2012; Revised 16 April 2012; Accepted 24 April 2012

Academic Editor: Olivier Bastien

Many living systems, from cells to brains to governments, are controlled by the activity of a small subset of their constituents. It has been argued that coherence is of evolutionary advantage and that this active subset of constituents results from competition between two processes, a Next process that brings about coherence over time, and a Now process that brings about coherence between the interior and the exterior of the system at a particular time. This competition has been termed competitive coherence and has been implemented in a toy-learning program in order to clarify the concept and to generate—and ultimately test—new hypotheses covering subjects as diverse as complexity, emergence, DNA replication, global mutations, dreaming, bioputing (computing using either the parts of biological system or the entire biological system), and equilibrium and nonequilibrium structures. Here, we show that a program using competitive coherence, Coco, can learn to respond to a simple input sequence 1, 2, 3, 2, 3, with responses to inputs that differ according to the position of the input in the sequence and hence require competition between both Next and Now processes.

1. Introduction

The quest for universal laws in biology and other sciences has led to the development—and sometimes the acceptance—of concepts such as tensegrity [1], edge of chaos [2, 3], small worlds [4], and self-organised criticality [5]. This quest has also led to the pioneering (N, k) model developed by Kauffman in which N is the number of nodes in an arbitrarily defined Boolean network and k is the fixed degree of connectivity between them [6]. The actual use of the (N, k) model to the microbiologist, for example, is that it might help explain how a bacterium negotiates the enormity of phenotype space so as to generate a limited number of reproducible phenotypes on which natural selection can act. Although the (N, k) model successfully generates a small number of short state cycles from an inexplorable vast number of combinations—which might be equated to generating a few phenotypes from the vast number apparently available to the cell—the model has its limitations for the microbiologist as, for example, it has a fixed connectivity, it does not evolve, and it does not actually do anything.

In a different attempt to find a universal law in biology, one of us began working on the idea of network coherence in the seventies. This idea is related to neural networks (though the idea was developed with no knowledge of them) which have indeed been proposed as important in generating phenotypes [7]. The network coherence idea turned out to be scale-free and to address one of the most important problems that confronts bacteria, eukaryotic cells, collections of cells (including brains), and even social organisations. This problem is how a system can behave in (1) a coherent way over time so as to maintain historical continuity and (2) a coherent way at a particular time that makes sense in terms of both internal and environmental conditions. A possible solution would be for these systems to operate using the principle of *competitive coherence* [8, 9]. Competitive coherence can be used to describe the way that a key subset of constituents—the Active Set—are chosen to determine the behaviour of an organisation at a particular level. This choice results from a competition for inclusion in the Active Set between elements with connections that confer coherence over time (i.e., continuity) and connections that confer internal and external coherence at the present time. Given that living systems are complex or rather *hypercomplex* systems, characterised by emergent properties, we have speculated that competitive coherence has parameters useful for clarifying emergent properties and, perhaps, for classifying and quantifying types of complexity

[10]. We have further argued that competitive coherence might even be considered as the hallmark of life itself.

One of the objections to universal laws such as competitive coherence is that without testing they are mere handwaving. To try to overcome this objection and to make the central idea and related ideas clear, we have implemented competitive coherence into a toy-learning program in which the competitive coherence part, Coco, learns or fails to learn when playing against an environment that rewards or punishes Coco by changing connections between elements. In what follows, we describe the structure of the program, results from this toy version, and optional extras that could be added to or manipulated within the program. Some of these optional extras are wildly speculative but they are included to show that Coco might be useful as a generator, editor, and test-bed for new and/or woolly concepts including those that might find a home in biology-based computing [11].

2. Principle of the Program

The central idea is that an *active* subset of elements determines the behaviour of a system: the majority of elements are inactive. The members of this active subset, the size of which is fixed, are chosen by a competition between two processes, *Next* and *Now* (Figure 1). The active subset corresponds to those elements that have their addresses in the current line of the Activity Register. The subset of elements that are active at the same time constitutes the state of the system at that time and each line in the Activity Register says what state is at a particular time; consecutive lines correspond to consecutive states of the system. How is this state obtained? An overview of the program is given in Figure 2 to show the order of the essential subroutines, which are explained in detail in the following subsections. COMPUTE is the cyclical core of the system. INITIALISE sets up the initial conditions by filling at random the Now and Next fields (two separate sets of weightings) of each element with the addresses of other elements, and by filling at random the first line (which corresponds to the state of the system) of the Activity Register with addresses. INPUT provides one input at a time (by loading the address into the Activity Register of a specific element); these inputs are taken in order from a defined sequence. DOWNTIME prevents an element whose address has been loaded into the Activity Register from having its address loaded again within a brief time. MUTATE changes at random the contents of the Now and Next fields (in a sense, the weightings). LOAD is responsible for extracting and ranking the scores of the elements' Now and Next fields and then comparing these scores so as to decide which address to load to the Activity Register. LOAD is helped by REVERSE DOWNTIME, which stops an element likely to be useful later from being used too soon, and by COMPATIBILITY-EMERGENCE, which increases the probability that some groups of elements have their addresses loaded at the same time (corresponding to these elements being active together). REWARD-PUNISH routines detect and evaluate outputs and reward or punish the elements involved by strengthening or weakening the Now/Next links between them. The results are displayed every loop of the COMPUTE routine, which corresponds to a line in the Activity Register being filled (see the following).

2.1. The Biological Elements. An individual gene or neurone or another biological building block is represented in the program as an element. In the version presented here, there are a thousand elements. Only ten elements (again in the version presented here) can be active—that is, determine the state of the system—at any one time. The composition of this subset of active elements determines whether this active subset is successful or not.

Each element contains, in the version presented here, two fields: Now and Next (Figure 3). Each element has an address. Each field contains the addresses of other elements with which the element has been associated successfully during learning (see the following). The Now field contains the addresses of those elements that have been successful in the same time step in which the element itself was successful whilst the Next field contains the addresses of those elements that have been successful in the time step following the step in which the element was active. A time step corresponds to a single line in the Activity Register, hence the active subset of elements corresponds to those elements that have their addresses in the current line of the Activity Register. In other words, each line says what the state of the system is in terms of *activity* in that particular time step.

2.2. The Activity Register. Each line in the Activity Register contains the addresses of the small subset of elements that are active at a particular time (Figure 4). Each line is filled according to a set of rules (see the following), and when the register is full, the first line is filled again (so the register operates cyclically). A line is set to zero before it is filled.

2.3. Coherence via the Now Process. Biological systems are characterised by coherence. At the level of human society, the composition of a successful football team reflects the importance of coherence in the manager's choice of players: the team must contain players who can take on the roles of goalkeeper, defenders, midfield players and attackers. Choosing the players is a progressive process: the choice of the goalkeeper influences the choice of the defenders which then influences the choice of the defenders. At the level of a bacterium, the composition of a bacterium reflects its growth strategy: if *Escherichia coli* is to grow rapidly, it needs to express the genes that encode ribosomes, tRNA synthetases, RNA polymerases, and the proteins that drive the cell cycle whereas if it is to survive in harsh conditions and in the absence of nutrients, it needs to express the genes that, for example, encode proteins that compact and protect its genome such as Dps [12]. Biological systems from sports' teams and brains to bacteria that fail to achieve this *internal* coherence risk elimination in a world in which natural selection operates. The coherence that a biological system must achieve also has an *external* component. The football manager may choose his team as a function of the

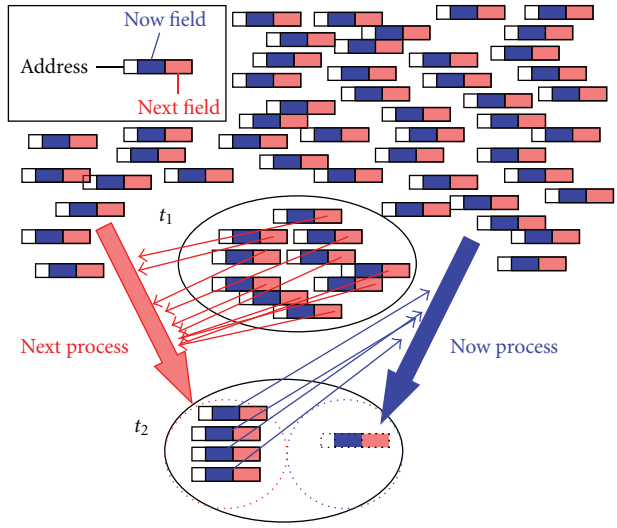


Figure 1: The principle of Now-Next competition. The elements active at time t_1 (inside black ellipse) are used to select the elements to be active at time t_2. This selection uses the Next fields of the elements active at time t_1 (red arrows). As the new elements are activated at time t_2 (inside the red dotted circle), the Now fields of these new elements are also used to select and activate more elements (blue arrows) such as the element inside the dotted blue circle. The elements to be activated are selected from a large set of inactive elements by a competition between the Now and Next processes. The inset shows an element which has an address and two fields.

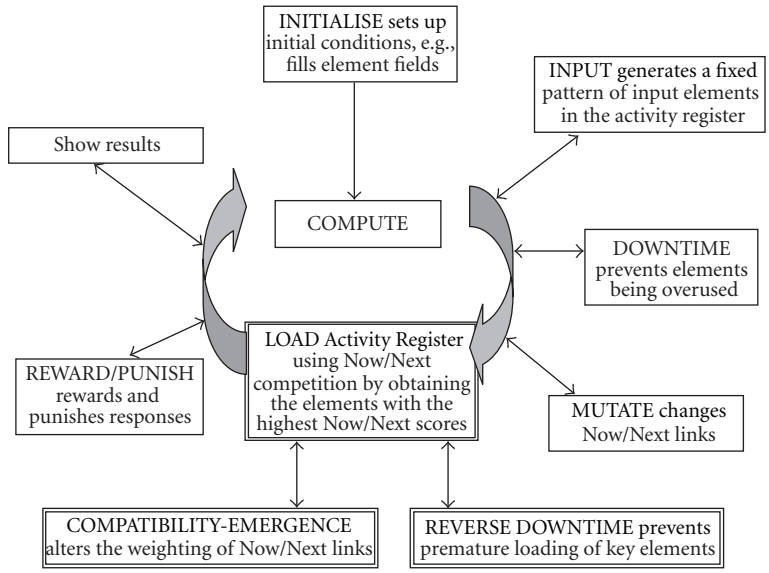


Figure 2: Overview of essential modules. A single cycle of the COMPUTE routine results in the addresses of a subset of the elements being loaded into a line of the Activity Register; this corresponds to *activating* these elements. Certain elements are inputs and others are outputs. The INPUT routine creates an input by loading the address of an input element into the Activity Register. The learning part of the program, Coco, eventually responds to an input by loading the address of an output element into the Activity Register; this corresponds to an output. The actual loading of addresses results from a competition between Now and Next links; the scores obtained from counting these links may be modified by an EMERGENCE routine. The DOWNTIME and REVERSE DOWNTIME routines may make certain elements ineligible for loading, depending on the history of these elements. Outputs are detected, evaluated, and rewarded or punished by REWARD/PUNISH routines. Each time a line of the Activity Register has been filled, the results are displayed.

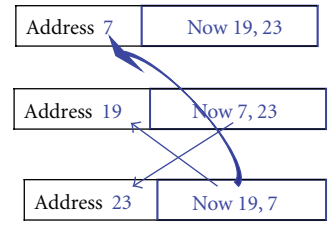
pitch, the weather, their position in a league table, and the opposing team (which may contain players known for their "physical" approach). The bacterium should not express the full complement of genes needed for fast growth if it is in conditions in which there are few nutrients and in which physical conditions such as temperature and humidity require stress responses. The Now process is intended to represent the way biological systems achieve both internal and external coherence. For example, suppose that at high temperatures, phospholipids with long, saturated fatty acids

Address	Now field					Next field				
1	878	63	50	533	119	43	227	136	19	256
2	13	218	516	923	121	724	447	225	134	15
3	611	577	488	818	63	70	337	722	297	98
4	437	316	218	726	413	124	126	117	747	299
...										
999	165	534	183	612	626	667	879	355	649	220
1000	388	431	566	435	924	654	255	818	921	33

Figure 3: The Elements table. The Now and Next fields of an element contain the addresses of the elements with which the element in question has been successfully active (for convenience of representation each field here contains only 5 addresses).

Present state t	10	5	18	11	6	4	23
Developing state $t+1$	7	22	*16*				

Figure 4: The Activity Register. Two lines are shown (corresponding to two successive active states), one full and the other (italics) in the process of being filled (for convenience, only 7 entries and only two lines are shown).



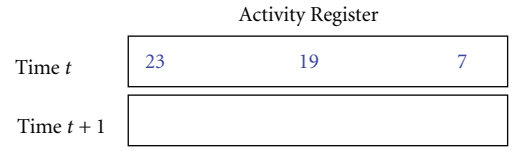


Figure 5: The Now Process. Out of a large set of elements (genes), a few, well-connected ones are chosen (positioned) to be active (expressed) at a particular time. For example, if gene 23 encodes a protein with the same lipid preferences as the proteins encoded by genes 19 and 7 the expression of these genes may be synergistic (for convenience, only two addresses are shown in the Now field).

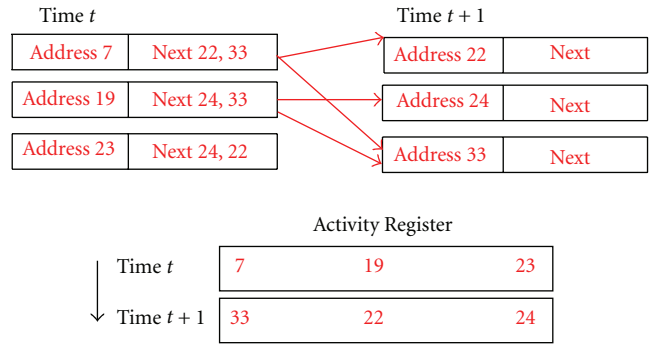


Figure 6: The Next Process. The elements (genes) active in time step t determine the elements that will be active in time step $t+1$.

are needed to confer stability to the membrane; suppose that genes 7, 19, and 23 encode membrane proteins with an affinity for such phospholipids (Figure 5); the expression of one of these genes (e.g., 23) may help to create a domain on the membrane enriched in both the protein encoded by 23 and the long chain phospholipid; the existence of this domain then helps the expression of 19 which has a protein product that also contributes to the size and stability of the domain; this in turn helps the expression of 7 (for the possible biological significance see [13]).

2.4. Coherence via the Next Process. The Now process on its own is not enough to allow a biological system to adapt effectively to its environment. This is because these environments often require different responses to the same stimulus because the historical contexts are different. One may not respond in the same way to an invitation from Human Resources following a successful sales campaign as following the news that the company is about to go out of business. A Next process is required to take account of the dependency of a correct response on the history of the system. This is made easier by the fact that environments are often unchanging for long periods and, when they do change, change in predictable ways; rules in sport, for example, do not change very often. And even when different environments exist at the same time, it would not be advisable for a sports' team that wants to compete at a good level for it to play football one week, switch to hockey the next, then rugby, and so forth. It would not be a winning strategy for a bacterium to switch phenotypes either (we ignore here events at the level of the population of bacteria which needs to explore phenotypic heterogeneity). For example, three genes that are active in one time step, 7, 19, and 23, may have connections via their Next fields to the same subset of genes, 22, 24, and 33 (Figure 6) with 7 encoding a transcriptional activator being produced in one time step and 22 and 23 being the genes under its control and hence expressed in the following time step.

2.5. Competition between the Now and Next Processes. Modelling competitive coherence in the program consists of the competition between the Now and Next processes for choosing the subset of elements that are to be active (i.e., determine the state of the system). This activation takes the form of loading the addresses of elements into the new line of the Activity Register. The competition is on the basis of the scores of the elements. To load a new line, first, the Next fields of the elements present in the current line are consulted and the number of occurrences of the addresses in these fields is counted to give Next Scores (Figure 7). These Next Scores are then ranked in HighestNext to give, in Figure 7, element 7 with the highest score (of 8).

Once the address of this first element has been loaded into the new line, the following highest score in the

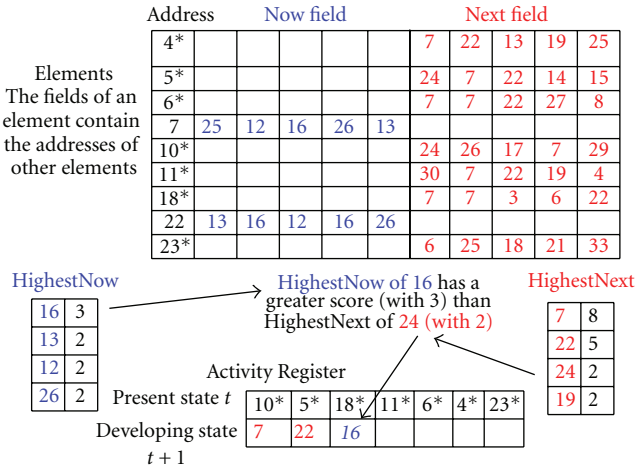


FIGURE 7: Competition between Now and Next Processes. The diagram shows the operation of filling the third position in the new line of the Activity Register (corresponding to the developing state of the system). The elements in the current line of the Activity Register are labelled with an asterisk and only their Next fields are shown. The two elements loaded into the new line at the start of the competition for third position have their Now fields in bold; their Next fields are not shown. The Now and Next scores are in the second column of HighestNow and HighestNext, respectively. See text for full explanation.

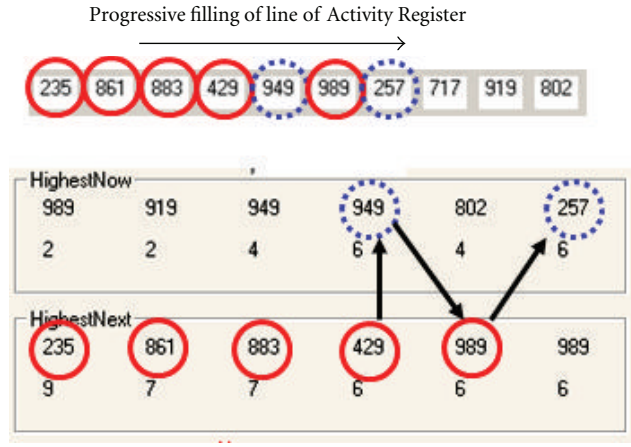


FIGURE 8: Differential Contributions of the Next and Now processes. A line of the Activity Register that has just been filled is shown along with part of the contents of the HighestNext and HighestNow counters at this time. The addresses in these counters are on the top and the corresponding scores on the bottom. The red circles show the addresses of elements contributed by the Next process and the dotted blue circles show those contributed by the Now process. The arrows show where there is a clear change in the contributions of the processes to filling the Activity Register.

HighestNext list is consulted: this is 22 with a score of 5. It is not, however, loaded straightaway because 7 has been loaded and 7 has a Now field. The addresses of the elements in 7's Now field are counted and ranked in HighestNow; in this example, the ranking is at random because the scores of these addresses are all the same (here, 1). The highest scores in HighestNext and HighestNow are then compared and the address of the element with the higher score is then loaded into the Activity Register (if they have the same score, the Now element is preferred).

There are now two addresses in the Activity Register, 7 and 22. The addresses in the Now fields of both 7 and 22 are counted and ranked to give at the top of HighestNow 16 with a score of 3. The address at the top of HighestNext is 24 with a score of 2. Comparison of the two scores shows that the address of element 16 has the greater score and this address is therefore loaded to the Activity Register. Once an address has been loaded, its score is set to zero to prevent it from being loaded a second time.

This competition between Next and Now processes continues until the line of the Activity Register has been filled. Note that, during the determination of the new state of the system, the relative contribution of the two processes changes since, first, the number of Now fields increases as addresses are progressively loaded into the Activity Register (i.e., as more and more elements become active), and, second, each time an element is chosen from the HighestNext, the following one has a lower score. In other words, the Next process dominates at the beginning and the Now process at the end (Figure 8).

2.6. Inputs and Outputs. Both inputs and outputs correspond to specific elements each of which has a Now and a Next field. In this version, there are three inputs, 1, 2, and 3, and three outputs, 998, 999, and 1000. An input is generated when the INPUT subroutine inserts one (and only one) of the three inputs into the new line of the Activity Register. An output is generated when Coco loads an output element into the new line of the Activity Register. The lines between the input line and the output line constitute an *input-output module*.

A new input is generated in the line immediately following an output. A new output must occur in the four lines following an input; if an output is not generated in these time steps, an arbitrarily chosen output is inserted into the fourth line. A maximum number of lines before forcing an output is needed if the program is to run rapidly when the proportion of output elements to the total number of elements becomes small (since the initial probability of generating an output is similarly low). The choice of four lines as a maximum is also arbitrary.

Inputs are not related to outputs: it does not matter whether the output is right or wrong. The sequence of inputs is fixed: $(1, 2, 3, 2, 3)_n$. Elements in lines containing outputs that correspond to inputs are rewarded—or punished if the outputs are wrong (see the following). Coco is considered to have succeeded when it has learnt to generate $(1000, 999, 999, 1000, 998)_n$ in response to the input sequence (Figure 9). This simple input-output relationship was chosen because neither the Now nor the Next process alone is sufficient to lead to Coco learning. The Now process fails because two different outputs—999 and 1000—are required when 2 is the input (and also because two different outputs—999 and 998—are required when 3 is the input).

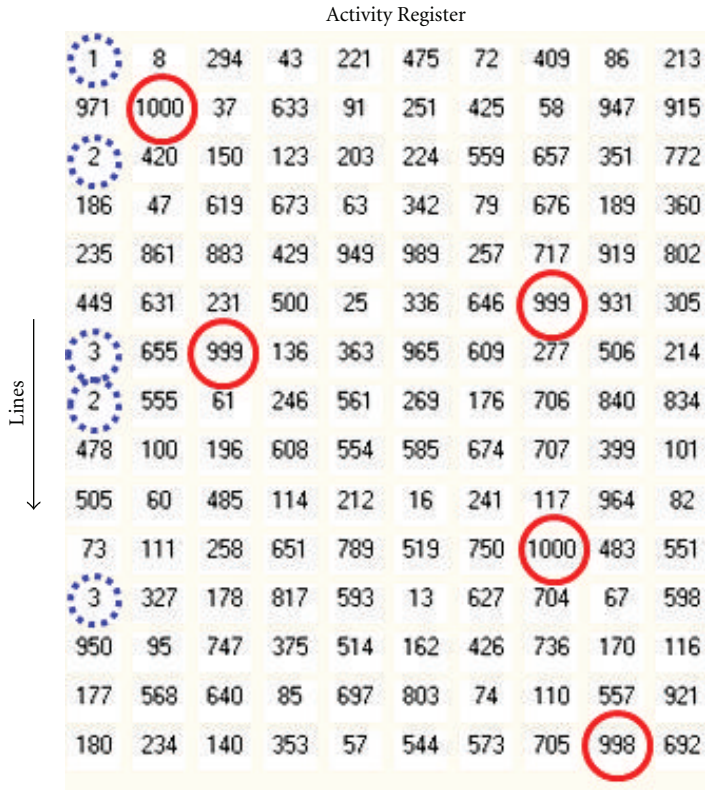


Figure 9: Contents of Activity Register after learning. The screen print shows consecutive lines in the Activity Register. Inputs are circled with dotted blue lines and outputs with continuous red lines.

The Next process fails because two different outputs—999 and 998—are required when 3 follows 2.

2.7. Downtime. An element that has been active (i.e., its address has been loaded into a line of the Activity Register) cannot be activated again for ten more timesteps. Inputs that are generated by the environment subroutine are not affected by downtimes (artefactual inputs generated by the dynamics of Coco do have downtimes). Outputs do not have downtimes. Downtimes are taken into account when the Now and Next scores are worked out. Downtime can be problem when the size of the Activity Register is increased since Coco can run out of elements to load; for example, if there are only 1000 elements, if the Activity Register has a line containing 100 elements, and if Downtime is set to 10, there are not enough elements to load. There is an echo here of the *E. coli* cell cycle in which, after initiation of chromosome replication, the constituents of the initiation hyperstructure are inactivated or sequestered to prevent a second initiation event [14], whilst after cell division, the constituents of the division hyperstructure are presumably also disabled to prevent repeated divisions in the bacterial poles [15].

2.8. Rewarding and Punishing. Rewarding entails strengthening the connections between successful states (and series of successful states). Briefly, this is achieved by (1) taking an element with its address in a line of the Activity Register between and including the input and output lines, which we term a module (see the previous part) and (2) writing this address into the Now or Next fields of other elements in the same or in the preceding lines of the Activity Register (Figure 10). First, the environmental part of the program detects whether there is an output in the new line and, if so, decides whether there is more than one output (more than one output is punished, see the following). Then, if the output is correct, two addresses are chosen from the same line in the successful module and the second address is written into the Now field of the first element. This is done for every element in the entire module (i.e., each line is treated). These connections are not made when it would entail connecting an element to itself (i.e., self-referral) or overwriting the address of another element that is actually already in the same line. The Next connections are rewarded in much the same way except that elements are taken from one line of the module and the addresses that are written into their Next fields are taken from the following line. Self-referral and overwriting of successful elements is avoided as in the case of rewarding via the Now connections. It should be noted that a connection is made between the previous module and the rewarded module insofar as the Next fields of the last line of the previous module are connected to the first line of the rewarded module (note too that the Next rewarding has to stop at the penultimate line of the rewarded module because the future line is not

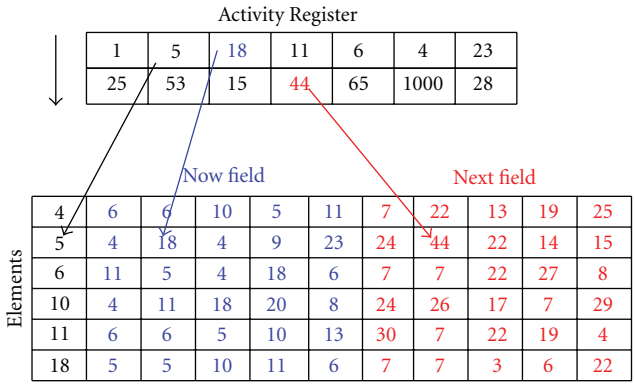


Figure 10: Rewarding by overwriting. The connections of element 5, whose address is in a line of a successful module in the Activity Register, are strengthened (black arrow). A randomly chosen address in the Now field of 5 is overwritten with 18, which is another address present in the same line as 5 (blue arrow), whilst a randomly chosen address in the Next field of 5 is overwritten with 44, which is an address present in the following line (red arrow).

known!). A small section of the elements, along with a couple of lines of the Activity Register and the HighestNow and HighestNext registers (Figure 11) shows the pattern of connectivity resulting from learning to couple an input with the appropriate output for Coco with a 1000 elements, Anumber of 10 (size of Active Set) and Knumber of 7 (size of Now and Next fields).

Punishing entails taking a *single* Activity Register line at random from the failed input-output module or taking the last line of the previous module. The Now fields of the elements whose addresses are in this line are then consulted. A randomly chosen address is then used to overwrite one of these Now addresses. The Next fields of these elements (which helped determine the following line) are altered in a similar way. There is a mutation aspect to punishment (Section 4.3).

2.9. Synchrony. There is a synchrony in the program that results from it being based on successive lines in the Activity Register, each of which is examined in a time step. Reward and punish decisions are then made in this time step to change the connections between the elements by altering the contents of their Now and Next fields; these alterations are made together in a synchronous fashion. The actual loading of addresses into a new line is a mixture of synchronous and asynchronous events: the Next scores are obtained together at the end of one time step (and the process exhibits synchrony) whilst the Now scores are obtained progressively during the loading of the Activity Register (and the process exhibits asynchrony).

3. Results

3.1. With Next Alone. The relative contributions of Now and Next scores to the loading of addresses into the Activity Register can be altered by a constant factor, NowNextWeighting. If this factor is set very low or very high, it switches the program so that it operates with either just Nexts or just Nows. Generally, this factor equals 1. By setting NowNextWeighting to 1/100, the Next scores dominate. The graphs show the fraction (total outputs-correct outputs)/total outputs, so a line towards the top of the figure corresponds to a failure to learn effectively. When the Nexts alone determine the loading of the Activity Register (red circles), learning does not occur (Figure 12). Note that only a truncated part of HighestNext and HighestNow is shown and that the scores shown in the bottom row are before scaling by NowNextWeighting.

3.2. With Now Alone. Setting NowNextWeighting to 100 allows the Now scores to dominate. The graphs in Figure 13 show the proportion of incorrect outputs (failures to learn) to total outputs, so a series of successful outputs corresponds to a negative slope. In one case, learning has not occurred after 20000 timesteps (Figure 13(a)). In another case, learning has actually occurred at a late stage (Figure 13(b)). The explanation is that the Next scores are still operating even with this NowNextWeighting because the first address chosen for the NewLine of the Activity Register is chosen from the Next element with the HighestNext score (unless an input is loaded). This initial choice does not depend on the NowNextWeighting. The Activity Register corresponding to this surprising learning confirms that the Nows have contributed the addresses (compare contents of last line of Activity Register and with contents of HighestNow and HighestNext in Figure 13(c)).

3.3. Without Downtime. The address of an element that has loaded into the Activity Register cannot be loaded again within the next ten timesteps. In the absence of DOWNTIME, Coco does not learn (Figure 14(a)). One evident reason for this is that false inputs can be loaded into the Activity Register (Figure 14(b)).

3.4. With Now, Next, and Downtime. Three independent runs of the program show that Coco learns the task albeit sometimes with difficulty (Figure 15). Inspection of the Activity Register and the HighestNext and HighestNow confirms that the initial contribution to the last line in the register comes first from the Next (red circles) and then from the Now (blue circles). Note that only a truncated part of HighestNext and HighestNow is shown. Note too the limited size of the Activity Register (Section 4.14). What might DOWNTIME correspond to in a biological system? In a bacterium, it might correspond to the state of a gene x in an operon which requires an activator but which also encodes an unstable repressor Y of this operon; activating the gene would then lead to both production of X and of Y; Y would then switch off the operon for a Downtime until it was degraded. A more interesting possible example is that of the sequestration of newly replicated, hemimethylated DNA in *E. coli* (which occurs when the SeqA protein recognises that a GATC sequence in the old strand is methylated whilst its complement in the new strand has yet to be replicated); since a gene may have a greater chance of being expressed when

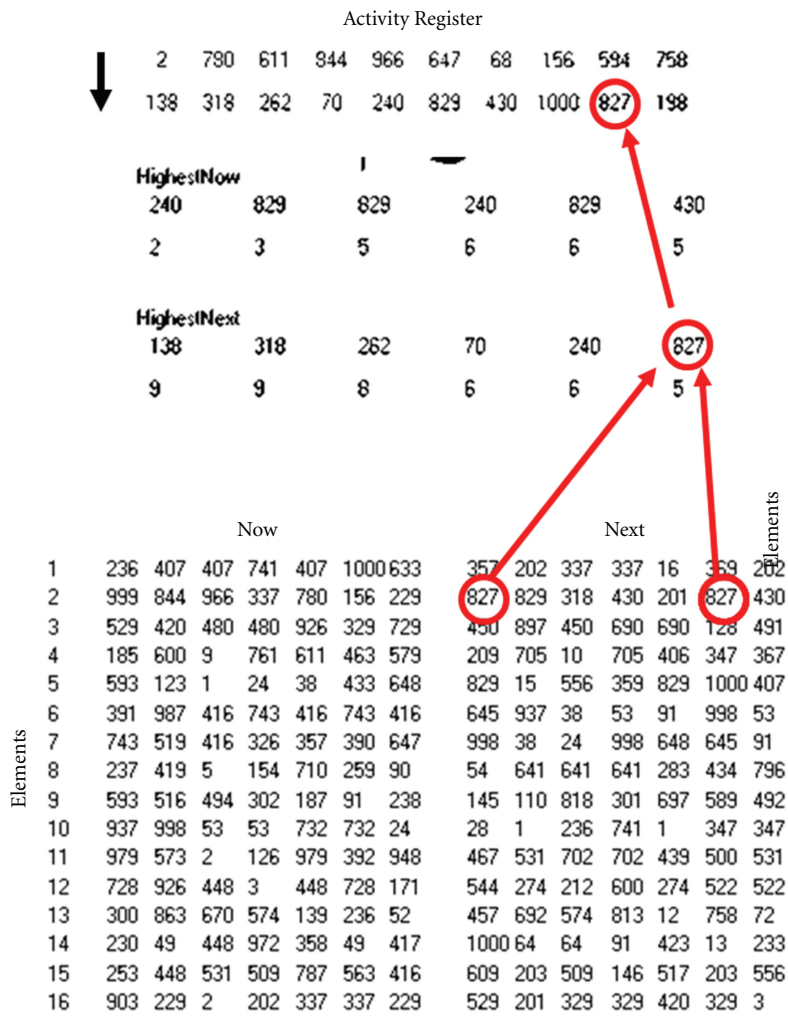


Figure 11: The connectivity after successful learning. An input line and the following, correct, output line in the Activity Register are shown. The total scores of some of the elements whose addresses are in the Next fields of the elements in the top line of the Activity Register can be seen in the bottom row of HighestNext. If the Next field of element 2 is inspected, two references to element 827 are found; these two references are part of the total of five references to element 827; this score of five is sufficient for the address of element 827 to be loaded into the new line of the Activity Register (so activating element 827). Note the presence of the output itself, 1000.

the region within which it lies is being replicating (perhaps due to greater accessibility to RNA polymerase), some genes with GATC sequences may be both switched on and switched off by the act of replication. This latter possibility might be modelled specifically by confining DOWNTIME to those elements that are activated by the CYCLE subroutine, as mentioned in Section 4.2. What then might the inputs and outputs correspond to in a biological system? As shown here, Coco readily learns to give the same output to different inputs. It also learns to give a different output to the same input depending on the history of the inputs; this could correspond to a low concentration of nutrients having a different meaning for a bacterium if this concentration follows a period of starvation or a period of plenty; in the former case it means that conditions are improving and in the latter case it means that they are getting worse, and the appropriate response of the bacterium would be to grow or to sporulate, respectively.

3.5. An Oscillatory Input Gives an Oscillatory Output. The environment gives the inputs (1, 2, 3) in a cycle $(1, 2, 3, 2, 3)_n$ and immediately Coco responds with an output the environment gives the next input. Hence, Coco learns to respond to an oscillating input pattern with an oscillating output pattern. This can be confirmed after learning has occurred by removing inputs and following the pattern active elements (i.e., the addresses in the Activity Register). It does indeed maintain the output pattern in the absence of inputs and absence of changes to connectivity normally caused by rewarding, punishing, and noise (data not shown). This shows the extent to which the state of the system reflects the interdependency of the elements loaded to the Activity Register. Note though that if the Activity register is small (low Anumber) and the connections are weak (low Knumber), inputs must be continued (along with rewarding) to maintain the oscillating output pattern. We discuss further the significance of

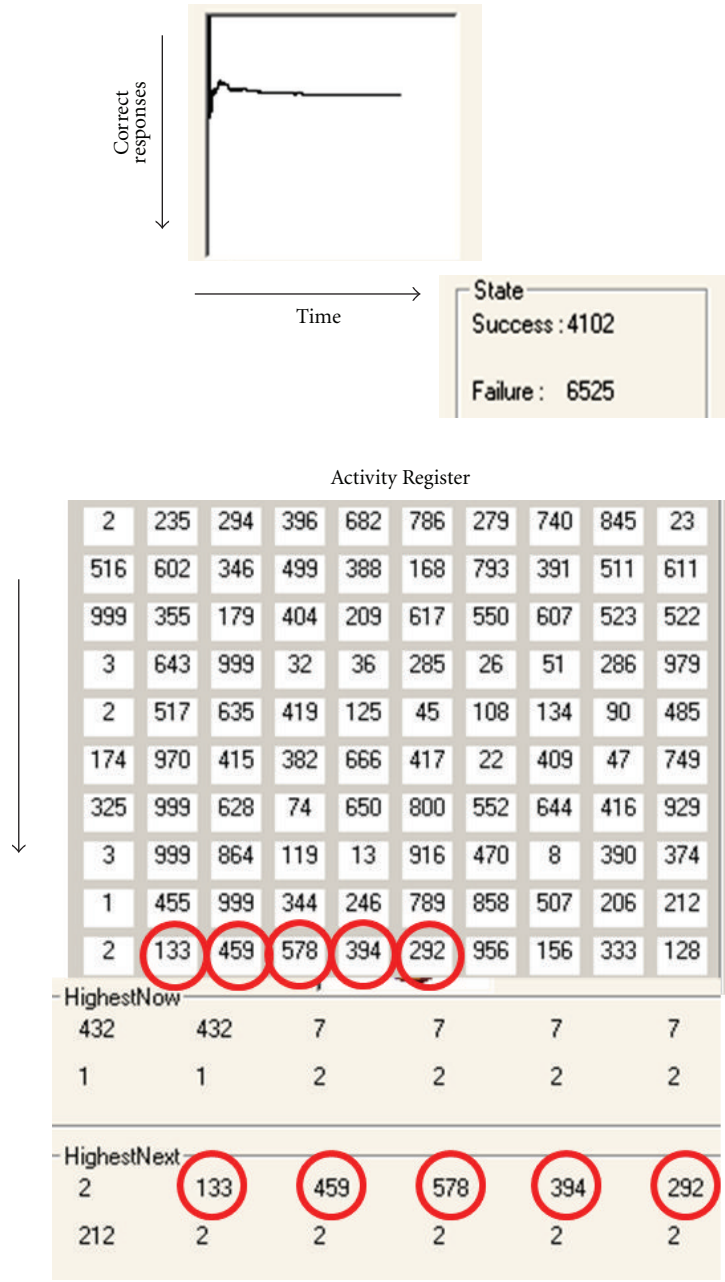


Figure 12: Running the program depending on the Nexts. No learning has occurred after 20000 time steps (the time taken to fill, analyse, and respond to a single line in the Activity Register); the contents of the Activity Register and the HighestNow and HighestNext confirm the Next contribution (red circles). Correct responses is (total number of responses–correct responses)/total number of responses. Note that the total number of successes and failures (10627) would only equal the number of timesteps or lines (here 20000) if the program were constrained to give an output in the same line as it had received an input (rather than up to four lines later).

a noncyclical input-output pattern—and how to achieve this—in Section 4.18.

4. Optional Extras

4.1. Positive and Negative Links. Biological systems generally have both activators and repressors. Simulation suggests that the ratio between them is a major influence on the dynamics [16]. In the program, elements can be connected positively and negatively. When the Now and Next scores are calculated, each address present in a field with a positive link counts as +1 whilst an address with a negative link counts as −1. The positive or negative nature of a link is defined at random at the start of the program and is not modified afterwards. In the present version, 10% of the signs are negative (though it learns when none of them are negative).

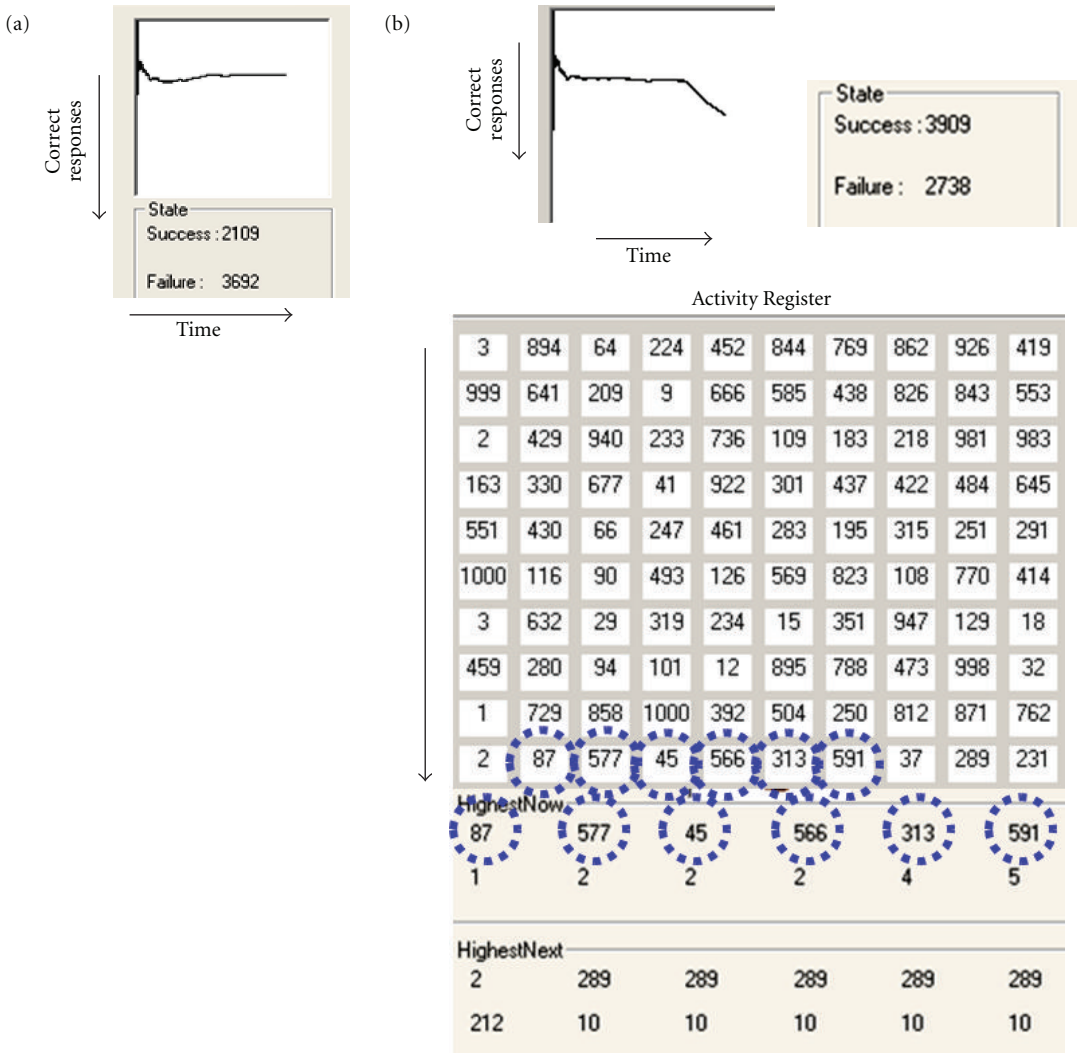


Figure 13: Running the program depending on the Nows. (a) No learning has occurred after 20000 timesteps, and (b) Learning has been delayed in this case after 20000 timesteps and the contents of the Activity Register and the HighestNow and HighestNext confirm the Now contribution (dotted blue circles).

4.2. DNA Replication. Growing bacteria replicate their DNA. It has been proposed that the replication of a gene affects the probability that the gene and its physical neighbours may have an altered probability of transcription [17, 18]. One consequence of this could be to enable the bacterium to avoid getting trapped in a very limited state cycle of phenotypes. In other words, DNA replication itself might constitute a way of exploring phenotype space. Such exploration could even constitute a coherent exploration of phenotype space if the position of the gene on the chromosome were close to those of other genes with related functions (and far from those with opposed functions). To introduce this parameter into the program, a CYCLE subroutine allows an element to be loaded into the Activity Register irrespective of the Now and Next connections. This element is chosen in order from the elements (e.g., first 17 is inserted, then 18, then 19, etc.). It is not inserted into the Activity Register if Coco has just given the correct output (which corresponds in this version of the program to CyclePermission = 0).

4.3. Mutations and Noise. Mutations occur as part of the PUNISH subroutines. In fact, the overwriting of the Now and Next fields (of the elements with addresses in the line to be punished) is a mutation process insofar as the new addresses that are written into these fields are chosen at random. This overwriting is done at a frequency determined by the MutationThreshold which, in the version presented here, is set so that overwriting occurs on one out of ten occasions.

There is more to the mutation story than this though. After the program has run for 200 timesteps, a RunningScore is kept of how many of the last ten outputs have been correct (this involves a sliding window, RunningScoreWindow, set to ten). Depending on this RunningScore, mutations are either made at different frequencies or not made at all. A mutation is made by taking an element with an address that is rarely found in the fields of the other elements (i.e., a lonely element) and writing it into a field of any one of the other elements chosen at random. This is done ten

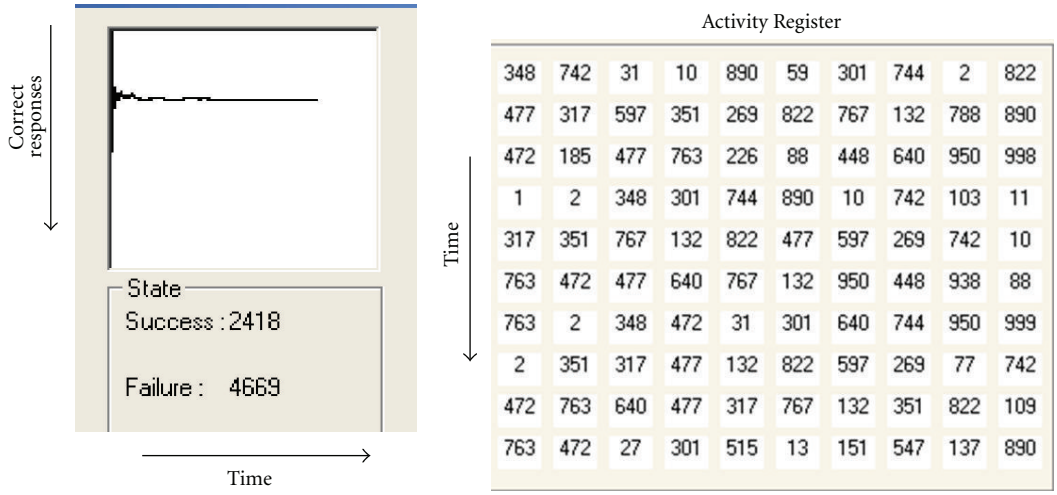


Figure 14: No learning without DOWNTIME. (a) After 20000 timesteps, Coco shows no sign of learning. (b) The Activity Register shows that inputs are being generated inappropriately.

times per output if the RunningScore is low. If the last ten outputs were all correct, no mutations are made. Clearly, RunningScoreWindow is a parameter that the program itself could modify during running but, in the version presented here, it is held constant at ten.

Noise is not present in the basic version of the program presented here but is easy to introduce. For example, in Figure 16 the NoiseLevel has been set to insert a random address for every twenty or so (on average) addresses loaded into the Activity Register (i.e., around every three lines). The results show that the learning displayed by Coco is fairly robust and, even when it is forced to *forget*, it can relearn rapidly.

Finally, a SCRAMBLE subroutine may be a source of *mutations*. This routine operates in a democratic way to ensure that when elements have the same score, they are chosen at random to be ranked in HighestNow and HighestNext. This may result in an address being loaded into a position in the Activity Register that it had not occupied previously and bring to an end a winning streak, at least, temporarily. The fact that learning is often stable despite scrambling is again indicative of the robustness of this learning.

4.4. Reverse Downtime. Input-output modules can readily become compressed so that, for example, an input in line n of the Activity Register followed by an output in line $n + 4$ can be shortened such that the output occurs in line $n + 2$ or indeed in line n itself (Figure 17). The idea behind REVERSE DOWNTIME is to avoid this compression by preventing the addresses of elements in line $t + 1$ from being loaded into the previous line t. This entails using the REVERSE DOWNTIME subroutine to examine progressively the elements of addresses loaded into the NewLine and ensuring that the addresses in their Nexts, which may correspond to elements often active in the following line, are not loaded via the Now process. Preventing such addresses from being loaded is done via the NOW EXTRACTION subroutine which sets their scores to zero (in this version, it is not done by the NEXT EXTRACTION subroutine too—but could be). Although it is not clear to us that the REVERSE DOWNTIME subroutine has an equivalent in cells, it might be argued that REVERSE DOWNTIME resembles checkpoints, which prevent late cell cycle events from occurring until earlier ones have been completed [19].

4.5. Uptime. It would be easy to introduce an uptime in which an element that had already participated successfully would have a greater chance of being loaded again. This would be a variant of the Matthew effect in which the rich get richer (and the poor, poorer), which has been explored in connectivity studies [20].

4.6. Emergence. One of the characteristics of an emergent property is that it resists attempts to predict or deduce it [21]. An emergent property could be, for example, the affinity of certain membrane proteins for the phospholipid, cardiolipin, so that they assemble into a membrane domain enriched in cardiolipin where these proteins then function together. In the framework of competitive coherence, emergence is related to the formation of the new state, the subset of elements that are active together because their addresses are loaded together into the Activity Register [10]. Suppose that a subset of the elements (e.g., 27, 37, 47, 57, 67, and 77) correspond to proteins with a strong affinity for cardiolipin. Similarly, another subset of elements (e.g., 23, 33, 43, 53, 63, and 73) might correspond to proteins with an affinity for another phospholipid, phosphatidylethanolamine. This could be done for a hundred different phospholipids. In the program, these affinities could correspond to a hardwiring done in the INITIALISE subroutine such that the probability that 27 and 37 and so forth are loaded into the Activity

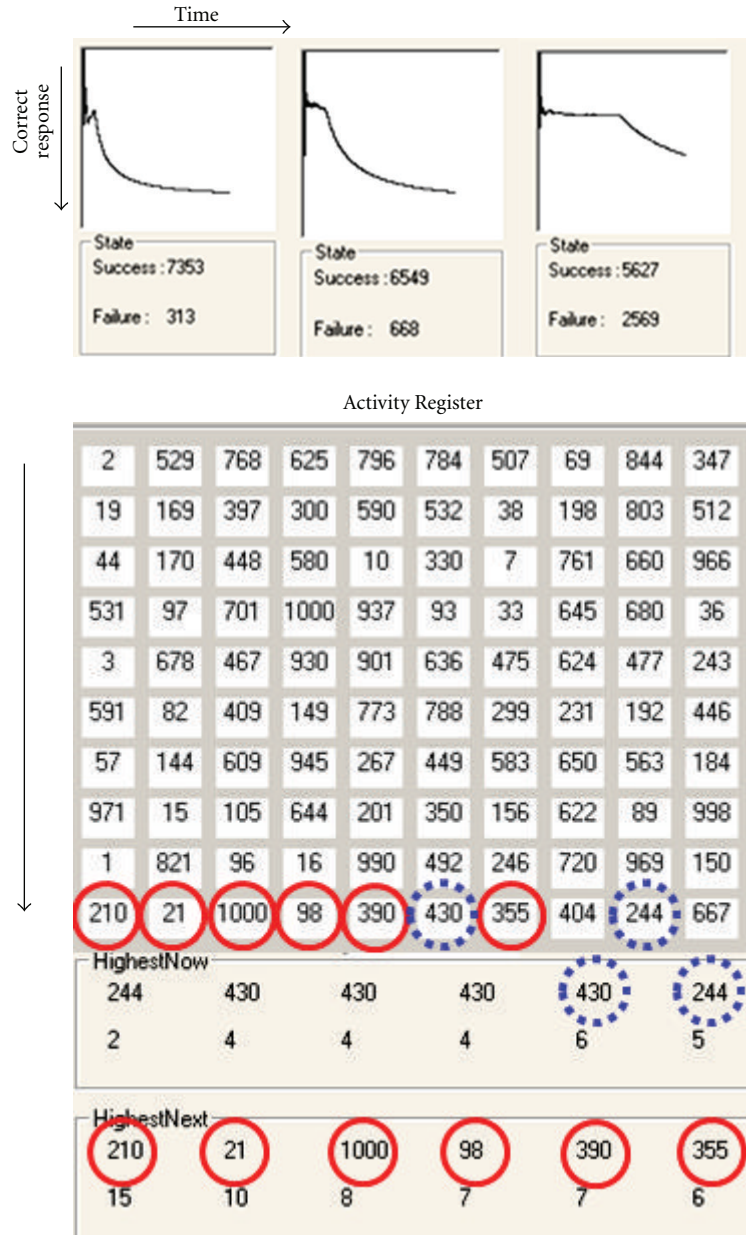


Figure 15: Successful learning with Now, Next, and Downtime processes working. Three, consecutive, independent examples are shown after 20000 timesteps. The contents of the Activity Register, HighestNext, and HighestNow of one of them are shown (now addresses are circled with a dotted blue line and Next addresses with a red line).

Register is greater if one of them is already present. If this combination of elements turns out to be a successful one, this might be considered as the emergence of the property of an affinity for cardiolipin.

The aforementioned approach to emergence can be modelled to some extent in the program via a Compatibility Table, which is a 2D matrix of the elements in which the Compatibility of Element(i) $\times$ Element(j) is a factor. During the loading of the Activity Register, this factor is used to multiply the Now and Next scores of the elements so as to help determine which is the highest. The factor in (i, j) is determined once and for all at the start of the program (i.e., it is hardwired). It would be possible to have a large number of sets of factors (e.g., a hundred sets of factors each linking ten elements) and then explore the effect—for example, it might significantly reduce the combinatorial space. In the present version, all the factors in the Compatibility Table are set to 1 (so they have no effect) with the exception of the inputs with one another which are set to zero (e.g., Compatibility(2,3) = 0).

4.7. Global Changes via Yin-Yang. A bacterium like *E. coli* can be exposed suddenly to an environmental change that

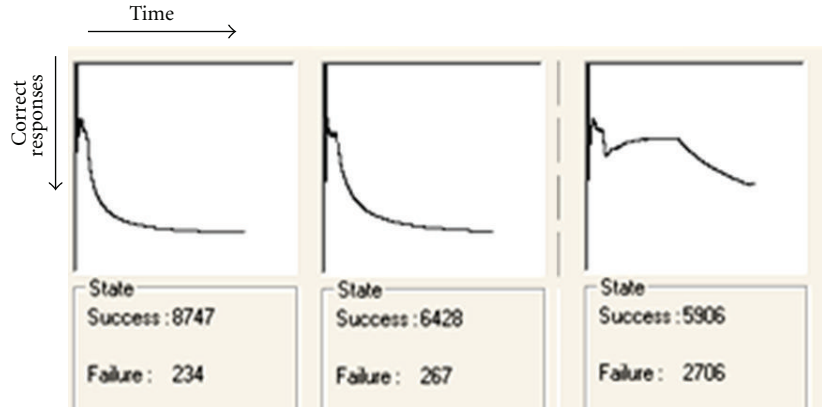


FIGURE 16: Robustness to noise. Three separate runs of the program, each for 20000 timesteps, show how allowing noise has in the third run perturbed Coco twice and caused it to forget a previously learnt sequence.

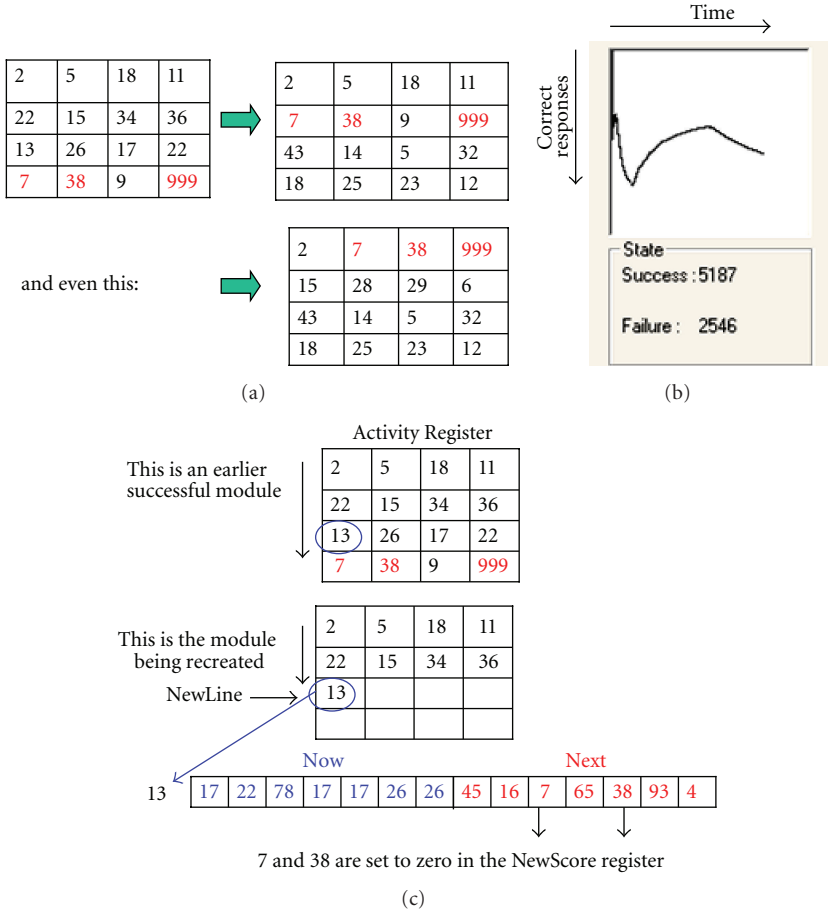


FIGURE 17: Principle of REVERSE DOWNTIME. Without this rule to reduce the compression of input-output modules, addresses can move from one line to another line closer to the input line. (a) The Activity Register is shown here with only four elements (for simplicity) and initially has addresses of certain elements (red) in a successful output line; these addresses can become displaced towards the input line. (b) The results of 20000 timesteps without REVERSE DOWNTIME. (c) Implementing REVERSE DOWNTIME during the loading of a NewLine requires the Next field of 13 to be consulted; this shows that 7 and 38 often follow 13, so, to avoid compression by 7 and 38 being loaded into the NewLine, the NowScore register for 7 and 38 is set to zero.

affects a great many of its systems. Typically, such global changes might include those in temperature or calcium concentration. Global changes can also result from internal changes such as those resulting from alterations in DNA supercoiling, mediated by topoisomerases like DNA gyrase, which then affect the expression of many genes [22]. Such changes distort the entire phenotype but in a way that is related to the original phenotype. It seems reasonable to imagine that mutations that affect the activity of enzymes like gyrase have a different role in evolution from those that affect enzymes involved in the metabolism of lactose. One way to explore this in the program is to choose two elements (e.g., 6 and 7) to represent two different global conditions (such as a high calcium level and a low calcium level alias Yin and Yang). When a 6 is loaded into the Activity Register, the Compatibility Table is modified temporarily such that different addresses may be loaded into the Activity Register under these Yin conditions. The effect of loading a 6 can be carried over several time steps before the Compatibility Table is reset. When a 7 is loaded into the Activity Register, the Compatibility Table is modified in a different way to take into account the Yang conditions. In the present version, no such changes are allowed to the Compatibility Table.

4.8. Equilibrium and Nonequilibrium Structures. It has been proposed that bacteria and other cells are confronted with the task of reconciling surviving harsh conditions, which requires quasi-equilibrium structures (thick, cross-linked walls, and liquid crystalline DNA), with growing in favourable conditions, which requires nonequilibrium structures (such as those formed by the dynamic, ATP/GTP consuming, dynamically coupled processes of transcription and translation) [23–26]. To put it picturesquely, cells are confronted with life on the scales of equilibria and, conceivably, use the cell cycle in their balancing act [23, 27]. It might therefore be interesting to explore what happens when the equivalent of the energy currencies of the cell, ATP, GTP, and polyphosphate, is introduced into the program. This might be achieved by attributing the role of ATP to an element and giving this element special properties. For example, this element (e.g., element 77) might have to be present in the Activity Register for a subset of other nonequilibrium elements to be loaded (which could be done via the Compatibility Table); if 77 were absent from the Activity Register, this subset could not be loaded although another subset of equilibrium structures could be; the probability with which 77 could be loaded might depend on a combination in a line of the Activity Register of an input element (corresponding to glucose) and other elements (corresponding to glycolytic enzymes).

4.9. Several Activity Registers. The Yin-Yang approach (Section 4.7) could be adapted to study some of the under-appreciated implications of DNA being double stranded [28, 29]. This might be done by employing two Activity Registers running in parallel, one using the odd numbers and the other the even numbers. Loading one Activity Register (corresponding to creating a nonequilibrium hyperstructure) would require ATP whilst loading the other Activity Register (corresponding to creating an equilibrium hyperstructure) would require the absence of ATP (Section 4.8). This might allow two phenotypes to be selected simultaneously so as to balance the scales of equilibria (Section 4.8). More interesting still would be to create a hierarchy of Activity Registers or to allow Coco itself to create them during learning.

4.10. Free-Running, Dreaming, and Looking Ahead. If Coco's environment were partially disconnected, Coco can continue running—more exactly, free-running—in the absence of inputs. Such free-running would occur if the environment were not to respond immediately to an output. Periods of free-running could be used to play in a sandbox or to dream. A sandbox is a concept that refers to a software environment where potentially dangerous operations can be tested in isolation, thus reducing the chances of damaging the primary program. Using a sandbox allows risk-free exploratory behaviour and, in a sense, corresponds to Coco operating in a look-ahead mode. In this mode, Coco might be disconnected from the environment and run through different combinations of stored input and output modules (where modules are sequential lines of the Activity Register). These modules might be associated with special elements 12 and 13 to represent pleasure and pain, respectively, depending on whether they have been rewarded or punished. Starting from the present state, Coco might load the Activity Register with different addresses for different runs (e.g., 20 timesteps) and compare the pleasure index (e.g., sum of $12/(12 + 13)$) for these runs (in which inputs from the environment are replaced by those stored in the modules). The initial loading of the Activity Register corresponding to the most successful run would then be adopted and environmental inputs once again were allowed.

Hypotheses about the function of dreaming might be explored via the creation of a parallel set of objects copied from the normal ones, namely, a DreamActivity Register running in dream time with DreamElements, DreamNow, and DreamNext. Perhaps this could be used to discover and remedy pathogenic connectivities that lead the system to get stuck in deep basins of attraction and so forth. Such action might be based on a characterisation of the states in the Activity Register. For example, for each line in the Activity Register, the ten addresses have elements where each contains seven Now addresses and seven Next addresses, hence a total for the line of 70 Now addresses and 70 Next addresses. Each line, alias the state of the system at that time, can therefore be characterised by a pair of coordinates (different Now addresses, different Next addresses). Intuitively, a successful state should tend towards (length of line, length of line)—here (10, 10)—whilst an unsuccessful state should tend towards (length of line $\times$ size of Now field, length of line $\times$ size of Next field)—here (70, 70). (This is not strictly speaking correct, but it gives the flavour.) The sequence of these coordinates then constitutes a function that might be recognised and used during dreaming.

A different approach would be to use dreaming to make a landscape of the connection space by loading the Activity Register with different elements and then letting it run so as to determine state cycles and basins of attraction; when the program has done, it might be possible to modify the connections so as to maximise the use of the landscape and connect basins. One attractive possibility is that Coco acts during free-running to minimise conflict between the Next and Now processes. This would take the form of changing the connections so that in loading addresses into a line of the Activity Register the same elements are scored highly by both processes. It amounts to making the Next and Now processes coherent with one another. Indeed, insofar as incoherence results in unhappiness, it could even be argued that this would create a state of happiness in systems as different as men, bacteria, and machines!

How far away is all this from real biology? Is the dreaming envisaged here only relevant to higher organisms or does it extend, for example, to bacteria? If that were the case, related questions include how we would know that a bacterium was dreaming and what it would mean for the bacterium. Showing that dreaming had a role in the learning of Coco might cast some light on its potential evolutionary value for all organisms.

4.11. Varying the Size of the Now and Next Fields. In the version presented here, the Now and Next fields are of constant size (each contains 7 addresses). This is far from biological reality where networks generally contain nodes with very different connectivities, including hubs and "driver" nodes [30–32]. These connectivities can take the form of protein activators and repressors of gene expression [33], small molecules acting on functioning-dependent structures [34], ions travelling along charged filaments such as microtubules or DNA [35], ions and molecules moving along and through pili and nanotubes [36, 37], convergence on common frequencies of oscillation [38, 39], joining a hyperstructure [25], and so forth. Coco would be much closer to modelling reality if the size of the fields were to vary as a function of learning (one reason for this is that increasing the size of a field permits element X to have stronger connections to element Y because X's field can hold more copies of Y's address). This may prove relatively easy to implement: for example, rewarding and punishing might entail increasing and decreasing the fields, respectively (as well as overwriting).

The relative strengths of the Now and Next connections can also be modified via the NowNextWeighting. As shown in Sections 3.1 and 3.2, setting NowNextWeighting very low or very high makes Coco run with either just Nexts or just Nows. Insofar as Next connections can be equated with local connections and Now connections with global connections, changing the value of NowNextWeighting can result in a phase transition in connectivity with similarity perhaps to the great deluge algorithm [40] or to Dual-Phase Evolution [41] or even, in the world of microbiology, to maintaining the right ratio of nonequilibrium to equilibrium hyperstructures within cells [23].

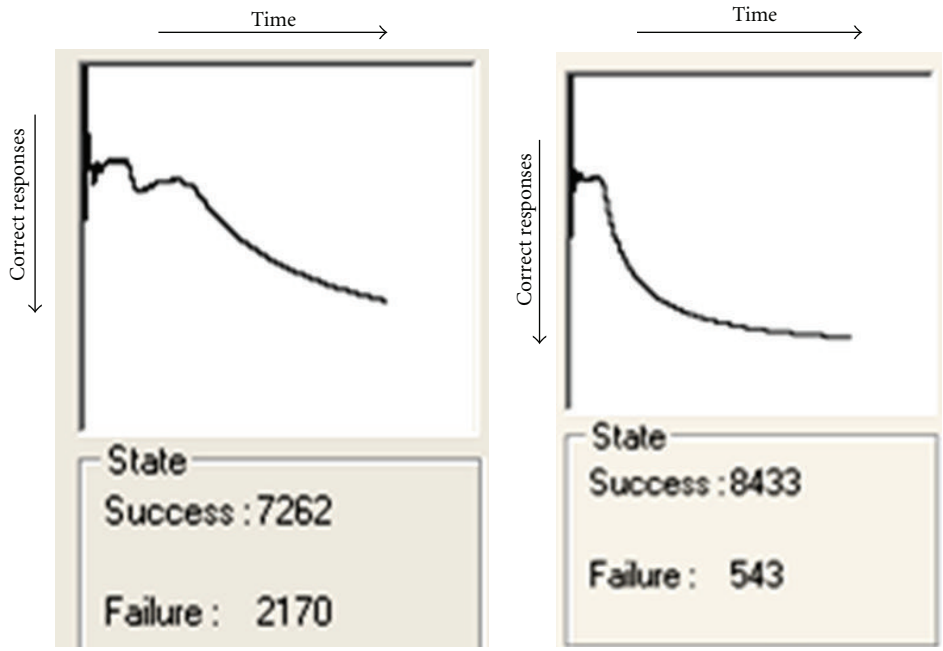
4.12. An Interactive Environment. An output is immediately followed by an environmental input that does not depend on the nature of the output. There is no possibility therefore for the present version of the program to influence the environment. This excludes the richness of connections that may emerge from dialogues between a learning system such as Coco and its environment. Two-way connections between a biological system such as a bacterium and its environment are fundamental and trying to understand them using Coco might take the form of Coco learning to play a simple game such as Noughts and Crosses (Tic-Tac-Toe).

4.13. Three or More Fields: from, Now, and Next and So Forth. The addition of a From field might add a new dimension to Coco. A From field would record the addresses of elements which preceded successful states in which the element was active. This would allow Coco to run backwards during Dreamtime (Section 4.10), analogous perhaps to the way humans mull over the day's events. In this speculation, such running might then allow input-output modules with similar characteristics to be identified (Section 4.10) and eventually connected via yet another, higher-level field involving a higher-level Activity Register. This, we would like to think, might be the equivalent of generating concepts.

4.14. Size of the Active Set. The size of the Active Set is an important parameter that can be changed and, in particular, increased, to take into account biological systems in which many elements can be active at the same time. In the present version of the program, the maximum size of the Activity Register is limited by the number of elements and by DOWNTIME (Section 2.7). This limits the size of the Active Set to around 80. An example of results obtained with an Activity Register containing 60 addresses is shown in Figure 18. Increasing the number of elements to, for example, 4000, allows learning with an Activity Register containing 100 addresses (Figure 18). It may prove important under some circumstances for Coco itself to modify the size of the Active Set. This might be the case if an input were to arrive "out of the blue" when Coco is in free-running mode; if an input were to trigger a sudden change in the size of the Activity Register, this could have a major effect, perhaps similar to that reported for the effects of a stimulus on connectivity in the cortex (for references see [41]) and perhaps similar too to the greater receptivity of bacteria to their environment when conditions start to worsen [42].

4.15. Collaborative Coherence. It will not have escaped the attention of the reader that, instead of separating the scores of the elements in competition for inclusion in the Active Set into Next and Now scores, these scores could be added together or even be combined synergistically to yield a kind of collaborative coherence. Philosophically, it would be nice to escape competition but we have no preliminary evidence that collaborative coherence leads to learning.

4.16. Pain and Pleasure. The present version punishes incorrect responses by randomly overwriting connections.



(a) (b)

Figure 18: Examples of successful learning with (a) 1000 elements and an Activity Register containing 60 addresses. (b) 4000 elements and an Activity Register containing 100 addresses.

This squanders a lot of information. A potentially better approach would be to retain and reuse information about mistakes by, for example, making use of specific elements such as 12 and 13 to represent pleasure and pain, respectively, in combination with a LOOK AHEAD subroutine (Section 4.10). Rewarding successful states might then entail writing a 12 into the Now and Next fields of the elements whose addresses are in the rewarded line of the Activity Register and, reciprocally, punishing might entail writing a 13 into them. This information could then be exploited using the LOOK AHEAD subroutine sketched out before that would count and compare the total of the addresses of these two elements when looking into possible futures (Section 4.10). An interesting question here is what would end up in the Now and Next fields of elements 12 and 13 themselves. Presumably, these fields would reflect connections to common, strong sources of pleasure and pain; these connections might then be used to drive the system towards or away from these sources.

4.17. Long-Term Memory. One way to obtain a long-term memory would be via a connections' matrix (number of elements $\times$ number of elements) that would record Now connections between elements that had participated in the same successful state (there could be similar ones for Next connections and, perhaps, further two matrices for unsuccessful states). A SUCCESSFUL CONNECTIONS subroutine could then be used to prevent overwriting successful connections or, at least, to alter the probability of overwriting these connections.

4.18. Noncyclical Input Sequences. Cycles are of major importance in biology. A primary example is the cell cycle which still remains to be fully understood [43]. The learning task presented here is based on an input sequence that is presented cyclically. Each output is immediately followed by a new input; at no time does Coco "run on its own" or run freely (Section 4.10). In responding to an input that "comes out of nowhere," as in the case of the first input in a linear series of inputs, the size of the Activity Register may be important (Section 4.14). One might envisage that during free-running the Activity Register would be small but would increase greatly on receiving an input; such increase might enable an input to make a decisive contribution to the composition of the Activity Register, particularly in the case of a variable Knumber since in such conditions, and when learning has occurred, the Now and Next fields of inputs become large (Section 4.11, unpublished data). It would also be interesting to explore the effects (on responding to an unannounced input) of changes to connectivity made during free-running (Section 4.10), changes that might even include modification of the weights associated with the Now and Next fields of inputs; such temporary modifications could be made during looking ahead.

Free-running whilst waiting for an input would entail Coco filling the Activity Register and wandering through the enormity of the combinatorial space in a state cycle [6]. The nature and length of such cycles may prove an important parameter in learning to respond to inputs that are given at random intervals from one another and from the outputs. This is because it is of little value in learning to connect (via the Next process) an active state containing an input to the

active state that immediately precedes it if this preceding state is hardly ever repeated. A possible solution would be for Coco to enter a short state cycle whilst waiting for the next input such that each input in the sequence was accessible from its own preceding state cycle. It is conceivable that these cycles might again be generated by the changes in connectivity occurring during dreaming (Section 4.10).

5. Relationship to Existing Systems

Hopfield's network model [44] uses the same learning rule as used by Hebb in which learning occurs as a result of the strengthening of the weights of the links between nodes [45]; this resembles the strengthening of links in Coco by the writing of addresses into the Now and Next fields of elements. In the Hopfield model it is assumed that the individual units preserve their individual states until they are selected at random for a new update; in Coco, the elements also preserve their identity until they are selected, but this selection is confined to members of the Active Set and occurs during rewarding and punishing. In a Hopfield network, each unit is connected to all other units (except itself); in Coco, each element can only be connected to a few others (the Knumber) via the Now and Next fields. A Hopfield network is symmetric because the weight of the link between unit i and unit j equals that between unit j and unit i; the network in Coco is asymmetric. In a Hopfield network, all the nodes contribute to the change in the activation of any single node at any one time; in Coco, only the elements (nodes/units) in the previous Active Set and in the developing Active Set contribute to the activation of an element (node/unit). In a Hopfield network, there is one type of connection between the nodes; in Coco, there are two types of connection—Next and Now. In a Hopfield network, the units can be in a state of either 1 or 0; similarly, in Coco, the elements can be either active or inactive.

It might be argued that a Hopfield network is a type of the Coco program. For example, if an attempt were to be made to turn Coco into a Hopfield network, (1) the size of the Knumber would be set similar to the size of the Enumber (i.e., the total number of elements) to make it closer to a weighting factor that takes into account all elements, (2) the activity of an element would be determined by its absolute score (using a threshold) rather than by its score relative to a limited number of competing elements, (3) the Anumber (the size of the Active Set) would therefore become a variable whose size would vary with the number of elements deemed to be active, (4) the Next links would correspond to the links between nodes but the Now links would have no equivalent, (5) the asymmetrical Coco network would tend towards symmetry if changes in the links to element i in the Next field of element j were accompanied by reciprocal changes in the links to element j in the Next field of element i (of course, Coco would then no longer run in the same way), and (6) some of the biologically relevant developments of Coco would have to be implemented in a Hopfield network which would be hard since Coco lends itself to the study of types of links with different properties; see Sections 4.6 and 4.7. In this context, it should be stressed that the weights in Coco are discrete, transparent, and easy to study and to manipulate.

Boolean networks have been extensively used to model biological systems. Thomas and collaborators have developed logical analysis which they have used both to study specific systems [46] and to derive general principles [47]. From such analyses, predictions can be made for experimental biologists to test. Logical analysis is not, however, a learning system like Coco. Reciprocally, Coco is not designed at present to model specific biological systems. As mentioned in Section 1, the (N, k) Boolean network of Kauffman [6] has given insight into the dynamics of biological systems and, in particular, into the concept of cells as living on the "edge of chaos" [2, 3]. But again, it is not a learning system like Coco.

6. Discussion

Few would deny that living systems are rich, complicated, and (hyper)complex. Such systems are often, almost necessarily, modelled and simulated by invoking Occam's Razor and adopting a reductionist approach. Life may, however, have originated as a rich, complicated, and diverse system, as, in other words, a prebiotic ecology [48]. In attempting to capture some of the characteristics of living system in a program, we have therefore adopted the holist approach of putting everything in and seeing what, if anything, emerges. To try to create a test-bed for concepts and to ensure that these concepts have some substance, we have written a program with a learning part, Coco. Coco contains parameters that may have very loose equivalents in aspects of evolution via global (as opposed to local) mutation, DNA replication, emergence, life on the scales of equilibria, and even the generation of concepts and dreaming. Most importantly, Coco is based on coherence.

Coherence characterises living systems. Coherence mechanisms operating—or suspected by some to operate—at the level of cells include tensegrity, ion condensation, DNA supercoiling, and a variety of oscillations [25, 48–52] plus, of course, mechanisms based on the usual activators and repressors of transcription along with DNA packaging proteins. Competitive coherence is an attempt to describe how bacterial phenotypes are created by a competition between maintaining a consistent story over time and creating a response that is coherent with respect to both internal and external conditions. Previously, it has been proposed that the bacterium *E. coli* can be considered as passing through a series of states in which a distinct set of its constituent molecules or "elements" are active [8]. The activity of these elements is determined by a competition between two processes. One of these processes depends on the previous cell state whilst the other depends on the internal coherence of the developing state. The simultaneous operation of these two processes is competitive coherence. Competitive coherence is in fact a scale-free concept. It can be applied to a population of bacteria such as a colony in which each cell is an element with its own Now and Next

fields. In this case, a typical Now process might involve global connections via a sonic vibration created by the combined metabolic activity of all the growing cells in the colony (which constitute the Active Set) [38] whilst the Next process might involve essentially local connections via diffusible molecules, sex pili, and lipid nanotubes [37]. We have invoked competitive coherence at higher levels to explain, for example, how a football team is selected. It is perhaps no longer original in computer science since the idea of two competing processes staggered in time can be found elsewhere in Simple Recurrent Networks [53]. What may be new is the possibility that the implementation of competitive coherence into a learning program could give rise to parameters suitable for describing the rich form of complexity found in living systems which depends on the interaction between many types of connection and which we have termed hypercomplexity [10]. The preliminary results from the toy program presented here encourage us to think that this may be the case.

One of these preliminary results is on the maximum size of the Activity Register that can result in Coco learning (Section 4.14). In Section 1, we mentioned the problem of how cells manage to negotiate the enormity of phenotype space in a reproducible and selectable way [6]; if, for example, a phenotype were determined by a simple combination of on-off expression of genes, a bacterium like *E. coli* with over 4000 genes would have the difficult task of exploring 2^{4000} combinations. (It should be noted though that no one knows how many genes are expressed at one time in an individual cell, let alone how many of these expressed genes are actually determining the phenotype, that is, form part of the Active Set.) However, if the phenotype were determined not directly by genes but at a higher level by a hundred or so extended macromolecular assemblies or *hyperstructures* comprising many different macromolecules—for which there is good evidence [25, 54, 55]—the number of on-off combinations would fall to 2^{100}. As we show here, a system with 4000 elements, of which a hundred form an Active Set, can learn via competitive coherence.

Finally, if there is any substance to our claim that competitive coherence is a fundamental to life, perhaps even its defining characteristic [56], the concept should be of value in novel approaches to computing inspired by and reliant on the way real cells behave [11].

7. Conclusion

Competitive coherence is a concept used to describe how a subset of elements out of a large set is activated to determine behaviour. It has been proposed as operating at many levels in biology. The results of the toy version of a type of neural network, based on competitive coherence, are presented here and show that it can learn. This is consistent with competitive coherence playing a central role in living systems. The parameters responsible for the functioning of the competitive coherence part of the program, which, for example, are related to complexity and emergence, may be of interest to biologists and others.

Acknowledgments

For helpful discussions, the authors thank Abdallah Zemirline, Alex Grossman, Francois Kepes, Jacques Ninio, and Michel Thellier. They also thank the anonymous referees for many valuable comments. For support they thank the Epigenomics Project, Evry, and the University of Brest. This paper is dedicated to the memory of Maurice Demarty.

References

[1] D. E. Ingber, "The origin of cellular life," *Bioessays*, vol. 22, no. 12, pp. 1160–1170, 2000.

[2] J. P. Crutchfield and K. Young, "Computation at the edge of chaos," in *Complexity, Entropy and the Physics of Information: SFI Studies in the Sciences of Complexity*, W. H. Zurek, Ed., pp. 223–269, Addison-Wesley, Reading, Mass, USA, 1990.

[3] C. G. Langton, "Computation at the edge of chaos: phase transitions and emergent computation," *Physica D*, vol. 42, no. 1–3, pp. 12–37, 1990.

[4] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world9 networks," *Nature*, vol. 393, no. 6684, pp. 440–442, 1998.

[5] P. Bak, *How Nature Works: The Science of Self-Organized Criticality*, Copernicus, New York, NY, USA, 1996.

[6] S. Kauffman, *At Home in the Universe, the Search for the Laws of Complexity*, Penguin, London, UK, 1996.

[7] D. Bray, "Intracellular signalling as a parallel distributed process," *Journal of Theoretical Biology*, vol. 143, no. 2, pp. 215–231, 1990.

[8] V. Norris, "Modelling *Escherichia coli* The concept of competitive coherence," *Comptes Rendus de l'Academie des Sciences*, vol. 321, no. 9, pp. 777–787, 1998.

[9] V. Norris, "Competitive coherence," in *Encyclopedia of Sciences and Religions*, N. P. Azari, A. Runehov, and L. Oviedo, Eds., Springer, New York, NY, USA, 2012.

[10] V. Norris, A. Cabin, and A. Zemirline, "Hypercomplexity," *Acta Biotheoretica*, vol. 53, no. 4, pp. 313–330, 2005.

[11] V. Norris, A. Zemirline, P. Amar et al., "Computing with bacterial constituents, cells and populations: from bioputing to bactoputing," *Theory in Biosciences*, pp. 1–18, 2011.

[12] S. G. Wolf, D. Frenkiel, T. Arad, S. E. Finkeil, R. Kolter, and A. Minsky, "DNA protection by stress-induced biocrystallization," *Nature*, vol. 400, no. 6739, pp. 83–85, 1999.

[13] V. Norris and M. S. Madsen, "Autocatalytic gene expression occurs via transertion and membrane domain formation and underlies differentiation in bacteria: a model," *Journal of Molecular Biology*, vol. 253, no. 5, pp. 739–748, 1995.

[14] T. Katayama, S. Ozaki, K. Keyamura, and K. Fujimitsu, "Regulation of the replication cycle: conserved and diverse regulatory systems for DnaA and oriC," *Nature Reviews Microbiology*, vol. 8, no. 3, pp. 163–170, 2010.

[15] V. Norris, C. Woldringh, and E. Mileykovskaya, "A hypothesis to explain division site selection in *Escherichia coli* by combining nucleoid occlusion and Min," *FEBS Letters*, vol. 561, no. 1–3, pp. 3–10, 2004.

[16] Y. Grondin, D. J. Raine, and V. Norris, "The correlation between architecture and mRNA abundance in the genetic regulatory network of *Escherichia coli*," *BMC Systems Biology*, vol. 1, p. 30, 2007.

[17] C. L. Woldringh and N. Nanninga, "Structure of the nucleoid and cytoplasm in the intact cell," in *Molecular Cytology of*

Escherichia coli, N. Nanninga, Ed., pp. 161–197, Academic Press, London, UK, 1985.

[18] V. Norris, L. Janniere, and P. Amar, "Hypothesis: variations in the rate of DNA replication determine the phenotype of daughter cells," in *Modelling Complex Biological Systems in the Context of Genomics*, EDP Sciences, Evry, France, 2007.

[19] L. H. Hartwell and T. A. Weinert, "Checkpoints: controls that ensure the order of cell cycle events," *Science*, vol. 246, no. 4930, pp. 629–634, 1989.

[20] A. L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, no. 5439, pp. 509–512, 1999.

[21] M. H. V. Van Regenmortel, "Emergence in Biology," in *Modelling and Simulation of Biological Processes in the Context of Genomics*, P. Amar, V. Norris, G. Bernot et al., Eds., pp. 123–132, Genopole, Evry, France, 2004.

[22] K. S. Jeong, Y. Xie, H. Hiasa, and A. B. Khodursky, "Analysis of pleiotropic transcriptional profiles: a case study of DNA gyrase inhibition.," *PLoS Genetics*, vol. 2, no. 9, p. e152, 2006.

[23] V. Norris, M. Demarty, D. Raine, A. Cabin-Flaman, and L. Le Sceller, "Hypothesis: hyperstructures regulate initiation in *Escherichia coli* and other bacteria," *Biochimie*, vol. 84, no. 4, pp. 341–347, 2002.

[24] A. Minsky, E. Shimoni, and D. Frenkiel-Krispin, "Stress, order and survival," *Nature Reviews Molecular Cell Biology*, vol. 3, no. 1, pp. 50–60, 2002.

[25] V. Norris, T. Den Blaauwen, R. H. Doi et al., "Toward a hyperstructure taxonomy," *Annual Review of Microbiology*, vol. 61, pp. 309–329, 2007.

[26] V. Norris, "Speculations on the initiation of chromosome replication in *Escherichia coli*: the dualism hypothesis," *Medical Hypotheses*, vol. 76, no. 5, pp. 706–716, 2011.

[27] V. Norris and P. Amar, "Life on the scales: initiation of replication in *Escherichia coli*," in *Modelling Complex Biological Systems in the Context of Genomics*, EDF Sciences, Evry, France, 2012.

[28] E. P. C. Rocha, J. Fralick, G. Vediyappan, A. Danchin, and V. Norris, "A strand-specific model for chromosome segregation in bacteria," *Molecular Microbiology*, vol. 49, no. 4, pp. 895–903, 2003.

[29] M. A. White, J. K. Eykelenboom, M. A. Lopez-Vernaza, E. Wilson, and D. R. F. Leach, "Non-random segregation of sister chromosomes in *Escherichia coli*," *Nature*, vol. 455, no. 7217, pp. 1248–1250, 2008.

[30] D. J. Raine and V. Norris, *Metabolic cycles and self-organised criticality*. Interjournal of complex systems, Paper 361, http://www.interjournal.org/, 2000.

[31] A. L. Barabási and Z. N. Oltvai, "Network biology: understanding the cell's functional organization," *Nature Reviews Genetics*, vol. 5, no. 2, pp. 101–113, 2004.

[32] Y. Y. Liu, J. J. Slotine, and A. L. Barabási, "Controllability of complex networks," *Nature*, vol. 473, no. 7346, pp. 167–173, 2011.

[33] N. Guelzim, S. Bottani, P. Bourgine, and F. Képès, "Topological and causal structure of the yeast transcriptional regulatory network," *Nature Genetics*, vol. 31, no. 1, pp. 60–63, 2002.

[34] M. Thellier, G. Legent, P. Amar, V. Norris, and C. Ripoll, "Steady-state kinetic behaviour of functioning-dependent structures," *FEBS Journal*, vol. 273, no. 18, pp. 4287–4299, 2006.

[35] C. Ripoll, V. Norris, and M. Thellier, "Ion condensation and signal transduction," *BioEssays*, vol. 26, no. 5, pp. 549–557, 2004.

[36] G. Reguera, K. D. McCarthy, T. Mehta, J. S. Nicoll, M. T. Tuominen, and D. R. Lovley, "Extracellular electron transfer via microbial nanowires," *Nature*, vol. 435, no. 7045, pp. 1098–1101, 2005.

[37] G. P. Dubey and S. Ben-Yehuda, "Intercellular nanotubes mediate bacterial communication," *Cell*, vol. 144, no. 4, pp. 590–600, 2011.

[38] M. Matsuhashi, A. N. Pankrushina, K. Endoh et al., "Bacillus carboniphilus cells respond to growth-promoting physical signals from cells of homologous and heterologous bacteria," *Journal of General and Applied Microbiology*, vol. 42, no. 4, pp. 315–323, 1996.

[39] G. Reguera, "When microbial conversations get physical," *Trends in Microbiology*, vol. 19, no. 3, pp. 105–113, 2011.

[40] G. Dueck, "New optimization heuristics; The great deluge algorithm and the record-to-record travel," *Journal of Computational Physics*, vol. 104, no. 1, pp. 86–92, 1993.

[41] G. Paperin, D. G. Green, and S. Sadedin, "Dual-phase evolution in complex adaptive systems," *Journal of the Royal Society Interface*, vol. 8, no. 58, pp. 609–629, 2011.

[42] J. Vohradský and J. J. Ramsden, "Genome resource utilization during prokaryotic development," *The FASEB Journal*, vol. 15, no. 11, pp. 2054–2056, 2001.

[43] X. Wang, C. Lesterlin, R. Reyes-Lamothe, G. Ball, and D. J. Sherratt, "Replication and segregation of an *Escherichia coli* chromosome with two replication origins," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 108, no. 26, pp. E243–E250, 2011.

[44] J. J. Hopfield, "Neural networks and physical systems with emergent collective computational abilities," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 79, no. 8, pp. 2554–2558, 1982.

[45] D. O. Hebb, *The Organization of Behavior*, Wiley and Sons, New York, NY, USA, 1949.

[46] M. Kaufman, J. Urbain, and R. Thomas, "Towards a logical analysis of the immune response," *Journal of Theoretical Biology*, vol. 114, no. 4, pp. 527–561, 1985.

[47] R. Thomas, D. Thieffry, and M. Kaufman, "Dynamical behaviour of biological regulatory networks I. Biological role of feedback loops and practical use of the concept of the loop-characteristic state," *Bulletin of Mathematical Biology*, vol. 57, no. 2, pp. 247–276, 1995.

[48] A. Hunding, F. Kepes, D. Lancet et al., "Compositional complementarity and prebiotic ecology in the origin of life," *BioEssays*, vol. 28, no. 4, pp. 399–412, 2006.

[49] H. Fröhlich, "Long range coherence and energy storage in biological systems," *International Journal of Quantum Chemistry*, vol. 42, no. 5, pp. 641–649, 1968.

[50] V. Norris and G. J. Hyland, "Do bacteria sing? Sonic intercellular communication between bacteria may reflect electromagnetic intracellular communication involving coherent collective vibrational modes that could integrate enzyme activities and gene expression," *Molecular Microbiology*, vol. 24, no. 4, pp. 879–880, 1997.

[51] D. E. Ingber, "The architecture of life," *Scientific American*, vol. 278, no. 1, pp. 48–57, 1998.

[52] A. Travers and G. Muskhelishvili, "DNA supercoiling—a global transcriptional regulator for enterobacterial growth?" *Nature Reviews Microbiology*, vol. 3, no. 2, pp. 157–169, 2005.

[53] J. L. Elman, "An alternative view of the mental lexicon," *Trends in Cognitive Sciences*, vol. 8, no. 7, pp. 301–306, 2004.

[54] R. Narayanaswamy, M. Levy, M. Tsechansky et al., "Widespread reorganization of metabolic enzymes into reversible assemblies upon nutrient starvation," *Proceedings of the National Academy of Sciences of the United States of America*, vol. 106, no. 25, pp. 10147–10152, 2009.

[55] P. M. Llopis, A. F. Jackson, O. Sliusarenko et al., "Spatial organization of the flow of genetic information in bacteria," *Nature*, vol. 466, no. 7302, pp. 77–81, 2010.

[56] V. Norris, P. Amar, G. Bernot et al., "Questions for cell cyclists," *Journal of Biological Physics and Chemistry*, vol. 4, pp. 124–130, 2004.

Hindawi Publishing Corporation
Advances in Artificial Neural Systems
Volume 2012, Article ID 457590, 7 pages
doi:10.1155/2012/457590

Research Article

Unsupervised Neural Techniques Applied to MR Brain Image Segmentation

A. Ortiz,[1] J. M. Gorriz,[2] J. Ramirez,[2] and D. Salas-Gonzalez[2]

[1] *Department of Communication Engineering, University of Malaga, 29071 Malaga, Spain*
[2] *Department of Signal Theory, Networking and Communications, University of Granada, 18071 Granada, Spain*

Correspondence should be addressed to A. Ortiz, aortiz@ic.uma.es

Received 17 February 2012; Accepted 14 April 2012

Academic Editor: Anke Meyer-Baese

The primary goal of brain image segmentation is to partition a given brain image into different regions representing anatomical structures. Magnetic resonance image (MRI) segmentation is especially interesting, since accurate segmentation in white matter, grey matter and cerebrospinal fluid provides a way to identify many brain disorders such as dementia, schizophrenia or Alzheimer's disease (AD). Then, image segmentation results in a very interesting tool for neuroanatomical analyses. In this paper we show three alternatives to MR brain image segmentation algorithms, with the Self-Organizing Map (SOM) as the core of the algorithms. The procedures devised do not use any a priori knowledge about voxel class assignment, and results in fully-unsupervised methods for MRI segmentation, making it possible to automatically discover different tissue classes. Our algorithm has been tested using the images from the Internet Brain Image Repository (IBSR) outperforming existing methods, providing values for the average overlap metric of 0.7 for the white and grey matter and 0.45 for the cerebrospinal fluid. Furthermore, it also provides good results for high-resolution MR images provided by the Nuclear Medicine Service of the "Virgen de las Nieves" Hospital (Granada, Spain).

1. Introduction

Nowadays, magnetic resonance imaging (MRI) systems provide an excellent spatial resolution as well as a high tissue contrast. Nevertheless, since actual MRI systems can obtain 16-bit depth images corresponding to 65535 gray levels, the human eye is not able to distinguish more than several tens of gray levels. On the other hand, MRI systems provide images as slices which compose the 3D volume. Thus, computer-aided tools are necessary to exploit all the information contained in an MRI. These are becoming a very valuable tool for diagnosing some brain disorders such as Alzheimer's disease [1–5]. Moreover, modern computers, which contain a large amount of memory and several processing cores, have enough process capabilities for analyzing the MRI in reasonable time.

Image segmentation consists in partitioning an image into different regions. In MRI, segmentation consists of partitioning the image into different neuroanatomical structures which corresponds to different tissues. Hence, analyzing the neuroanatomical structures and the distribution of the tissues on the image, brain disorders or anomalies can be figured out. Hence, the importance of having effective tools for grouping and recognizing different anatomical tissues, structures and fluids is growing with the improvement of the medical imaging systems. These tools are usually trained to recognize the three basic tissue classes found on a healthy brain MR image: white matter (WM), gray matter (GM), and cerebrospinal fluid (CSF). All of the nonrecognized tissues or fluids are classified as suspect, to be pathological.

The segmentation process can be performed in two ways. The first consists of manual delimitation of the structures present within an image by an expert. The second consists of using an automatic segmentation technique. As commented before, computer image processing techniques allow exploiting all the information contained in an MRI.

There are several automatic segmentation techniques. Some of them use the information contained in the image histogram [6–11]. This way, since different contrast areas should correspond with different tissues, the image histogram can be used for partitioning the image. Nevertheless, variations on the contrast of the same tissue are found in

an image due to RF noise or shading effects due to magnetic field variations, resulting in tissue misclassification. Other methods use statistical classifiers based on the expectation-maximization (EM) algorithms [12–14], maximum likelihood (ML) estimation [15], or Markov random fields [16, 17]. Other segmentation techniques are based on artificial neural network classifiers [8, 18–21] such as *self-organizing maps* (SOMs) [18, 19, 21–23].

In this paper we present three segmentation alternatives based on SOMs, which provide good results over the internet brain image repository (IBSR) [16] images.

2. SOM Algorithm

SOM is an unsupervised classifier proposed by Kohonen and it has been used for a large number of applications regarding classification or modelling [24]. The self-organizing process is based on the distance (usually the Euclidean distance) computation among each training sample and all the units on the map as a part of a competitive learning process. On the other hand, several issues such as topological map, number of units on the map, initialization of weights, and the training process on the map are decisive for the classification quality. Regarding the topology, a 2D hexagonal grid was selected since it fitted better in the feature space as shown in the experiments.

The SOM algorithm can be summarized as follows. Let $X \subset \mathbb{R}^d$ be the data manifold. In each iteration, the winning unit is computed according to

$$U_\omega(t) = \arg\min_i \{\|x(t) - \omega_i(t)\|\}, \tag{1}$$

where $x(t), x \in X$, is the input vector at time t and $\omega_i(t)$ is the prototype vector associated with the unit i. The unit closer to the input vector $U_\omega(t)$ is referred to as *winning unit* and the associated prototype is updated. To complete the adaptive learning process on the SOM, the prototypes of the units in the neighborhood of the *winning unit* are also updated according to:

$$\omega_i(t+1) = \omega_i(t) + \alpha(t)h_{Ui}(t)(x(t) - \omega_i(t)), \tag{2}$$

where $\alpha(t)$ is the exponential decay learning factor and $h_{Ui}(t)$ is the neighborhood function associated with the unit i. Both, the learning factor and the neighborhood function decay with time; thus the prototypes adaptation becomes slower as the neighborhood of the unit i contains less number of units:

$$h_{Ui}(t) = e^{(-\|r_U - r_i\|^2/2\sigma(t)^2)}. \tag{3}$$

Equation (3) shows the neighbourhood function, where r_i represents the position on the output space and $\|r_U - r_i\|$ is the distance between the winning unit and the unit i on the output space. The neighbourhood is defined by a Gaussian function which shrinks in each iteration as shown in (4). In this competitive process, the winning unit is named the best matching unit (BMU). On the other hand, $\sigma(t)$ controls the reduction of the Gaussian neighborhood in each iteration. τ_1 is a time constant which depends on the number of iterations and the map radius and computed as $\tau_1 =$ number_of_iterations/map_radius:

$$\sigma(t) = \sigma_0 e^{(-t/\tau_1)}. \tag{4}$$

The quality of the trained map can be computed by the means of two measures. These two measures are the quantization error (t_e), which determines the average distance between each data vector and its best matching unit (BMU) and the topological error (q_e), which measures the proportion of all data vectors for which first and second BMUs are not adjacent units. Both, the quantization error and the topological error are defined by the following:

$$t_e = \frac{1}{N}\sum_{i=1}^{N} u(\vec{x}_i), \tag{5}$$

$$q_e = \sum_{i=1}^{N} \left\| \vec{x}_i - \vec{b}_{\vec{x}_i} \right\|. \tag{6}$$

In (5), N is the total number of data vectors, and $u(\vec{x}_i)$ is 1 if the first and the second BMU for $\vec{x}_i$ are nonadjacent and 0 otherwise. In (6) the quantization error is defined where $\vec{x}_i$ is the ith data vector on the input space and $b_{\vec{\omega}_i}$ is the weight (prototype) associated with the best matching unit for the data vector $\vec{x}_i$. Therefore, lower values of t_e and q_e imply a better topology preservation, which is equivalent to a better clustering result. That is to say, the lower the values on the quantization error (q_e) and the topological error (t_e), the better the goodness of the SOM [25, 26]. In this paper, *SOM toolbox* [27] has been used to implement SOM.

3. MR Image Segmentation with SOM

In this section we present two image segmentation algorithms based on unsupervised SOM. The first uses the histogram to segment the whole volume (i.e., classify all the voxels on the volumetric image). The second extracts a set of features from each image slice and uses an SOM to classify the feature vectors into clusters using the devised entropy gradient clustering method. Thus, Figure 1 shows the block diagram of the presented segmentation algorithms.

3.1. Image Preprocessing. Once the MR image has been acquired, a preprocessing is performed in order to remove noise and to homogenize the image background. The brain extraction for undesired structures removal (i.e., skull and scalp) can be done at this stage. There are several algorithms for this purpose such as brain surface extractor (BSE), brain extraction tool (BET) [8], Minneapolis consensus strip (McStrip), or hybrid watershed algorithm (HWA) [2]. Since IBSR 1.0 images have these undesired structures already removed, brain extraction is not required. Nevertheless, images provided by IBSR 2.0 are distributed without the scalp/skull already removed. In these images, the brain has been extracted in the preprocessing stage using BET.

3.2. Segmentation Using the Volume Image Histogram (HFS-SOM). The first step after preprocessing the image consists

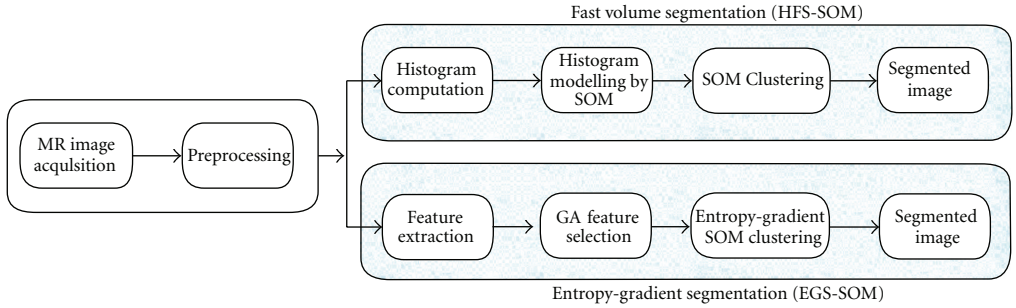


Figure 1: Block diagram of the segmentation methods.

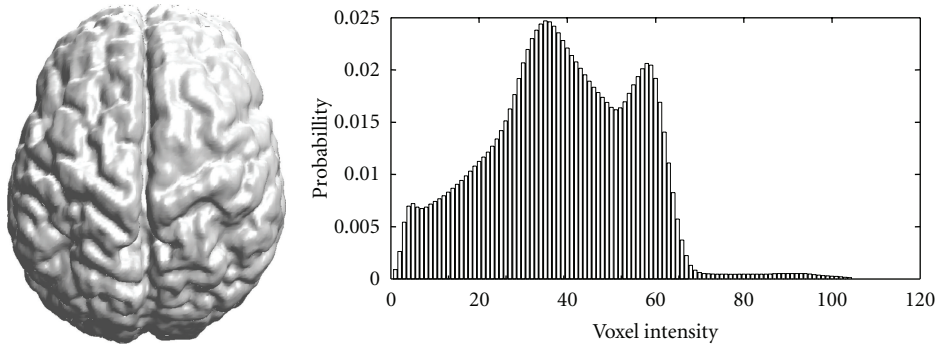


Figure 2: Rendered brain surface extracted from IBSR_12 volume (a) and computed histogram (b).

in computing the volume image histogram which describes the probability of occurrence of voxel intensities in the volume image and provides information regarding different tissues. A common approach to avoid processing the large number of voxels present on MR images consists in modelling the intensity values as a finite number of prototypes, which deals to improve the computational effectiveness. After computing the histogram, the bin 0 is removed since it contains all the background voxels. Thus, only information corresponding to the brain is stored.

Figure 2 shows the rendered brain surface from the IBSR volume 12, and its histogram.

Histogram data including the intensity occurrence probabilities (p_i) and the relative position (bin number), b_i, are used to compose the feature vectors $\vec{F} = (p_i, b_i)$, $p_i \in \mathbb{R}, \mathrm{b}_i \in \mathbb{Z}$, to be classified by the SOM.

On a trained SOM, the output layer is composed by a reduced number of prototypes (the number of units on the output layer) modelling the input data manifold. In addition, the most similar prototypes are closely located in the output map at the time the most dissimilar are located apart. Nevertheless, since all the units have an associated prototype, it is necessary to cluster the SOMs in order to define the borders between clusters. In other words, each prototype is grouped so that it belongs to a cluster. Thus, the k-means algorithm is used to cluster the SOMs, grouping the prototypes into a number of different classes, and the DBI [28], which gives lower values for better clustering results, is computed for different k values to provide a measurement of the clustering validity.

The clusters on the SOM group the units so that they belong to a specific class. As each of these units will be the BMU of a specific set of voxels, the clusters define different voxel classes. This way, each voxel is labeled as belonging to a class (i.e., segment).

3.3. MR Image Segmentation with SOM and the Entropy-Gradient Algorithm (EGS-SOM). The method described in this section is also based on SOM for voxel classification, but histogram information from the image volume is replaced by computing a set of features, selecting the most discriminant ones. After that, SOM clustering is performed by the EGS-SOM method described here in after, which allows us to obtain higher-resolution images providing good segmentation results as shown in the experiments.

3.3.1. Feature Extraction and Selection. In this stage some significant features from the MR image are extracted to be subjected to classification. As commented before, we perform the image processing slice by slice on each plane. Thus, the feature extraction is carried out by using an overlapping and sliding window of 7×7 pixels on each slice of a specific plane.

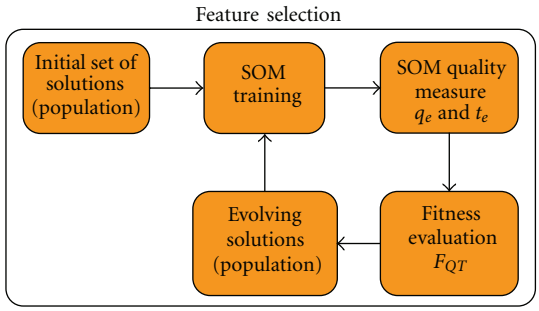


Figure 3: Feature selection process with genetic algorithm.

In the feature extraction process, window size plays an important role since smaller windows are not able to capture the second-order features, that is, texture information. The use of higher window sizes results in loosing resolution. Therefore the 7×7 size gives a good trade-off between complexity and performance.

In this paper we use first- and second-order statistical features [29]. The first-order features we extract from the image are intensity, mean, and variance. The intensity is referred to the gray level of the center pixel on the window. The mean and variance are calculated taking into account the gray level present on the window. On the other hand, we additionally use second-order features such as textural features. Haralick et al. [30] proposed the use of 14 features for image classification, computed using the gray level coocurrence matrix (GLCM) method. The set of second-order features we have used are energy, entropy, contrast, angular second moment (ASM), sum average, autocorrelation, correlation, inverse difference moment, maximum probability, cluster prominence, cluster shade, dissimilarity, and second-order variance as well as moment invariants [31].

In order to select the most discriminant features, a genetic algorithm is used to minimize the topological and the quantization error on the SOM through the fitness function shown in (7)

$$F_{QT} = (0.5q_e + 0.5t_e). \tag{7}$$

The feature selection process is summarized in Figure 3.

The stop criterion is reached when the performance of the proposed solutions does not improve the performance significantly (1%) or the maximum number of generations is reached (500).

Once the dimension of the feature space has been reduced, we use the vectors of this space for training a SOM. The topology of the map and the number of units on the map are decisive for the SOM quality. In that sense, we use a hexagonal grid since it allows better fitting the prototypes to the feature space vectors. Each BMU on the SOM has an associated pixel on the image. This association is made through a matrix computed during the feature extraction phase which stores the coordinates of the central pixel on each window. This allows associating a feature vector to an image pixel.

Nevertheless, these clusters roughly define the different areas (segments) on the image, and a further fine-tuning phase is required. This fine-tuning phase is accomplished by the entropy-gradient method.

Entropy-Gradient Method. The procedure devised consists of using the feature vectors associated with each BMU to compute a similarity measurement among the vectors belonging with each BMU and the vectors associated to each other BMU. Next, the BMUs are sorted in ascending order of the contrast. Finally, the feature vectors of each BMU are included on a cluster. For each map unit, we compute the accumulated entropy:

$$H_{m_i} = \sum_{n=1}^{N_p} H_n, \tag{8}$$

where i is the map unit index and N_p the number of pixels belonging to the map unit in the classification process. This means that the unit i has a number of N_p-associated pixels. Since the output layer on the SOM is a two-dimensional space, we calculate the entropy-gradient vector from each map unit (8) and move to the opposite direction for clustering.

4. Results and Discussion

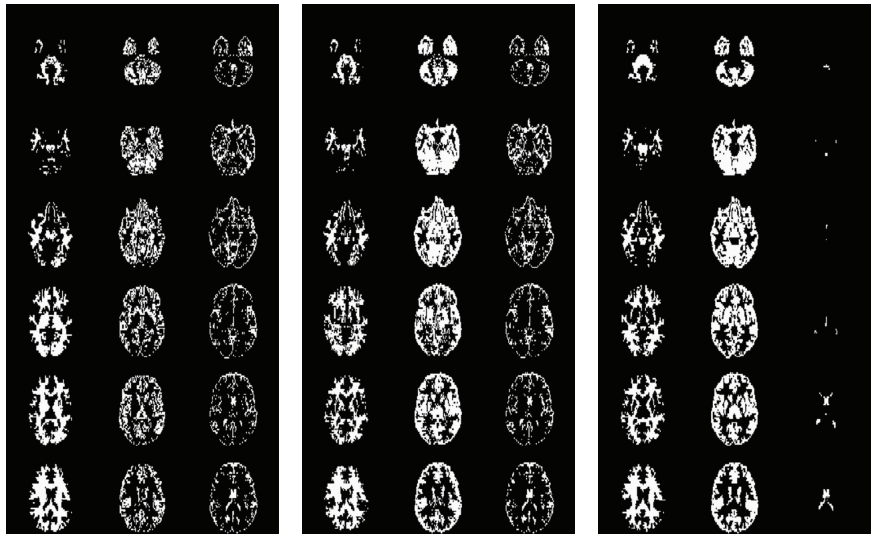
In this section we show the segmentation results obtained using real MR brain images from two different sources. One of these sources is the IBSR database [32] in two versions, IBSR and IBSR 2.0.

Figures 4(a) and 4(b) show the segmentation results for the IBSR volume 100_23 using the HFS-SOM algorithm and the EGS-SOM algorithm, respectively. In these images, WM, GM, and CSF are shown for slices 120, 130, 140, 150, 160, and 170 on the axial plane. Expert segmentation from IBSR database is shown in Figure 4(c).

Figure 5(a) shows the segmentation results for the IBSR 2.0 volume 12 using the fast volume segmentation algorithm. In this figure, each row corresponds to a tissue and each image column corresponds to a different slice. In the same way, Figure 5(b) shows the same slices of Figure 5(b) but the segmentation is performed using the EGS-SOM algorithm. Figure 5(c) shows the segmentation performed by expert radiologists provided by the IBSR database (ground truth).

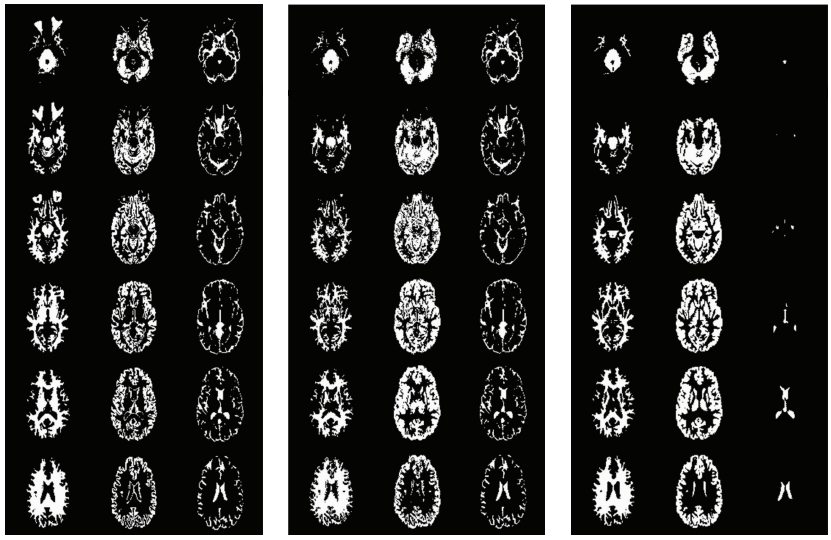
Visual comparison between automatic segmentation and the ground truth points up that the EGS-SOM method outperforms the fast volume segmentation method.

This fact is also stated in Figure 6 where Tanimoto's index is shown for different segmentation algorithms, where SSOM corresponds to our entropy-gradient algorithm, BMAP is biased map [33], AMAP is adaptative map [33], MAP is maximum a posteriori probability [34], MLC is maximum likelihood [35], FUZZY is fuzzy k-means [36] and TSKMEANS is tree-structured k-means [36]. The performance of the presented segmentation techniques has been evaluated by computing the average overlap rate through Tanimoto's index, as it has been widely used by other authors



(a) (b) (c)

Figure 4: Segmentation of the IBSR volume 100_23 using the HFS-SOM algorithm (a) and the EGS-SOM algorithm (b). Ground Truth is shown in (c). Slices 120, 130, 140, 150, 160, and 170 on the axial plane are shown on each column. First column corresponds to WM, second column to GM, and third column to CSF.



(a) (b) (c)

Figure 5: Segmentation of the IBSR 2.0 volume 12 using the HFS-SOM algorithm (a) and the EGS-SOM algorithm (b). Ground Truth is shown in (c). Slices 110, 120, 130, 140, 150, and 160 on the axial plane are shown on each column. First column corresponds to WM, second column to GM, and third column to CSF.

to compare the segmentation performance of their proposals [13, 16, 17, 21, 26, 37–41]. Tanimoto's index can be defined as

$$T(S_1, S_2) = \frac{|S_1 \cap S_2|}{|S_1 \cup S_2|}, \tag{9}$$

where S_1 is the segmentation set and S_2 is the ground truth.

5. Conclusions

In this paper we presented fully unsupervised segmentation methods for MR images based on hybrid artificial intelligence techniques for improving the feature extraction process and self-organizing maps for pixel classification. The use of a genetic algorithm provides a way for training the Self-Organizing map used as a classifier in the most efficient way. This is because the dimension of the training samples

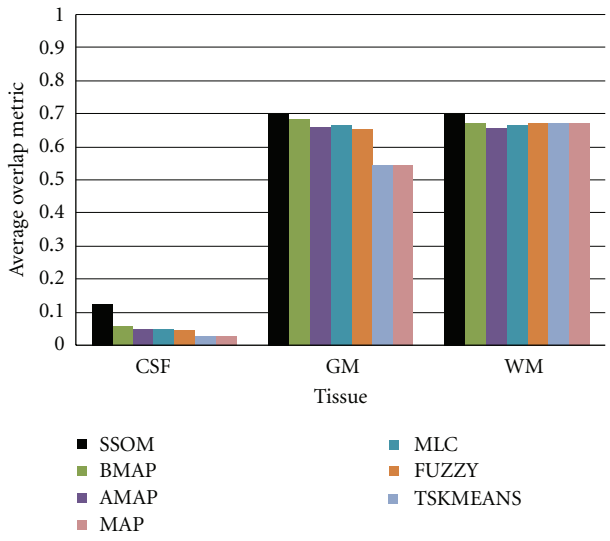


Figure 6: Average overlap metric comparison for different segmentation methods.

(feature vectors) has been reduced in order to be enough discriminant but not redundant. As a result, the number of units (neurons) on the map is also optimized as well as the classification process. Thus, we take advantage of the competitive learning model of the SOM which groups the pixels into clusters. This competitive process discovers discriminant among the pixels, resulting in an unsupervised way to segment the image. Moreover, the clusters' borders are redefined by using the entropy-gradient method presented on this paper. The whole process allows figuring out the segments present on the image without using any a priori information.

The results shown in Section 4 have been compared with the segmentations provided by the IBSR database that outperform the results obtained by other algorithms such as *k*-means or fuzzy *k*-means. The number of segments or different tissues found in an MR image is figured out automatically making possible to find out tissues which could be identified with a pathology.

Acknowledgment

This work was partly supported by the *Consejería de Innovación, Ciencia y Empresa* (Junta de Andalucía, Spain) under the Excellence Projects TIC-02566 and TIC-4530.

References

[1] I. A. Illán, J. M. Górriz, J. Ramírez et al., "18F-FDG PET imaging analysis for computer aided Alzheimer's diagnosis," *Information Sciences*, vol. 181, no. 4, pp. 903–916, 2011.

[2] I. A. Illán, J. M. Górriz, M. M. López et al., "Computer aided diagnosis of Alzheimer's disease using component based SVM," *Applied Soft Computing Journal*, vol. 11, no. 2, pp. 2376–2382, 2011.

[3] J. M. Górriz, F. Segovia, J. Ramírez, A. Lassl, and D. Salas-Gonzalez, "GMM based SPECT image classification for the diagnosis of Alzheimer's disease," *Applied Soft Computing Journal*, vol. 11, no. 2, pp. 2313–2325, 2011.

[4] M. Kamber, R. Shinghal, D. L. Collins, G. S. Francis, and A. C. Evans, "Model-based 3-D segmentation of multiple sclerosis lesions in magnetic resonance brain images," *IEEE Transactions on Medical Imaging*, vol. 14, no. 3, pp. 442–453, 1995.

[5] J. Ramírez, J. M. Górriz, D. Salas-Gonzalez et al., "Computer-aided diagnosis of Alzheimer's type dementia combining support vector machines and discriminant set of features," *Information Sciences*. In press.

[6] D. N. Kennedy, P. A. Filipek, and V. S. Caviness, "Anatomic segmentation and volumetric calculations in nuclear magnetic resonance imaging," *IEEE Transactions on Medical Imaging*, vol. 8, no. 1, pp. 1–7, 1989.

[7] A. Khan, S. F. Tahir, A. Majid, and T. S. Choi, "Machine learning based adaptive watermark decoding in view of anticipated attack," *Pattern Recognition*, vol. 41, no. 8, pp. 2594–2610, 2008.

[8] Z. Yang and J. Laaksonen, "Interactive retrieval in facial image database using self-organizing maps," in *Proceedings of the MVA*, 2005.

[9] M. García-Sebastián, E. Fernández, M. Graña, and F. J. Torrealdea, "A parametric gradient descent MRI intensity inhomogeneity correction algorithm," *Pattern Recognition Letters*, vol. 28, no. 13, pp. 1657–1666, 2007.

[10] E. Fernández, M. Graña, and J. R. Cabello, "Gradient based evolution strategy for parametric illumination correction," *Electronics Letters*, vol. 40, no. 9, pp. 531–532, 2004.

[11] M. García-Sebastián, A. Isabel González, and M. Graña, "An adaptive field rule for non-parametric MRI intensity inhomogeneity estimation algorithm," *Neurocomputing*, vol. 72, no. 16-18, pp. 3556–3569, 2009.

[12] T. Kapur, L. Grimson, W. M. Wells, and R. Kikinis, "Segmentation of brain tissue from magnetic resonance images," *Medical Image Analysis*, vol. 1, no. 2, pp. 109–127, 1996.

[13] Y. F. Tsai, I. J. Chiang, Y. C. Lee, C. C. Liao, and K. L. Wang, "Automatic MRI meningioma segmentation using estimation maximization," in *Proceedings of the 27th Annual International Conference of the Engineering in Medicine and Biology Society (IEEE-EMBS '05)*, pp. 3074–3077, September 2005.

[14] J. Xie and H. T. Tsui, "Image segmentation based on maximum-likelihood estimation and optimum entropy-distribution (MLE-OED)," *Pattern Recognition Letters*, vol. 25, no. 10, pp. 1133–1141, 2004.

[15] Y. Zhang, M. Brady, and S. Smith, "Segmentation of brain MR images through a hidden Markov random field model and the expectation-maximization algorithm," *IEEE Transactions on Medical Imaging*, vol. 20, no. 1, pp. 45–57, 2001.

[16] N. A. Mohamed, M. N. Ahmed, and A. Farag, "Modified fuzzy c-mean in medical image segmentation," in *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '99)*, pp. 3429–3432, March 1999.

[17] W. M. Wells III, W. E. L. Crimson, R. Kikinis, and F. A. Jolesz, "Adaptive segmentation of mri data," *IEEE Transactions on Medical Imaging*, vol. 15, no. 4, pp. 429–442, 1996.

[18] D. Tian and L. Fan, "A brain MR images segmentation method based on SOM neural network," in *Proceedings of the 1st International Conference on Bioinformatics and Biomedical Engineering (ICBBE '07)*, pp. 686–689, July 2007.

[19] I. Güler, A. Demirhan, and R. Karakiş, "Interpretation of MR images using self-organizing maps and knowledge-based expert systems," *Digital Signal Processing*, vol. 19, no. 4, pp. 668–677, 2009.

[20] P. K. Sahoo, S. Soltani, and A. K. C. Wong, "A survey of thresholding techniques," *Computer Vision, Graphics and Image Processing*, vol. 41, no. 2, pp. 233–260, 1988.

[21] W. Sun, "Segmentation method of MRI using fuzzy Gaussian basis neural network," *Neural Information Processing*, vol. 8, no. 2, pp. 19–24, 2005.

[22] J. Alirezaie, M. E. Jernigan, and C. Nahmias, "Automatic segmentation of cerebral MR images using artificial neural networks," *IEEE Transactions on Nuclear Science*, vol. 45, no. 4, pp. 2174–2182, 1998.

[23] A. Ortiz, J. M. Górriz, J. Ramírez, and D. Salas-Gonzalez, "MR brain image segmentation by hierarchical growing SOM and probability clustering," *Electronics Letters*, vol. 47, no. 10, pp. 585–586, 2011.

[24] T. Kohonen, *Self-Organizing Maps*, Springer, 2001.

[25] E. Arsuaga and F. Díaz, "Topology preservation in SOM," *International Journal of Mathematical and Computer Sciences*, vol. 1, no. 1, pp. 19–22, 2005.

[26] K. Taşdemir and E. Merényi, "Exploiting data topology in visualization and clustering of self-organizing maps," *IEEE Transactions on Neural Networks*, vol. 20, no. 4, pp. 549–562, 2009.

[27] E. Alhoniemi, J. Himberg, J. Parhankagas, and J. Vesanta, "SOM Toolbox for Matlab v2.0," 2005, http://www.cis.hut.fi/projects/somtoolbox.

[28] M. O. Stitson, J. A. E. Weston, A. Gammerman, V. Vork, and V. Vapnik, "Theory of support vector machines," Tech. Rep. CSD-TR-96-17, Department of Computer Science, Royal Holloway College, University of London, 1996.

[29] M. Nixson and A. Aguado, *Feature Extraction and Image Processing*, Academic Press, 2008.

[30] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 3, no. 6, pp. 610–621, 1973.

[31] M. Hu, "Visual pattern recognition by moments invariants," *IRE Transactions on Information Theory*, vol. 8, pp. 179–187, 1962.

[32] Internet Brain Database Repository, Massachusetts General Hospital, Center for Morphometric Analysis, 2010, http://www.cma.mgh.harvard.edu/ibsr/data.html.

[33] J. C. Rajapakse and F. Kruggel, "Segmentation of MR images with intensity inhomogeneities," *Image and Vision Computing*, vol. 16, no. 3, pp. 165–180, 1998.

[34] J. L. Marroquin, B. C. Vemuri, S. Botello, F. Calderon, and A. Fernandez-Bouzas, "An accurate and efficient Bayesian method for automatic segmentation of brain MRI," *IEEE Transactions on Medical Imaging*, vol. 21, no. 8, pp. 934–945, 2002.

[35] J. C. Bezdek, L. O. Hall, and L. P. Clarke, "Review of MR image segmentation techniques using pattern recognition," *Medical Physics*, vol. 20, no. 4, pp. 1033–1048, 1993.

[36] L. P. Clarke, R. P. Velthuizen, M. A. Camacho et al., "MRI segmentation: methods and applications," *Magnetic Resonance Imaging*, vol. 13, no. 3, pp. 343–368, 1995.

[37] C. T. Su and H. C. Lin, "Applying electromagnetism-like mechanism for feature selection," *Information Sciences*, vol. 181, no. 5, pp. 972–986, 2011.

[38] K. Tan, E. Khor, and T. Lee, *Multiobjective Evolutionary and Applications*, Springer, 1st edition, 2005.

[39] T. Tasdizen, S. P. Awate, R. T. Whitaker, and N. L. Foster, "MRI tissue classification with neighborhood statistics: a nonparametric, entropy-minimizing approach," in *Proceedings of the International Conference on Medical Image Computing and Computer Assisted Intervention (MICCAI '05)*, 2005.

[40] I. Usman and A. Khan, "BCH coding and intelligent watermark embedding: employing both frequency and strength selection," *Applied Soft Computing Journal*, vol. 10, no. 1, pp. 332–343, 2010.

[41] Y. Wang, T. Adali, S. Y. Kung, and Z. Szabo, "Quantification and segmentation of brain tissues from MR images: a probabilistic neural network approach," *IEEE Transactions on Image Processing*, vol. 7, no. 8, pp. 1165–1181, 1998.

Hindawi Publishing Corporation
Advances in Artificial Neural Systems
Volume 2012, Article ID 962105, 13 pages
doi:10.1155/2012/962105

Research Article

Measuring Non-Gaussianity by Phi-Transformed and Fuzzy Histograms

Claudia Plant,[1] Son Mai Thai,[2] Junming Shao,[3] Fabian J. Theis,[4] Anke Meyer-Baese,[1] and Christian Böhm[2]

[1] *400 Dirac Science Library, Florida State University, Tallahassee, FL 32306-4120, USA*
[2] *Department for Informatics, Research Unit for Database Systems, University of Munich, Oettingenstraße 67, 80538 Munich, Germany*
[3] *Klinikum rechts der Isar der TUM, Ismaninger Straße 22, 81675 Munich, Germany*
[4] *Helmholtz Zentrum München, Ingolstädter Landstraße 1, 85764 Neuherberg, Germany*

Correspondence should be addressed to Claudia Plant, cplant@fsu.edu

Received 14 February 2012; Accepted 1 April 2012

Academic Editor: Juan Manuel Gorriz Saez

Independent component analysis (ICA) is an essential building block for data analysis in many applications. Selecting the truly meaningful components from the result of an ICA algorithm, or comparing the results of different algorithms, however, is nontrivial problems. We introduce a very general technique for evaluating ICA results rooted in information-theoretic model selection. The basic idea is to exploit the natural link between non-Gaussianity and data compression: the better the data transformation represented by one or several ICs improves the effectiveness of data compression, the higher is the relevance of the ICs. We propose two different methods which allow an efficient data compression of non-Gaussian signals: Phi-transformed histograms and fuzzy histograms. In an extensive experimental evaluation, we demonstrate that our novel information-theoretic measures robustly select non-Gaussian components from data in a fully automatic way, that is, without requiring any restrictive assumptions or thresholds.

1. Introduction

Independent component analysis (ICA) is a powerful technique for signal demixing and data analysis in numerous applications. For example, in neuroscience, ICA is essential for the analysis of functional magnetic resonance imaging (fMRI) data and electroencephalograms (EEGs). The function of the human brain is very complex and can be only imaged at a very coarse spatial resolution. Millions of nerve cells are contained in a single voxel of fMRI data. The neural activity is indirectly measured by the so-called BOLD-effect, that is, by the increased supply of active regions with oxygenated blood. In EEG, the brain function can be directly measured by the voltage fluctuations resulting from ionic current flows within the neurons. The spacial resolution of EEG, however, is even much lower than that of fMRI. Usually, an EEG is recorded using an array of 64 electrodes distributed over the scalp. Often, the purpose of acquiring fMRI or EEG data is obtaing a better understanding of brain function while the subject is performing some task. An example for such an experiment is to show subjects images while they are in the scanner to study the processing of visual stimuli, see Section 4.1.4. Recent results in neuroscience, for example [1], confirm the organization of the human brain into distinct functional modules. During task processing, some functional modules are actively contributing to the task. However, many other modules are also active but not involved into task-specific activities. Due to the low resolution of fMRI and EEG data, we observe a partial volume effect: the signal at one particular voxel or electrode consists of task-related activities, nontask-related activities, and a lot of noise. ICA is a powerful tool for signal demixing and, therefore, in principle very suitable to reconstruct the interesting task-related activity.

Many ICA algorithms use the non-Gaussianity as implicit or explicit optimization goal. The rationale behind this decision is due to a reversion of the central limit theorem: the sum of a sufficiently large number of independent random

variables, each with finite mean and variance, will approximate a normal distribution. Therefore, an algorithm for the demixing of signals has to optimize for non-Gaussianity in order to obtain the original signals. We adopt this idea in this paper and define a data compression method which yields a high compression rate exactly if the data distribution is far away from Gaussianity and no compression in the data distribution is exactly gaussian.

However, the evaluation and interpretation of the result of ICA is often difficult for two major reasons. First, most ICA algorithms always yield a result, even if the underlying assumption (e.g., non-Gaussianity for the algorithm FastICA [2]) is unfulfilled. Thus, many ICA algorithms extract as many independent sources as there are mixed signals in the dataset, no matter how many of them really fulfill the underlying assumption. Second, ICA has no unique and natural evaluation criterion to assess the relevance or strength of the detected result (like, e.g., the variance criterion for principle component analysis (PCA)). Different ICA algorithms use different objective functions, and to select one of them as an *overall objective*, or *neutral* criterion would give unjustified preference to the result of that specific algorithm. Moreover, if the user is interested in comparing an ICA result to completely different modeling techniques like PCA, regression, mixture models, and so forth, these ICA-internal criteria are obviously unsuitable. Depending on the actual intension of the user, different model selection criteria for ICA might be appropriate. In this paper, we investigate the *compressibility* of the data as a more neutral criterion for the quality of single component or the overall ICA result.

2. Related Work

2.1. Model Selection for ICA. Model selection for ICA or automatically identifying the most interesting components is an active research question. Perhaps the most widely used options for model selection are measures like Kurtosis, Skewness, and approximations of neg-entropy [3]. However, these measures are also applied as optimization criteria by some ICA algorithms. Thus, a comparison of the results across algorithms is impossible. Moreover, these measures are very sensitive with respect to noise points and single outliers.

In [4], Rasmussen et al. propose an approach for model selection of epoched EEG signals. In their model order selection procedure, the data set is split into two sets, training- and test set, to ensure an unbiased measure of generalization. With each model hypothesis, the negative logarithm of the likelihood function is then calculated using a probabilistic framework on the training and test set. The model having minimal generalization error is selected. This approach, however, is based on certain assumptions about source autocorrelation and tends to be sensitive to noise.

The most common method for model order selection is based on principal component analysis (PCA) of the data covariance matrix, which is proposed by Hyvärinen et al. [3]. The choice of number of sources to be selected is based on the number of dominant eigenvalues which significantly contribute to the total variance. This approach is fast and simple to implement, however, it suffers from a number of problems, for example, an inaccurate eigenvalue decomposition of the data covariance matrix in the noise-free case with fewer numbers of sources than sensors and sensitivity to noise. Moreover, there are no reasons to say that the subspace spanned by dominant principal components contains the source of interest [5]. Another approach proposed by James and Hesse [5] is to do the step-wise extraction of the sources until it reaches a predefined accuracy. However, the choice of reasonable accuracy level is also one drawback of this algorithm.

Related to model selection but still a different problem is the reliability of ICA results. The widely used iterative fix-point algorithm FastIca [6] converges towards different local optima of the optimization surface. The technique Icasso [7] combines Bootstrapping with a visualization to allow the user to investigate the relationship between different ICA results. Reliable results can be easily identified as dense clusters in the visualization. However, no information on the quality of the results is provided, which is the major focus of our work. Similar to Icasso, Meinecke et al. [8] proposed a resampling method to assess the quality ICA results by computing the stability of the independent subspaces. First, they create surrogate datasets by randomly selecting independent components from an ICA decomposition and apply the ICA algorithm for each of the surrogate data sets. Then, they separate the data space into one or multidimensional subspaces by their block structure and compute the uncertainty for each subspace. This proposed reliability estimation can be used to choose the appropriate BSS-model, to enhance the separation performance and, most importantly, to flag components which have a physical meaning.

2.2. Minimum Description Length for Model Selection. The minimum description length (MDL) principle is based on the simple idea that the best model to describe the data is one with the overall shortest description of the data and model itself, and it is essentially the same as Occam's razor.

The MDL principle has been successfully applied for model selection for a large variety of tasks, ranging from linear regression [9], image segmentation [10] to polyhedral surface models [11].

In data mining, the MDL principle has recently attracted some attention enabling parameter-free algorithms to graph mining [12], clustering, for example [13–15], and outlier detection [16]. Sun et al. [12] proposed GraphScope, a parameter-free technique to mine information from streams of graphs. This technique used MDL to decide how and when to form and modify communities automatically. Böhm et al. [15] proposed OCI, a novel fully automatic algorithm to clustering non-Gaussian data with outliers, based on MDL to control the splitting, filtering, and merging phase in a parameter-free and very efficient top-down clustering approach. CoCo [16], a technique for parameter-free outlier detection, is based on the ideas of data compression and coding costs. CoCo used MDL to define an intuitive outlier factor together with a novel algorithm for outlier detection.

This technique is parameter free and can be applied to a wide range of data distributions. OCI combines local ICA with clustering and outlier filtering. Related to this idea, in [17], Gruber et al. propose an approach for automated image denoising combining local PCA or ICA with model selection by MDL.

In this paper, we propose a model selection criterion based on the MDL principle suitable for measuring the quality of single components as well as complete ICA results.

3. Independent Component Analysis and Data Compression

One of the fundamental assumptions of many important ICA algorithms is that independent sources can be found by searching for maximal non-Gaussian directions in the data space. Non-Gaussianity leads to a decrease in entropy, and therefore, to a potential improvement of the efficiency of data compression. In principle, the achievable compression rate of a dataset, after ICA, is higher compared to the original dataset. The principle of minimum description length (MDL) uses the probability $P(x)$ of a data object x to represent it according to Huffman coding. Huffman coding gives a lower bound for the compression rate of a data set D achievable by any concrete coding scheme, as follows: $\sum_{x\in D} -\log_2(P(x))$. If x is taken from a continuous domain (e.g., the vector space $\mathcal{R}^d$ for the blind source separation of a number d of signals), the relative probability given by a probability density function $p(x)$ is applied instead of the absolute probability. The relative and absolute log-likelihoods (which could be obtained by discretizing x) are identical up to a constant value which can be safely ignored, as we discuss in detail in Section 3.1. For a complete description of the dataset (allowing decompression), the parameters of the probability density function (PDF) such as mean and variance for Gaussian PDFs need to be coded and their code lengths added to the negative log-likelihood of the data. We call this term the code book. For each parameter, a number of bits equal to $(1/2)\log_2(n)$ where n is the number of objects in D, is required, as fundamental results from information theory have proven [18]. Intuitively, the term $(1/2)\log_2(n)$ reflects the fact that the parameters need to be coded more precisely when a higher number n of data objects is modeled by the PDF. The MDL principle is often applied for model selection of parametric models like Gaussians, or Gaussian mixture models (GMMs). Gaussian mixture models vary in the model complexity, that is, the number of parameters needed for modeling. MDL-based techniques are well able to compare models of different complexity. The main purpose of the code book is to punish complexity in order to avoid overly complex, over-fitted models (like a GMM having one component exactly at the position of each data object: such a model would yield a minimal Huffman coding, but also a maximal code-book length). By the two concepts, Huffman coding using the negative log-likelihood of a PDF and the code-book for the parameters of the PDF, the principle of MDL provides a very general framework which allows the comparison of very different modeling techniques like principal component analysis (based on a Gaussian PDF model), clustering [13], regression [19] for continuous domains, but, in principle, also for discrete or mixed domains. At the same time, model complexity is punished and, therefore, overfitting avoided. Related criteria for general model selection include, for example, the Bayesian information criterion and the Aikake information criterion. However, these criteria are not adapted to the ICA model. In the following section, we discuss how to apply the MDL principle in the context of ICA.

3.1. General Idea of the Minimum Description Length Principle. The minimum description length (MDL) principle is a well-established technique for selecting the best model out of a finite or infinite number of possible models for a given data set D (in our case, a signal). The model is usually given in terms of a probability function $f(x)$ which assigns to every element $x \in D$ a probability that this element occurs in the dataset. For continuous domains, $f(x)$ is a probability density function satisfying $\int_{-\infty}^{+\infty} f(x)\mathbf{d}x = 1$. The idea of MDL is that $f(x)$ can be used as a basis to compress the data set D using Huffman coding and to exploit that this coding becomes the more efficient (w.r.t. the achievable compression rate, or more precisely the code length after compression) the better $f(x)$ represents the true data distribution. According to Huffman coding, the minimum code length corresponds to the negative log-likelihood of the data set, that is,

$$\mathrm{NLLH}_f(D) = -\sum_{x\in D} \log_2 f(x). \tag{1}$$

While only for discrete domains the values of $f(x)$ are scaled between 0 and 1, this negative log-likelihood is also applied for continuous domains, but then some caveats apply. Basically, we can always reduce the continuous case to the noncontinuous case by discretizing the data (e.g., by a regular grid with a fixed resolution g). In this case,

$$F_g(x) = \int_{g\cdot\lfloor x/g\rfloor}^{g\cdot\lfloor x/g+1\rfloor} f(\xi)\mathbf{d}\xi \tag{2}$$

is a probability function scaled between 0 and 1 with

$$\lim_{g\to 0} \frac{F_g(x)}{g} = f(x). \tag{3}$$

For the negative log-likelihood of the so-discretized dataset D_g, we get

$$\begin{aligned} \mathrm{NLLH}_f\left(D_g\right) &= -\sum_{x\in D} \log_2 F_g(x) \\ &\approx -\sum_{x\in D} \log_2 g \cdot f(x) \\ &= \mathrm{NLLH}_f(D) - n\log_2 g, \end{aligned} \tag{4}$$

where in the case $g \to 0$ we have exact equality and also $-n\log_2 g \to \infty$, corresponding to the obvious fact that we need an infinite number of bits to represent a real number with infinite precision. However, when comparing different

models of the data (i.e., different probability functions $f_1(x)$ and $f_2(x)$), we simply have to ensure that all data are basically discretized with the same (and sufficiently high) resolution g in the original data space and then ignore the term $-n\log_2 g$ which is equal in all compared models of the data (note that data in a computer is always represented with finite precision, and, therefore, always implicitly discretized):

$$\begin{aligned}\lim_{g\to 0}&\left(\mathrm{NLLH}_{f_2}\left(D_g\right)-\mathrm{NLLH}_{f_1}\left(D_g\right)\right)\\ &=\lim_{g\to 0}\Big(\left(\mathrm{NLLH}_{f_2}(D)-n\log_2 g\right)\\ &\qquad-\left(\mathrm{NLLH}_{f_1}(D)-n\log_2 g\right)\Big)\\ &=\mathrm{NLLH}_{f_2}(D)-\mathrm{NLLH}_{f_1}(D).\end{aligned}\tag{5}$$

We simply have to observe that (1) the resolution is not implicitly changed by any transformations of the dataset (like, e.g., a linear scaling $\alpha\cdot x$ of the data objects) and (2) that ignoring the term might lead to negative values of $\mathrm{NLLH}_f(D)$ (so we partly lose the nice intuition of a number of bits encoding the data objects, and particularly that 0 is a lower limit of this amount of information).

The principle of coding a signal with a pdf $f(x)$ is visualized in Figure 1 where a signal (a superposition of three sinuses) is coded. To represent one given point of the signal (at $t = 15.5$), we should actually use a discretization of x as indicated in the right part of the diagram to obtain an actual code length for the value x. However, we can also directly use the negative log-likelihood of the value x with the above-mentioned implications.

In addition to the amount of information which is caused by the negative log-likelihood of the data, we need also to code the function f. From an information-coding perspective, we need this information in order to be able to decode the data again after transferring it through a communication channel. The function f tells us which code words translate back to what original data objects, so it serves as a code book. From a statistical perspective, the coding of f is needed to avoid overfitting. The intention of f is to generalize the data, and not to anticipate it, as a weird function would do which has simply a peak at every position where a data object is available. For both purposes, the representation of the code book must require considerably less information than the data itself. In this paper, we will propose two different methods to represent the function f. In Figure 1, a kernel density estimator (KDE) was used. However, KDE needs a number of parameters which is the same as the number of points, and, therefore, it is not suitable for our purpose. The classical (parametric) method is to use a class of model functions (like a Gaussian pdf) and to code the parameters (for Gaussian, μ and σ) with an amount of information corresponding to $(1/2)\log_2 n$ per parameter. This number can be derived by an optimization process which takes into account that a small error in the parameters does not lead to a serious deterioration of the NLLH, particularly if n is small and only a few data objects are modeled by f. The number of $(1/2)\log_2 n$ bits represents an optimal trade-off (and includes this deterioration of NLLH already). Throughout this paper, we will use the minimum description length of a dataset as goal for minimization:

$$\mathrm{MDL}_f(D)=\mathrm{NLLH}_f(D)+\frac{1}{2}\#\mathrm{PAR}(f)\cdot\log_2 n.\tag{6}$$

We will in the following sections propose nonparametric methods which are both related to histograms. Therefore, the number of parameters in principle corresponds to the number of bins. We do not use histograms directly since our goal is to code the signals in a way that punishes the Gaussianity and rewards the non-Gaussianity. Thus, we propose two different methods which modify the histogram concept in a suitable way.

3.2. Phi-Transformed Histograms. Techniques like [15] or [20] successfully use the exponential power distribution (EPD), a generalized distribution function including Gaussian, Laplacian, uniform, and many other distribution functions for assessing the ICA result using MDL. The reduced entropy of non-Gaussian projections in the data allows a higher compression rate and thus favors a good ICA result. However, the selection of EPD is overly restrictive. For instance, multimodal and asymmetric distributions cannot be well represented by EPD but are highly relevant to ICA. In the following, we describe an alternative representation of the PDF which is efficient if (and only if) the data is considerably different from Gaussian. Besides the non-Gaussianity, we have no additional assumption (like for instance EPD) on the data. To achieve this, we tentatively assume Gaussianity in each signal of length n and transform the assumed Gaussian distribution into a uniform distribution in the interval $(0, 1)$ by applying the Gaussian cumulative distribution function $\Phi((x-\mu)/\sigma)$ (the Φ-transformation) to each signal of the data representation to be tested (e.g., after projection on the independent components). Then, the resulting distribution is represented by a histogram $(H_1,\ldots,H_b)$ with a number b of equidistant bins where b is optimized as we will show later. $H_j(1\le j\le b)$ is the number of objects falling in the corresponding half open interval $[(j-1)/b,\ j/b)$. If the signal is Gaussian indeed, then the signals after Φ-transformation will be uniform and the histogram bins will be (more or less) uniformly filled. Therefore, a trivial histogram with only one bin will in this case yield the best coding cost (and thus, no real data compression comes into effect). The Φ-transformation itself causes a change of the coding cost which is equivalent to the entropy of the Gaussian distribution function, as we show in as follows.

The negative log-likelihood of the signal before the Φ-transformation with the tentative assumption that the signals are compressed by the Gaussian pdf correspond to

$$\begin{aligned}\mathrm{NLLH}_{\mathrm{before}}&=\sum_{x\in D}-\log_2\left(\frac{1}{\sqrt{2\pi\sigma^2}}\mathbf{e}^{-(x-\mu)^2/2\sigma^2}\right)\\ &=n\cdot\log_2\left(\sqrt{2\pi\sigma^2}\right)+\frac{1}{2\sigma^2\ln 2}\sum_{x\in D}(x-\mu)^2,\end{aligned}\tag{7}$$

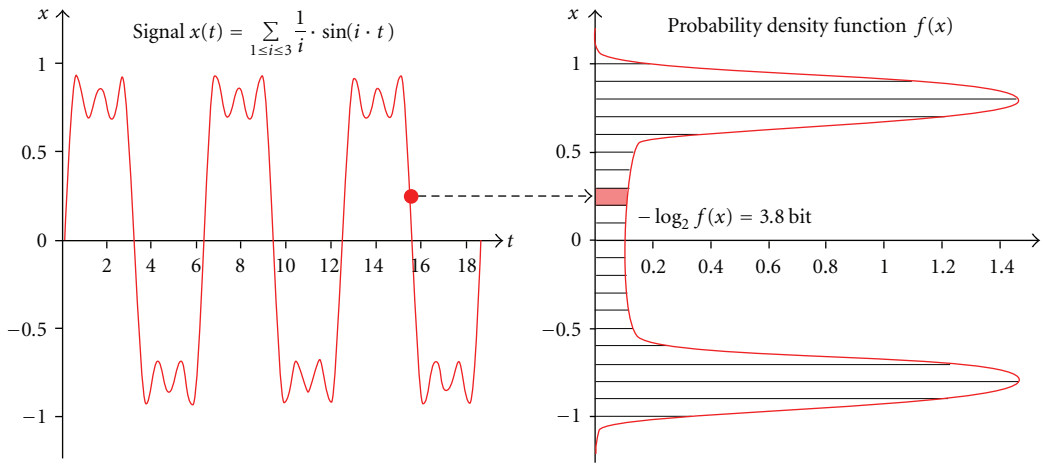


Figure 1: MDL-based compression of signals by Huffman coding.

and since $(1/n)\sum (x-\mu)^2$ is exactly the definition of the variance σ^2, we have

$$\mathrm{NLLH}_{\mathrm{before}} = n \cdot \log_2\left(\sqrt{2\pi \mathbf{e}\sigma^2}\right), \tag{8}$$

which is independent from the distribution which the signal x actually has. After the Φ-transformation, we code the signal under the assumption that it is uniformly distributed in $(0, 1)$. Thus, we obtain

$$\mathrm{NLLH}_{\mathrm{after}} = \sum_{x \in D} -\log_2(1) = 0, \tag{9}$$

and again, we do not worry that it appears as if no information is necessary to code the signals after the Φ-transformation. But the difference between coding of the signals before and after the Φ-transformation corresponds to

$$\mathrm{PRE} = \mathrm{NLLH}_{\mathrm{before}} - \mathrm{NLLH}_{\mathrm{after}} = n \cdot \log_2\left(\sqrt{2\pi \mathbf{e}\sigma^2}\right). \tag{10}$$

Representing the histogram $(H_1, \ldots, H_b)$ as a probability density function (integrating to 1) leads to

$$f_H(x) = H_{\lfloor b \cdot x + 1 \rfloor} \cdot \frac{b}{n}. \tag{11}$$

The negative log-likelihood of this Φ-transformed signal corresponds to

$$\begin{aligned} \mathrm{NLLH}_{f_H}(D) &= \sum_{x \in D} -\log_2\left(H_{\lfloor b \cdot x + 1 \rfloor} \cdot \frac{b}{n}\right) \\ &= \sum_{x \in D} \log_2 \frac{n}{H_{\lfloor b \cdot x + 1 \rfloor}} - \sum_{x \in D} \log_2 b, \end{aligned} \tag{12}$$

and since we have H_i objects in histogram bin i, we can change the first sum into the entropy of the histogram:

$$\mathrm{NLLH}_{f_H}(D) = \sum_{1 \le i \le b} \left(H_i \cdot \log_2 \frac{n}{H_i}\right) - n \cdot \log_2 b. \tag{13}$$

Using this coding scheme, the overall code length ($\mathrm{CLRG}(D, b)$, code length relative to Gaussianity) of the signal is provided by

$$\begin{aligned} \mathrm{CLRG}(D, b) = &\overbrace{\sum_{1 \le j \le b} H_j \cdot \log_2 \frac{n}{H_j}}^{\text{histogram entropy}} \\ &- \overbrace{n \cdot \log_2 b}^{\text{offset cost}} \\ &+ \overbrace{\frac{b-1}{2} \log_2 n}^{\text{code book}} \\ &+ \overbrace{\mathrm{PRE}}^{\text{preproc}}. \end{aligned} \tag{14}$$

As introduced in Section 3, the first two terms represent the negative log-likelihood of the data given the histogram. The first corresponds to the entropy of the histogram, and the second term, stemming from casting the histogram into a PDF, has also the following intuition: when coding the same data with a varying number of histogram bins, the resulting log-likelihoods are based on different basic resolutions of the data space (a grid with a number b of partitions). Although the choice of a particular basic resolution is irrelevant for the end-result, for comparability, all alternative solutions must be based on a common resolution. We choose $g = 1$ as basic resolution, and subtract for each object the number of bits by which we know the position of the object more precisely than in the basic resolution. The trivial histogram having $b = 1$ represents the case where the data is assumed to be Gaussian: since the Gaussian cumulative distribution function $\Phi((x-\mu)/\sigma)$ has been applied to the data, Gaussian data are transformed into uniform data, and our histograms have an implicit assumption of uniformity *inside* each bin. Therefore, we call it *offset cost* because it stands for coding the position of a value *inside* a histogram bin. If some choice of $b \neq 1$ leads to smaller $\mathrm{CLRG}(D, b)$, we have evidence that the signal is different from Gaussian. It is easy to see that in

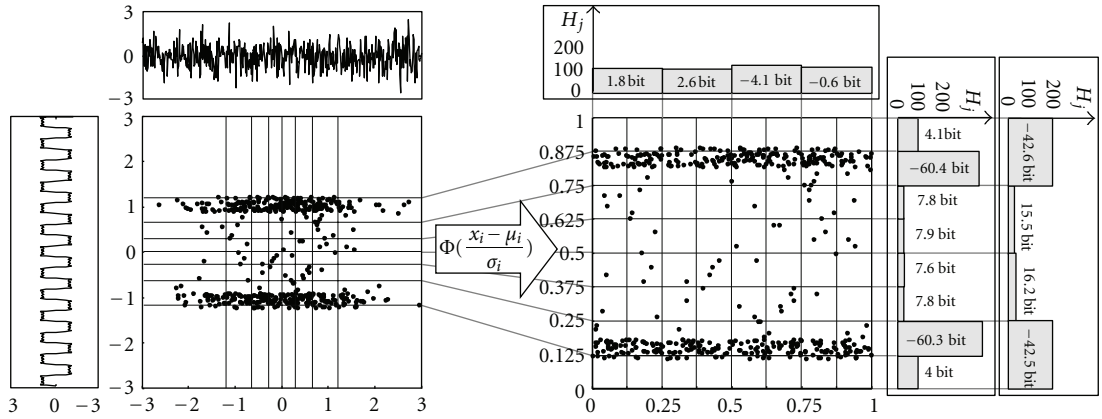


FIGURE 2: Overview of the computation of CLRG(D, b): left: scatter plot of two signals in original space. On the x-axis: Gaussian noise signal. On the y-axis: signal with a rectangular pattern. The lines represent the quantiles assuming a Gaussian distribution; right: the scatter plot after applying the Gaussian CDF. The quantiles now form an equidistant grid. On the axes: histograms and compression costs using Huffman coding. Since the Gaussian signal does not contain any pattern beyond Gaussianity, it cannot be compressed in CDF-space.

the case $b = 1$ the code length is CLRG(D, b) = PRE, which is a consequence of our definition of the offset cost. Therefore, we call our cost function code length relative to Gaussianity, (CLRG). If no choice that $b \neq 1$ leads to CLRG$(D, b) <$ PRE, then either the data is truly Gaussian or the number of data objects is not high enough to give evidence for non-Gaussianity. In the latter case, we use Gaussianity as the safe default-assumption. The third term is the cost required for the code book: to completely describe b histogram bins it is sufficient to use $b - 1$ codewords since the remaining probability is implicitly specified. The last term, PRE is for preprocessing, that is, taking the Φ-transform into account.

Figure 2 gives an overview and example of our method. On the left side, the result of an ICA run is depicted which has successfully separated a number $d = 2$ of signals (each having $n = 500$ points). The corresponding scatter-plot shows a Gaussian signal on the x-axis, a rectangular signal on the y-axis (note that the corresponding signal plots on the axes are actually transposed for better visibility). On the right side, we see the result after applying the Gaussian CDF. Some histograms with different resolutions are also shown. On the x-axis, the histogram with $b = 4$ bins is approximately uniformly filled (like also most other histograms with a different selection of b). Consequently, only a very small number of bits is saved compared to Gaussianity (e.g., only 4.1 bits for the complete signal part falling in the third bin H_3) by applying this histogram as PDF in Huffman coding (here, the cost per bin are reported including log-likelihood and offset-cost). The overall saving of 0.29 bit are contrasted by a code-book length of $(3/2)\log_2 n = 13.4$, so the histogram representation does not pay off. In contrast, the two histograms on the y-axis do pay off, since for $b = 8$, we have overall savings over Gaussianity of 81.3 bit by Huffman coding, but only $(7/2)\log_2 n = 31.4$ bits of codebook.

3.2.1. An Optimization Heuristic for the Histogram Resolution. We need to optimize b individually for each signal such that the overall coding cost CLRG(D, b) is minimized. As an efficient and effective heuristic, we propose to only consider histogram resolutions where b is a power of 2. This is time efficient since the number of alternative results is logarithmic in n (as we will show), and the next coarser histogram can be intelligently gained from the previous. In addition, the strategy is effective since a sufficient number of alternative results is examined.

We start with a histogram resolution based on the worst-case assumption that (almost) all objects fall into the same histogram bin of a histogram of very high resolution b_m. That means that the log-likelihood approaches 0. The offset cost corresponds to $-n\log_2 b_m$ but the parameter cost are very high: $((b_m - 1)/2)\log_2 n$. The other extreme case is the model with the lowest possible resolution $b = 1$ having no log-likelihood, no offset-cost, and no parameter cost. The histogram with resolution b_m can pay off only if the following condition holds:

$$n\log_2 b_m \geq \frac{b_m - 1}{2}\log_2 n, \tag{15}$$

which is certainly true if $b_m \leq n/2$. We use $b_m = 2^{\lfloor \log_2 n \rfloor}$, the first power of two less or equal n as starting resolution. Then, in each step, the algorithm generates a new histogram $H' = (H'_1, \ldots, H'_{b/2})$ from the previous histogram $H = (H_1, \ldots, H_b)$ by merging each pair of adjacent bins using $H'_j = H_{2j-1} + H_{2j}$ for all j having $1 \leq j \leq b/2$. The overall number of adding operations for histogram bins starting from the histogram $H^{\text{start}} = (H_1^{\text{start}}, \ldots, H_{b_m}^{\text{start}})$ to the final histogram $H^{\text{end}} = (H_1^{\text{end}})$ corresponds to

$$\sum_{1 \leq i \leq b_m/2} i = b_m - 1 = 2^{\lfloor \log_2 n \rfloor} - 1 \in O(n). \tag{16}$$

The coding cost of the data with respect to each alternative histogram is evaluated as described in Section 3.2 and the histogram with resolution b_{opt} providing the best compression is reported as result for dimension i. In the case of $b_{\text{opt}} = 1$, no compression was achieved by assuming

non-Gaussianity. After having optimized b_{opt} for each signal separately, $\text{CLRG}(D, b)$, the coding costs of the data are provided as in Section 3.2 applying b_{opt}. To measure the overall improvement in compression achieved by ICA $\text{CLRG}(D, b)$ is summed up across all dimensions i:

$$\text{CLRG}(D) = \sum_{1 \le i \le d} \left(\min_{0 \le \log_2 b \le \lfloor \log_2 n \rfloor} \text{CLRG}(D, b) \right). \tag{17}$$

3.3. Fuzzy Histograms. Often histograms are not a good description of data since they define a discontinuous function whereas the original data distribution often corresponds to a continuous function. Since we want to focus on non-Gaussianity without any other assumption on the underlying distribution function, a good alternative to histograms is fuzzy histograms. In statistics, often kernel density estimators (KDEs) are applied in cases where a continuous representation of the distribution function is needed. However, KDEs require a number of parameters which is higher than the number of objects, and, therefore, KDEs are not suitable for our philosophy of compressing the dataset according to the defined distribution function (although other information-theoretic KDEs exist). Therefore, we apply the simpler fuzzy histograms which extend histograms as follows.

We have a kernel function $\kappa_{\mu_i,\sigma}(x)$ which is assigned to each fuzzy histogram bin i. Like with ordinary histograms, the location parameters μ_i are equidistant, that is,

$$\mu_i = m \cdot i + t, \tag{18}$$

and the scale parameter σ is uniform for all bins (and also called bandwidth). In this paper, we use the normal distribution:

$$\kappa_{\mu_i,\sigma}(x) = n_{\mu_i,\sigma}(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu_i)^2}{\sigma^2} \right) \tag{19}$$

since it allows an elegant way to express the coding cost relatively to Gaussianity without explicitly transforming the dataset using the cumulative standard normal distribution $\Phi(x)$. To every histogram bin, a weight w_i, which indicates to which extent the bin is filled, is assigned. The sum over all w_i, is unity. The fuzzy histogram then defines the probability density function:

$$f(x) = \sum_{1 \le i \le b} w_i \cdot \kappa_{\mu_i,\sigma}(x), \tag{20}$$

which is continuous (as it is a sum of continuous functions) and integrates to 1 (as all w_i sum up to one and each kernel function integrates to 1).

We determine the positions $\mu_1, \ldots, \mu_b$ (or, actually the parameters m and t to determine all μ_i in an equidistant way) as well as the bandwidth σ in an iterative learning algorithm. We initialize the parameters such that

$$\begin{aligned} \mu_1 &= \min_{x \in D}(x), \qquad \mu_b = \max_{x \in D}(x), \\ w_i &= \frac{1}{b}, \quad (\forall i, 1 \le i \le b), \qquad \sigma = \frac{\mu_b - \mu_1}{b}. \end{aligned} \tag{21}$$

That means that we set the initial slope $m = (\mu_b - \mu_1)/b$ and $t = \mu_1 - m$.

Each point $x \in D$ may be assigned to more than one bin. It is gradually assigned and the sum of all assignments equals 1. The assignment is based on Bayes' theorem:

$$p(i \mid x) = \frac{w_i \cdot \kappa_{\mu_i,\sigma}(x)}{\sum_{1 \le j \le b} w_j \cdot \kappa_{\mu_j,\sigma}(x)}, \tag{22}$$

and the weights can be determined as

$$w_i = \frac{1}{|D|} \sum_{x \in D} p(i \mid x). \tag{23}$$

Then, we assign the points according to (22). We then determine each μ_i (calling it $\hat{\mu}_i$) individually (temporarily omitting the requirement that they are equi-distant) as

$$\hat{\mu}_i = \frac{1}{|D| \cdot w_i} \sum_{x \in D} p(i \mid x) \cdot x \tag{24}$$

and determine m and t as a weighted linear regression of the $\hat{\mu}_i$. Let $\overline{\mu} = \sum_{1 \le i \le b} w_i \cdot \hat{\mu}_i$ be the weighted average of all $\hat{\mu}_i$ and $\overline{i} = \sum_{1 \le i \le b} w_i \cdot i$ the weighted average of all i. Then, we obtain

$$m = \frac{\sum_{1 \le i \le b} w_i \cdot \left(i - \overline{i}\right) \cdot \left(\hat{\mu}_i - \overline{\mu}\right)}{\sum_{1 \le i \le b} w_i \cdot \left(i - \overline{i}\right)^2}, \qquad t = \overline{\mu} - m\overline{i}. \tag{25}$$

Finally, we determine the bandwidth parameter σ by the average variance which is caused by D in every bin:

$$\sigma^2 = \frac{1}{|D|} \sum_{x \in D} \sum_{1 \le i \le b} p(i \mid x) \cdot (x - \mu_i)^2. \tag{26}$$

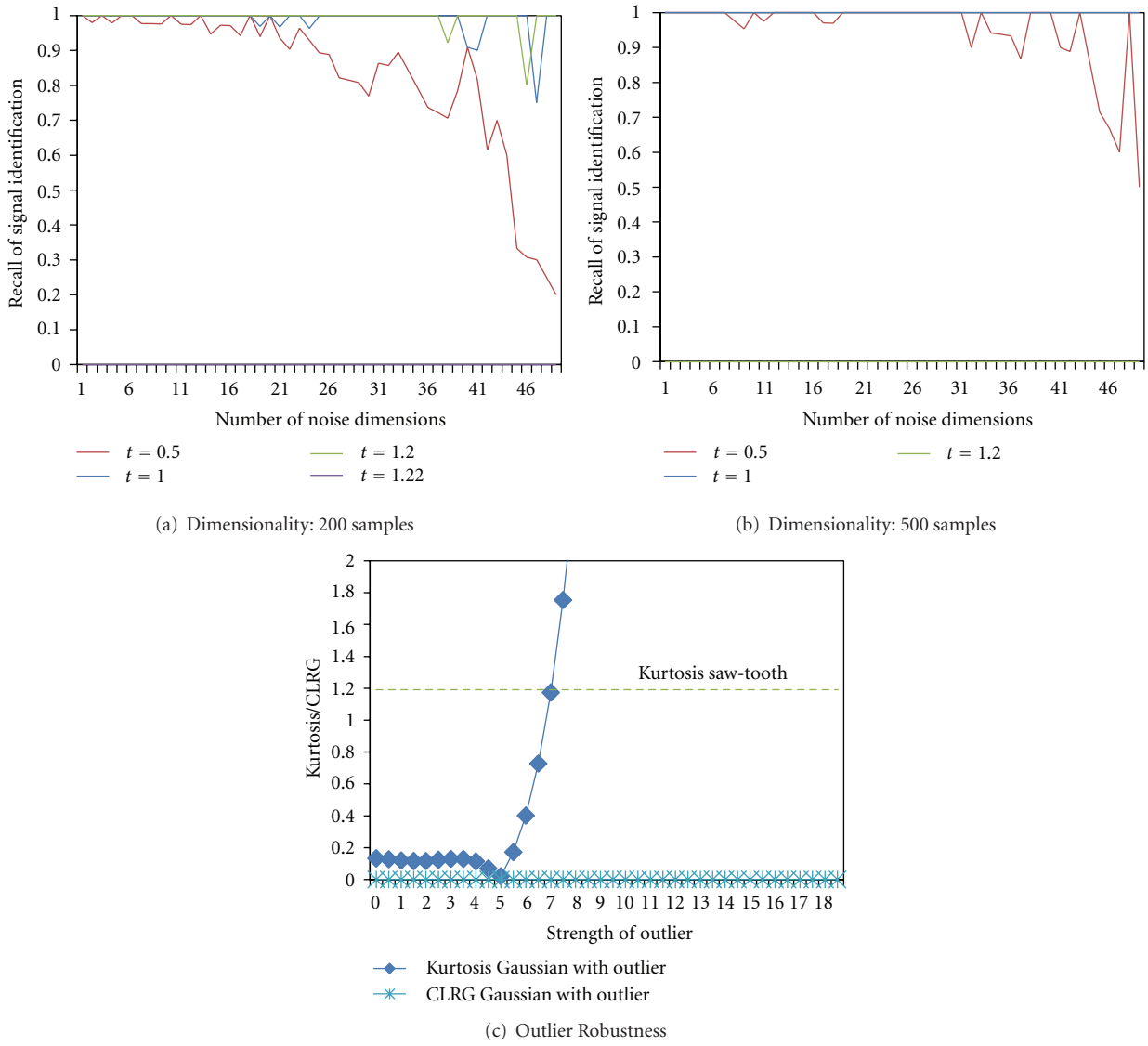
These steps starting from evaluation (22) are repeated until convergence.

4. Experiments

This section contains an extensive experimental evaluation. We start by a proof of concept demonstrating the benefits of information-theoretic model selection for ICA over established model selection criteria such as kurtosis in Section 4.1. Since in these experiments phi-transformed histograms and equidistant Gaussian Mixture Models perform very similar, for space limitations, we only show the results of phi-transformed histograms. In Section 4.2, we discuss the two possibilities of estimating the code length relative to Gaussianity.

4.1. Proof of Concept: Information-Theoretic Model Selection for ICA

4.1.1. Selection of the Relevant Dimensions. Which ICs truly represent meaningful signals? Measures like kurtosis, skewness, and other approximations of neg-entropy are often used for selecting the relevant ICs but need to be suitably thresholded, which is a nontrivial task. Figures 3(a) and 3(b)



(a) Dimensionality: 200 samples

(b) Dimensionality: 500 samples

(c) Outlier Robustness

Figure 3: Comparison of CLRG to kurtosis for selection of relevant ICs from high-dimensional data (a)-(b) and for outlier-robust estimation of IC quality (c).

display the recall of signal identification for a dataset consisting of highly non-Gaussian saw-tooth signals and a varying number of noise dimensions for various thresholds of kurtosis. Kurtosis is measured as the absolute deviation from Gaussianity. The recall of signal identification is defined as the number of signals which have been correctly identified by the selection criterion divided by the overall number of signals. Figure 3(a) displays the results for various thresholds on a dataset with 200 samples. For this signal length, a threshold of $t = 1.2$ offers the best recall in signal identification for various numbers of noise dimensions. A slightly higher threshold of 1.22 leads to a complete break down in recall to 0, which implies that all noise signals are rated as non-Gaussian by kurtosis. For the dataset of 500 samples, however, $t = 1.0$ is a suitable threshold and for $t = 1.2$, we can observe a complete breakdown in recall. Even on these synthetic examples with a very clear distinction into highly non-Gaussian signals and Gaussian noise, the range for suitable thresholding is very narrow. Moreover, the threshold depends on the signal length and of course strongly on the type of the particular signal. A reasonable approach to select a suitable threshold is to try out a wide range of candidate thresholds and to select the threshold maximizing the area under ROC. For most of our example datasets with 200 samples, a threshold of $t = 1.2$ maximizes the area under ROC. For the datasets with 20 to 36 noise dimensions, this threshold yields a perfect result with an area under ROC of 1.0. For 500 samples, however, a lower threshold is preferable on most datasets. Supported by information theory, CLRG automatically identifies the relevant dimensions without requiring any parameters or thresholds. For all examples, CLRG identifies the relevant dimensions as those dimensions allowing data compression with a precision and a recall of 100%.

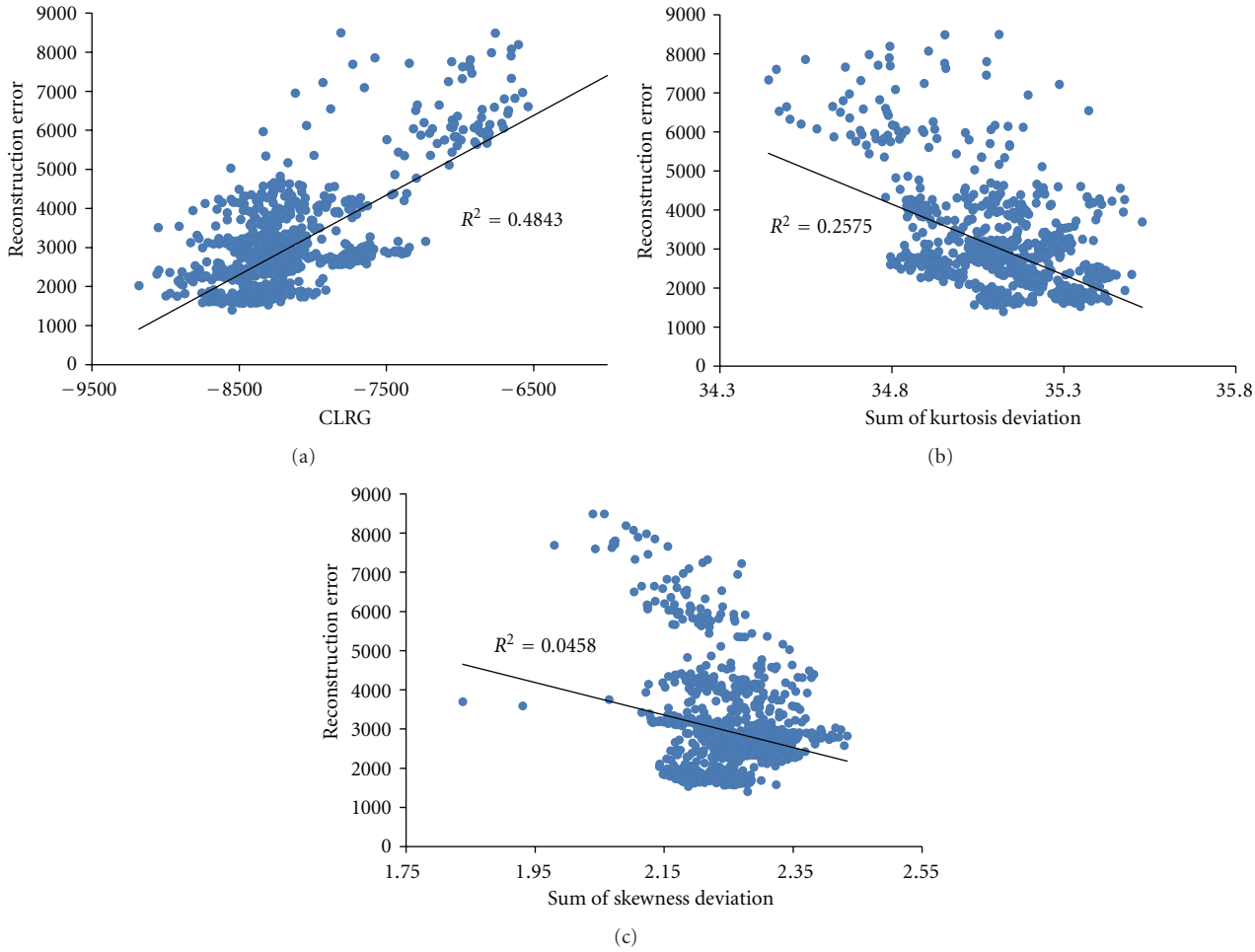


Figure 4: CLRG in comparison to kurtosis and skewness for assessing the quality of ICA-results. For 1,000 results obtained with FastIca on de-mixing speech signals, CLRG best correlates with the reconstruction error of the ICs.

4.1.2. Stable Estimation of the IC Quality. Commonly used approximations of neg-entropy are sensitive to single outliers. Outliers may cause an overestimation of the quality of the IC. CLRG is an outlier-robust measure for the interestingness of a signal. Figure 3(c) displays the influence of one single outlier on the kurtosis (displayed in terms of deviation from Gaussian) and CLRG of a Gaussian noise signal with 500 samples with respect to various outlier strengths (displayed in units of standard deviation). For reference, also the kurtosis of a highly non-Gaussian saw-tooth signal is displayed with a dotted line. Already for moderate outlier strength, the estimation of kurtosis becomes unstable. In case of a strong single outlier, kurtosis severely overestimates the interestingness of the signal. CLRG is not sensitive with respect to single outliers: even for strongest outliers, the noise signal is scored as not interesting with a CLRG of zero. For comparison, the saw-tooth curve allows an effective data compression with a CLRG of −553.

4.1.3. Comparing ICA Results. CLRG is a very general criterion for assessing the quality of ICA results which does not rely on any assumptions specific to certain algorithms. In this experiment, we compare CLRG to kurtosis and skewness on the benchmark dataset acspeec16 form ICALAB (http://www.bsp.brain.riken.go.jp/ICALAB/ICALABSignalProc/benchmarks/). This dataset consists of 16 speech signals which we mixed with a uniform random mixing matrix. Figure 4 displays 1,000 results of FastIca [2] generated with the nonlinearity tanh and different random starting conditions. For each result, we computed the reconstruction error as the sum of squared deviations of the ICs found by FastIca to the original source signals. For each IC, we used the best matching source signal (corrected for sign ambiguity) and summed up the squared deviations. Figure 4(a) shows that CLRG correlates best with the reconstruction error. In particular, ICA results with a low reconstruction error also allow effective data compression. For comparison, we computed the sum of kurtosis deviations and the sum of skewness deviations from Gaussianity. Kurtosis and even more skewness show only a slight correlation with the reconstruction error. As an example, Figure 5(a) shows the first extracted IC from the result best scored by CLRG and the corresponding IC (Figure 5(b)) from the result best scored by kurtosis. For each of the two ICs the scatter plots with

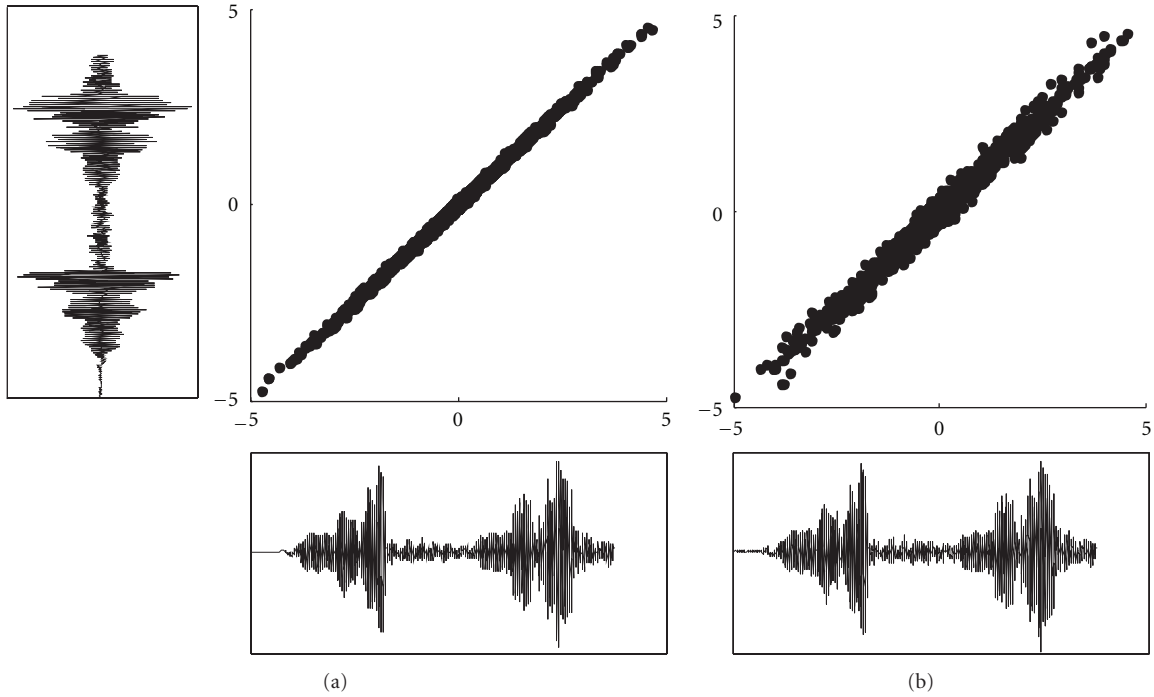


Figure 5: Reconstruction error of first IC from the result best scored by CLRG and matching IC from the result best scored by kurtosis.

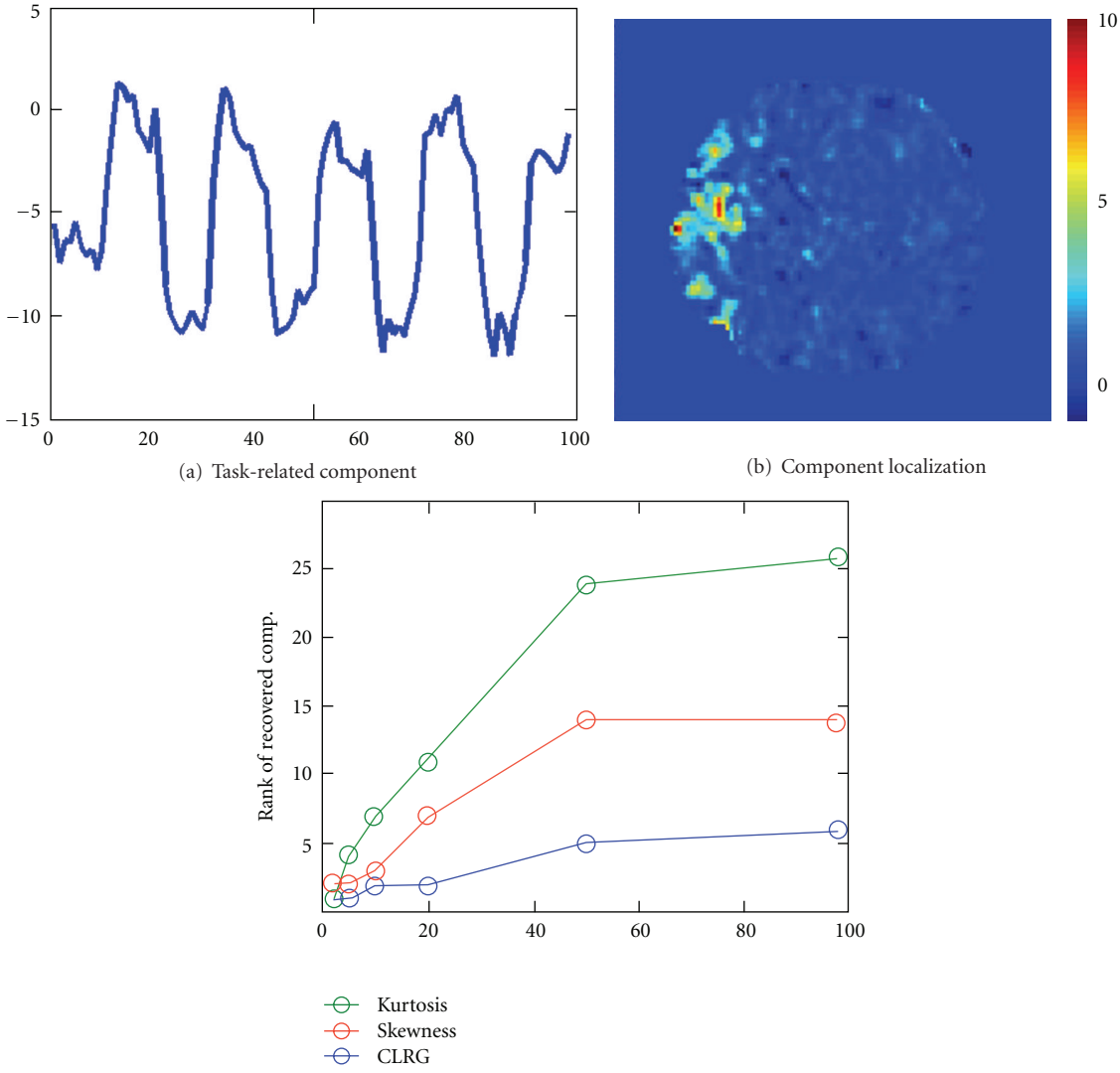
the original signal are displayed. Obviously, the left IC better matches the true signal than the right IC resulting in a lower reconstruction error.

4.1.4. Selecting Relevant Components from fMRI Data. Functional magnetic resonance imaging (fMRI) yields time series of 3-d volume images allowing to study the brain activity, usually while the subject is performing some task. In this experiment, a subject has been visually stimulated in a block-design by alternately displaying a checkerboard stimulus and a central fixation point on a dark background as control condition [21]. fMRI data with 98 images (TR/TE = 3000/60 msec) were acquired with five stimulation and rest periods having each a duration of 30 s. After standard preprocessing, the dimensionality has been reduced with PCA. FastIca has been applied to extract the task-related component. Figure 6(a) displays an example component with strong correlation to the experimental paradigm. This component is localized in the visual cortex which is responsible for processing photic stimuli, see Figure 6(b). We compared CLRG to kurtosis and skewness with respect to their scoring of the task-related component. In particular, we performed PCA reductions with varying dimensionality and identified the component with the strongest correlation to the stimulus protocol. Figure 6(c) shows that CLRG scores the task-related component much better than skewness and kurtosis. Regardless of the dimensionality, the task-related component is always among the top-ranked components by CLRG, in most cases among the top 3 to 5. By kurtosis and skewness, the interest of task-related component often rated close to the average.

4.2. Discussion of CLRG Estimation Techniques. Phi-transformed histograms and equidistant Gaussian mixture models represent different possibilities to estimate the code length relative to Gaussianity (CLRG). As elaborated in Section 3, to estimate the code length in bits, we need a probability density function (PDF) and the two variants differ in the way the PDF is defined. The major benefit of equidistant Gaussian mixture models over Phi-transformed histograms is that the PDF is defined by a continuous function which tends to represent some signals better than phi-transformed histograms. A better representation of the non-Gaussian characteristics of a signal results in more effective data compression expressed by a lower CLRG.

Figure 7 provides a comparison of phi-transformed histograms and equidistant Gaussian mixture models (eGMMs) regarding the CLRG estimated for the 16 signals of the aspeech16 dataset. For most signals, the CLRG estimated by both variants is very similar, for example, signals number 1 to 3, 10, and 16. Eight signals can be most effectively compressed using phi-transformed histograms, most evidently signals number 11 to 13. The other eight signals can be most effectively compressed using eGMM. In average on the aspeech16 dataset, the average CLRG 6,028 bits for phi-transformed histograms, 6,148 bits for eGMM.

We found similar results on other benchmark datasets also available at the ICALAB website: the 19 signals of the eeg19 dataset tend to be better represented by eGMM with an average CLRG of 17,691 (11 signals best represented by eGMM) followed by phi-transformed histograms with an average CLRG of 17,807 (8 signals best represented by phi-transformed).



(a) Task-related component

(b) Component localization

(c) Varing dim. reduction

Figure 6: fMRI experiment: (a) Task-related IC extracted by FastIca from a fMRI experiment where the subject performed a visual task while in the scanner. (b) Color-coded spatial activation pattern of this IC in an axial brain slice. (c) The rank of this IC according to CLRG and the comparison methods for varying dimensionality reduction. This interesting IC is always identified among the top-ranked components by CLRG.

Also, the abio7 data tends to be better represented by eGMM with an average CLRG of 6,480 (5 out of 7 signals best represented by GMM). Phi-transformed histograms perform with an average CLRG of 7,023 (2 best represented signals).

To summarize, we found only minor differences in performance among the two techniques estimating CLRG. Whenever a continuous representation of the PDF is required, the eGMM techniques should be preferred. A continuous representation allows, for example, incremental assessment of streaming signals. In this case, the CLRG can be reestimated periodically when enough novel data points have arrived from the stream.

5. Conclusion

In this paper, we introduced CLRG (code length relative to gaussianity) as an information-theoretic measure to evaluate the quality of single independent components as well as complete ICA results. Our experiments demonstrated that CLRG is an attractive complement to existing measures for non-Gaussianity, for example, kurtosis and skewness for the following reasons: relating the relevance of an IC to its usefulness for data compression, CLRG identifies the most relevant ICs in a dataset without requiring any parameters or thresholds. Moreover, CLRG is less sensitive to outliers

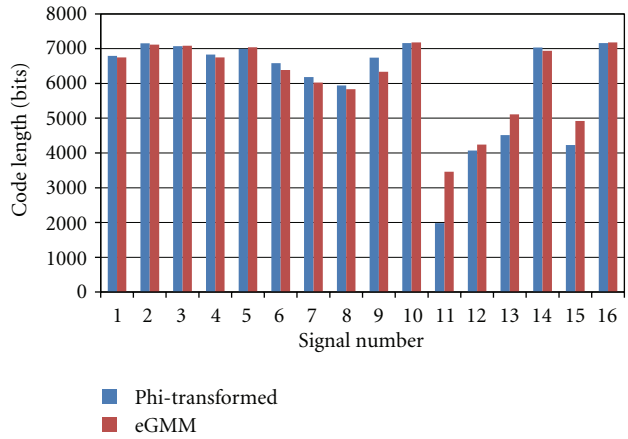


Figure 7: Comparison of CLRG estimation techniques regarding the compression of speech signals.

than comparison measures. On fMRI data, CLRG clearly outperforms the comparison techniques in identifying the relevant task-specific components.

The basic idea that a good model provides efficient data compression is very general. Therefore, not only different ICs and ICA results obtained by different algorithms can be unbiasedly compared. Given a dataset, we can also compare the quality completely different models, for example, obtained by ICA, PCA, and projection pursuit. Moreover, it might lead to the best data compression to apply different models to different subsets of the dimensions as well as different subsets of the data objects. In our ongoing and future work, we will extend CLRG to support various models and will explore algorithms for finding subsets of objects and dimensions which can be effectively compressed together.

Acknowledgment

Claudia Plant is supported by the Alexander von Humboldt Foundation.

References

[1] O. Sporns, "The human connectome: a complex network," *Annals of the New York Academy of Sciences*, vol. 1224, no. 1, pp. 109–125, 2011.

[2] A. Hyvärinen, "Fast and robust fixed-point algorithms for independent component analysis," *IEEE Transactions on Neural Networks*, vol. 10, no. 3, pp. 626–634, 1999.

[3] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis*, John Wiley & Sons, New York, NY, USA, 2001.

[4] P. M. Rasmussen, M. Mørup, L. K. Hansen, and S. M. Arnfred, "Model order estimation for independent component analysis of epoched EEG signals," in *Proceedings of the 1st International Conference on Bio-inspired Systems and Signal Processing (BIOSIGNALS '08)*, pp. 3–10, January 2008.

[5] C. J. James and C. W. Hesse, "Independent component analysis for biomedical signals," *Physiological Measurement*, vol. 26, no. 1, pp. R15–R39, 2005.

[6] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and applications," *Neural Networks*, vol. 13, no. 4-5, pp. 411–430, 2000.

[7] J. Himberg and A. Hyvärinen, "Icasso: software for investigating the reliability of ica estimates by clustering and visualization," in *Proceedings of the IEEE Workshop on Neural Networks for Signal Processing (NNSP '03)*, pp. 259–268, 2003.

[8] F. Meinecke, A. Ziehe, M. Kawanabe, and K. R. Müller, "A resampling approach to estimate the stability of one-dimensional or multidimensional independent components," *IEEE Transactions on Biomedical Engineering*, vol. 49, no. 12, pp. 1514–1525, 2002.

[9] G. Qian, "Computing minimum description length for robust linear regression model selection," in *Proceedings of the Pacific Symposium on Biocomputing*, pp. 314–325, 1999.

[10] S. R. Rao, H. Mobahi, A. Y. Yang, S. S. Sastry, and Y. Ma, "Natural image segmentation with adaptive texture and boundary encoding," in *Proceedings of the Asian Conference on Computer Vision (ACCV '09)*, vol. 5994 of *Lecture Notes in Computer Science*, pp. 135–146, 2009.

[11] T. Wekel and O. Hellwich, "Selection of an optimal polyhedral surface model using the minimum description length principle," in *Proceedings of the 32nd Symposium of the German Association for Pattern Recognition (DAGM '10)*, vol. 6376 of *Lecture Notes in Computer Science*, pp. 553–562, 2010.

[12] J. Sun, C. Faloutsos, S. Papadimitriou, and P. S. Yu, "GraphScope: parameter-free mining of large time-evolving graphs," in *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '07)*, pp. 687–696, August 2007.

[13] D. Pelleg and A. W. Moore, "X-means: extending k-means with efficient estimation of the number of clusters," in *Proceedings of the 17th International Conference on Machine Learning (ICML '00)*, pp. 727–734, 2000.

[14] C. Böhm, C. Faloutsos, J. Y. Pan, and C. Plant, "Robust information-theoretic clustering," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '06)*, pp. 65–75, August 2006.

[15] C. Böhm, C. Faloutsos, and C. Plant, "Outlier-robust clustering using independent components," in *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD '08)*, pp. 185–198, June 2008.

[16] C. Böhm, K. Haegler, N. S. Müller, and C. Plant, "CoCo: coding cost for parameter-free outlier detection," in *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '09)*, pp. 149–157, July 2009.

[17] P. Gruber, F. Theis, A. Tome, and E. Lang, "Automatic denoising using local independent component analysis," in *Proceedings of the 4th International ICSC Symposium on Engineering of Intelligent Systems (EIS '04)*, 2004.

[18] A. Barron, J. Rissanen, and B. Yu, "The Minimum Description Length Principle in Coding and Modeling," *IEEE Transactions on Information Theory*, vol. 44, no. 6, pp. 2743–2760, 1998.

[19] T. C. M. Lee, "Regression spline smoothing using the minimum description length principle," *Statistics and Probability Letters*, vol. 48, no. 1, pp. 71–82, 2000.

[20] T. W. Lee, M. S. Lewicki, and T. J. Sejnowski, "ICA mixture models for unsupervised classification of non-Gaussian classes and automatic context switching in blind signal separation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1078–1089, 2000.

[21] A. Wismüller, O. Lange, D. R. Dersch et al., "Cluster analysis of biomedical image time-series," *International Journal of Computer Vision*, vol. 46, no. 2, pp. 103–128, 2002.