Research Article

Parameters Optimization and Application to Glutamate Fermentation Model Using SVM

Xiangsheng Zhang and Feng Pan

Key Laboratory of Advanced Process Control for Light Industry (Ministry of Education), Jiangnan University, Wuxi 214122, China

Correspondence should be addressed to Xiangsheng Zhang; zxs@jiangnan.edu.cn and Feng Pan; pan_feng_63@163.com

Received 10 September 2014; Revised 23 December 2014; Accepted 23 December 2014

Academic Editor: Pandian Vasant

Copyright © 2015 X. Zhang and F. Pan. 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.

Aimed at the parameters optimization in support vector machine (SVM) for glutamate fermentation modelling, a new method is developed. It optimizes the SVM parameters via an improved particle swarm optimization (IPSO) algorithm which has better global searching ability. The algorithm includes detecting and handling the local convergence and exhibits strong ability to avoid being trapped in local minima. The material step of the method was shown. Simulation experiments demonstrate the effectiveness of the proposed algorithm.

1. Introduction

Glutamate fermentation is a complex microbial growth process. Glutamate bacterium draws raw material of nutrition and produces complex biochemical reactions in vivo specific enzyme [1]. The reaction process is highly nonlinear, time-variant, and uncertain. It is very difficult to establish dynamic model in the application of fermentation control [2, 3].

Modeling the glutamate fermentation process under normal or abnormal conditions is major challenge as an optimization strategy is applied from the initial stage to the final stage. It is difficult to measure online substrate, biomass, and product concentrations; hence pH, dissolved oxygen (DO) concentration, and CO₂ production are usually utilized in the bioprocess analysis. The process variables provide indications of the bioprocess condition. However, the information density of these data is usually low, and the multidimensional nature of the data usually makes it difficult to understand.

Support vector machines (SVMs), developed by Boser, Guyon, and Vapnik (1992), are a kind of machine learning method based on statistical learning theory [4–7]. It has become a hot research in the field of machine learning. SVM overcomes commendably such defects as dimensionality curse and overfitting that are apt to appear in some other conventional algorithms, such as neural networks [8, 9].

But there exists a problem in the practical application of SVM. This problem is how to select some SVM parameters so that the performance of SVM can be maximized. These SVM parameters mainly include the penalty constant C , the relaxation factor ξ , and the parameters in kernel function (e.g., the width in the RBF kernel function), and they affect the SVM performance more or less. Most existing approaches use leave-one-out (LOO) [10], gradient descent (GD) [11], genetic algorithm (GA) [12], particle swarm optimization (PSO) [13], and related parameter estimators. There does not exist a single universally effective method to select these SVM parameters. Generally, cross verification trial is used, but it involves *ad hoc* user interference factors or requires that the kernel function should be continuously differentiable, and the resulting SVM classifiers are prone to falling into local minima.

In order to overcome the shortcomings of the existing parameter selection approaches mentioned above, an attempt is made to jointly optimize the feature selection and the SVM parameters with some evolutionary algorithms such as improved particle swarm optimization (IPSO) algorithm, in hopes of improving the SVM performance in glutamate fermentation. This paper describes an IPSO algorithm which has better global search ability and then focuses on optimizing the parameters which can be used in the prediction of SVM modeling and state of glutamate fermentation.

The paper is organized as follows. In Section 2, the SVM with mixed kernel function is briefly reviewed. Section 3 describes the standard PSO algorithm. Then in Section 4 we present the new IPSO technique. The simulations results are provided in Section 4. Finally, concluding remarks are drawn in Section 5.

2. Mixed Kernel Function

The aim of this section is to introduce the notations and to review the concepts that are relevant to the development of the proposed model parameter selection method.

The development of SVM starts from the simplest case of two classes that is linearly separable. Its basic mathematical model can be given using the following training sample set:

$$T = \{x_i, y_i, i = 1, 2, \dots, l\}, \quad x_i \in R^N, \quad y_i \in R, \quad (1)$$

$$f(x) = \sum_{i=1}^l w_i \phi_i(x) + b,$$

where $\{\phi_i(x)\}_{i=1}^l$ is the data in features space and $\{w_i(x)\}_{i=1}^l$ and b are coefficients. They can be estimated by minimizing the regularized risk function

$$R(C) = C \frac{1}{l} \sum_{i=1}^l L_\varepsilon(y, f(x_i)) + \frac{1}{2} \|w\|^2, \quad (2)$$

where $L_\varepsilon(y, f(x_i))$ is the so-called loss function measuring the approximate errors between the expected output y_i and the calculated output $f(x_i)$ and C is a regularization constant determining the trade-off between the training error and the generalization performance. The second term, $(1/2)\|w\|^2$, is used as a measurement of function flatness. Introducing slack variables ξ, ξ^* transforms (2) into the following constrained function:

$$\begin{aligned} \min_{w, b, \xi} \quad & J = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l (\xi_i + \xi_i^*) \\ \text{s.t.} \quad & y_i - w\phi(x_i) - b \leq \varepsilon + \xi_i \\ & w\phi(x_i) + b - y_i \leq \varepsilon + \xi_i^* \\ & \xi, \xi_i^* \geq 0, \quad i = 1, 2, \dots, l. \end{aligned} \quad (3)$$

Using the duality principle, (3) could be changed to

$$\begin{aligned} \max_{\alpha, \alpha^*} \quad & \left\{ L_D = -\frac{1}{2} \sum_{i,j=1}^l (\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*) \langle x_i \cdot x_j \rangle \right. \\ & \left. - \varepsilon \sum_{i=1}^l (\alpha_i + \alpha_i^*) + \sum_{i=1}^l y_i (\alpha_i - \alpha_i^*) \right\} \\ \text{s.t.} \quad & \sum_{i=1}^l (\alpha_i - \alpha_i^*) = 0 \\ & \alpha_i, \alpha_i^* \in [0, C], \quad i = 1, 2, \dots, l. \end{aligned} \quad (4)$$

Although the nonlinear function ϕ is usually unknown, all computations related to ϕ can be reduced to the form $\phi(x)^T \phi(y)$, which can be replaced with a so-called kernel function $K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$ that satisfies Mercer's condition. Then, (1) becomes

$$f(x, \alpha_i, \alpha_i^*) = \sum_{x_i \in \text{SV}} (\alpha_i - \alpha_i^*) K(x_i, x) + b. \quad (5)$$

In (5), Lagrange multipliers α_i and α_i^* satisfy the equalities:

$$\alpha_i \times \alpha_i^* = 0, \quad \alpha_i \geq 0, \quad \alpha_i^* \geq 0, \quad i = 1, \dots, l. \quad (6)$$

Those vectors with $\alpha_i \neq 0$ are called support vectors, which contribute to the final solution.

The kernel functions are used to project the sample data into a high-dimensional feature space and then to find the optimal separation plane in it. Kernel functions used by SVM can be divided into global and local kernels. Smits and Jordaan studied representative mapping properties of global kernel function (polynomial kernel function) and local kernel function (RBF kernel) and proposed a mixed kernel function. It was shown that the mixed kernel function-based SVM has strong learning and generalization ability.

This paper selects two polynomial kernel functions and RBF kernel function to produce a hybrid kernel function given by

$$K_{\text{mix}} = \rho K_{\text{poly}} + (1 - \rho) K_{\text{RBF}}, \quad (7)$$

where $K_{\text{poly}} = (x \cdot x_i + 1)^2$ is the polynomial kernel function; $K_{\text{RBF}} = \exp(-\|x - x_i\|^2 / \sigma^2)$ is the RBF kernel function; ρ ($0 \leq \rho \leq 1$) adjusts the sizes of two kernel functions.

The parameters in the mixed kernel function SVM such as C and the width coefficient in the kernel function $K(x, x_i)$ exert a considerable influence on the performance of SVM. A large or small value of C may lead to degraded generalization ability of SVM. The value of C indicates the error expectation in the classification process of the sample data, and it affects the number of support vectors generated by the classifier, thereby affecting the generalization error of the classifier. If the value of C is too big, the separating error is high, the number of support vectors is small, and vice versa [14]. The parameters in the kernel function reflect the characteristics of the training data, and they also affect the generalization of SVM. Therefore, only after the choice of all these parameters is correctly made can the SVM achieve its best possible performance.

3. Improved Particle Swarm Optimization (IPSO)

The standard PSO algorithm has fast convergence speed, but the speed of particle gets more and more slowly later [15, 16]. A particle updates its velocity and location according to the following formula:

$$\begin{aligned} V_i^{(t+1)} &= \omega V_i^{(t)} + c_1 r_1 (P_{i\text{best}}^{(t)} - X_i^{(t)}) + c_2 r_2 (P_{g\text{best}}^{(t)} - X_i^{(t)}) \\ X_i^{(t+1)} &= X_i^{(t)} + V_i^{(t+1)}, \end{aligned} \quad (8)$$

FIGURE 1: Flowchart of IPSO.

where $V_i^{(t)}$ and $V_i^{(t+1)}$ represent the present and next velocity of particle i and $X_i^{(t)}$ and $X_i^{(t+1)}$ are the present and next position of particle i .

In order to reduce the possibility in the evolutionary process of particles leaving the search space, the particle velocity is usually limited to a certain range $V_i \in [-V_{\max}, V_{\max}]$; P_i is the best previous position of particle i ; P_g is the best position among all particles in the population $X = [X_1, X_2, \dots, X_N]$; r_1 and r_2 are random real numbers in the range of $[0, 1]$; c_1 and c_2 are acceleration constants; ω is the internal weight coefficient (i.e., the impact of the previous velocity of particle on its current one).

Because it is easy for the standard PSO to fall into local minima, which is called the phenomenon of being premature [17–19], improved particle swarm optimization algorithm is proposed which is the theory of detecting premature convergence. This algorithm combines the method of analyzing and treating premature convergence to avoid the premature phenomenon throughout the whole algorithm. The whole algorithm process is shown in Figure 1.

3.1. Premature Detection. The literature [20] pointed out that when the particle swarm has premature convergence, the particles in the swarm will appear as “aggregation” phenomenon, and the position of the particles determines the adaptability of particles, so the state of particle swarm can be tracked by the overall changes in the fitness of all particles.

The number of particles in the particle swarm is assumed to be m , F_i is the fitness of i particle, F_{avg} is the mean fitness, and δ^2 is the particle swarm colony fitness variance defined as

$$\delta^2 = \sum_{i=1}^m \frac{F_i - F_{\text{avg}}}{F}, \quad (9)$$

where F is a normalized scaling factor, and it can restrict the size of δ^2 . The value of the F is determined using the following formula:

$$F = \begin{cases} \max_{1 \leq i \leq m} |F_i - F_{\text{avg}}|, & \max_{1 \leq i \leq m} |F_i - F_{\text{avg}}| > 1 \\ 1, & \text{other.} \end{cases} \quad (10)$$

Equation (9) shows that the variance of the colony fitness is the reflection of all particles in the swarm “aggregation” degree. The smaller the δ^2 , the greater “aggregates” level particle swarm is. If the algorithm does not meet the termination condition, the “aggregation” will enable the group to lose diversity in early state. A premature detection can be made when $\delta^2 < H$ (H is a given constant).

3.2. Premature Treatment. For premature state of the particle, the method combined by chaos algorithm and particle swarm algorithm can set up the position and velocity of particles and make it jump out of local minima. Use the classical logistic equation to achieve chaos sequence via

$$y'_i = \lambda y_i (1 - y_i), \quad (11)$$

where y_i is the random number uniformly sampled from $(0, 1)$; λ is control parameter, when it takes the value of 4, the Logistic equation is in complete chaos [13]. Then, it will be introduced into the optimization space by the following formula:

$$x_i = a_i + y'_i (b_i - a_i), \quad (12)$$

where $[a_i, b_i]$ is the range of variables. A global optimal solution has no practical significance to the algorithm when particle swarm runs into local optimum. We can put forward the renewal equation of particle premature condition after combining (11) and (12):

$$\begin{aligned} V_i^{(t+1)} &= \omega V_i^{(t)} + c_1 r_1 (P_{i\text{best}} - X_i^{(t)}) + (-V_{\max} + 2y'_i V_{\max}), \\ X_i^{(t+1)} &= X_i^{(t)} + V_i^{(t+1)}, \end{aligned} \quad (13)$$

where $[-V_{\max}, V_{\max}]$ is the range of particle velocity. Thus, the particle can jump out of local optimum and back into the particle swarm optimization iteration.

3.3. Optimization Algorithm. In order to improve the precision and ability of generalization of the mixed kernel function, the following variance function can be used for the fitness function of IPSO algorithm which can respond to SVM regression performance directly. Consider

$$F_{\text{fitness}} = \frac{1}{n} \sum_{i=1}^n (f_i - y_i)^2, \quad (14)$$

where f_i is the predictive value, y_i is the measured value, and n is the number of samples. The flow diagram of optimization algorithm is shown in Figure 2. The detailed process of modeling is as follows.

FIGURE 2: Flowchart of optimization algorithm.

Step 1. Initialize particle swarm parameters (ρ, σ, C), swarm size, and the maximum number of iterations and determine the weight factor ω and particle swarm flight speed range $[-V_{\max}, V_{\max}]$. Set a loss function parameter ε of SVM and judgment criterion of global convergence and premature convergence.

Step 2. The individual value of each particle P_{ibest} is set to the current position, and then fitness is calculated for each particle, the fitness value of individual extreme corresponding to the best particle as the global extreme initial P_{gbest} .

Step 3. Judging the convergence criterion is satisfied. If the algorithm is satisfied, then execute Step 10; otherwise go to Step 4.

Step 4. Use (2) (the standard PSO algorithm) to execute the iterative calculation and update the position and velocity of particles.

Step 5. Compare $F(X_i)$ and $F(P_{ibest})$, and if the $F(X_i) < F(P_{ibest})$, update P_{ibest} .

Step 6. Compare the updated $F(P_{ibest})$ and $F(P_{gbest})$, and if $F(P_{ibest}) < F(P_{gbest})$ then update P_{gbest} .

Step 7. Determine whether the convergence criterion is satisfied. If it is satisfied, then execute Step 10; otherwise go to Step 8.

Step 8. Evaluate (3) and (4) to calculate the variance δ^2 of the population's fitness and then judge whether the establishment of $\delta^2 < H$, and if established turn to Step 9 for premature treatment; otherwise turn to Step 4.

Step 9. According to (7) for premature treatment of falling into local optimal particle, the particle swarm escapes from local optima; then turn to Step 3.

FIGURE 3: Simulation results of $f(x)$.

TABLE 1: Performance comparison of parameters selection of SVM by different methods.

Method	ρ	σ	C	Mean error
Particle swarm optimization	0.0102	0.2015	756	0.0306
Improved particle swarm optimization	0.0100	0.2836	884	0.0201

Step 10. Output the particle swarm optimal value, and the algorithm terminates.

4. Modeling and Simulation

4.1. Function Fitting Simulation. In order to test the effectiveness of the algorithm, one-dimensional function is selected to simulate

$$f(x) = \frac{\sin(x)}{x} + \xi, \quad x \in [-3, 3], \quad (15)$$

where ξ is the Gaussian noise with zero mean and variance of 0.1. In order to minimize the fitness function and optimize the mixed kernel function of SVM parameters using IPSO algorithm, we take 50 sets of data to constitute a hybrid kernel function training sample of SVM in the input variable domain. Among them, mixed kernel function SVM uses an insensitive loss function which is $\varepsilon = 0.1$; ρ, σ and C are initialized, respectively, in $[0, 1]$, $[0.01, 1.0]$, and $[1, 1000]$; the population size is 20, and the premature judgment constant H is 1. The maximum number of iterations is 50 times, and $F(P_{g_{best}}) \leq 10^{-3}$ which is the fitness value as global convergence. The simulation results are given in Figure 3 and Table 1.

The simulation results show that the mixed kernel function of SVM model has a high accuracy constructed by IPSO

FIGURE 4: Predictive result of glutamate concentration.

high parameter optimization. It shows the feasibility and effectiveness of the improved particle swarm optimization algorithm.

4.2. Modeling and Application Based on IPSO. In this paper, a forecast model of glutamate fermentation concentration can be built by a mixed kernel function of SVM based on IPSO; the input which is the current time in fermentation process (H), dissolved oxygen (DO), concentration of glutamate (g/L), cell concentration (g/L), and residual sugar concentration (g/L) can estimate next moment of glutamate and residual sugar concentration.

In the experiment, in the 6 batches of production data, each number represents a complete fermentation process, in which the 4 batches of data are for training, and the other 2, used as the test sample to test its generalization ability. Modeling mixed kernel function SVM using an insensitive loss function $\varepsilon = 0.08$, the general range of initializing particle swarm parameters is given: $\sigma = [0, 1]$, $\rho = [0, 1]$, $C = [100, 1000]$. Based on the training data for learning and the optimization and adjustment for mixed kernel function of SVM parameter by IPSO algorithm, it can get $\sigma = 0.4862$, $\rho = 0.0421$, $C = 861$. Using the parameter set of glutamate fermentation model trained to predict test data, the results are as shown in Figures 4 and 5.

Figure 4 shows the changing behavior in glutamate production from 6 glutamate fermentation experiments. The production of glutamate increased in a nonlinear way over the fermentation period from 2 h to 38 h; glutamate increased very slowly in the early period during 2–5 h. It increased quickly from 20 to 60 g/L in 9–20 h and then increased slowly after 20 h. After 34 h, the production of glutamate kept stable at 70–75 g/L. Figure 5 shows the changing behavior of residual sugar concentration in glutamate fermentation experiments. The early period (usually the former 5 hours) is bacteria growth stage and the sugar concentration is

FIGURE 5: Predictive result of residual sugar concentration.

higher. About 9 hours later fermentation turned into acid producing period, bacteria growth slowing down and a large number of glutamic acids accumulating. Most of sugar in the culture medium has been burned off, requiring a continuous consumption of nutrients.

In order to maintain stable environment for the growth of bacteria and prevent severe changes in sugar concentration which can disturb bacteria physiological metabolism, we can take a fed-batch flow rate strategy and keep the glucose concentration on a constant level, referring to the estimate results.

The average prediction error for glutamate concentration is 2.61%. The average prediction error residual sugar concentration is 3.82%. It can then be seen that this method has good modeling effect.

5. Conclusion and Future Work

In this paper, a new IPSO-based optimization algorithm was used to select the optimal parameters for the mixed kernel function of SVM. The simulation results show that the mixed kernel function-based SVM with optimally chosen parameters is accurate and reliable, and the optimum mixed kernel function of SVM model based on structural parameters has better learning precision.

Nomenclature

SVM: Support vector machine
 IPSO: Improved particle swarm optimization
 DO: Dissolved oxygen
 LOO: Leave-one-out
 GD: Gradient descent
 GA: Genetic algorithm

PSO: Particle swarm optimization
 RBF: Radial basis function
 T : Set of training data
 $R(C)$: Regularized risk function
 C : Regularization constant
 K_{mix} : Hybrid kernel function
 K_{poly} : Polynomial kernel function
 K_{RBF} : RBF kernel function
 $V_i^{(t)}$: Present velocity of particle i
 $V_i^{(t+1)}$: Next velocity of particle i
 $X_i^{(t)}$: Present position of particle i
 $X_i^{(t+1)}$: Next position of particle i
 P_i : Best previous position of particle i
 P_g : Best position of particle i among all particles
 c_1, c_2 : Acceleration constants
 ω : Internal weight coefficient
 m : Number of particles in particle swarm
 F_i : The fitness of i particle
 F_{avg} : The mean fitness
 σ^2 : Particle swarm colony fitness variance
 F : Normalized scaling factor
 y_i : Random number
 λ : Control parameter
 F_{fitness} : Fitness function
 f_i : Predictive value
 y_i : Measured value
 n : Number of samples
 ξ : Gaussian noise.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant no. 61273131) and the Fundamental Research Funds for the Central Universities (JUDCF13040).

References

- [1] J. Xiao, Z. P. Shi, P. Gao, H. Feng, Z. Duan, and Z. Mao, "On-line optimization of glutamate production based on balanced metabolic control by RQ," *Bioprocess and Biosystems Engineering*, vol. 29, no. 2, pp. 109–117, 2006.
- [2] X. Feng, T. Yu, and J. Wang, "Nonlinear GPC with in-place trained RLS-SVM model for DOC control in a fed-batch bioreactor," *Chinese Journal of Chemical Engineering*, vol. 20, no. 5, pp. 988–994, 2012.
- [3] X. J. Gao, P. Wang, Y. S. Qi, Y. Zhang, H. Q. Zhang, and A. J. Yan, "An optimal control strategy combining SVM with RGA for improving fermentation titer," *Chinese Journal of Chemical Engineering*, vol. 18, no. 1, pp. 95–101, 2010.
- [4] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Springer, Berlin, Germany, 1995.

- [5] Y. F. Li, Z. F. Wang, and J. Q. Yuan, "On-line fault detection using SVM-based dynamic MPLS for batch processes," *Chinese Journal of Chemical Engineering*, vol. 14, no. 6, pp. 754–758, 2006.
- [6] Z. W. Liu, H. R. Cao, X. F. Chen, Z. J. He, and Z. J. Shen, "Multi-fault classification based on wavelet SVM with PSO algorithm to analyze vibration signals from rolling element bearings," *Neurocomputing*, vol. 99, pp. 399–410, 2013.
- [7] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines*, Cambridge University Press, Cambridge, UK, 2000.
- [8] G. F. Smits and E. M. Jordaen, "Improved SVM regression using mixtures of kernels," in *Proceedings of the International Joint Conference on Neural Networks (IJCNN '02)*, pp. 2785–2790, May 2002.
- [9] S.-X. Du and T.-J. Wu, "Support vector machines for regression," *Journal of System Simulation*, vol. 15, no. 11, pp. 1580–1585, 1633, 2003.
- [10] J. H. Lee and C. J. Lin, *Automatic Model Selection for Support Vector Machines*, Taiwan University, Taipei, Taiwan, 2000.
- [11] O. Chapelle, V. Vapnik, O. Bousquet, and S. Mukherjee, "Choosing multiple parameters for support vector machines," *Machine Learning*, vol. 46, no. 1–3, pp. 131–159, 2002.
- [12] C. L. Zheng and L. C. Jiao, "Automatic parameters Selection for SVM based on GA," in *Proceedings of the 5th World Congress on Intelligent Control and Automation (WCICA '04)*, pp. 1869–1872, Piscataway, NJ, USA, June 2004.
- [13] A. P. Mariano, C. B. B. Costa, D. D. F. de Angelis et al., "Optimisation of a fermentation process for butanol production by particle swarm optimisation (PSO)," *Journal of Chemical Technology and Biotechnology*, vol. 85, no. 7, pp. 934–949, 2010.
- [14] W. Wang, X. M. Guo, S. Y. Wang et al., "On the method of Kernel function selection in support vector machine," *Journal of Liaoning Normal University (Natural Science Edition)*, vol. 31, no. 1, pp. 1–4, 2008.
- [15] Q. Wu, "A self-adaptive embedded chaotic particle swarm optimization for parameters selection of Wv-SVM," *Expert Systems with Applications*, vol. 38, no. 1, pp. 184–192, 2011.
- [16] X. Li, S.-D. Yang, and J.-X. Qi, "New support vector machine optimized by improved particle swarm optimization and its application," *Journal of Central South University of Technology*, vol. 13, no. 5, pp. 568–572, 2006.
- [17] Y. J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, pp. 1942–1948, Perth, Australia, December 1995.
- [18] R. Suzuki, F. Kawai, C. Nakazawa, T. Matsui, and E. Aiyoshi, "Parameter optimization of model predictive control by PSO," *Electrical Engineering in Japan*, vol. 178, no. 1, pp. 40–49, 2012.
- [19] Y.-B. Li, N. Zhang, and C.-B. Li, "Support vector machine forecasting method improved by chaotic particle swarm optimization and its application," *Journal of Central South University of Technology*, vol. 16, no. 3, pp. 478–481, 2009.
- [20] H. Y. Liu, Y. Lin, and J. S. Zhang, "A hybrid particle swarm optimization based on chaos strategy to handle local convergence," *Computer Engineering and Applications*, no. 13, pp. 77–79, 2006.