Supplier selection based on hierarchical potential support vector machine

Xuesong Guo a,*, Zhiping Yuan b, Bojing Tian c

a School of Public Policy and Administration, Xi’an Jiaotong University, Xi’an, 710049, PR China
b School of Management, Xi’an Jiaotong University, Xi’an, 710049, PR China
c School of Architecture, Xi’an University of Architecture and Technology, Xi’an, 710055, PR China

ARTICLE INFO

Keywords:
Supplier selection
Potential support vector machine
Decision tree

ABSTRACT

Supplier selection is an important and widely studied topic since it has significant impact on purchasing management in supply chain. Recently, support vector machine has received much more attention from researchers, while studies on supplier selection based on it are few. In this paper, a new support vector machine technology, potential support vector machine, is introduced and then combined with decision tree to address issues on supplier selection including feature selection, multiclass classification and so on. So, hierarchical potential support vector machine and hierarchical system of features are put forward in the paper, and experiments show the proposed methodology has much better generalization performance and less computation consumptions than standard support vector machine.

© 2008 Elsevier Ltd. All rights reserved.

0. Introduction

Supplier selection is one of the most critical activities of purchasing management in supply chain, because of the key role of supplier’s performance on cost, quality, delivery and service in achieving the objectives of a supply chain. Supplier selection is a multiple criteria decision making (MCDM) problem which is affected by several conflicting factors. Consequently, a purchasing manager must analyze the trade off among the criteria. And MCDM techniques support the decision-makers in evaluating a set of alternatives.

Among the methods supporting supplier selection, artificial intelligence (AI) based models play important role in the domains. AI based on computer aided systems can be “trained” by a purchasing expert or historic data. Subsequently, non-experts facing similar but new situations can consult the system. Examples of methods based on AI technologies that have been applied to supplier choice include neural networks and other new techniques. One of the strengths of the methods is that they do not require formulation of the decision making process. In this respect, AI technologies can cope better with complexity and uncertainty than “traditional methods”, because they are designed to be more like human judgment functioning.

The user of AI systems only has to provide the information on characteristics of current situation, e.g. performance of a supplier on the criteria The AI technologies subsequently make the actual trade off of the users, based on what they have “learned” from the experts or cases in the past. A decision support system based on neural networks is put forward in Albino and Garavelli (1998). Moreover, other technologies based on AI also have been applied in domains of supplier selection (Khoo, Tor, & Lee, 1998; Cook, 1997; Ng & Skitmore, 1995). The support vector machine (SVM) method is a new and promising classification and regression technique. SVM, a development in statistical learning theory, is recently of increasing interests of researchers, though researches on issues of supplier selection are few (Sun, Xie, & Xue, 2005; Wen & Li, 2006). It is not only well founded theoretically, but also superior in practical applications. Moreover, SVM has been successfully applied in a wide variety of domains including handwriting recognition (Park & Woo Kim, 2005), speaker identification (Campbell, Campbell, & Gleason, 2007; Wan & Renals, 2005), face detection (Li & Tang, 2007), and text categorization (Lin & Peng, 2006). In most of these cases, the performance of SVM is either similar or significantly better than that of traditional machine learning approaches, including neural networks. Nevertheless, SVM has some problems to be solved when applied to practice. Firstly, SVM is a binary classifier while many multiclass classifications are required to be accomplished in practice. In the domains of supplier selection, suppliers to be evaluated will usually be divided into more than two categories according to given criteria.

Secondly, feature selection is required to be performed while applied to classification. Feature selection is an important issue in building classification systems. It is advantageous to limit the number of input features in a classifier in order to have a good predictive and less computationally intensive model (Zhang, 2000). With a small feature set, the explanation of rationale for the classification decision can be realized more easily. In terms of supplier selection, supplier can be described by some attributes originally, which can be represented with features in the view of machine learning approaches.
learning. And a subset of attributes by which suppliers are described should be selected from original ones so as to establish a system of criteria because some features are too noisy or not conveying correct information and can’t be used in subsequent evaluation procedures. In order to accomplish feature selection when standard SVM is applied, many complex algorithms, i.e., GA (Huang & Wang, 2006), chaos optimization etc., have to be used only to improve computation consumption.

In this paper, a novel SVM approach, potential support vector machine (P-SVM) (Hochreiter & Obermayer, 2006) which can accomplish binary classifiers construction and feature selection simultaneously, is introduced to the study at first. And method proposed in the paper combines P-SVM and decision tree into a new algorithm, called as hierarchical potential support vector machine, which can accomplish multiclass classification and feature selection simultaneously. Therefore, a new system of criteria represented as hierarchical structure, which can be applied to supplier description, established with the method employed to solve problems of standard SVM mentioned above is put forward in the paper.

1. Proposed methodology

1.1. Introductions to P-SVM

In this section, a new technology of SVM called as potential support vector machine (P-SVM), which can perform feature selection and classification simultaneously is introduced. The main differences to previous approaches are just as follows:

**Sphering:** In order to judge the relevance of feature components, the variance should be normalized, that is, the data should be sphered (whitened). Therefore, an objective is formulated according to which the classifier is selected by maximizing the margin after sphering. It turns out that sphering has two advantages for support vector machine techniques. Firstly, the derived new support vector machine approach is invariant to linear transformation of the data, which are the margin bounds. Secondly, tighter margin bounds can be obtained.

**New constraints:** The constraints of the optimization problem are modified in order to ensure that the classifier is optimal with respect to the mean squared error between the classification function and the labels. In contrast to previous approaches where one constraint is associated with each of the m training examples, each constraint is now associated with one feature and the number of new constraints is equal to the number N of features.

**Support features:** The combination of the new objective with the new constraints allows assigning support vector weights to features, and the normal vector of the classification boundary is expanded in terms of these weights rather than in terms of support vector data points. This allows feature selection according to whether a feature is a support vector or not. As a side effect the dual optimization problem can be efficiently solved using a technique similar to sequential minimal optimization.

In summary, a classifier is selected from the set of all classifiers with minimal mean squared error which yields the largest margin after sphering the data. The new support vector machine removes irrelevant features, which lead to a minimal increase of the mean squared error when removed. More formally, feature selection is done by assigning support vector weights to features, the features which are support vectors are selected.

In the following subsections, we first briefly review the standard support vector machine (SVM). Then, we introduce a new objective for achieving scale-invariant support vector machine, present new constraints for correct classification, and combine the new objective and the new constraints into one framework. Finally, a summary of the new technique is given.

1.1.1. Standard support vector machine

Consider a set of m objects, which are described by feature vectors \( \mathbf{x} \in \mathbb{R}^d \), and let us represent the data set by the matrix \( \mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_m] \). We furthermore assume that every object belongs to one of two classes and the class membership is denoted by a binary label \( y_i \in \{+1, -1\} \). The labels for the m objects are summarized by a label vector \( \mathbf{y} \), where \( y_i \) is the label for \( \mathbf{x}_i \).

The goal of standard support vector machine is to construct a linear classifier based on the feature vector \( \mathbf{x} \). In standard support vector machine, the classifier is defined by taking the signs of the classification function shown

\[
f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b.
\]

where the weight vector \( \mathbf{w} \) has been normalized such that the margin \( \rho \), that is the distance between the classification boundary and the closest data point, is \( \rho = \frac{1}{\|\mathbf{w}\|} \).

Standard support vector machine constructs a classification function which maximizes the margin under the constraint that the training data is classified correctly, just as shown

\[
\min_{\mathbf{w}, b} \frac{1}{2} \|\mathbf{w}\|^2
\]

\[
y_i (\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1.
\]

Here, we assume that the data \( \mathbf{X} \) with label vector \( \mathbf{y} \) is linearly separable, otherwise slack variables have to be introduced. If the number of training samples \( m \) is larger than the VC dimensions \( h \), then one obtains the following bound on the generalization error \( R(f) \) of \( f \), just as shown

\[
R(f) \leq R_{emp}(f) + \frac{1}{m} h \left( \ln \left( \frac{2m}{\delta} \right) + 1 \right) \left( \ln \left( \frac{h \delta}{4} \right) \right)
\]

which holds with probability \( 1 - \delta \). \( \delta \) denotes the probability, that a training set \( \mathbf{X} \) of size \( m \) has been randomly drawn from the underlying distribution, for which the bound equation does not hold. \( R_{emp}(f) \) denotes the training error of \( f \) (also called “empirical risk of \( f \)”). For the set of all linear classifiers defined on \( \mathbf{X} \), for which \( \rho \geq \rho_{min} \) holds, one obtain

\[
h \leq \min \left\{ \frac{R^2}{\rho^2_{min}}, N \right\} + 1,
\]

where \( [\cdot] \) denotes the integer part and \( R \) is the radius of the smallest sphere in the data space, which contains all the training data. The fact that the bounds become smaller for increasing \( \rho \) and decreasing \( N \) motivates the maximum margin principle and the concept of feature selection (Vapnik, 1995, 1998).

1.1.2. A scale invariant objective function

Both the selection of a classifier using the maximum margin principle and the values obtained for the bounds on the generalization error described in the last section suffer from the problem addressed in Hochreiter and Obermayer (2006); Scholkopf, Shawe-Taylor, and Smola (1999) that they are not invariant under linear transformations.

Here, we suggest scale the training data such that the margin \( \rho \) remains constant while the radius \( R \) of the sphere containing all training data becomes as small as possible. The scaling results derive a new sphere with radius \( R \) which still contains all training data and which leads to a tight margin-based bound for the generalization error. Optimality is achieved when all directions orthogonal the normal vector \( \mathbf{w} \) is scaled to zero and \( R = \min_{\mathbf{x} \in \mathbb{R}^d} \max_{\mathbf{x} \neq \mathbf{x}_i} |\mathbf{w} \cdot \mathbf{x}_i + b| \). Note that with offset \( b \) the classification function the sphere must not be centered at the origin. Unfortunately, above formulation does not lead to an optimization problem. In this paper, the form of data points input to P-SVM is illustrated in Fig. 1.
So, a set of data points illustrated in Fig. 1 can be represented as matrix $X$.

$$X = \begin{bmatrix} x_{11} & x_{21} & x_{31} & \cdots & x_{i1} \\ x_{12} & x_{22} & x_{32} & \cdots & x_{i2} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{1j} & x_{2j} & x_{3j} & \cdots & x_{ij} \end{bmatrix},$$  \hspace{1cm} (6)

where $i$ and $j$ in $x_{ij}$ represent data point $i$ and its feature $j$, respectively. Therefore, we suggest minimize the upper bound

$$R^2_{\text{emp}} = R^2 \|w\|^2 \leq \max \|w \cdot x_i\|^2 \leq \sum ||w \cdot x_i||^2 = ||X'w||^2.$$  \hspace{1cm} (7)

In Hochreiter and Obermayer (2003b), it is shown that replacing the objective in Eq. (7) by the upper bound

$$w'XX'w = ||X'w||^2.$$  \hspace{1cm} (8)

The new objective leads to separating hyperplanes which are invariant to transformations of the data. Consequently, the bounds and the performance of the derived classifier no longer depend on scale factors. Note the kernel trick carries over to the P-SVM as shown in Hochreiter and Obermayer (2003b). The P-SVM can also be applied for kernels which are not positive definite, that is, which are not Mercer kernels (Hochreiter & Obermayer, 2002).

### 1.1.3. New constraints

To assign support vector weights to the feature components, the constraints enforcing correct classification have to be transformed into $N$ constraints associated with the features. The idea of the transformation is to compute the correlation between the residual error and a feature component. If these correlations are zero, the empirical risk is minimal.

Thus, we define a residual error $r_i$ for a data point $x_i$ as the difference between its label class $y_i$ and the value of the classification function $f$, where

$$f(x) = w \cdot x + b.$$  \hspace{1cm} (9)

$$r_i = w \cdot x_i + b - y_i.$$  \hspace{1cm} (10)

In terms of every feature component $j$, we then compute the mixed moments $\sigma_j$

$$\sigma_j = \frac{1}{m} \sum_{i=1}^{m} c(x_i, y_i, f(x_i)),$$  \hspace{1cm} (11)

between the residual $r_i$ and the measured values $(x_i)$. These mixed moments $\sigma_j$ should be made small or zero. The rationale behind minimal values for $\sigma_j$ is that, given quadratic loss function, they lead to an optimal classifier.

Consider the quadratic loss function as

$$c(x_i, y_i, f(x_i)) = \frac{1}{2} r_i^2$$  \hspace{1cm} (12)

and the empirical loss (the mean squared error) is shown as

$$R_{\text{emp}}(f) = \frac{1}{m} \sum_{i=1}^{m} c(x_i, y_i, f(x_i)).$$  \hspace{1cm} (13)

The mixed moments $\sigma_j$ are equal to the derivative of the empirical loss with respect to $w_j$

$$\sigma_j = \frac{\partial R_{\text{emp}}(f)}{\partial w_j}.$$  \hspace{1cm} (14)

That is the empirical error is minimal if

$$\sigma_j = \frac{1}{m} \sum_i (x_{ij}) r_i = 0.$$  \hspace{1cm} (15)

Note that there exists only one minimum since the squared error is a convex function is the parameters $w$. And these considerations motivate a new set of constraints shown as follows:

$$X_i = X(X'w + b1 - y) = 0.$$  \hspace{1cm} (16)

which an optimal classifier must fulfill, because

$$X_i = \sum_{j=1}^{m} (x_{ij}) r_j = m \sigma_j = m \frac{\partial R_{\text{emp}}(f)}{\partial w_j}.$$  \hspace{1cm} (17)

However, measurement noise may lead to high value of $\sigma_j$ which, then minimized, will lead to strong over fitting. Therefore, we introduce a “correlation threshold” $\varepsilon$ which separates the signal part, and we modify the constraints according to

$$\begin{cases} X_1 \varepsilon + b1 - y - \varepsilon^+ & \leq 0 \\ X_1 \varepsilon + b1 - y + \varepsilon^- & \geq 0 \\ \varepsilon^+, \varepsilon^- & \geq 0. \end{cases}$$  \hspace{1cm} (18)

### 1.1.4. Potential support vector machine

The new objective from Eq. (8) and the new constraints shown as Eq. (18) can be combined. And the procedures of selecting a classifier are called potential support vector machine. The combination of new objective and new constraints leads to an expansion of the normal vector of the classification boundary into a sparse set of features, hence allows expressing the relevance of feature via support vector weights.

$$\min_{w, b} \frac{1}{2} \|X'w\|^2 + C \frac{1}{2} (\varepsilon^+ + \varepsilon^-)$$  \hspace{1cm} (19)

Subject to

$$\begin{cases} X_1 \varepsilon + b1 - y - \varepsilon^+ & \leq 0 \\ X_1 \varepsilon + b1 - y + \varepsilon^- & \geq 0, \varepsilon^+ , \varepsilon^- \geq 0. \end{cases}$$  \hspace{1cm} (20)

If the row vectors of $X$ are normalized to mean zero, then $X_1 = 0$ and the term $b$ vanishes. The parameters $\varepsilon^+$ and $\varepsilon^-$ serve as two important purposes.

Large values of $\varepsilon$ lead to a sparse expansion of the weight vector through the support features.

If $XX'$ is singular and $w$ is not uniquely determined, $\varepsilon$ enforces a unique solution, which is characterized by the sparsest representation through features.

The interpretation of $\varepsilon$ as a sparseness property is known from support vector regression.

Optimization is usually performed using the Wolfe dual of Eqs. (19) and (20).

$$L = \frac{1}{2} w'XX'w + C \frac{1}{2} (\varepsilon^+ + \varepsilon^-) - (\varepsilon^+)\frac{1}{2} (X'XX'w + b1 - y)$$
$$\varepsilon^+) + (\varepsilon^-)\frac{1}{2} (X'XX'w + b1 - y) - \varepsilon^-) - (\varepsilon^-)\varepsilon^+$$
$$- (\varepsilon^+)\varepsilon^-$$  \hspace{1cm} (21)

And the optimal conditions are shown
\[
\begin{align*}
\nabla_w \hat{f} &= XX^t w - XX^t (x^+ - x^-) = 0, \\
\nabla_b &= 1^t X (x^+ - x^-) = 0.
\end{align*}
\]

where the quantities \( z = x^+ - x^- \) denote the Lagrange parameters. \( 1^t X (x^+ - x^-) = 0 \) is automatically satisfied, if the row vectors of \( X \) are normalized to zero mean. The dual problem is solved by a Sequential Minimal Optimization technique. A fast solver for P-SVM is shown in Hochreiter and Obermayer (2006), where the advantages can be taken of the fact that the equality constraint vanishes for zero mean row vectors of \( X \).

Finally, the classification function \( f \) has to be constructed using the optimal values of the Lagrange parameters \( z \).

\[
w = z.
\]

In contrast to standard support vector machine expansion of \( w \) by its support vectors, the weight vector \( w \) is expanded into a sparse set of features components which serve as the support vectors in this case. The value of \( b \) can be derived from the condition that the average residual error \( r \) is equal to zero.

\[
b = -\frac{1}{m} \sum_{i=1}^{m} (w \cdot x_i - y_i).
\]

(24)

Note that \( b \) is chosen so that

\[
\frac{\partial \text{R}_{\text{emp}}(f)}{\partial b} = \frac{1}{m} \sum_{i=1}^{m} r_i = b + \frac{1}{m} \sum_{i=1}^{m} (w \cdot x_i - y_i) = 0.
\]

(25)

That means \( b \) takes on its optimal value of the empirical risk (as was ensured for \( w \) through its constraints). If the row vectors of \( X \) are normalized to zero, we obtain

\[
b = \frac{1}{m} \sum_{i=1}^{m} y_i.
\]

(26)

And the classification function is then given by

\[
f(v) = \text{sign}(w \cdot v + b) = \text{sign} \left( \sum_{j=1}^{n} z_j (v \cdot e_j) + b \right)
\]

(27)

The classifier based on Eq. (27) depends on the weighting coefficients \( z_j \) and \( b \), which was determined during optimization, and the measured values \( v \) of the selected features for the object to be classified. The weighting coefficients \( z_j \) can be interpreted as class indicators, because they separate the features into the ones which are relevant for class +1 and -1, according to the sign of \( z_j = x_{j+} - x_{j-} \). If the value of \( x \) is large enough during learning, the expansion of the weight vector contains only a few “most informative” features, hence most of the components of \( w \) are zero. The other features are discarded because of being too noisy or not conveying information about the class labels.

The comparison between support vector in standard SVM and support feature in P-SVM is illustrated in Fig. 2.

In Fig. 2a, the grey columns represent support vectors selected to construct classification function in standard SVM, while the grey rows marked with dashed lines in Fig. 2b represent support features selected to construct classification function in P-SVM. In terms of P-SVM, the features (the rows which are white in Fig. 2b) not selected as support features are not considered as the most “informative” ones, and will not used to construct classifier. Therefore, feature selection is accomplished automatically with selection of optimal classifier by exchanging the roles of features and data points.

Application of Kernel function in P-SVM is similar to that of standard SVM, so it will not be addressed in details.

1.2. Hierarchical structure for multiclass classification

For many problems of classification where large datasets are used and the information contained is complex and may contain errors, decision tree provides a useful solution. A definition of decision tree was given in Russell et al. (2002). Decision trees are rapid and effective methods of classifying data set entries, and can provide good decision support capabilities. In past few years, applications of classification based on decision trees have predominated in various domains (Levow, 1998; Markey, Tourassi, & Floyd, 2003; Zmazek, Todorovski, & Dz eroski, 2003). Moreover, the hierarchical structures represented as decision trees have become an alternative multiclass problem solution with binary predictors including standard support vector machine. In a structure of decision tree, these structures are composed of nodes and ramifications. Internal nodes correspond to binary classifiers; the ramifications are the outputs of these classifiers and the leaf nodes represent the problem classes. To obtain a pattern class, the nodes and ramifications are followed accordingly until a leaf is reached. This is the reconstruction scheme usually employed with hierarchical approaches.

Generally, it can be stated that the hierarchical approaches have lower prediction times than the code-matrix ones. This occurs because, in a pattern classification, only a part of the binary predictors in the hierarchical structure are consulted, while the decoding of code-matrix approaches requires the evaluation of all binary classifier outputs. To define the binary partitions of classes in the hierarchy, which is equivalent to decompose the multiclass problem, several strategies can be employed.

In Platt, Cristian, and Shawe-Taylor (2000), it is suggested that the classifiers produced by all-against-all (AAA) (Krebell, 1999) decomposition are displaced in a decision directed acyclic graph (DDAG). Thus, each node of the graph corresponds to a binary predictor for a pair of classes. In the classification of a new pattern with this structure, \( k - 1 \) binary classifiers are evaluated. A disadvantage of the DDAG, pointed in Kjissirikul and Ussivakul (2002), is that different configurations of the binary classifiers in the graph influence the results obtained. Other deficiency of DDAG pointed in Kjissirikul and Ussivakul (2002) is that, depending of the position of the correct class in the graph and the number of evaluations with it may be unnecessarily high, resulting in a high cumulative
error. For instance, if the correct class of a pattern is in the root node, for example, it will be tested against others \( k - 1 \) times. If there is a 1% probability of error in each node, this will cause cumulative error rate of \( 1 - 0.99^{k-1} \).

These facts motivated Kijsirikul and Usivakul (2002) to develop a new hierarchical structure to combine the outputs of binary classifiers. The new structure, named adaptive DAG (ADAG), corresponds to a DAG with reverse structure. The ADAG has \( k - 1 \) nodes and each one of them corresponds to a binary classifier for a pair of classes. The first layer has \( \frac{k}{2} \) nodes, followed by \( \frac{k}{4} \) nodes in the second layer and so on, until a layer with one unique node is reached, giving the final classification. To classify a new pattern using the ADAG, first it is evaluated by the binary classifiers in the first level, which distinguish pairs including all classes. Each binary predictor produces as output its preferred class. Based on the choices of previous levels, the classifiers of the next levels are determined. By this process, at each level the number of candidate classes is reduced by half. If the number of classes in a level is odd, one of the nodes in this layer contains one unique class, which is directly passed to the next level. Like in the DDAG, the ADAG requires \( k - 1 \) node evaluations for each prediction. However, the correct class is tested against others \( \lceil \log_2 k \rceil \) times or less. The ADAG thus minimizes the occurrence of cumulative errors. Although the ADAGs have also demonstrated less dependency in relation to the order of binary classifiers in the graph, there are still results disparities between distinct structures. Another common type of hierarchical structure is a tree, in which, apart from the root, each node presents one unique parent. The tree requires the training of \( k - 1 \) binary classifiers, the lowest number among all decomposition approaches already mentioned. In the test phase, in the best case, it is possible to classify a new pattern in the first node. In the worst case, the \( k - 1 \) classifiers have to be consulted. Thus, the number of evaluated predictors in each classification is at least 1 and at most \( k - 1 \). Like in the DDAG and ADAG, the tree structure influences its results. The works that use this type of structure differentiate on how binary partitions of classes in each node of the tree are defined and, consequently, on its structure determination. Generally a specific criterion is applied recursively to subsets of classes, partitioning them into two until one unique class remains.

In Takahashi and Abe (2002), it is proposed that nodes in superior levels of the tree divide those classes that are more separable. The argument is that if superior nodes produce fewer errors, the final result can be improved. The authors proposed four methods to obtain the trees. One of them recursively separates the classes according to their centroid distances.

In Schwenker and Palm (2001), the authors used the concept of confusion classes to define the binary partitions of the classes in the tree. A confusion class is a subset of the classes for which there is a high similarity between their data samples. To define these sets, the alternative employed was to recursively use the \( k \)-means clustering algorithm (Mitchell, 1997), with \( k \) equaling to 2.

In Vural (2004), three methods were suggested to define the binary partitions of classes. The first one consists of grouping classes more similar in each level of the tree with the algorithm \( k \) means, adopting \( k \) equal to 2. In the second, the classes are divided according to the mean distance of their data samples to the origin. In the third method, the classes are divided into subsets for which the difference in the number of examples is minimal. Based on some concepts introduced by these works, in Lorena and Carvalho (2005) Lorena and Carvalho proposed an alternative algorithm based on graphs and minimum spanning trees to obtain the hierarchical structure from information collected from data. Given an undirected graph \( G = (V, E) \) with \( |V| \) vertices, \( |E| \) edges and a cost or weight associated to each edge, a minimum spanning tree (MST) \( T \) is a connected acyclic subgraph that spans all vertices of \( G \) with the smallest total cost of edges (Ahuja, Magnanti, & Orlin, 1993).

In the proposed algorithm, initially, information collected from the training dataset is used to obtain a weighted graph, which has \( k \) vertices and \( k \times (k - 1)/2 \) edges connecting all pairs of vertices. A MST is then extracted from this graph. The MST algorithm iteratively groups subsets of classes that have low weight connections in the graph. Taking advantage of this characteristic, the multiclass tree partitions are defined.

To build the weighted graph, various methods can be used to assign costs to the edges. In this work, the following approaches were investigated:

(1) Centroid distances: Each class is first represented by a centroid \( u_i \). The weight of an arc \((i, j)\) is then given by the Euclidean distance between \( u_i \) and \( u_j \). Using this criterion, the MST will group in each level of the hierarchical subsets of one or more classes that are similar to each their centroids.

(2) Balanced subsets: Inspired by ideas presented in Vural (2004), this criterion acts by grouping that have similar data distribution. The weight of an arc \((i, j)\) is then given by the difference among the number of patterns from classes \( i \) and \( j \).

(3) Confusion matrix: Given the concept of confusion classes from (Schwenker & Palm, 2001), a confusion matrix can be employed in the graph weights definition. The confusion matrix offers an idea of which classes a classifier has more difficulty to distinguish. For a dataset with \( k \) classes, it has \( k \times k \) dimension, and each element \( c_{ij} \) represents the number of examples from class \( i \) that were misclassified as belonging to the class \( j \) (Mitchell, 1997). To obtain this matrix, the whole \( k \) class problem has to be solved first. In this research, DTs were used in the confusion matrix generation. The weight of an arc \((i, j)\) is then calculated by Eq. (28), where \( n_i \) is the number of examples from class \( i \). Applied to these weights, the MST will group classes that present less confusion with each other.

\[
d_{CM}(i, j) = \frac{c_{ij}}{n_i} + \frac{c_{ij}}{n_j}.
\] (28)

Given the obtained weighted graph, an adapted version of the Kruskal algorithm for finding a MST (Ahuja et al., 1993) was applied in the multiclass tree determination. The proposed algorithm uses the defined groups to determine the binary partitions in the tree formation process. A pseudocode of this algorithm is shown in Appendix, originally presented in Lorena and Carvalho (2005). The generation of the tree structure operates in a bottom–up iterative way. The given algorithm is efficient and allows a totally automatic determination of the multiclass tree structure using binary predictors.

1.3. Hierarchical P-SVM algorithm

The problems confronted by standard SVM in domains of supplier selection can be solved by a new method, called as hierarchical P-SVM, which combines P-SVM and decision tree based on adapted Kruskal algorithm. And a hierarchical system of features is derived with the new method, whose procedures can be illustrated by the following steps:

Step 1. Generate decision tree DT by adapted Kruskal algorithm based on training samples.
Step 2. Beginning with root of DT, every node except leaves implies a binary classification problem. So, accomplish binary classifiers construction and support feature selection from original ones based on P-SVM at related nodes until last leaf of DT is reached.

Step 3. When new samples arrive, just input the values to binary classifiers based on P-SVM according to features selected at related nodes of DT in step 2, and accomplish supplier evaluation according to the thoughts of decision tree.

So, following the procedures proposed in the subsection, a subset of original features will be obtained as support features at each node except leaves by using P-SVM. Then hierarchical features can be obtained by using hierarchical P-SVM. And structure of hierarchical P-SVM is illustrated in Fig. 3.

In the hierarchical structure shown in Fig. 3, binary classifiers based on P-SVM are constructed at nodes including A, B and C, with subsets of features selected. For example, samples belonging to class 2 are discriminated further at node B and C until every sample is assigned to correct class. So, at every node, new samples only have to be described by a subset of original features at every node. Moreover, there will be some subsets of features obtained by P-SVM to form a hierarchical system. For example, when samples from class 2 are separated from those belonging to class 1, 3 and 4 which are classified further at node B and C until every sample is assigned to correct classes. So, at every node, new samples only have to be described by a subset of original features at every node. Moreover, there will be some subsets of features obtained by P-SVM to form a hierarchical system. For example, when samples from class 2 are separated from those belonging to class 1, 3 and 4, they should be described with subset of features marked with A. However, when samples from class 3 are separated from those from class 1 and 4, they should be described with subset of features marked with B. Therefore, subsets of features marked with A, B and C, which may be different from each other, form a hierarchical system of features which can be used to describe samples to be classified at related nodes.

2. Supplier selection using proposed methodology

2.1. Original criteria applied to supplier selection

Issues on supplier selection criteria have been addressed in much literature (Dickson, 1966; Abratt, 1986; Billesbach, Harrison, & Croom-Morgan, 1991; Lehmann & Shaughnessy, 1974; Min & Galle, 1999; Perreault & Russ, 1976; Weber, Current, & Benton, 1991; Wind, Green, & Robinson, 1968; Stavropolous, 2000), on which a set of criteria has been derived as original features, just as listed in Table 1.

2.2. Ranks of supplier selection

All variables used in Table 1 are used as the factors of supplier selection. Performance of supplier is measured on a seven-point scale, just as listed in Table 2.

3. Experiments and results

The samples from China which have been addressed in Song (2001) are used to test the proposed methodology. Therefore, 64 training samples described with original features listed in Table 2 are employed in the experiment. With training samples input to model based on hierarchical P-SVM, decision tree based on adapted Kruskal algorithm is generated and binary classifiers are constructed at the nodes to accomplish supplier evaluation. The implementation was carried out on Matlab 6.5 development environment by extending the toolbox on P-SVM originally designed by Hochreiter and Obermayer (xxxx, xxxx). The empirical evaluation was performed on Intel Pentium IV CPU running at 2.6 GHz and 512 MB RAM. The hierarchical structure derived by the method is illustrated in Fig. 4.

Just as shown in Fig. 4, there are only 6 binary classifiers marked with A, B, C, D, E and F to be trained to establish the system of supplier selection, while standard SVM with one-against-one strategy (Collobert & Bengio, 2001; Krebel, 1999), which has been proved to one of the most efficient and effective multiclass classification strategies of standard SVM and applied to supplier selection (Sun et al., 2005), has to train 21 binary classifiers to solve the same multiclass classification problem.

Numbers of selected features at every node are listed in Table 3, and average numbers of selected features and discarded ones by hierarchical P-SVM are illustrated in Fig. 5, respectively.

In the process of feature selection based on P-SVM, the most “informative” features are selected, while the ones which are too
Table 3
Number of selected features at every node

<table>
<thead>
<tr>
<th>Binary classifier</th>
<th>Number of selected features (support features)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>12</td>
</tr>
<tr>
<td>B</td>
<td>14</td>
</tr>
<tr>
<td>C</td>
<td>17</td>
</tr>
<tr>
<td>D</td>
<td>20</td>
</tr>
<tr>
<td>E</td>
<td>19</td>
</tr>
<tr>
<td>F</td>
<td>21</td>
</tr>
</tbody>
</table>

Fig. 5. Average number of selected features by hierarchical P-SVM.

Fig. 6. Generalization performance of hierarchical P-SVM and standard SVM.

References


