Research Article

Unsupervised Topographic Learning for Spatiotemporal Data Mining

Guénaél Cabanes and Younès Bennani

LIPN-CNRS, UMR 7030, Université de Paris 13, 99, avenue J-B. Clément, 93430 Villetaneuse, France

Correspondence should be addressed to Guénaél Cabanes, guenael.cabanes@lipn.univ-paris13.fr

Received 14 June 2010; Revised 5 September 2010; Accepted 7 September 2010

Academic Editor: Abbes Amira

Copyright © 2010 G. Cabanes and Y. Bennani. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In recent years, the size and complexity of datasets have shown an exponential growth. In many application areas, huge amounts of data are generated, explicitly or implicitly containing spatial or spatiotemporal information. However, the ability to analyze these data remains inadequate, and the need for adapted data mining tools becomes a major challenge. In this paper, we propose a new unsupervised algorithm, suitable for the analysis of noisy spatiotemporal Radio Frequency Identification (RFID) data. Two real applications show that this algorithm is an efficient data-mining tool for behavioral studies based on RFID technology. It allows discovering and comparing stable patterns in an RFID signal and is suitable for continuous learning.

1. Introduction

In recent years, the size of datasets has shown an exponential growth. A study exhibits that the amount of data doubles every three years [1]. In application areas such as robotics, computer vision, mobile computing, and traffic analysis, huge amounts of data are generated and stored in databases, explicitly or implicitly containing spatial or spatiotemporal information. For instance, the proliferation of location-aware devices gives rise to vast amounts of frequently updated telecommunication and traffic data, and satellites generate terabytes of image data daily. These huge collections of spatiotemporal data often hide possibly interesting information and valuable knowledge. It is obvious that a manual analysis of these data is impossible, and data mining might provide useful tools and technology in this setting. Spatiotemporal data mining is an emerging research area that is dedicated to the development of novel algorithms and computational techniques for the successful analysis of large spatiotemporal databases and the disclosure of interesting knowledge in spatiotemporal data. However, the ability to analyze these data remains inadequate and the need for adapted data mining tools becomes a major challenge.

As the study of data streams and large databases is a difficult problem because of the computing costs and the big

storage volumes involved, two issues appear to play a key role in such an analysis: (i) a good condensed description of the data properties [2, 3] and (ii) a measure capable of detecting changes in the data structure [4, 5]. In this paper, we propose a new algorithm which combines two methods [6, 7] to perform these two tasks. The solution we propose consists of an algorithm, which constructs an abstract representation of the datasets, performs automatically a clustering of the data based on their representations, and is able to evaluate the dissimilarity between two distinct datasets. The abstract representation is based on the learning of a variant of Self-Organizing Map (SOM) [8], which is enriched with information extracted from the data. Then, the underlying data density function is estimated from the abstract representation. The dissimilarity is a measure of the divergence between two estimated densities. A great advantage of this method is that each enriched SOM is at the same time a very informative and a highly condensed description of the data distribution that can be stored easily for a future use. Moreover, as the algorithm is effective both in terms of computational complexity and in terms of memory requirements, it can be used for comparing large datasets or for detecting changes in data streams.

In this work, we focus on the analysis of RFID data. RFID is an advanced tracking technology. The RFID tags, which

consist of a microchip and an antenna, must be used with a reader that can detect simultaneously a lot of tags in a single scan. A computer is used to store the data about the position of each tag for each scan in a database. This allows different analyses. RFID, thanks to miniaturization, offers the advantage of automation and overcomes the constraints imposed by video analyzes. The evolution of these data over time and their spatial position require the exploration of multiple data sets described in high dimension spaces.

The proposed algorithm presents some interesting properties to deal with RFID data.

- (i) It allows a compact representation of each trajectory in a linear computational cost. This is important regarding that the total amount of recorded data can be high. Indeed, each tag's position is recorded very frequently (sometime less than each second) during several hours and many tags are followed simultaneously.
- (ii) It is able to deal with noisy data and to find automatically a suitable number of clusters, without constraints about the clusters' shape. Moreover, it presents good clustering performances comparing to traditional algorithms. This is appreciable as RFID trajectories are very noisy in our application and the extraction of general patterns become a difficult task. The algorithm is thus used to perform a clustering of each trajectory from its representation.
- (iii) It is suitable for trajectories comparisons. As new trajectory record can be added to the database at any time, it is important to compare patterns of new trajectories with older ones. The algorithm can evaluate the similarity of two trajectories' representations using information about underlying distribution of the tags' movements. This approach is very resistant to noise and is more reliable than distance-based methods.

Thus, the algorithm combines all needed properties—good abstraction performances, low memory, and computational cost, suitable for noisy experimental data analysis—to be a good candidate for RFID data mining.

The remainder of this paper is organized as follows. Section 2 presents the algorithm. Properties analysis and results are described in Section 3. Section 4 introduces the adaptation of the algorithm to deal with RFID data then describes two real application (mining customers trajectories in a supermarket and analyzing migration behavior of an ant's colony). A conclusion is given in Section 5.

2. Proposed Algorithm

2.1. General Algorithm Schema. The basic assumption in this work is that it is possible to define prototypes in the data space and to calculate a distance measure between data points and prototypes. First, each dataset is modeled using an enriched SOM model, constructing an abstract representation which is supposed to capture the essential properties of the data. Then, a clustering of the data is

computed, in order to catch the global structure of these data. Finally, the density function of each dataset is estimated from the abstract representation and different datasets can be compared using a dissimilarity measure based upon these density functions.

The idea is to combine the dimension reduction and the fast learning capabilities of SOM to construct a new vector space then apply other analysis in this space. These are called two-level methods. The two-level methods are known to reduce greatly the computational time, the effects of noise, and the “curse of dimensionality” [6]. Furthermore, it allows some visual interpretation of the result using the two-dimensional map generated by the SOM.

The algorithm proceeds in three steps.

- (1) The first step is the learning of the enriched SOM. During the learning, each SOM prototype is extended with novel information extracted from the data. These information will be used in the following step to find clusters in the data and to infer the density function. More specifically, the attributes added to each prototype are the following.
 - (a) *Density modes.* It is a measure of the data density surrounding the prototype (local density). The local density is a measure of the amount of data present in an area of the input space. We use a Gaussian kernel estimator [9] for this task.
 - (b) *Local variability.* It is a measure of the data variability that is represented by the prototype. It can be defined as the average distance between the prototypes and the represented data.
 - (c) *The neighborhood.* This is a prototype's neighborhood measure. The neighborhood value of two prototypes is the number of data that are well represented by each one.
- (2) The second step is the clustering of the data using density and connectivity information so as to detect low-density boundary between clusters.
- (3) The third step is the construction, from each cluster (i.e., a set of enriched prototypes in a SOM), of a density function which will be used to estimate the density in the input space. This function is constructed by induction from the information associated to the prototypes of the SOM and is represented as a mixture model of spherical normal functions.
- (4) The last step accomplishes the comparison of two different datasets (e.g., clusters from different databases) using a dissimilarity measure able to compare the two density functions constructed in the previous steps.

2.2. Data Structure Modeling with Enriched SOM. Kohonen SOM can be defined as a competitive unsupervised learning neural network [8]. When an observation is recognized, the activation of an output cell—competition layer—inhibits the activation of other neurons and reinforce itself. It is said that

it follows the so called “Winner Takes All” rule. Actually, neurons are specialized in the recognition of one kind of observation. An SOM consists of a two-dimensional map of neurons which are connected to n -inputs according to n weights connections and to their neighbors with topological links. The training set is used to organize these maps under topological constraints of the input space. Thus, a mapping between the input space and the network space is constructed; two close observations in the input space would activate two close units of the SOM. An optimal spatial organization is determined by the SOM from the input data, and when the dimension of the input space is less than three, both position of weights vectors (also called prototypes) and direct neighborhood relations between cells can be represented visually. Thus, a visual inspection of the map provides qualitative information about the map and the choice of its architecture. The winner neuron updates its prototype vector, making it more sensitive for later presentation of that type of input. This allows different cells to be trained for different types of data. To achieve a topological mapping, the neighbors of the winner neuron can adjust their prototype vector towards the input vector as well, but at a lesser degree, depending on how far away they are from the winner. Usually, a radial symmetric Gaussian neighborhood function K_{ij} is used for this purpose.

In our algorithm, the SOM’s prototypes will be “enriched” by adding new numerical values extracted from the dataset.

The enrichment algorithm proceeds in three phases.

Input:

- (i) the data $X = \{x_k\}_{k=1}^N$.

Output:

- (i) the density D_i and the local variability s_i associated to each prototype w_i ,
- (ii) the neighborhood values $v_{i,j}$ associated with each pair of prototype w_i and w_j .

Algorithm:

(1) Initialization:

- (i) initialize the SOM parameters,
- (ii) For all i, j initialize to zero the local densities (D_i), the neighborhood values ($v_{i,j}$), the local variability (s_i) and the number of data represented by $w_i(N_i)$.

(2) Choose randomly a data $x_k \in X$:

- (i) compute $d(w, x_k)$, the distance between the data x_k and each prototype w_i ,

- (ii) find the two closest prototypes (BMUs: Best Match Units) w_{u^*} and $w_{u^{**}}$

$$\begin{aligned} u^* &= \arg \min_i (d(w_i, x_k)), \\ u^{**} &= \arg \min_{i \neq u^*} (d(w_i, x_k)). \end{aligned} \quad (1)$$

(3) Update structural values:

- (i) number of data: $N_{u^*} = N_{u^*} + 1$,
- (ii) variability: $s_{u^*} = s_{u^*} + d(w_{u^*}, x_k)$,
- (iii) density: For all i , $D_i = D_i + \frac{1}{\sqrt{2\pi}h} e^{-d(w_i, x_k)^2/2h^2}$,
- (iv) neighborhood: $v_{u^*, u^{**}} = v_{u^*, u^{**}} + 1$.

(4) Update the SOM prototypes w_i as defined in [8].

(5) Repeat T times step (2) to (4).

(6) Final structural values: For all i , $s_i = s_i/N_i$ and $D_i = D_i/N$.

In this study, we used the default parameters of the SOM Toolbox [10] for the learning of the SOM, and we use $T = \max(N, 50 \times M)$ as in [10]. The number M of prototypes must neither be too small (the SOM does not fit the data well) nor too large (time consuming). If N is not too big, to choose M close to $\sqrt{N}$ seems to be a good trade off [10]. The last parameter to choose is the bandwidth h . The choice of h is important for good results, but its optimal value is difficult to calculate and time consuming (see [11]). A heuristic that seems relevant and gives good results consists of defining h as the average distance between a prototype and its closest neighbor [6].

At the end of this process, each prototype is associated with a density and a variability value, and each pair of prototypes is associated with a neighborhood value. The substantial information about the distribution of the data is captured by these values. Then, it is no longer necessary to keep data in memory.

2.3. Data Clustering. Various prototypes-based approaches have been proposed to solve the clustering problem [12, 13]. However, the obtained clustering is never optimal, since part of the information contained in the data is not represented by the prototypes. In [6], a new method of prototypes’ clustering is proposed, the Density-based Simultaneous 2-Level—SOM algorithm (DS2L-SOM) that uses density and neighborhood information to optimize the clustering. The main idea is that the core part of a cluster can be defined as a region with high density. Then, in most cases, the cluster borders are defined either by low-density region or “empty” region between clusters (i.e., large intercluster distances) [12].

Here, DS2L-SOM uses information learned by the enriched SOM. Figure 1 shows an example of the different stages of the clustering algorithm: prototypes have been learned with the enriched SOM algorithm; they are represented by hexagons with preservation of the neighborhood.

FIGURE 1: Example of a sequence of the different stages of the clustering algorithm.

At the end of the enrichment process (Section 2.2), each set of prototypes linked together by a neighborhood value $\nu > \text{threshold}$ define well-separated clusters. This is useful to detect borders defined by large intercluster distances (Figure 1(b)). The estimation of the local density (D) is used to detect cluster borders defined by low density. Each cluster is defined by a local maximum of density (density mode, Figure 1(c)). Thus, a “Watersheds” method [14] is applied on prototypes’ density for each well-separated cluster to find low-density area inside these clusters, in order to characterize density defined subclusters (Figure 1(d)). For each pair of adjacent subgroups, we use a density-dependent index [15] to check if a low-density area is a reliable indicator of the data structure, or whether it should be regarded as a random fluctuation in the density (Figure 1(e)). This process is very fast because generally the number of prototypes is small. The combined use of these two types of group definition can achieve very good results despite the low number of prototypes in the map and is able to detect automatically the number of cluster.

2.4. Comparisons of Data Distributions. Each dataset is modeled using an enriched Self-Organizing Map (SOM) model, constructing an abstract representation which is supposed to capture the essential data structure. Each of the datasets is partitioned using the DS2L-SOM algorithm. In order to be able to compare different clusters from different databases, the algorithm first estimate the underlying density function of each clusters, then use a dissimilarity measure based upon the density functions for the comparison.

The first objective of this step is to estimate the density function which associates a density value to each point of the input space. An estimation of some values of this function have been calculated (i.e., D_i) at the position of the prototypes representing a cluster. An approximation of the function must now be inferred from these values.

The hypothesis here is that this function may be properly approximated in the form of a mixture of Gaussian kernels. Each kernel K is a Gaussian function centered on a prototype. The density function can; therefore; be written as:

$$f(x) = \sum_{i=1}^M \alpha_i K_i(x), \quad (2)$$

with

$$K_i(x) = \frac{1}{N\sqrt{2\pi}h_i} e^{-d(w_i, x)^2/2h_i^2}. \quad (3)$$

The most popular method to fit mixture models (i.e., to find h_i and α_i) is the expectation-maximization (EM) algorithm [16]. However, this algorithm needs to work in the data input space. As here we work on enriched SOM instead of dataset, we cannot use EM algorithm.

Thus, we propose the heuristic to choose h_i

$$h_i = \frac{\sum_j (v_{i,j}/(N_i + N_j)) (s_i N_i + d_{i,j} N_j)}{\sum_j v_{i,j}}, \quad (4)$$

$d_{i,j}$ is the distance between w_i and w_j . The idea is that h_i is the standard deviation of data represented by K_i . These data are

FIGURE 2: “Engytime” dataset and the related estimated density function.

FIGURE 3: “Rings” dataset and the related estimated density function.

also represented by w_i and their neighbors. Then, h_i depends on the variability s_i computed for w_i and the distance $d_{i,j}$ between w_i and its neighbors, weighted by the number of data represented by each prototype and the connectivity value between w_i and his neighborhood.

Now, since the density D for each prototype w is known ($f(w_i) = D_i$), a gradient-descent method can be used to determine the weights α_i . The α_i are initialized with the values of D_i , then these values are reduced gradually to better fit $D = \sum_{i=1}^M \alpha_i K_i(w)$. To do this, the following criterion is optimized:

$$\alpha = \arg \min_{\alpha} \frac{1}{M} \sum_{i=1}^M \left[\sum_{j=1}^M (\alpha_j K_j(w_i)) - D_i \right]^2. \quad (5)$$

Thus, we have a density function that is a model of the dataset represented by the enriched SOM. Some examples of estimated density are shown on Figures 2 and 3.

A measure of dissimilarity between two clusters A and B can now be defined, represented by two models: $M_A = [\{w_i^A\}_{i=1}^{N^A}, f^A]$ and $M_B = [\{w_j^B\}_{j=1}^{N^B}, f^B]$, with N^A and N^B the number of prototypes representing A and B and f^A and f^B the density function of A and B .

The dissimilarity between A and B is given by

$$CBd(A, B) = \frac{\sum_{i=1}^{N^A} f^A(w_i^A) \log(f^A(w_i^A)/f^B(w_i^A))}{N^A} + \frac{\sum_{j=1}^{N^B} f^B(w_j^B) \log(f^B(w_j^B)/f^A(w_j^B))}{N^B}. \quad (6)$$

The idea is to compare the density functions f^A and f^B for each prototype w of A and B . If the distributions are identical, these two values must be very close. This measure is an adaptation of the weighted Monte Carlo approximation of the symmetrical Kullback-Leibler measure (see [17]), using the prototypes of an SOM as a sample of the database.

3. Results and Analysis

3.1. Validity of the Clustering. The effectiveness of the proposed two-level clustering method have been demonstrated in [6] by testing the performances on 10 databases presenting various clustering difficulties. DS2L-SOM was compared to S2L-SOM (similar to DS2L-SOM but using only neighborhood information) and to some traditional two-level methods, in term of clustering quality (Jaccard and Rand indexes [18]) and stability (subsampling-based method [19]). The selected traditional algorithms for comparison are K -means and Ascendant Hierarchical Clustering (AHC) applied (i) to the data and (ii) to the prototypes of the trained SOM. The Davies and Bouldin [20] index was used to determine the best cutting of the dendrogram (AHC) or the optimal number K of centroids for K -means. S2L-SOM and DS2L-SOM determine the number of clusters automatically and do not need to use this index. In AHC, the proximity of two clusters was defined as the minimum of the distance between any two objects in the two different clusters. The results for the external indexes show that for all the databases DS2L-SOM is able to find without any error the expected data segmentation and the right number of clusters. This is not the case of the other algorithms, when the groups have an arbitrary form, when there is no structure (i.e., only one cluster) in the data or when clusters are in contact. Considering the stability, the DS2L-SOM algorithm shows excellent results, whatever the dimension of data or the clusters' shape. It is worth noticing that in some cases the clustering obtained by the traditional methods can be extremely unstable.

To summarize, DS2L-SOM presents some interesting qualities in comparison to other clustering algorithms

- (i) the number of cluster is automatically detected by the algorithm,
- (ii) no linearly separable clusters and nonhyperspherical clusters can be detected, and
- (iii) the algorithm can deal with noise (i.e., touching clusters) by using density estimation.

3.2. Validity of the Comparisons. In order to demonstrate the performance of the proposed dissimilarity measure, nine artificial datasets generators and two real datasets were used in [7].

The main idea to test the quality of the comparison measure is that a low dissimilarity value is only consistent with a similar distribution and does, of course, give an indication of the similarity between the two sample distributions. On the other hand, a very high dissimilarity does show, to the given level of significance, that the distributions are different.

TABLE 1: Value of the Dunn index obtained from various dissimilarity measures to compare various data distributions.

Distributions	Average	Minimum	Ward	Proposed
Ring 1-3 + Spiral 1-2	0.4	0.9	0.5	1.6
Noise 1-4	1.1	1.4	22.0	115.3
Shuffle 1-2	1.1	16.5	6.3	27.6

Then, if the measure of dissimilarity is efficient, it should be possible to compare different datasets (with the same attributes) to detect the presence of similar distributions, that is the dissimilarity of datasets generated from the same distribution law must be much smaller than the dissimilarity of datasets generated from very different distribution. This is measured using a generalized index of Dunn [21]: the higher is this index, the better is the comparison measure (see [7]).

The results was compared with some distance-based measures usually used to compare two sets of data (here, we compare two sets of prototypes from the SOMs). These measures are the average distance (Ad: the average distance between all pair of prototypes in the two SOMs), the minimum distance (Md: the smallest Euclidean distance between prototypes in the two SOMs), and the Ward distance (Wd: The distance between the two centroids, with some weight depending on the number of prototypes in the two SOMs) [22].

As shown in Table 1, the new dissimilarity measure using density is much more effective than measures that only use the distances between prototypes. Indeed, the Dunn index for density-based measure is much higher than distance-based ones for the three kinds of datasets tested: nonconvex data, noisy data, and real data. This means that the new dissimilarity measure is much more effective for the detection of similar datasets.

3.2.1. Algorithm Complexity. The complexity of the algorithm is scaled as $O(T \times M)$, with T the number of steps and M the number of prototypes in the SOM. It is recommended to set at least $T > 10 \times M$ for a good convergence of the SOM [8]. In this study, we use $T = \max(N, 50 \times M)$ as in [10] (N is the number of data). This means that if $N > 50 \times M$ (large database), the complexity of the algorithm is $O(N \times M)$, that is, is linear in N for a fixed size of the SOM. Then, the whole process is very fast and is suited for the treatment of large databases. Also, very large databases can be handled by fixing $T < N$ (this is similar as working on a random subsample of the database).

This is much faster than traditional density estimator algorithms as the Kernel estimator [9] (that also needs to keep all data in memory) or the Gaussian mixture model [23] estimated with the EM algorithm (as the convergence speed can become extraordinarily slow [24]).

4. Case Study

In this section, an adaptation of the algorithm is introduced to deal with RFID data, and two applications are described. The first application aims at mining customers trajectories

in a supermarket during their shopping. The second is an analyze of migration behavior in an ant's colony.

4.1. Proposed Method. The proposed method is to analyze RFID signal proceeds in three steps.

- (1) Data postprocessing: The aim is to analyze the variation of the spatial behavior over time. To do that, a current spatiotemporal behavior must be defined for each instant during the following. This behavior must be inferred from the complex and noisy RFID signal. To do this, we compute for each x seconds of the tag's trajectory, a vector representing some relevant spatiotemporal information during a y minutes time window centered on the current time ($x \ll y$).

Obviously, this definition implies some correlations between the description of two time windows if they are separated by less than y minutes, as they overlap. This will allow us to detect when a customer moves from one area to another, by detecting sudden change in the description of current location.

- (2) Detection of individual homogeneous patterns: In order to regroup similar behaviors and to detect changes over time, an enriched SOM is applied on time windows from each individual sequences. DS2L-SOM is then applied on the enriched SOM as in Section 2.3, then it uses the density information to detect sudden change in the location windows, so as to regroup similar windows. The goal is to detect time-stable patterns inside the RFID signal. The signal is thus segmented in a succession of homogeneous patterns.

- (3) Detection of similar patterns: The method used in step (3) allows to analyze efficiently the spatiotemporal behavior of each tag. Now, a method is needed to compare all these individual sequences so as to perform an analysis at the collective level.

The idea is to define a similarity measure between two set of prototypes (from the enriched SOM) that represent two individual subsequences (i.e., clusters). For this job, the related density function is computed for each stable pattern as in Section 2.3. Then, a similarity matrix between each pattern of each RFID signal is computed. Finally, DS2L-SOM is applied on this matrix in order to find similar pattern in different RFID signal.

4.2. Mining Customers Trajectories. In this application, we aim at studying the individuals' spatiotemporal activity during their shopping in a supermarket. Until now, little research has been undertaken in this way. Usual questions are: how do customers really travel through the store, do they go through every area or do they skip from one area to another in a more direct manner, do they follow a single, dominant pattern, or are they rather heterogeneous?

FIGURE 4: The RFID experimental device. Yellow box represents readers positions.

Tag ID	Time of the recording
0c00011629 (FF)	172.30.149.10 ; 19/02/2009 ; 10:02:52
IP address of the RFID reader	

FIGURE 5: Example of a recorded scan in the data file.

4.2.1. Context. The purpose of this work is to explore data recorded via an RFID device to model and analyze the purchasing behavior of customers. In particular, we would know the time spend in each area of the store so as to detect hot spots and cold spots. We also aim at analyzing the customers trajectory patterns.

The movement of customers during their shopping was monitored using RFID device. To do this, some plastic basket are at the disposal of the customers. Each basket have an RFID tag glued on is back.

The supermarket used for this experiment is a store specializing in the sale of decorative objects (6000 m²), with an average of 2,500 visits each month. An RFID device with four readers was installed in the different area of the store to analyze the movement of customers. The first tests carried out on site confirm that the positioning of readers is very delicate and only allow testing to find the optimum location. Figure 4 shows the final position of the readers inside the shop, noting that the readers numbers (1, 4, 9 and 10) represent the last number in their IP address. The information recorded by readers are handled by an RFID electronics and then sent to a computer which creates and stores the data files.

The data files are in text format. They indicate, for each scan (about one scan per second), the ID number of the tag detected, the IP address of the reader that have detected the tag, and the date and time of the detection (Figure 5). If, during a scan, none is detected, nothing appears in the data file.

As a customer moves inside the store, they are detected successively by different readers. However, depending of the crossed area, one tag can be detected by more than one reader, approximatively at the same time. This adds much more information about the actual location of the customer, but this also make the moving sequence much

more hard to understand. Furthermore, data recorded a very noisy, because of perturbations of the RFID signal by all the metallic structures and by human bodies (see Figure 6 for an example).

4.2.2. Analyses. Here, we want to analyze the variation of the customers spatial behavior over time. To do that, a current location must be defined. The current location represent the area where the customer is in a given time. For each 10 seconds of the customer trajectory, a vector is computed representing how many times and how long each RFID reader have detected the customer's tag during a 3-minute time window centered on the current time. This will allow to detect when a customer moves from one area to another, by detecting sudden change in the description of current location.

In order to regroup similar current location and to detect changes over time, the algorithm DS2L-SOM is applied on time windows from each individual sequences.

By using the similarity measure to compare all the subsequences of all the recorded customers in the first day of recording, we found six clusters that represent six well-defined homogeneous locations. The similarity measure can now be used to label each new customer's sequence recording after the first day. This is fast enough to be made in real time during the customer's shopping.

4.2.3. Main Results. The analysis method allowed to find six well-defined homogeneous locations (named sectors). This means that we were able to define more well-localized area than the number of reader (50% more), this is a good information extraction. The sectors can be described as follows (see also Figure 7):

- (S1) detected by reader 9 only, it correspond to the entrance of the store. Baskets waiting for new customers are detected in this sector.
- (S2) detected by reader 1 only. In this sector, customers can find flowers and vases.
- (S3) mainly detected by reader 4 and 10. In this sector, customers can find wrought iron objects.
- (S4) mainly detected by reader 1, sometime by reader 9 (wood furnitures).
- (S5) mainly detected by reader 9 and 10, sometime by reader 4 or 1 (dishes and small objects).
- (S6) mainly detected by reader 1 and 4, sometime by reader 9 (Mirrors and linens).

Figure 7 shows an estimation of the location of the different sectors. The frequency of transition between one sector to another is also computed. This gives an idea of how customers move inside the store. We can see for example that customers always take the same way in the first part of the store ((S1) and (S2)), but act more freely at the bottom of the store ((S3) to (S6)). (S5) seems to be a key area as there is many transitions from and to it, and it is connected to four over the five other sectors.

FIGURE 6: Example of a moving sequence. On the abscissa is the time (minutes). On the ordinate are the readers detecting the tag.

FIGURE 7: Estimation of sectors location. The thickness of arrows is proportional to transitions frequency.

Finally, the mean time spent in each sector is computed so as to find hot and cold spot. This shows us that (S5) is a very hot spot (48% of the time) unlike (S2) (6%) and (S4) (2%) which are very cold spot. Note that we do not use (S1) for this analysis, as this sector include waiting baskets.

4.3. Monitoring of an Ant Colony Migration. Animal societies are dynamic systems characterized by numerous interactions between individual members. Such dynamic structures stem from the synergy of these interactions, the individual capacities in information processing, and the diversity of individual responses.

Ants, often caricatured and little known, have nevertheless a huge ecological impact, and they are considered as major energy catalysts. Their complex underground nests contributes to soil ventilation, and because of their predatory and detritivore diets, they contribute to ecosystem equilibrium. Ant colonies face rapid changes of environmental conditions and constraints through an important individual flexibility. The following study aims at studying the mechanisms leading to a colony migrations (change of nest). Migration is a widespread phenomenon in many species, but it remains a risky event because during the movement the queen and brood will be particularly vulnerable. The strategies used in nest choice and movement organization are therefore crucial for group survival.

4.3.1. Context. An RFID device has been developed for this study. Based on marketed products, it requires little development. It consists of a network of RFID readers in a constrained space with compulsory passageways in an

FIGURE 8: Ant with RFID tag.

FIGURE 9: The RFID experimental device.

artificial nest. These readers are connected to a detector which sends the information to a computer.

For this study, we chose a big-sized tropical ant *Pachycondyla tarsata*, which establishes subterranean nests distributed in several interconnected chambers distributed over 10 m. Colonies of these species are typically composed of ten to a few thousand ants. RFID tag consists of a chip attached to an antenna weighting under 40 mg (i.e., 25% of an ant weight), glued on the animal thorax (Figure 8). Preliminary tests showed that the tags do not disturb the ants behavior and the colony dynamic significantly.

The movement between nests of a colony of 55 *Pachycondyla tarsata* workers was monitored in the RFID device (about four hours recording). Each worker had a tag attached to its thorax.

The experimental device for this experiment consists of two artificial nests (N1 and N2) of three rooms each (Room 1, 2, and 3) and a foraging area, linearly connected by six tunnels (Figure 9). At the beginning of the experiment, the brood is located in Room 3 of the first nest, the farthest from the foraging area. Each tunnel is equipped with two RFID readers (number 1 to 12 from Room 3 in Nest 2 to Room 3 in Nest 1) that detect the passage and the direction of tagged individuals between rooms. The information recorded by readers are handled by two RFID electronics and then sent to a computer which creates and stores the data files.

At time = 0, we switch on a strong neon light over the first nest, and we open the entrance of the second nest, then we record the colony movement until the entire brood is moved into the second nest (~4 hours).

The data files are in text format. They indicate, for each antenna scan (about three scans per second), the scan number, the date, time, and, for each individual (i.e., for each tag), which antenna is activated (Figure 10). If, during

FIGURE 10: Example of a recorded scan in the data file.

a scan, none is detected, nothing appears in the data file. If an ant moves from one room to another, it is detected by two successive antennas, and this allows us to infer the exact position of each ant at any moment.

4.3.2. Analyses. We used this information to produce the individual moving sequence of each ant. This sequence is a function that gives the ant's location at any time during the move.

However, what we would like to analyze is the variation of the ant's current spatial behavior over time. To do that, a current spatial behavior must be defined. Here, the current location cannot just be chosen, because this way, we would lose all dynamic information such as "the ant is moving quickly" or "the ant makes round trips between two rooms". Therefore, the current behavior is defined as the time spent in each location (static information) and the number of exits from each location (dynamic information) during a 10-minute time window centered on the current time. As there are 19 locations in the RFID device (7 Rooms and 12 readers), each temporal windows is coded in a vectorial form of 38 normalized features (one static and one dynamic feature for each location).

In order to regroup similar current behaviors and to detect changes in current behaviors over time, the DS2L-SOM algorithm is applied on time windows vectors modeled by an enriched SOM from each individual sequences. Then, the similarity measure is used to compute a similarity matrix with all the subsequences of all the ants. DS2L-SOM is used to find clusters of homogeneous subsequences. This allows to compare the behaviors of different ants. These clusters are then used to rename all the subsequences, so as to give the same name to subsequences that belong to the same cluster.

The RFID apparatus only provides a partial observation of the individuals. No information is provided concerning what happens inside a room, but only the duration of the permanence of an ant inside it can be known. Moreover, sensors are not reliable having a missing detection rate ranging from 5% to 15%. Thus, a HMM variant called S-HMM [25] is used to reconstruct for each ant the most probable evolution of the different activities during the migration.

4.3.3. Main Results. At the end of learning procedure, eight groups of behaviors A_i ($1 \leq i \leq 8$) emerged. A last

FIGURE 11: Examples of detected transportation patterns.

TABLE 2: Corresponding between visual observation and automatic analysis (number of ants).

	Transportation	Others
Activity A_5	8	2
Others	0	42

group (A_0) has been defined, which corresponds to activity segments where a clear pattern is not detectable. The activity of each ant have been labeled according to those behaviors.

In order to check the validity of the obtained results, we compared them with some visual observations from a video record. A movie camera was placed over the foraging area and every ant moving across this area was filmed. This allowed us to detect only one apparent behavior: the transportation of larva and cocoon. Each ant can be identified visually thanks to some color painted on their tag. So, we know which ant does transport and at what time this behavior occurred. We compared this with the results of the automatic analysis, and we found that all the transportations subsequences were grouped into only one activity (Activity A_5 , see Table 2 and Figure 11). Moreover, only two ants have a A_5 subsequence without having been seen transporting (i.e., less than a 5% error). This result shows that there is some reliability in the clustering found with our automatic method.

The analysis of the spatiotemporal structure of each activity have been used by ethologist for hypothesize a plausible explanation for each behavior.

- (A_0) *Unstructured patterns*. The ants are not participating into any activity and their path through the rooms is not structured.
- (A_1) *Quick exploration of the new nest*. The ants have just discovered the new nest and start a fast and active exploration of the new site.
- (A_2) *Panic movements*. This behavior is mostly expressed by nurses: due to the new environmental conditions—strong light and increase of the temperature—the old nest is no more suitable for a good brood care.
- (A_3) *Panic movements in the old nest*. It is very similar to (A_2), but the ants are more feared (this behavior seems to be expressed mainly by the youngest ants).
- (A_4) *General patrolling*. This is a general exploration of the environment.
- (A_5) *Transportation*. The ants is transporting something, that is, the queen, a cocoon, a larva, or an egg.
- (A_6) *Preparation of the new nest*. The new nest is now known to be safe, but some works are needed to prepare the nest for an optimal brood care.
- (A_7) *Patrolling in the old nest*. This behavior is a defensive patrolling inside the old nest in react to the disturbance.
- (A_8) *Patrolling in the new nest and foraging area*. This behavior is a defensive patrolling inside the new environment.

FIGURE 12: Activity analysis: evolution of the number n_{ij} of individuals involved in the activities $A_0 \cdots A_2$.

Now, the abstracted individual activity during emigration can be used to compute the collective dynamic of emigration. Figure 12 is a representation of this dynamic, showing the number of ants expressing each activity during the migration process. As we can see, colony emigration follows a typical pattern: when the light is switched on, the first event is a panic exploration of the old nest that is expressed by most of the ants. Slower patrolling will remain constant during all the process and concern not only the old nest but also the foraging area. The second event is the discovery of new nests followed by the brood transport. Afterwards, a more constant exploration of the new nest occur. The last activity, which appears gradually, is the settlement in the new nest.

From an ethological point of view, these results are of great help for understanding how tasks are distributed during a nest relocation. In fact, we obtained a very accurate description of the dynamic of the whole colony during all the emigration phase allowing us to emit some strong hypothesis about the function of the different behavior during the nest relocation phase. Some results are in accordance with previous works, especially the behaviors that can be observed in the foraging area. For example, the dynamic of the transportation behavior detected by the system match up the results presented in [26]. These hypotheses should now be validated by repeating the experiment with different colonies and different species. A complete understanding of the emigration process based on systematic experimentations would be an important step ahead for the research in social insects.

5. Conclusions

In this paper, a new algorithm is proposed for modeling data structure, based on the learning of an SOM, and a measure of dissimilarity between cluster structures. The advantages of this algorithm are not only the low computational cost and the low memory requirement, but also the high accuracy achieved in fitting the distribution of the modeled datasets. The results obtained on the basis of artificial and real datasets are very encouraging. The new unsupervised algorithm used in this paper is an efficient data mining tool for behavioral studies based on RFID technology. It allows discovering and comparing stable patterns in a RFID signal and is suitable for continuous learning.

Here, it was possible to highlight some characteristics of spatial organization of customers during their shopping in a big store from complex and noisy spatiotemporal database. Moreover, the characteristics of spatiotemporal organization in ant colonies during migration were described, and these results are perfectly compatible with the results of previous works using classic methods [27].

Acknowledgment

This work was partly supported by the ANR (Agence National de la Recherche) CADI 07 TLOG 003 and SILLAGE 05 BLAN 017701.

References

- [1] P. Lyman and H. R. Varian, "How Much Information, 2003," <http://www.sims.berkeley.edu/how-much-info-2003>.
- [2] J. Gehrke, F. Korn, and D. Srivastava, "On computing correlated aggregates over continual data streams," *Special Interest Group on Management of Data Record Record*, vol. 30, no. 2, pp. 13–24, 2001.
- [3] G. S. Manku and R. Motwani, "Approximate frequency counts over data streams," in *Very Large Data Base*, pp. 346–357, 2002.
- [4] F. Cao, M. Ester, W. Qian, and A. Zhou, "Density-based clustering over an evolving data stream with noise," in *Proceedings of the 6th SIAM International Conference on Data Mining*, pp. 328–339, April 2006.
- [5] C. Aggarwal and P. Yu, "A survey of synopsis construction methods in data streams," in *Data Streams: Models and Algorithms*, C. Aggarwal, Ed., pp. 169–207, Springer, 2007.
- [6] G. Cabanes and Y. Bennani, "A local density-based simultaneous two-level algorithm for topographic clustering," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '08)*, pp. 1176–1182, June 2008.
- [7] G. Cabanes and Y. Bennani, "Comparing large datasets structures through unsupervised learning," in *Proceedings of the International Conference on Neural Information Processing*, pp. 546–553, 2009.
- [8] T. Kohonen, *Self-Organizing Maps*, Springer, Berlin, Germany, 2001.
- [9] B. Silverman, "Using kernel density estimates to investigate multimodality," *Journal of the Royal Statistical Society, Series B*, vol. 43, pp. 97–99, 1981.
- [10] J. Vesanto, "Neural network tool for data mining: SOM Toolbox," 2000.

- [11] S. Sain, K. Baggerly, and D. Scott, "Cross-validation of multivariate densities," *Journal of the American Statistical Association*, vol. 89, pp. 807–817, 1994.
- [12] A. Ultsch, "Clustering with SOM: U*C," in *Proceedings of the Workshop on Self-Organizing Maps*, pp. 75–82, 2005.
- [13] E. E. Korkmaz, "A two-level clustering method using linear linkage encoding," in *Proceedings of the International Conference on Parallel Problem Solving From Nature*, vol. 4193 of *Lecture Notes in Computer Science*, pp. 681–690, 2006.
- [14] L. Vincent and P. Soille, "Watersheds in digital spaces: an efficient algorithm based on immersion simulations," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 6, pp. 583–598, 1991.
- [15] S.-H. Yue, P. Li, J.-D. Guo, and S.-G. Zhou, "Using Greedy algorithm: DBSCAN revisited II," *Journal of Zhejiang University Science*, vol. 5, no. 11, pp. 1405–1412, 2004.
- [16] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society, Series B*, vol. 39, pp. 1–38, 1977.
- [17] J. R. Hershey and P. A. Olsen, "Approximating the Kullback Leibler divergence between Gaussian mixture models," in *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '07)*, vol. 4, pp. 317–320, 2007.
- [18] M. Halkidi, Y. Batistakis, and M. Vazirgiannis, "Cluster validity methods," *Special Interest Group on Management of Data Record Record*, vol. 31, no. 2-3, pp. 40–45, 2002.
- [19] A. Ben-Hur, A. Elisseeff, and I. Guyon, "A stability based method for discovering structure in clustered data," in *Proceedings of the Pacific Symposium on Biocomputing*, vol. 7, pp. 6–17, 2002.
- [20] D. L. Davies and D. W. Bouldin, "A cluster separation measure," *IEEE Transactions on Pattern Recognition and Machine Intelligence*, vol. 1, no. 2, pp. 224–227, 1979.
- [21] J. C. Dunn, "Well separated clusters and optimal fuzzy partitions," *Journal of Cybern*, vol. 4, pp. 95–104, 1974.
- [22] A. K. Jain and R. C. Dubes, *Algorithms for Clustering Data*, Prentice-Hall, Upper Saddle River, NJ, USA, 1988.
- [23] C. Fraley and A. E. Raftery, "Model-based clustering, discriminant analysis, and density estimation," *Journal of the American Statistical Association*, vol. 97, no. 458, pp. 611–631, 2002.
- [24] H. Park and T. Ozeki, "Singularity and slow convergence of the em algorithm for gaussian mixtures," *Neural Processing Letters*, vol. 29, no. 1, pp. 45–59, 2009.
- [25] U. Galassi, A. Giordana, C. Julien, and L. Saitta, "Modeling temporal behavior via structured hidden markov models: an application to keystroke dynamics," in *Proceedings of the Indian International Conference on Artificial Intelligence*, pp. 2140–2154, 2007.
- [26] A. Pezon, D. Denis, P. Cerdan, J. Valenzuela, and D. Fresneau, "Queen movement during colony emigration in the facultatively polygynous ant *Pachycondyla obscuricornis*," *Naturwissenschaften*, vol. 92, no. 1, pp. 35–39, 2005.
- [27] D. Fresneau and P. Dupuy, "Behavioural study of the primitive ant *Neoponera apicalis*," *Animal Behaviour*, vol. 36, no. 5, pp. 1389–1399, 1988.