Chapter 5
Knowledge-incorporated Multiple Criteria Linear Programming Classifiers

Classification is a main data mining task, which aims at predicting the class label of new input data on the basis of a set of pre-classified samples. Multiple Criteria Linear Programming (MCLP) is used as a classification method in data mining area, which can separate two or more classes by finding discriminate hyperplane. Although MCLP shows good performance in dealing with linear separable data, it is no longer applicable when facing with nonlinear separable problem. Kernel-based Multiple Criteria Linear Programming (KMCLP) model is developed to solve nonlinear separable problem. In this method, kernel function is introduced to project the data into a higher-dimensional space in which the data will have more chance to be linear separable. KMCLP performs well in some real applications. However, just as other prevalent data mining classifiers, MCLP and KMCLP learn only from training examples. In traditional machine learning area, there are also classification tasks in which data sets are classified only by prior knowledge, i.e. expert system. Some works combine the above two classification principle to overcome the defaults of each approach. In this section, we combine the prior knowledge and MCLP or KMCLP model to solve the problem when input consists of not only training example, but also prior knowledge.

5.1 Introduction

Multiple Criteria Linear Programming (MCLP) is used as a classification method which is based on a set of classified training examples (Kou et al. 2003). By solving a linear programming problem, MCLP can find a hyperplane to separate two classes. The principle of MCLP classifier is to train on the training set then get some separation model that can be used to predict the label of the new data. However, MCLP model is only applicable for linear separable data. To facilitate its application on nonlinear separable data set, kernel-based multiple criteria linear programming (KMCLP) method was proposed by Zhang et al. (2009), which introduces kernel function into the original MCLP model to make it possible to solve nonlinear separable problem. Likewise, there are also many other prevalent classifiers, such

Y. Shi et al., *Intelligent Knowledge,* SpringerBriefs in Business,
DOI 10.1007/978-3-662-46193-8_5

as Support Vector Machine, Neural Networks, Decision Tree etc., which share the same principle of learning solely from training examples. This inevitably can bring out some disadvantages. One problem is that noisy points may lead to poor result. The other more important one is that when training samples are hard to get or when sampling is costly, these methods will be inapplicable.

Different from the above empirical classification methods, another commonly used method in some area to classify the data is to use prior knowledge as the classification principle. Two well-known traditional methods are Rule-Based reasoning and Expert System. In these methods, prior knowledge can take the form of logical rule which is well recognized by computer. However, these methods also suffer from the fact that pre-existing knowledge cannot contain imperfections (Towell et al. 1990). Whereas, as is known to all, most of the knowledge is tacit in that it exists in people's mind. Thus, it is not an easy task to acquire perfect knowledge.

Recent works combine the above two classification principles to overcome the defaults of each approach. Prior knowledge can be used to aid the training set to improve the classification ability; also training example can be used to refine prior knowledge. In such combination methods, Knowledge-Based Artificial Neural Networks (KBANN) and Knowledge-Based Support Vector Machine (KBSVM) are two representatives. KBANN is a hybrid learning system which firstly inserts a set of hand-constructed, symbolic rules into a neural network. The network is then refined using standard neural learning algorithms and a set of classified training examples. The refined network can function as a highly-accurate classifier (Towell and Shavlik 1994). KBSVM provides a novel approach to incorporate prior knowledge into the original support vector classifier. Prior knowledge in the form of polyhedral knowledge sets in the input space of the given data can be expressed into logical implications. By using a mathematical programming theorem, these logical implications can work as a set of constraints in support vector machine formulation. It is also a hybrid formulation capable of generating a classifier based on training data and prior knowledge (Fung et al. 2002; Mangasarian 2005). Some works are focused on incorporating nonlinear knowledge into nonlinear kernel classification problem (Mangasarian and Wild 2008), because nonlinear prior knowledge is more general in practical application. In addition to the application in classification problem, (Mangasarian et al. 2004) has shown the effectiveness of introduce prior knowledge into function approximation.

In this chapter, we summarize the relevant works which combine the prior knowledge and MCLP or KMCLP model. Such works can extend the application of MCLP or KMCLP model to the cases where prior knowledge is available. Specifically, knowledge-incorporated MCLP model deals with linear knowledge and linear separable classification problem. The prior knowledge in the form of polyhedral knowledge sets can be expressed into logical implications, which can further be converted into a series of equalities and inequalities. Incorporating such kind of constraints to original MCLP model, we then obtain the final knowledge-incorporated MCLP model. It is supposed to be necessary and possible that KMCLP model make better use of knowledge to achieve better outcomes in classifying nonlinear separable data. Linear knowledge can also be introduced into kernel-based MCLP

model by transforming the logical implication into the expression with kernel. With this approach, nonlinear separable data with linear knowledge can be easily classified. Concerning the nonlinear prior knowledge, by writing the knowledge into logical expression, the nonlinear knowledge can be added as constraints to the kernel-based MCLP model. It then helps to find the best discriminate hyperplane of the two classes. Numerical tests on the above models indicate that they are effective in classifying data with prior knowledge.

5.2 MCLP and KMCLP Classifiers

5.2.1 MCLP

Multiple criteria linear programming (MCLP) is a classification method (Olson and Shi 2007). Classification is a main data mining task. Its principle is to use the existing data to learn some useful knowledge that can predict the class label of other unclassified data. The purpose of classification problem can be described as follows:

Suppose the training set of the classification problem is $\boldsymbol{X}$, which has n observations in it. Of each observation, there are r attributes (or variables) which can be any real value and a two-value class label G (Good) or B (Bad). Of the training set, the *ith* observation can be described by $\boldsymbol{X}_i=(X_i1,\ldots, X_{ir})$, where i can be any number from 1 to n. The objective of the classification problem is to learn from the training set and get a classification model that can classify these two classes, so that when given an unclassified sample $z=(z_1,\ldots, z_r)$, we can predict its class label with the model.

So far, many classification methods have been developed and widely used in data mining area. Specifically, MCLP is an efficient optimization-based method in solving classification problem. The framework of MCLP is based on the linear discriminate analysis models. In linear discriminate analysis, the purpose is to determine the optimal coefficients (or weights) for the attributes, denoted by $W=(w_1, \ldots, w_r)$ and a boundary value (scalar) b to separate two predetermined classes: G (Good) and B (Bad); that is

$$\begin{aligned} X_{i1}w_1+\cdots+X_{ir}w_r \le b, X_i \in B(Bad) \\ and \;\; X_{i1}w_1+\cdots+X_{ir}w_r \ge b, X_i \in G(Good) \end{aligned} \tag{5.1}$$

To formulate the criteria and constraints for data separation, some variables need to be introduced. In the classification problem, $X_i w = X_{i1} w_1 + \cdots + X_{ir} w_r$ is the score for the *ith* observation. If all records are linear separable and a sample $\boldsymbol{X}_i$ is correctly classified, then let β_i be the distance from $\boldsymbol{X}_i$ to b, and consider the linear system, $X_i w = b + \beta_i$, $\forall X_i \in G$ and $X_i w = b - \beta_i$, $\forall X_i \in B$. However, if we consider the case where the two groups are not linear separable because of mislabeled

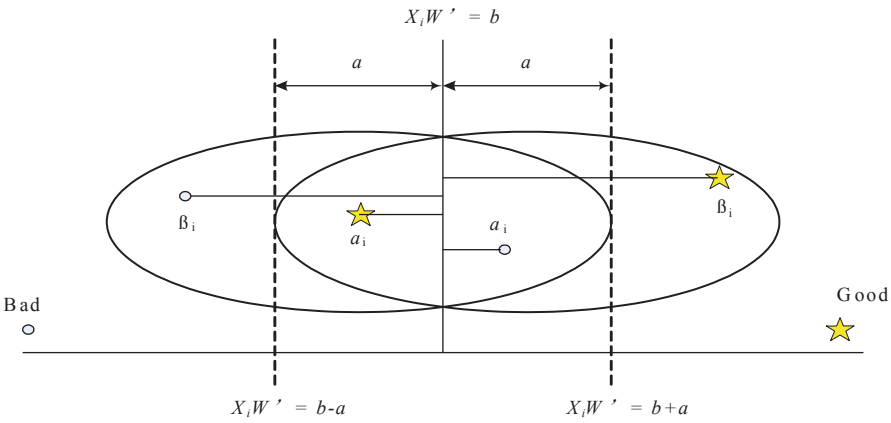

Fig. 5.1 Overlapping of two-class Linear Discriminate Analysis

records, a "soft margin" and slack distance variable α_i need to be introduced. α_i is defined to be the overlapping of the two-class boundary for mislabeled case X_i. Previous equations now can be transformed to $X_i w = b - \alpha_i + \beta_i,\ \forall X_i \in G$ and $X_i w = b + \alpha_i - \beta_i,\ \forall X_i \in B$. To complete the definitions of β_i and α_i, let $\beta_i = 0$ for all misclassified samples and $\alpha_i = 0$ for all correctly classified samples. Figure 5.1 shows all the above denotations in two-class discriminate problem.

A key idea in linear discriminate classification is that the misclassification of data can be reduced by using two objectives in a linear system. One is to maximize the minimum distances (MMD) of data records from a critical value and another is to separate the data records by minimizing the sum of the deviations (MSD) of the data from the critical value. In the following we give the two basic formulations of MMD and MSD (Olson and Shi 2007):

MSD

$$
\begin{aligned}
&\text{Minimize } \alpha_1 + \ldots + \alpha_n \\
&\text{Subject to:} \\
&\quad X_{11} w_1 + \ldots + X_{1r} w_r = b + \alpha_1, \quad for\ \ X_1 \in B, \\
&\quad\quad \ldots \\
&\quad X_{n1} w_1 + \ldots + X_{nr} w_r = b - \alpha_n, \quad for\ \ X_n \in G, \\
&\quad \alpha_i \geq 0, \quad i = 1, \ldots, n, \\
&\quad w \in R^r.
\end{aligned}
\tag{5.2}
$$

MMD

$$\begin{aligned}
&\text{Minimize } \beta_1 + ... + \beta_n \\
&\text{Subject to:} \\
&\quad X_{11}w_1 + ... + X_{1r}w_r = b - \beta_1, \quad for \;\; X_1 \in B, \\
&\quad\quad ... \\
&\quad X_{n1}w_1 + ... + X_{nr}w_r = b + \beta_n, \quad for \;\; X_n \in G, \\
&\quad \beta_i \geq 0, \quad i = 1, ..., n, \\
&\quad w \in R^r.
\end{aligned} \tag{5.3}$$

Instead of maximizing the minimizing distances of data records from a boundary b or minimizing the sum of the deviations of the data from b in linear discriminate analysis models, MCLP classification considers all of the scenarios of tradeoffs and finds a compromise solution. So, to find the compromise solution of the two linear discriminate analysis models MMD and MSD for data separation, MCLP wants to minimize the sum of α_i and maximize the sum of β_i simultaneously, as follows:

Two-Class MCLP model (Olson and Shi 2007):

$$\begin{aligned}
&\text{Minimize } \alpha_1 + \cdots + \alpha_n \quad and \quad \text{Maximize } \beta_1 + \cdots + \beta_n \\
&\text{Subject to :} \\
&\quad X_{11}w_1 + \cdots + X_{1r}w_r = b + \alpha_1 - \beta_1, \quad for \;\; X_1 \in B \\
&\quad\quad\quad\quad \cdots\cdots \\
&\quad X_{n1}w_1 + \cdots + X_{nr}w_r = b - \alpha_n + \beta_n, \; for \;\; X_n \in G \\
&\quad \alpha_1, \cdots, \alpha_n \geq 0, \qquad \beta_1, \cdots, \beta_n \geq 0
\end{aligned} \tag{5.4}$$

To facilitate the computation, a compromise solution approach (Olson and Shi 2007) has been employed to modify the above model so that we can systematically identify the best trade-off between $-\sum\alpha_i$ and $\sum\beta_i$ for an optimal solution. The "ideal value" of $-\sum\alpha_i$ and $\sum\beta_i$ are assumed to be $\alpha^* > 0$ and $\beta^* > 0$ respectively. Then, if $-\sum\alpha_i > \alpha^*$, we define the regret measure as $-d_\alpha^+ = \sum\alpha_i + \alpha^*$; otherwise, it is 0. If $-\sum\alpha_i < \alpha^*$, the regret measure is defined as $d\alpha^- = \alpha^* + \sum\alpha_i$; otherwise, it is 0. Thus, we have (i) $\alpha^* + \sum\alpha_i = d_\alpha^- - d_\alpha^+$, (ii) $|\alpha^* + \sum\alpha_i| = d_\alpha^- + d_\alpha^+$, and (iii) $d_\alpha^-, d_\alpha^+ \geq 0$. Similarly, we derive $\beta^* - \sum\beta_i = d_\beta^- - d_\beta^+$, $|\beta^* - \sum\beta_i| = d_\beta^- + d_\beta^+$, and $d_\beta^-, d_\beta^+ \geq 0$. The two-class MCLP model has been gradually evolved as:

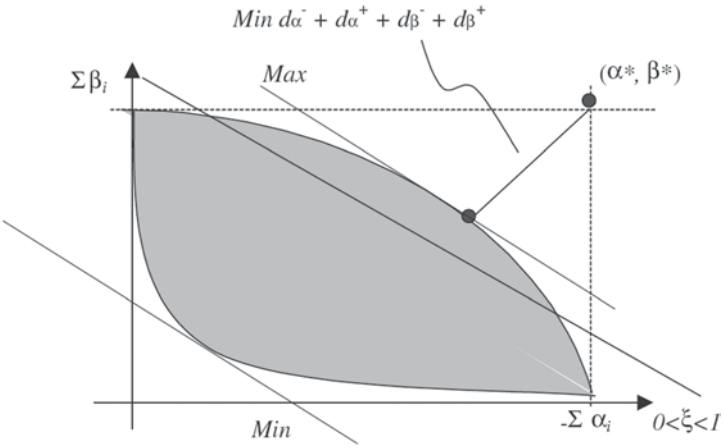

Fig. 5.2 Compromised and Fuzzy Formulations

$$
\begin{aligned}
&\text{Minimize} \quad d_{\alpha}^{+} + d_{\alpha}^{-} + d_{\beta}^{+} + d_{\beta}^{-} \\
&\text{Subject to:} \\
&\alpha^{*} + \sum_{i=1}^{n} \alpha_i = d_{\alpha}^{-} - d_{\alpha}^{+} \\
&\beta^{*} - \sum_{i=1}^{n} \beta_i = d_{\beta}^{-} - d_{\beta}^{+} \\
&X_{11} w_1 + \cdots + X_{1r} w_r = b + \alpha_1 - \beta_1, \quad for \quad X_1 \in B \\
&\qquad\qquad \cdots\cdots \\
&X_{n1} w_1 + \cdots + X_{nr} w_r = b - \alpha_n + \beta_n, \quad for \quad X_n \in G \\
&\alpha_1, \cdots, \alpha_n \geq 0, \quad \beta_1, \cdots, \beta_n \geq 0, \quad d_{\alpha}^{+}, d_{\alpha}^{-}, d_{\beta}^{+}, d_{\beta}^{-} \geq 0
\end{aligned}
\tag{5.5}
$$

Here α^* and β^* are given in advance, w and b are unrestricted. With the optimum value of w and b, a discriminate line is constructed to classify the data set.

The geometric meaning of the model is shown as in Fig. 5.2.

To better and clearly understand the methods, we now sum up the notations involved in the models above.

- X the training set of the classification problem with n observations and r attributes,
- W the optimal coefficients (or weights) for the attributes, $W=(w_1, \ldots, w_r)$,
- b a boundary value (scalar) to separate two predetermined classes, the discrimination function is $Wx=b$,
- α_i the overlapping of the two-class boundary for mislabeled case X_i. $\alpha_i = 0$ for all correctly classified samples,

β_i the distance from X_i to b, $\beta_i = 0$ for all misclassified samples,
α^* and β^* the "ideal value" of $-\Sigma\alpha_i$ and $\Sigma\beta_i$ for solving the two-criteria model (4),
d_α^-, d_α^+ the regret measure, if $-\Sigma\alpha_i > \alpha^*$, $-d_\alpha + = \Sigma\alpha_i + \alpha^*$; otherwise, it is 0. If $-\Sigma\alpha_i < \alpha^*$, $d_\alpha^- = \alpha^* + \Sigma\alpha_i$; otherwise, it is 0.
d_β^-, d_β^+ the regret measure, if $\Sigma\beta_i > \beta^*$, $d_\beta + = \Sigma\beta_i - \beta^*$; otherwise, it is 0. If $\Sigma\beta_i < \beta^*$, $d_\beta^- = \beta^* - \Sigma\beta_i$; otherwise, it is 0.

5.2.2 KMCLP

MCLP model is only applicable for the linear problem. To extend its application, kernel-based multiple criteria linear programming (KMCLP) method was proposed by (Zhang et al. 2009). It introduces kernel function into the original MCLP model to make it possible to solve nonlinear separable problem. The process is based on the assumption that the solution of MCLP model can be described in the following form:

$$w = \sum_{i=1}^{n} \lambda_i y_i X_i \tag{5.6}$$

here n is the sample size of data set. X_i represents each training sample. y_i is the class label of *ith* sample, which can be + 1 or − 1. Put this w into two-class MCLP model (5.5), the following model is formed:

Minimize $d_\alpha^+ + d_\alpha^- + d_\beta^+ + d_\beta^-$

Subject to:

$$\alpha^* + \sum_{i=1}^{n} \alpha_i = d_\alpha^- - d_\alpha^+$$

$$\beta^* - \sum_{i=1}^{n} \beta_i = d_\beta^- - d_\beta^+$$

$$\lambda_1 y_1 (X_1 \cdot X_1) + \ldots + \lambda_n y_n (X_n \cdot X_1) = b + \alpha_1 - \beta_1, \quad for \;\; X_1 \in B$$

$$\ldots\ldots$$

$$\lambda_1 y_1 (X_1 \cdot X_n) + \ldots + \lambda_n y_n (X_n \cdot X_n) = b - \alpha_n + \beta_n, \quad for \;\; X_n \in G$$

$$\alpha_1, \ldots, \alpha_n \geq 0, \beta_1, \ldots, \beta_n \geq 0, \lambda_1, \ldots, \lambda_n \geq 0, d_\alpha^+, d_\alpha^-, d_\beta^+, d_\beta^- \geq 0 \tag{5.7}$$

In above model, each X_i is included in the expression $(X_i \cdot X_j)$ which is the inner product of two samples. But with this model, we can only solve linear separable problem. In order to extend it to be nonlinear model, $(X_i \cdot X_j)$ in the model can be replaced with $K(X_i, X_j)$, then with some nonlinear kernel, i.e. RBF kernel, the above model can be used as a nonlinear classifier. The formulation of RBF kernel is $k(x, x') = \exp(-q \,||\, x - x' \,||^2)$.

Kernel-based multiple criteria linear programming (KMCLP) nonlinear classifier:

$$
\begin{aligned}
&\text{Minimize} \quad d_\alpha^+ + d_\alpha^- + d_\beta^+ + d_\beta^- \\
&\text{Subject to:} \\
&\alpha^* + \sum_{i=1}^{n} \alpha_i = d_\alpha^- - d_\alpha^+ \\
&\beta^* - \sum_{i=1}^{n} \beta_i = d_\beta^- - d_\beta^+ \\
&\lambda_1 y_1 K(X_1, X_1) + \ldots + \lambda_n y_n K(X_n, X_1) = b + \alpha_1 - \beta_1, \quad for \;\; X_1 \in B \\
&\qquad\qquad \ldots\ldots \\
&\lambda_1 y_1 K(X_1, X_n) + \ldots + \lambda_n y_n K(X_n, X_n) = b - \alpha_n + \beta_n, \quad for \;\; X_n \in G \\
&\alpha_1, \ldots, \alpha_n \geq 0, \beta_1, \ldots, \beta_n \geq 0, \lambda_1, \ldots, \lambda_n \geq 0, d_\alpha^+, d_\alpha^-, d_\beta^+, d_\beta^- \geq 0
\end{aligned}
\tag{5.8}
$$

With the optimal value of this model (λ, b, α, β), we can obtain the discrimination function to separate the two classes:

$$
\begin{aligned}
&\lambda_1 y_1 K(X_1, z) + \ldots + \lambda_n y_n K(X_n, z) \leq b, \quad then \;\; z \in B, \\
&\lambda_1 y_1 K(X_1, z) + \ldots + \lambda_n y_n K(X_n, z) \geq b, \quad then \;\; z \in G,
\end{aligned}
\tag{5.9}
$$

where z is the new input data which is the evaluated target with r attributes. X_i represents each training sample. y_i is the class label of *ith* sample.

We notice here that a set of optimization variable w is substituted by a set of variables λ in the new model, which is the result of introduction of formulation (6) and thus lead to the employment of kernel function. KMCLP is a classification model which is applicable for nonlinear separable data set. With its optimal solution λ and b, the discrimination hyperplane is then constructed, and the two classes can be separated by it.

5.3 Linear Knowledge-incorporated MCLP Classifiers

5.3.1 Linear Knowledge

Prior knowledge in some classifiers usually consists of a set of rules, such as, if A then $x \in G$ (or $x \in B$), where condition A is relevant to the attributes of the input data. One example of such form of knowledge can be seen in the breast cancer recurrence or nonrecurrence prediction. Usually, doctors can judge if the cancer recur or not in terms of some measured attributes of the patients. The prior knowledge

used by doctors in the breast cancer dataset includes two rules which depend on two features of the total 32 attributes: tumor size (T) and lymph node status (L). The rules are (Fung et al. 2005):

If $L \geq 5$ and $T \geq 4$ Then RECUR and If $L = 0$ and $T \leq 1.9$ Then NONRECUR

The conditions *$L \geq 5$ and $T \geq 4$ ($L = 0$ and $T \leq 1.9$)* in the above rules can be written into such inequality as $Cx \leq c$, where C is a matrix driven from the condition, x represents each individual sample, c is a vector. For example, if each sample x is expressed by a vector $[x_1,\ldots,x_L,\ldots,x_T,\ldots,x_r]^T$, for the rule: *if $L \geq 5$ and $T \geq 4$ then RECUR*, it also means: *if $x_L \geq 5$ and $x_T \geq 4$, then $x \in RECUR$*, where x_L and x_T are the corresponding values of attributes L and T of a certain sample data, r is the number of attributes. Then its corresponding inequality $Cx'' c$ can be written as:

$$\begin{bmatrix} 0 & \ldots & -1 & \ldots & 0 & \ldots & 0 \\ 0 & \ldots & 0 & \ldots & -1 & \ldots & 0 \end{bmatrix} x \leq \begin{bmatrix} -5 \\ -4 \end{bmatrix}.$$

where x is the vector with r attributes include two features relevant to prior knowledge.

Similarly, the condition *$L = 0$ and $T \leq 1.9$* can also be reformulated to be inequalities. With regard to the condition *$L = 0$*, in order to express it into the formulation of $Cx \leq c$, we must replace it with the condition *$L \geq 0$ and $L \leq 0$*. Then the condition *$L = 0$ and $T \leq 1.9$* can be represented by two inequalities: $C^1 x \leq c^1$ and $C^2 x \leq c^2$, as follows:

$$\begin{bmatrix} 0 & \ldots & -1 & \ldots & 0 & \ldots & 0 \\ 0 & \ldots & 0 & \ldots & 1 & \ldots & 0 \end{bmatrix} x \leq \begin{bmatrix} 0 \\ 1.9 \end{bmatrix} \text{ and } \begin{bmatrix} 0 & \ldots & 1 & \ldots & 0 & \ldots & 0 \\ 0 & \ldots & 0 & \ldots & 1 & \ldots & 0 \end{bmatrix} x \leq \begin{bmatrix} 0 \\ 1.9 \end{bmatrix}$$

We notice the fact that the set $\{x \mid Cx \leq c\}$ can be regarded as polyhedral convex set. In Fig. 5.3, the triangle and rectangle are such sets.

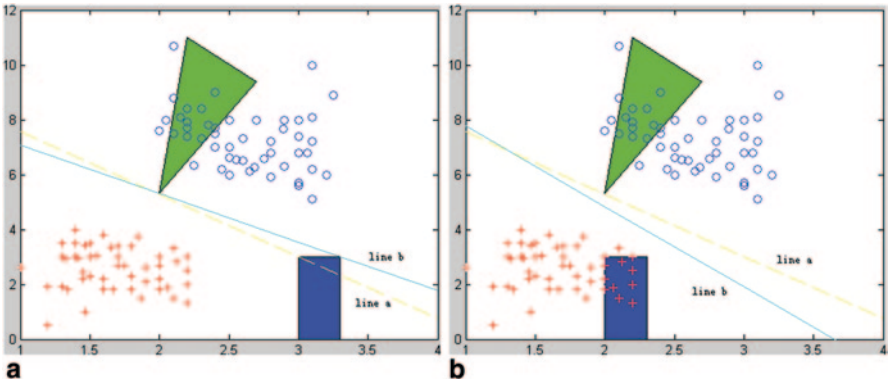

Fig. 5.3 The classification result by MCLP(line a) and Knowledge-Incorporated MCLP(line b)

In two-class classification problem, the result RECUR or NONRECUR is equal to the expression $x \in B$ or $x \in G$. So according to the above rules, we have:

$$Cx \le c \Rightarrow x \in G \quad (or \quad x \in B) \tag{5.10}$$

In MCLP classifier, if the classes are linearly separable, then $x \in G$ is equal to $x^T w \ge b$, similarly, $x \in B$ is equal to $x^T w \le b$. That is, the following implication must hold:

$$Cx \le c \Rightarrow x^T w \ge b \quad (or \quad x^T w \le b) \tag{5.11}$$

For a given (w,b), the implication $Cx \le c \Rightarrow x^T w \ge b$ holds, this also means that $Cx \le c, x^T w < b$ has no solution x. According to nonhomogeneous Farkas theorem, we can conclude that $C^T u + w = 0, c^T u + b \le 0, u \ge 0,$ has a solution (u, w) (Fung et al. 2002).

The above statement is able to be added to constraints of an optimization problem. In this way, the prior knowledge in the form of some equalities and inequalities in constraints is embedded to the original multiple linear programming (MCLP) model. The knowledge-incorporated MCLP model is described in the following.

5.3.2 Linear Knowledge-incorporated MCLP

Now, we are to explain the knowledge-incorporated MCLP model. This model is to deal with linear knowledge and linear separable data. The combination of the two kinds of input can help to improve the performances of both methods.

Suppose there are a series of knowledge sets as follows:

If $C^i x \le c^i, i = 1,\ldots,k$ Then $x \in G$

If $D^j x \le d^j, j = 1,\ldots,l$ Then $x \in B$

This knowledge also means the convex sets $\{x \mid C^i x \le c^i\}, i = 1,\ldots,k$ lie on the G side of the bounding plane, the convex sets $\{x \mid D^j x \le d^j\}, j = 1,\ldots,l$ on the B side.

Based on the above theory in the last section, we converted the knowledge to the following constraints:

There exist $u^i,\ i = 1,\ldots,k,\ v^j, j = 1,\ldots,l,$ such that:

$$\begin{aligned} &C^{iT} u^i + w = 0, \quad c^{iT} u^i + b \le 0, \quad u^i \ge 0, \quad i = 1,\ldots,k \\ &D^{jT} v^j - w = 0, \quad d^{jT} v^j - b \le 0, \quad v^j \ge 0, \quad j = 1,\ldots,l \end{aligned} \tag{5.12}$$

However, there is no guarantee that such bounding planes precisely separate all the points. Therefore, some error variables need to be added to the above formulas. The constraints are further revised to be:

There exist $u^i,\ r^i,\ \rho^i,\ i = 1,\ldots,k \ \ and \ \ v^j,\ s^j,\ \sigma^j,\ j = 1,\ldots,l,$ such that:

$$\begin{aligned} -r^i \le C^{iT}u^i + w \le r^i, \quad c^{iT}u^i + b \le \rho^i, \quad u^i \ge 0, \quad i = 1,\ldots,k \\ -s^j \le D^{jT}v^j - w \le s^j, \quad d^{jT}v^j - b \le \sigma^j, \quad v^j \ge 0, \quad j = 1,\ldots,l \end{aligned} \tag{5.13}$$

After that, we embed the above constraints to the MCLP classifier, and obtained the knowledge-incorporated MCLP classifier:

$$\begin{aligned} &\text{Minimize} \quad d_\alpha^+ + d_\alpha^- + d_\beta^+ + d_\beta^- + C(\sum(r^{\mathrm{i}} + \rho^{\mathrm{i}}) + \sum(s^{\mathrm{j}} + \sigma^{\mathrm{j}})) \\ &\text{Subject to:} \\ &\quad \alpha^* + \sum_{i=1}^{n}\alpha_i = d_\alpha^- - d_\alpha^+ \\ &\quad \beta^* - \sum_{i=1}^{n}\beta_i = d_\beta^- - d_\beta^+ \\ &\quad x_{11}w_1 + \ldots + x_{1\mathrm{r}}w_{\mathrm{r}} = \mathrm{b} + \alpha_1 - \beta_1\ , \quad \text{for } \mathrm{A}_1 \in \mathrm{B}, \\ &\qquad \vdots \\ &\quad x_{\mathrm{n}1}w_1 + \ldots + x_{\mathrm{nr}}w_{\mathrm{r}} = \mathrm{b} - \alpha_{\mathrm{n}} + \beta_{\mathrm{n}}\ , \quad \text{for } \mathrm{A}_{\mathrm{n}} \in \mathrm{G}, \\ &\quad -r^{\mathrm{i}} \le \mathrm{C}^{\mathrm{i}'}u^{\mathrm{i}} + w \le r^{\mathrm{i}}, \qquad \mathrm{i}=1,\ldots,\mathrm{k} \\ &\quad \mathrm{c}^{\mathrm{i}'}u^{\mathrm{i}} + b \le \rho^{\mathrm{i}} \\ &\quad -s^{\mathrm{j}} \le \mathrm{D}^{\mathrm{j}'}v^{\mathrm{j}} - w \le s^{\mathrm{j}}, \qquad \mathrm{j}=1,\ldots,l \\ &\quad \mathrm{d}^{\mathrm{j}'}v^{\mathrm{j}} - b \le \sigma^{\mathrm{j}} \\ &\quad \alpha_1,\ldots,\alpha_{\mathrm{n}} \ge 0, \quad \beta_1,\ldots,\beta_{\mathrm{n}} \ge 0, \quad (u^{\mathrm{i}}, v^{\mathrm{j}}, r^{\mathrm{i}}, \rho^{\mathrm{i}}, s^{\mathrm{j}}, \sigma^{\mathrm{j}}) \ge 0 \end{aligned} \tag{5.14}$$

In this model, all the inequality constraints are derived from the prior knowledge. The last objective $\mathrm{C}(\sum(r^{\mathrm{i}} + \rho^{\mathrm{i}}) + \sum(s^{\mathrm{j}} + \sigma^{\mathrm{j}}))$ is about the slack error variables added to the original knowledge equality constraints. The last objective attempts to drive the error variables to zero. We want to get the best bounding plane (w, b) in formula (1) by solving this model to separate the two classes.

We notice the fact that if we set the value of parameter C to be zero, this means to take no account of knowledge. Then this model will be equal to the original MCLP model. Theoretically, the larger the value of C, the greater impact on the classification result of the knowledge sets.

5.3.3 Linear Knowledge-Incorporated KMCLP

If the data set is nonlinear separable, the above model will be inapplicable. We need to figure out how to embed prior knowledge into the KMCLP model, which can solve nonlinear separable problem.

As is shown in the above part, in generating KMCLP model, we suppose:

$$w = \sum_{i=1}^{n} \lambda_i y_i X_i \tag{5.15}$$

If expressed by matrix, the above formulation will be:

$$w = X^T Y \lambda \tag{5.16}$$

where Y is $n*n$ diagonal matrix, the value of each diagonal element depends on the class label of the corresponding sample data, which can be $+1$ or -1. X is the $n*r$ input matrix with n samples, r attributes. λ is a n-dimensional vector $\lambda=(\lambda_1, \lambda_2,\ldots, \lambda_n)^T$.

$$Y = \begin{bmatrix} y_1 & 0 & \cdots & 0 \\ 0 & y_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & y_n \end{bmatrix}, \qquad X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1r} \\ x_{21} & x_{22} & \cdots & x_{2r} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{nr} \end{bmatrix}$$

Therefore, w in the original MCLP model is replaced by $X^TY\lambda$, thus forming the KMCLP model. And in this new model, the value of each λ_i is to be worked out by the optimization model.

In order to incorporate prior knowledge into KMCLP model, the inequalities about the knowledge must be transformed to be the form with λ_i instead of w. Enlightened by the KMCLP model, we also introduce kernel to the expressions of knowledge. Firstly, the equalities in (5.12) are multiplied by input matrix X (Fung et al. 2003). Then replacing w with $X^TY\lambda$, (5.12) will be:

$$\begin{aligned} &XC^{iT}u^i + XX^TY\lambda = 0, \quad c^{iT}u^i + b \le 0, \quad u^i \ge 0, \quad i = 1,\ldots,k \\ &XD^{jT}v^j - XX^TY\lambda = 0, \quad d^{jT}v^j - b \le 0, \quad v^j \ge 0, \quad j = 1,\ldots,l \end{aligned} \tag{5.17}$$

Kernel function is introduced here to replace XC^{iT} and XX^T. Also slack errors are added to the expressions, then such kind of constraints are formulated:

$$\begin{aligned} &-r^{\mathrm{i}} \le K(X, \mathrm{C}^{\mathrm{i}T})u^{\mathrm{i}} + K(X, X^T)Y\lambda \le r^{\mathrm{i}}, \quad \mathrm{i}=1,\ldots,\mathrm{k} \\ &\mathrm{c}^{\mathrm{i}\mathrm{T}}u^{\mathrm{i}} + b \le \rho^{\mathrm{i}} \\ &-s^{\mathrm{j}} \le K(X, \mathrm{D}^{\mathrm{j}T})v^{\mathrm{j}} - K(X, X^T)Y\lambda \le s^{\mathrm{j}}, \quad \mathrm{j}=1,\ldots,l \\ &\mathrm{d}^{\mathrm{j}T}v^{\mathrm{j}} - b \le \sigma^{\mathrm{j}} \end{aligned} \tag{5.18}$$

These constraints can be easily embedded to KMCLP model (5.8) as the constraints acquired from prior knowledge.

Knowledge-incorporated KMCLP classifier:

$$
\begin{aligned}
& \mathrm{Min}(d_\alpha^+ + d_\alpha^- + d_\beta^+ + d_\beta^-) + C(\sum_{i=1}^{k}(r^{\mathrm{i}} + \rho^{\mathrm{i}}) + \sum_{j=1}^{l}(s^{\mathrm{j}} + \sigma^{\mathrm{j}})) \\
\text{s.t.} \quad & \lambda_1 y_1 K(X_1, X_1) + \ldots + \lambda_n y_n K(X_n, X_1) = b + \alpha_1 - \beta_1, \ , \quad \text{for } X_1 \in \mathrm{B}, \\
& \qquad \vdots \\
& \lambda_1 y_1 K(X_1, X_n) + \ldots + \lambda_n y_n K(X_n, X_n) = b - \alpha_n + \beta_n \ , \quad \text{for } X_{\mathrm{n}} \in \mathrm{G}, \\
& \alpha^* + \sum_{i=1}^{n} \alpha_i = d_\alpha^- - d_\alpha^+, \\
& \beta^* - \sum_{i=1}^{n} \beta_i = d_\beta^- - d_\beta^+, \\
& -r^{\mathrm{i}} \le K(X, \mathrm{C}^{\mathrm{i}T})u^{\mathrm{i}} + K(X, X^T)Y\lambda \le r^{\mathrm{i}}, \qquad \mathrm{i}=1,\ldots,\mathrm{k} \\
& \mathrm{c}^{\mathrm{i}T}u^{\mathrm{i}} + b \le \rho^{\mathrm{i}} \\
& -s^{\mathrm{j}} \le K(X, \mathrm{D}^{\mathrm{j}T})v^{\mathrm{j}} - K(X, X^T)Y\lambda \le s^{\mathrm{j}}, \qquad \mathrm{j}=1,\ldots,l \\
& \mathrm{d}^{\mathrm{j}T}v^{\mathrm{j}} - b \le \sigma^{\mathrm{j}} \\
& \alpha_1, \ldots, \alpha_{\mathrm{n}} \ge 0, \quad \beta_1, \ldots, \beta_{\mathrm{n}} \ge 0, \quad \lambda_1, \ldots, \lambda_{\mathrm{n}} \ge 0, \\
& (u^{\mathrm{i}}, v^{\mathrm{j}}, r^{\mathrm{i}}, \rho^{\mathrm{i}}, s^{\mathrm{j}}, \sigma^{\mathrm{j}}) \ge 0 \\
& d_\alpha^-, d_\alpha^+, d_\beta^-, d_\beta^+ \ge 0
\end{aligned}
\tag{5.19}
$$

In this model, all the inequality constraints are derived from prior knowledge. u^i, $v^j \in R^p$, where p is the number of conditions in one knowledge. For example, in the knowledge *if* $x_L \ge 5$ *and* $x_T \ge 4$, *then* $x \in RECUR$, the value of p is 2. r^{i}, ρ^{i}, s^{j} and σ^{j} are all real numbers. And the last objective $\mathrm{Min}\sum(r^{\mathrm{i}} + \rho^{\mathrm{i}}) + \sum(s^{\mathrm{j}} + \sigma^{\mathrm{j}})$ is about the slack error variables added to the original knowledge equality constraints. As we talked in last section, the larger the value of C, the greater impact on the classification result of the knowledge sets.

In this model, several parameters need to be set before optimization process. Apart from C we talked about above, the others are parameter of kernel function q (if we choose RBF kernel) and the ideal compromise solution α* and β*. We want to get the best bounding plane (λ, b) by solving this model to separate the two classes. And the discrimination function of the two classes is:

$$
\begin{aligned}
& \lambda_1 y_1 K(X_1, z) + \ldots + \lambda_n y_n K(X_n, z) \le b, \quad \textit{then} \quad z \in B \\
& \lambda_1 y_1 K(X_1, z) + \ldots + \lambda_n y_n K(X_n, z) \ge b, \quad \textit{then} \quad z \in G
\end{aligned}
\tag{5.20}
$$

where z is the input data which is the evaluated target with r attributes. X_i represents each training sample. y_{i} is the class label of *ith* sample.

5.4 Nonlinear Knowledge-Incorporated KMCLP Classifier

5.4.1 Nonlinear Knowledge

In the above models, the prior knowledge we deal with is linear. That means the conditions in the above rules can be written into such inequality as $Cx \leq c$, where C is a matrix driven from the condition, x represents each individual sample, c is a vector. The set $\{x|\ Cx \leq c\}$ can be viewed as polyhedral convex set, which is a linear geometry in input space. But, if the shape of the region which consists of knowledge is nonlinear, for example, $\{x|\ \|x\|^2 \leq c\}$, how to deal with such kind of knowledge?

Suppose the region is nonlinear convex set, we describe the region by $g(x) \leq 0$. If the data is in this region, it must belong to class B. Then, such kind of nonlinear knowledge may take the form of:

$$\begin{aligned} g(x) \leq 0 \ &\Rightarrow\ x \in B \\ h(x) \leq 0 \ &\Rightarrow\ x \in G \end{aligned} \tag{5.21}$$

Here $g(x)$: $R^r \rightarrow R^p$ $(x \in \Gamma)$ and $h(x)$: $R^r \rightarrow R^q$ $(x \in \Delta)$ are functions defined on a subset Γ and Δ of R^r which determine the regions in the input space. All the data satisfied $g(x) \leq 0$ must belong to the class B and $h(x) \leq 0$ to the class G.

With KMCLP classifier, this knowledge equals to:

$$\begin{aligned} g(x) \leq 0 \ &\Rightarrow\ \lambda_1 y_1 K(X_1, x) + \ldots + \lambda_n y_n K(X_n, x) \leq b, (x \in \Gamma) \\ h(x) \leq 0 \ &\Rightarrow\ \lambda_1 y_1 K(X_1, x) + \ldots + \lambda_n y_n K(X_n, x) \geq b, (x \in \Delta) \end{aligned} \tag{5.22}$$

This implication can be written in the following equivalent logical form (Mangasarian and Wild 2007):

$$\begin{aligned} g(x) \leq 0 \quad & \lambda_1 y_1 K(X_1, x) + \ldots + \lambda_n y_n K(X_n, x) - b > 0, \text{has no solution } x \in \Gamma. \\ h(x) \leq 0 \quad & \lambda_1 y_1 K(X_1, x) + \ldots + \lambda_n y_n K(X_n, x) - b < 0, \text{ has no solution } x \in \Delta. \end{aligned} \tag{5.23}$$

The above expressions hold, then there exist $v \in R^p$, $r \in R^q$, $v, r \geq 0$ such that:

$$\begin{aligned} -\lambda_1 y_1 K(X_1, x) - \ldots - \lambda_n y_n K(X_n, x) + b + v^T g(x) \geq 0, (x \in \Gamma) \\ \lambda_1 y_1 K(X_1, x) + \ldots + \lambda_n y_n K(X_n, x) - b + r^T h(x) \geq 0, (x \in \Delta) \end{aligned} \tag{5.24}$$

Add some slack variables on the above two inequalities, then they are converted to:

$$\begin{aligned} -\lambda_1 y_1 K(X_1, x) - \ldots - \lambda_n y_n K(X_n, x) + b + v^T g(x) + s \geq 0, (x \in \Gamma) \\ \lambda_1 y_1 K(X_1, x) + \ldots + \lambda_n y_n K(X_n, x) - b + r^T h(x) + t \geq 0, (x \in \Delta) \end{aligned} \tag{5.25}$$

The above statement is able to be added to constraints of an optimization problem.

5.4.2 Nonlinear Knowledge-incorporated KMCLP

Suppose there are a series of knowledge sets as follows:

If $g_i(x) \le 0$, Then $x \in B$ $(g_i(x): R^r \to R^p{}_i\ (x \in \Gamma_i), i = 1,\ldots,k)$

If $h_j(x) \le 0$, Then $x \in G$ $(h_j(x): R^r \to R^q{}_j\ (x \in \Delta_j), j = 1,\ldots,l)$

Based on the above theory in last section, we converted the knowledge to the following constraints:

There exist $v_i \in R^p{}_i$, $i = 1,\ldots,k$, $r_j \in R^q{}_j$, $j = 1,\ldots,l$, $v_i, r_j \ge 0$ such that:

$$\begin{aligned} &-\lambda_1 y_1 K(X_1, x) - \ldots - \lambda_n y_n K(X_n, x) + b + v_i^T g_i(x) + s_i \ge 0, (x \in \Gamma) \\ &\lambda_1 y_1 K(X_1, x) + \ldots + \lambda_n y_n K(X_n, x) - b + r_j^T h_j(x) + t_j \ge 0, (x \in \Delta) \end{aligned} \tag{5.26}$$

These constraints can be easily imposed to KMCLP model (4.8) as the constraints acquired from prior knowledge.

Nonlinear knowledge in KMCLP classifier (Zhang et al. 2002):

$$\begin{aligned} &\mathrm{Min}\left(d_\alpha^+ + d_\alpha^- + d_\beta^+ + d_\beta^-\right) + C\left(\sum_{i=1}^{k} s_i + \sum_{j=1}^{l} t_j\right) \\ \text{s.t.}\quad &\lambda_1 y_1 K(X_1, X_1) + \ldots + \lambda_n y_n K(X_n, X_1) = b + \alpha_1 - \beta_1, \ , \quad \text{for } X_1 \in \mathrm{B}, \\ &\qquad \vdots \\ &\lambda_1 y_1 K(X_1, X_n) + \ldots + \lambda_n y_n K(X_n, X_n) = b - \alpha_n + \beta_n \ , \quad \text{for } X_n \in \mathrm{G}, \\ &\alpha^* + \sum_{i=1}^{n} \alpha_i = d_\alpha^- - d_\alpha^+, \\ &\beta^* - \sum_{i=1}^{n} \beta_i = d_\beta^- - d_\beta^+, \\ &-\lambda_1 y_1 K(X_1, x) - \ldots - \lambda_n y_n K(X_n, x) + b + v_i^T g_i(x) + s_i \ge 0, \qquad \mathrm{i}=1,\ldots,\mathrm{k} \\ &s_i \ge 0, \qquad \mathrm{i}=1,\ldots,\mathrm{k} \\ &\lambda_1 y_1 K(X_1, x) + \ldots + \lambda_n y_n K(X_n, x) - b + r_j^T h_j(x) + t_j \ge 0, \qquad \mathrm{j}=1,\ldots,l \\ &t_j \ge 0, \qquad \mathrm{j}=1,\ldots,l \\ &\alpha_1,\ldots,\alpha_n \ge 0, \quad \beta_1,\ldots,\beta_n \ge 0, \quad \lambda_1,\ldots,\lambda_n \ge 0, \\ &(v_i, r_j) \ge 0 \\ &d_\alpha^-, d_\alpha^+, d_\beta^-, d_\beta^+ \ge 0 \end{aligned} \tag{5.27}$$

In this model, all the inequality constraints are derived from the prior knowledge. The last objective $C(\sum_{i=1}^{k} s_i + \sum_{j=1}^{l} t_j)$ is about the slack error. Theoretically, the larger the value of C, the greater impact on the classification result of the knowledge sets.

The parameters need to be set before optimization process are C, q (if we choose RBF kernel), α^* and β^*. The best bounding plane of this model decided by (λ, b) of the two classes is the same with formula (5.20).

5.5 Numerical Experiments

All above models are linear programming models which are easily solved by some commercial software such as SAS LP and MATLAB. In this paper, MATLAB6.0 is employed in the solution process. To prove the effectiveness of these models, we apply them to four data sets which consist of knowledge sets and sample data. Among them, three are synthetic examples, one is real application.

5.5.1 A Synthetic Data Set

To demonstrate the geometry of the knowledge-incorporated MCLP, we apply the model to a synthetic example with 100 points. These points are marked by "o" and "+" in Fig. 5.3 which represent two different classes. Original MCLP model (5) and knowledge-incorporated MCLP model (14) are applied to get the separation lines of the two classes. Figure 5.3 depicts the results of the separation lines (***line a*** and ***line b***) generated by the two models.

The rectangle and the triangle in Fig. 5.3 are two knowledge sets for the classes. ***Line a*** is the discriminate line of the two classes by the origional MCLP model (C = 0), then ***line b*** is generated by the Knowledge-Incorporated MCLP model (C = 1). From the above figure, we can see that the separation line changed when we incorporated prior knowledge into MCLP(C is set to be 1), thus results in two different lines ***a*** and ***b***. And when we change the rectangle knowledge set's position, the line b is also changed with it. This means that the knowledge does have effect on the classifier, and our new model seems valid to deal with the prior knowledge.

5.5.2 Checkerboard Data

For knowledge-incorporated KMCLP which can handle nonlinear separable data, we construct a checkerboard dataset (Fig. 5.4) to test the model. This data set consists of 16 points, and no neighboring points belong to one class. The two squares in the bottom of the figure are prior knowledge for the classes (Fung et al. 2003). In this case, we can see the impressive influence of the knowledge on the separation curve.

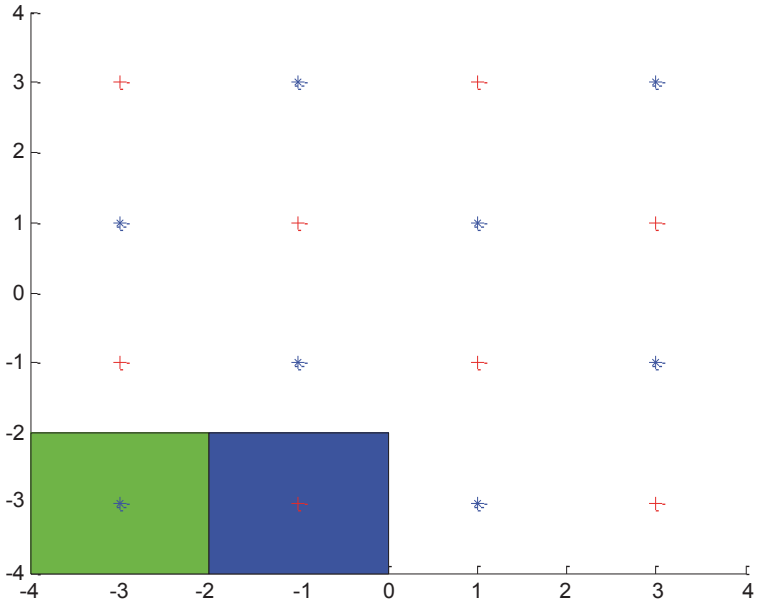

Fig. 5.4 The checkerboard data set

Experiments are conducted with the knowledge-incorporated KMCLP model with $C=0$, 0.001, 0.01, 0.1 and 1. And, after grid search process, we choose the best suitable value for parameters: $q=1$, $\alpha^*=10^{-5}$, $\beta^*=10^6$. The results of the separation curve generated by knowledge-incorporated KMCLP are showed in Fig. 5.5.

We notice the fact that when $C=0.01$(Fig. 5.5a) or even smaller value, the separation curve can not be as sharp as that of a bigger value of C like in Fig. 5.5b. And bigger C means more contribution of prior knowledge to the optimization result. Obviously in this checkerboard case, sharper line will be more preferable, because it can lead to more accurate separation result when faced with larger checkerboard data.

However in Fig. 5.5b, we also find when set $C=0$ the separation curve can also be sharp. It seems to have no difference with $C=0.1$ and 1. This demonstrates the original KMCLP model can achieve a preferable result by itself even without knowledge.

5.5.3 Wisconsin Breast Cancer Data with Nonlinear Knowledge

Concerning real word cases, we apply the nonlinear knowledge model (27) to Wisconsin breast cancer prognosis data set for predicting recurrence or nonrecurrence of the disease. This data set concerns 10 features obtained from a fine needle aspirate (Mangasarian and Wild 2007; Murphy and Aha 1992). Of each feature, the mean, standard error, and worst or largest value were computed for each image, thus resulting in 30 features. Besides, two histological features, tumor size and lymph

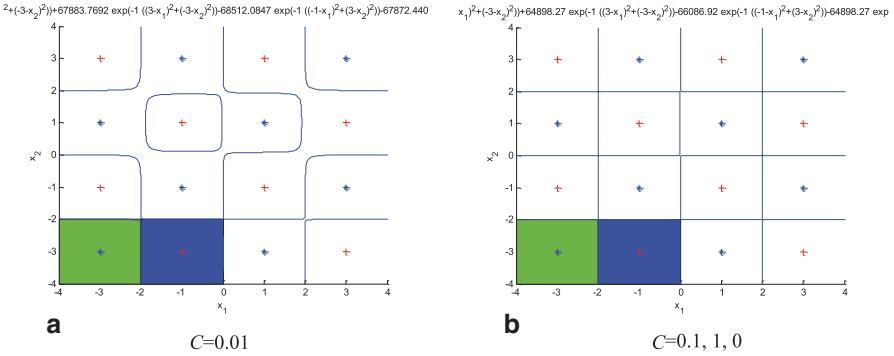

Fig. 5.5 The classification results by Knowledge-Incorporated KMCLP on checkerboard data set. **a** $C=0.01$. **b** $C=0.1, 1, 0$

node status, obtained during surgery for breast cancer patients, are also included in the attributes. According to the characteristic of the data set, we separate the features into four groups F1, F2, F3 and F4, which represent the mean, standard error, worst or largest value of each image and histological features, respectively. We plotted each point and the prior knowledge in the 2-dimensional space in terms of the last two attributes in Fig. 5.6. The three geometric regions in the figure are the corresponding knowledge. And the points marked by "o" and "+" represent two different classes. With the three knowledge regions, we can only discriminate a part of "o" data. So we need to use multiple criteria linear programming classification method plus prior knowledge to solve the problem.

The prior knowledge involved here is nonlinear knowledge. The whole knowledge consists of three regions, which correspond to the following three implications:

$$\left\|\begin{pmatrix} 5.5\times x_{iT} & 5.5\times 7 \\ x_{iL} & 9 \end{pmatrix}\right\| + \left\|\begin{pmatrix} 5.5\times x_{iT} & 5.5\times 4.5 \\ x_{iL} & 27 \end{pmatrix}\right\| - 23.0509 \le 0 \Rightarrow X_i \in RECUR$$

$$\begin{pmatrix} -x_{iL} + 5.7143\times x_{iT} - 5.75 \\ x_{iL} - 2.8571\times x_{iT} - 4.25 \\ -x_{iL} + 6.75 \end{pmatrix} \le 0 \Rightarrow X_i \in RECUR$$

$$\frac{1}{2}(x_{iT} - 3.35)^2 + (x_{iL} - 4)^2 - 1 \le 0 \Rightarrow X_i \in RECUR$$

Here, x_{iT} is the tumor size, and x_{iL} is the number of lymph nodes of training sample X_i. In Fig. 5.6, the ellipse near to the upper-right corner is about the knowledge of the first implication. The triangular region corresponds to the second implication. And the ellipse in the bottom corresponds to the third implication. The red circle points represent the recurrence samples, while the blue cross points represent non-recurrence samples.

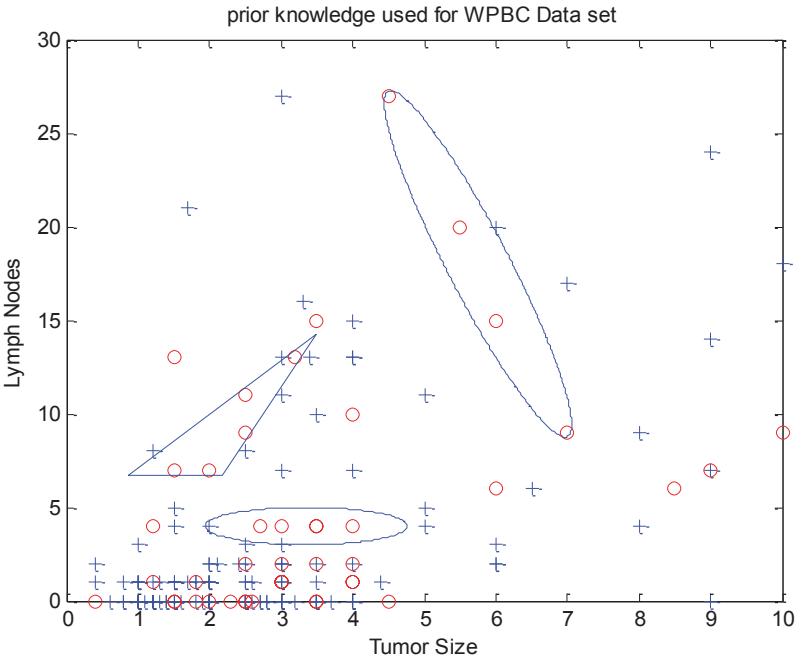

Fig. 5.6 WPBC data set and prior knowledge

Before classification, we scaled the attributes to [0, 1]. And in order to balance the samples in the two classes, we need to randomly choose 46 samples, which is the exact number of the recurrence samples, from the nonrecurrence group. We choose the value of q from the range $[10^{-6}, \ldots, 10^{6}]$, and find the best value of q for RBF kernel is 1. Leave-one-out cross-validation method is used to get the accuracy of the classification of our method.

Experiments are conducted with respect to the combinations of four subgroups of attributes. $C=0$ means the model takes no account of knowledge. The results are shown here.Tab 5.1

The above table shows that classified by our model with knowledge ($C=1$), the accuracies are higher than the results without knowledge ($C=0$). The highest improvement of the four attributes groups is about 6.7%. Although it is not as much as we expected, we can see the knowledge dose make good results on this classification problem. Probably, the knowledge here is not as precise as can pro-

Table 5.1 The accuracies of classification on Wisconsin breast cancer data set.

	F1 and F4 (%)	F1, F3 and F4 (%)	F3 and F4 (%)	F1,F2,F3 and F4 (%)
$C=0$	51.807	59.783	57.609	63.043
$C=1$	56.522	66.304	63.043	64.13

duce noticeable improvement to the precision. But it does have influence on the classification result. If we have much more precise knowledge, the classifier will be more accurate.

5.6 Conclusions

In this section, we summarize the relevant works which combine the prior knowledge and MCLP or KMCLP model to solve the problem when input consists of not only training example, but also prior knowledge. Specifically, how to deal with linear and nonlinear knowledge in MCLP and KMCLP model is the main concerning of this paper. Linear prior knowledge in the form of polyhedral knowledge sets in the input space of the given data can be expressed into logical implications, which can further be converted into a series of equalities and inequalities. These equalities and inequalities can be imposed to the constraints of original MCLP and KMCLP model, then help to generate the separation hyperplane of the two classes. In the same way, nonlinear knowledge can also be incorporated as the constraints into the KMCLP model to make it possible to separate two classes with help of prior knowledge. All these models are linear programming formulations, which can be easily solved by some commercial software. With the optimum solution, the separation hyperplane of the two classes can be formulated. Numerical tests indicate that these models are effective when combining prior knowledge with the training sample as the classification principle.