DEVELOPMENT OF ONLINE INDUCTIVE SYSTEMS USING SUPPORT VECTOR MACHINES

Gorden T. Jemwa and Chris Aldrich

Institute of Mineral Processing and Intelligent Process Systems,
Department of Process Engineering, University of Stellenbosch
Stellenbosch, Private Bag X1, Matieland, South Africa 7602
Fax +27(21) 808 2059

Abstract: Process monitoring, fault detection and diagnosis and supervisory control, among many other control methods, are based on statistical pattern recognition, where the objective is to eliminate assignable causes in process behaviour. However, it has been shown that the majority of process and manufacturing quality problems are attributable to common causes. Historical data collected under predictable and stable operating conditions contain information that can be used to reduce process variability. In this paper we propose the use of support vector machines (SVMs) in the identification of informative patterns from a process under statistical control. SVMs possess a number of advantages over other methods, such as the capability to describe nonlinear functions, to derive sparse descriptions of solution sets from training set patterns, as well as the ability to deal with noisy data. A symbolic classifier is then used to search and formulate process improvement opportunities by partitioning the decision space by using the previously identified support vectors.

Keywords: pattern recognition, decision trees, support vector machines.

1. INTRODUCTION

Technological advancements have seen an increase in data collection and processing activities in industrial processing plants. State-of-the-art industrial plants routinely measure a large number of variables using online sensors. Analysis of historical operating data can uncover potentially useful information to provide insight into and monitor process behaviour. While extraction of useful information is an ongoing research problem, statistical pattern recognition has inspired a number of applications now widely used in process monitoring, fault detection and diagnosis, supervisory control, etc. (Kresta *et al*., 1991; Raich and Çinar, 1996). These methods monitor process trends to detect abnormal events, provide troubleshooting guidelines when a similar problem recurs and, if possible, to eliminate the causes. Unfortunately, these approaches are aimed at restricting the process variability in a bounded region of predictable and stable operation. In other words, variability attributable to common and sustained causes is considered unavoidable.

However, studies have indicated that the majority of process and manufacturing quality problems arise from common causes (Deming, 1989). Hence, process improvements with significant impacts can be realised by complementing statistical process control (SPC) and related methods with efforts focussed on the reduced dispersion of performance or quality variables.

In this paper, we discuss an online support vector-based framework for formulation of process improvement opportunities using operating data collected from a process under a statistical control. Pivotal data points containing all the information required to separate objects belonging to different classes are identified using supports vector machines. Subsequently, a symbolic and modularised description of the decision boundaries is obtained by induction of decision trees using the pivotal data points, called support vectors. Suggestions for process improvement are extracted for verification, validation, and implementation by the operator or process engineer.

In the following, an overview of the general learning problem is presented, from which the algorithm for support vector machine learning is formulated. The use of classification decision trees, which provide the symbolic module in the methodology, is briefly described. A simulated continuous stirred tank reactor (CSTR) is then used to discuss and illustrate the functionality and features of the system.

2. EMPIRICAL LEARNING

Statistical pattern recognition is one of three specific instances of the general learning problem and forms the core part of the methodology (Vapnik, 1998). An understanding of the basic learning problem is therefore important.

2.1 The General Learning Problem

In the general learning problem, the idea is to *learn* a function from which the correct output can be computed given input data. A particular approach in solving this problem is supervised learning, which involves the use of known input/output pairs of vectors to estimate the function. Thus, for a two-class classification problem, given independent and identically distributed input-output training data pairs $(\mathbf{x}_i, y_i), i = 1,...,N;$ where $\mathbf{x} \in \Re^d$ and $y \in \{-1,+1\}$, a function f is sought such that the following holds:

$$y_i = f(\mathbf{x}_i), \ i = 1,...,N. \tag{1}$$

Based on the available information, an estimate of f is obtained by using induction. Conventional approaches, e.g. least-squares minimisation and maximum-likelihood, find such an estimate using the *empirical* risk minimisation (ERM) principle, which finds a consistent hypothesis by minimizing the training error.

Another alternative and more principled approach, *structural* risk minimisation (SRM), seeks a hypothesis that minimises the error of misclassifying yet-to-be-seen objects (Burges, 1997; Cristianini and Shawe-Taylor, 2000). In other words, the generalisation capacity of the classifier is maximised. Support Vector Machines (SVMs) employ the SRM principle, which has been shown to be superior to the ERM principle (Gunn, et al., 1997).

2.2 Classification with SVMs

Basic Theoretical Framework. The basic idea in SVM classification is to learn the optimal decision boundary or separating hyperplane in a high dimensional feature space using simple linear machines. If the separating hyperplane is defined by a weight parameter (**w**) and bias (*b*), then the hyperplane that minimises the solution of the problem, is found by constrained minimisation of an objective function $\varphi(\mathbf{w})$, as follows

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \varphi(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{subject to} \quad & y_i\left(\langle \mathbf{w}, \mathbf{x}_i \rangle + b\right) \geq 1, \ i = 1,...,N. \end{aligned} \tag{2}$$

A Lagrangian formulation of (2) yields the primal problem:

$$L(\mathbf{w},b,\alpha) = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{i=1}^{N}\alpha_i\left[y_i\left(\langle \mathbf{w}, \mathbf{x}_i \rangle + b\right)\right] \tag{3}$$

where $\alpha_i \geq 0$ are the Lagrange multipliers. Using Lagrangian duality, the primal problem in (2) can be transformed into an equivalent but easier to solve dual problem:

$$\max_{\alpha_i}\left(\min_{\mathbf{w},b} L(\mathbf{w},b,\alpha\right). \tag{4}$$

The solution of equations (2), (3), and (4) is given by among other Burges (1997) and Cristianini and Shawe-Taylor (2000):

$$\begin{aligned} \bar{\alpha} = \min_{\alpha} & \frac{1}{2}\sum_{i,j=1}^{N}\alpha_i\alpha_j y_i y_j \langle \mathbf{x}_i, \mathbf{x}_j \rangle - \sum_{k=1}^{N}\alpha_k \\ \text{subject to} \quad & \begin{cases} \sum_{i=1}^{N}\alpha_i y_i = 0 \\ \alpha_i \geq 0, \quad i = 1,...,N. \end{cases} \end{aligned} \tag{5}$$

The weight vector is then given by the expression $\bar{\mathbf{w}} = \sum_{i=1}^{N} y_i \bar{\alpha}_i$, which is the optimal maximal margin hyperplane. The bias *b* is found from the primal Karush-Kuhn-Tucker conditions:

$$\alpha_i\left[y_i\left(\langle \mathbf{w}, \mathbf{x}_i \rangle + b - 1\right)\right] = 0, \quad i = 1,...,N \tag{6}$$

Also, from (6) it can be seen that the corresponding Lagrangian multipliers exceed zero ($\alpha_i > 0$), only for inputs which lie closest to the hyperplane, while the rest of the multipliers are zero. Hence, in the description of the weight vector solution, only these Lagrangian multipliers are involved. Therefore, although the entire training set is used in the formulation and optimisation of the problem, the solution is found in terms of a few informative patterns only, also referred to as support vectors. The rest are degenerate and their omission in the formulation does not affect the solution.

Noisy Data and Complex Nonlinear Functions. Real data are invariably noisy, with decision functions appropriately defined by complex nonlinear functions. The SVM formulation as presented above cannot handle these data sets. Fortunately, the basic formulation described above can be extended. In particular, Cortes and Vapnik (1995) introduced an error term $C\sum_{i=1}^{N}\xi_i$ in the formulation of equation (2), where *C* is a regularisation constant and ξ_i a slack variable:

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \varphi(\mathbf{w}) = \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i \\ \text{subject to} \quad & y_i\left(\langle \mathbf{w}, \mathbf{x}_i \rangle + b\right) \geq 1, \ i = 1,...,N \end{aligned} \tag{7}$$

The solution of the optimisation problem is then a trade-off between maximisation of the margin term $\varphi(\mathbf{w})$ and level of allowable misclassification errors.

To allow for both linear models and nonlinear decisions, we map the input into a higher dimensional feature space *F* and formulate a *linear* algorithm in the new space:

$$\begin{aligned} \phi : \Re^d &\rightarrow F \\ \mathbf{x} &\mapsto \phi(\mathbf{x}) \end{aligned} \tag{8}$$

Kernel mapping (equation 9),

$$K(\mathbf{x},\mathbf{z}) = \langle \phi(\mathbf{x}), \phi(\mathbf{z}) \rangle \qquad (9)$$

sidesteps the potential computational complexity introduced by working in higher dimensions (Cristianini and Shawe-Taylor, 2000). Additionally, use of kernels negates the need to explicitly know the underlying feature map. However, a function must satisfy certain properties to ensure it is a kernel and construction of kernels is a separate issue that will not be discussed here. In all the experiments described below, we used standard Gaussian kernels, used successfully in many applications.

In summary, SVMs are learning machines that find a linear decision boundary in an implicitly defined feature space, corresponding to a nonlinear decision boundary in input space. The solution space is defined in terms of a sparse subset of the training data with good generalisation properties.

2.3 Classification Decision Trees

Although powerful techniques now exist that learn classification tasks, most are not amenable to an explicit description the decision space. An important feature of self-contained decision support systems is the capability to extract descriptive knowledge from data and expressing these decision rules in a comprehensible language.

Decision trees are top-down symbolic classifiers that express classification rules as a modularised description of the input space. A tree may be a terminal node or leaf associated with a single class, or a test node with a disjoint set of possible outcomes. Each outcome in turn is associated with a subsidiary tree, while the tests are designed to reduce an otherwise heterogeneous set of objects. The classification or decision rules are expressed as complexes or conjunctions of the operating conditions (input space). Thus the classification rules are represented in an understandable and concise language, similar to the way operators describe physical systems, such as processing plants. Decision trees are also flexible and can deal with categorical, discrete, and continuous values in a natural way.

These properties in particular are used to integrate a symbolic reasoning component into the proposed framework for process improvement, as discussed later. The theory and implementation of classification decision trees are well-established and the reader is referred to the literature such as Quinlan (1987) and Breiman *et al.* (1993).

3. DISCOVERING PROCESS IMPROVEMENT OPPORTUNITIES

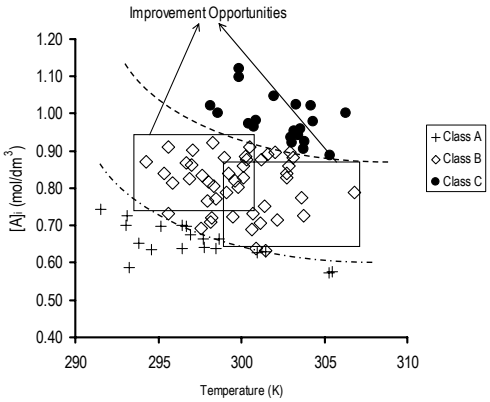

Fig. 1. Decision boundaries induced using a support vector classifier with Gaussian kernels.

As mentioned before in section 1, it is not enough to accept process behaviour under statistical control. Current performance levels need to be challenged continuously to reduce dispersion of quality variables. Fig. 1 illustrates the basic idea for the search and formulation of improvement opportunities for a process under statistical control. The process inputs consist of the two variables, viz. temperature and concentration of reactant Ai, and in this case class B is the desirable operating range. Data points that lie close to class separating hyperplanes provide critical information useful for identifying improvement opportunities. The challenge is to isolate these patterns for subsequent extraction of classification rules using a symbolic classifier.

Saraiva and Stephanopoulos (1992) have proposed the nonparametric identification of these pivotal patterns by using *Tomek links*. However, Tomek links are piecewise linear classifiers and may not be appropriate for nonlinear decision boundaries. Moreover, for sparse data it is not always guaranteed that pairs that form Tomek links remain so in higher dimensions. Support vectors avoid these potential limitations, apart from their other advantages.

With SVMs, the number of support vectors can be controlled, outlier detected and the data filtered, in addition to the advantages mentioned earlier. Fig. 2 gives an overview of the complete methodology. The framework closely resembles that previously proposed by Saraiva and Stephanopoulos (1992), except for the identification of points closest to the separating hyperplanes. The use of support vector classifiers also makes for interesting differences, which we discuss in a tutorial form in the following. First, it is necessary to describe the system used in the illustrations.

Problem Formulation. Consider a first-order irreversible reaction ($A \rightarrow B$) occurring in a CSTR whose reaction kinetics are governed by an Arrhenius relationship. The reactor volume and volumetric flow rate are maintained at constant values, while the feed concentration of reactant *A* and

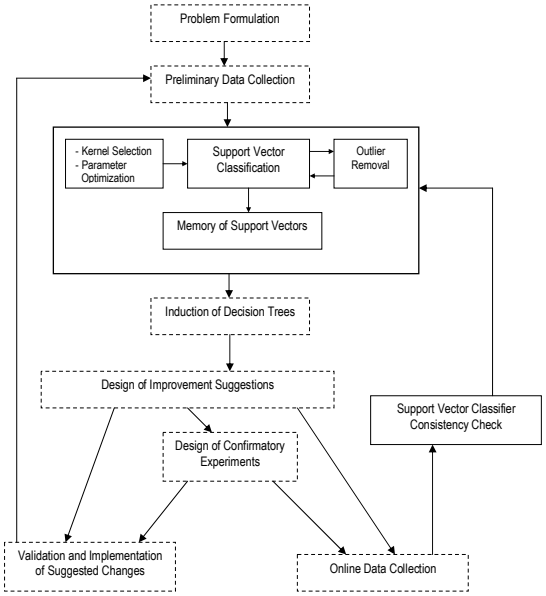

Fig 2: Search and formulation of process improvement opportunities. The pattern recognition scheme (elements with solid lined boxes) is central in the methodology.

the reactor temperature were generated according to normal probability distribution functions, as shown in Table 1 below. As Fig. 3 shows, the process can safely be assumed to be under statistical control.

Table 1 Parameter Values for the CSTR Monte Carlo Simulator

Variable	Mean	Std. deviation
$[A]_i$ (mol/dm^3)	0.8	0.1
T (K)	300	3.5

The two-variable restriction is only to allow for visualisation of the decision hyperplanes. The objects or examples are classified into one of three classes using distribution statistics of the process performance variable, [B]: Values more than one standard deviation lower than the mean of [B] are labelled class A; values falling within one standard deviation as class B, and values more than one standard deviation greater than the mean as class C. Fig. 4a shows a typical partitioning of the input space for 750 data points.

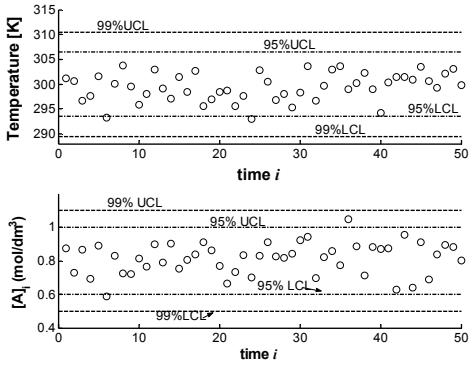

Fig 3. Data description showing SPC limits used for abnormal event monitoring.

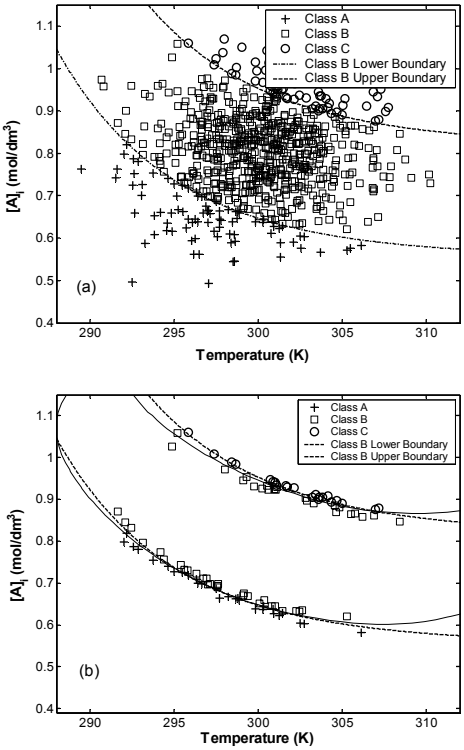

Fig 4: (a) Problem formulation using distribution statistics from historical operating data. (b) Decision boundaries induced using a support vector classifier with a Gaussian kernel with parameter $\gamma = 0.5$ and $C = 10$.

Support Vector Classification. The SVM approach outlined above is subsequently used to identify the support vectors lying in the proximity of the separating hyperplanes for the specified parameters, as indicated in Fig. 4(b). The support vectors act as input to the symbolic representation, where a search and formulation of improvement opportunities is done.

Formulating Improvement Opportunities.
Symbolic classification permits the partitioning of the input space into hyper-rectangular zones. A search through these zones allows for adjusting process variables to lie within ranges, which mostly result in desirable process performance or product quality. Based on the information in the support vectors identified earlier, a typical delineation of the input space is shown in Fig. 5. For example, if the indicated feed concentration of reactant A is in the range of [0.73, 0.97] mol/dm^3, then restricting the reactor temperature to the range $295\text{K} \leq T \leq 300\text{K}$ will reduce the dispersion of [B] values. The potency of the approach is especially important for high dimensional processes commonly encountered on process plants where such decisions may not be possible to visualise as in this 2D case.

Detection and removal of outliers. Sensor failures and other operational problems frequently result in incorrect measurement records. An essential characteristic of automated pattern-based systems is

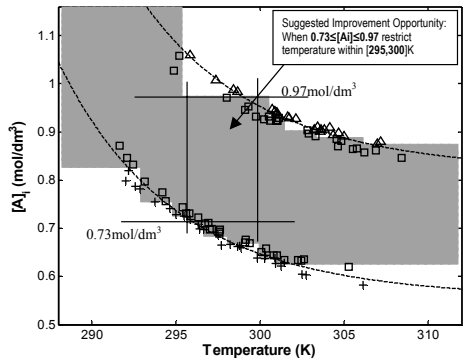

Fig 5. Formulation of Improvement Opportunities.

reliability and robustness to a few inconsistent data. The formulation of the SVM results in misclassified points or errors being included in the definition of the decision function. To use these errors in determining hyperrectangular zones unnecessarily complicates the decision process. Thus, the automated system preferably has to be able to identify the occurrence of a mismatch in a measurement.

The values of the slack variables in the solution of the soft-margin SVM formulation (6) can be used to identify and remove potential outliers. Patterns with $|\xi_i| \le 1$ lie within the margin and are correctly classified and therefore convey important information on the decision boundary. However, patterns with $1 < |\xi_i| < 2$ are incorrectly classified, but still within the margin. Patterns with $|\xi_i| > 2$ are totally embedded in a different class and are therefore immediate targets for elimination. The threshold value of ξ_i indicating the removal of patterns can be decided by the process experts familiar with the dynamics of the operation, providing an additional control on the decision support system (choice of kernel, kernel parameter, and regularisation constant being the others). For the purposes of this application, we consider such an approach more appropriate than just using the α_i values, a property which is often used.

Adaptive evolution of the memory of the support vectors. The nature of empirical learning is such that as new data becomes available over time, better estimates of the decision function are obtained. An online system must be able to learn continuously as data are collected, and make adjustments if necessary. These adjustments improve the smoothness of the current decision function or shift the decision function to a different decision space. In particular, a change in process parameters may shift the decision hyperplanes. In this case, it is important for the system to note and, therefore, exploit the information in the incoming data, rather than throw it away as outliers. In the proposed scheme, an efficient way to detect such changes is implemented using information on the update rate of the hyperplanes and the growth rate of the memory of support vectors.

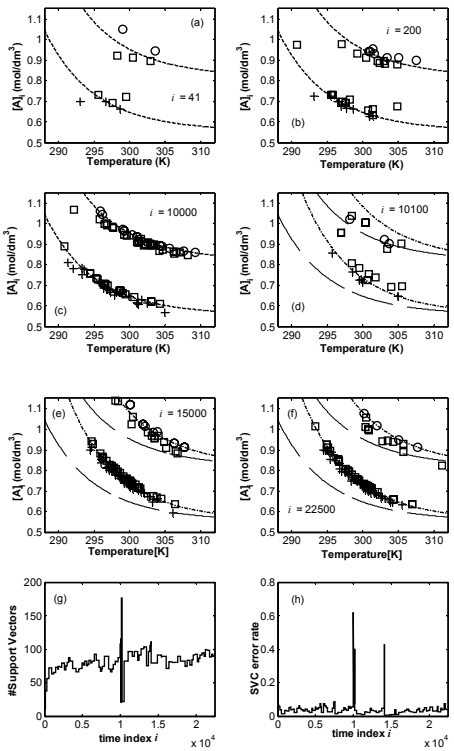

Fig 6: (a)-(f) Snapshots of the online evolution of support vectors at the indicated times. (g) Evolution of the size of the support vector memory. (h) Support vector classifier performance check using a test set of 200 taken from the "future".

3.2 Identification of Online Process Improvement Opportunities for a Simulated CSTR system.

To demonstrate the dynamic properties of the online decision support system for process improvement, we used data generated from the simulated CSTR described earlier. Additionally, a process shift was induced after 10 000 time units by changing the activation energy constant to 101.43 kJ/mol. This change shifts the decision boundaries upward.

Fig. 6(a)-(f) show support vectors at different times. The first 40 points were used in initializing the support vector classifier. The size of the support vector set increases rapidly initially, but gradually stabilises after approximately 200 time units. Interestingly, for most of the time all the information in the entire data set (before process drift) is captured by approximately 80 data points. As the rate of misclassification and the number of support vectors *abruptly* increase (as measured using an independent test set) after 10 000 time units, it is clear that the present support vector classifier is no longer consistent. To capture only relevant information, the learning methodology has to *unlearn* past information, retaining information within a specified time window. This reconfiguration takes a few steps, before stabilisation is achieved. Although in this

example final stabilisation occurs with more or less the same number of support vectors as before, this is an exception rather the rule. The number of support vectors is influenced by the form of the nonlinear function describing the hyperplane in the input space, which in this case happens to be similar for the different process parameters.
The rapid identification of process shifts has major implications on real plants, where the cost of nonconforming product could be high and recycle or rework expensive, such as in the pharmaceutical industries.

4. CONCLUSIONS

An online system for the formulation of process improvement opportunities was proposed, which uses parametric support vector classifiers in the selection of pivotal data points. Unlike traditional parametric methods, support vector classifiers use a bias from statistical learning theory, which maximises the generalisation capacity of the learning machine. Furthermore, the decision function is expressed in terms of a subset of the training points, thus inducing attractive nonparametric characteristics. The online decision support system for process improvement uses support vectors, appropriately screened for outliers, in the inductive learning of decision trees. Eventually, a modularised description of the input space is obtained that can be used in searching and formulating process improvement opportunities. These characteristics were illustrated using a simulated CSTR system in which a first-order irreversible reaction occurred.

REFERENCES

Bakshi, B.R. and G. Stephanopoulos (1994). Representation of Process Trends – IV. Induction of Real-Time Patterns from Operating Data for Diagnosis and Supervisory Control. *Computers chem. Engng*, 18, 303-332.

Breiman, L., J.H. Friedman, R.A. Olshen and C.J. Stone (1993). *Classification and Regression Trees*, Chapman & Hall, New York.

Burges, C.J.C. (1998). A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2, 121-167.

Cortes, C. and Vapnik, V. (1995). Support Vector Networks. *Machine Learning*, 20, 273-297.

Cristianini, N. and J. Shawe-Taylor (2000). *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods.* Cambridge University Press, Cambridge.

Deming, W. (1986). *Out of the Crisis.* MIT, Center for Advanced Engineering Study, Cambridge, Massachusetts.

Gunn, S.R., M. Brown, and K.M. Bossely (1997), Network Performance Assessment for Neurofuzzy Data Modelling. In: *Intelligent Data Analysis,* Vol. 1208, Lecture Notes in Computer Science (X. Liu, P. Cohen, and M. Berthold (Ed.)), 313-323.

Kresta, J.V., J.F. MacGregor and T.E. Marlin (1991). Multivariate Statistical Monitoring of Process Operating Performance. *Can. J Chem Engng*, 69, 35-47.

Quinlan, J.R. (1986). Induction of Decision Tress. *Machine Learning,* 1, 81-106.

Raich, A. and A. Çinar (1996). Statistical Process Monitoring and Disturbance Diagnosis in Multivariable Continuous Processes. *AIChE J.,* 42, 995-1009.

Saraiva, P.M., and G. Stephanopoulos (1992). Continuous Process Improvement through Inductive and Analogical Learning. *AIChE J.,* 38, 161-183.

Vapnik, V.N. (1998), *Statistical Learning Theory*, Wiley, New York.