8

Generating Support Vector Machines Using Multi-Objective Optimization and Goal Programming

Hirotaka Nakayama[1] and Yeboon Yun[2]

[1] Konan University, Dept. of Information Science and Systems Engineering
8-9-1 Okamoto, Higashinada, Kobe 658-8501, Japan
nakayama@konan-u.ac.jp

[2] Kagawa University, Kagawa 761-0396, Japan
yun@eng.kagawa-u.ac.jp

Summary. Support Vector Machine (SVM) is gaining much popularity as one of effective methods for machine learning in recent years. In pattern classification problems with two class sets, it generalizes linear classifiers into high dimensional feature spaces through nonlinear mappings defined implicitly by kernels in the Hilbert space so that it may produce nonlinear classifiers in the original data space. Linear classifiers then are optimized to give the maximal margin separation between the classes. This task is performed by solving some type of mathematical programming such as quadratic programming (QP) or linear programming (LP). On the other hand, from a viewpoint of mathematical programming for machine learning, the idea of maximal margin separation was employed in the multi-surface method (MSM) suggested by Mangasarian in 1960's. Also, linear classifiers using goal programming were developed extensively in 1980's. This chapter introduces a new family of SVM using multi-objective programming and goal programming (MOP/GP) techniques, and discusses its effectiveness throughout several numerical experiments.

8.1 Introduction

For convenience, we consider pattern classification problems. Let X be a space of conditional attributes. For binary classification problems, the value of $+1$ or -1 is assigned to each pattern $\boldsymbol{x}_i \in X$ according to its class $\mathcal{A}$ or $\mathcal{B}$. The aim of machine learning is to predict which class newly observed patterns belong to on the basis of the given training data set $(\boldsymbol{x}_i, y_i)$ $(i = 1, \dots, \ell)$, where $y_i = +1$ or -1. This is performed by finding a discriminant function $f(\boldsymbol{x})$ such that $f(\boldsymbol{x}) \geqq 0$ for $\boldsymbol{x} \in \mathcal{A}$ and $f(\boldsymbol{x}) < 0$ for $\boldsymbol{x} \in \mathcal{B}$. Linear discriminant functions, in particular, can be expressed by the following linear form

$$f(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{x} + b$$

H. Nakayama and Y. Yun: *Generating Support Vector Machines Using Multi-Objective Optimization and Goal Programming*, Studies in Computational Intelligence (SCI) **16**, 173–198 (2006)
www.springerlink.com

with the property

$$\begin{aligned} \boldsymbol{w}^T\boldsymbol{x}+b \geqq 0 \quad &\text{for} \quad \boldsymbol{x} \in \mathcal{A} \\ \boldsymbol{w}^T\boldsymbol{x}+b < 0 \quad &\text{for} \quad \boldsymbol{x} \in \mathcal{B}. \end{aligned}$$

For such a pattern classification problem, artificial neural networks have been widely applied. However, the back propagation method is reduced to nonlinear optimization with multiple local optima, and hence difficult to apply to large scale problems. Another drawback in the back propagation method is in the fact that it is difficult to change the structure adaptively according to the change of environment in incremental learning. Recently, Support Vector Machine (SVM, for short) is attracting interest of researchers, in particular, people who are engaged in mathematical programming, because it is reduced to quadratic programming (QP) or linear programming (LP). One of main features in SVM is that it is a linear classifier with maximal margin on the feature space through nonlinear mappings defined implicitly by kernels in the Hilbert space.

The idea of maximal margin in linear classifiers is intuitive, and its reasoning in connection with perceptrons was given in early 1960's (e.g., Novikoff [17]). The maximal margin is effectively applied for discrimination analysis using mathematical programming, e.g., MSM (Multi-Surface Method) by Mangasarian [11]. Later, linear classifiers with maximal margin were formulated as linear goal programming, and extensively studied through 1980's to the beginning of 1990's. The pioneering work was given by Freed-Glover [9], and a good survey can be seen in Erenguc-Koehler *et al.* [8]. This chapter discusses SVMs using techniques of multi-objective programming (MOP) and goal programming (GP), and proposes several extensions of SVM along MOP/GP.

8.2 Support Vector Machine

Support vector machine (SVM) was developed by Vapnik *et al.* [6], [22] (see also Cristianini and Shawe-Taylor [7], Schölkopf-Smola [20]) and its main features are

1) SVM maps the original data set into a high dimensional feature space by nonlinear mapping implicitly defined by kernels in the Hilbert space,

2) SVM finds linear classifiers with maximal margin on the feature space,

3) SVM provides an evaluation of the generalization ability using VC dimension.

Namely, in cases where training data set X is not linearly separable, we map the original data set X to a feature space Z by some nonlinear map ϕ.

Increasing the dimension of the feature space, it is expected that the mapped data set becomes linearly separable. We try to find linear classifiers with maximal margin in the feature space. Letting $\boldsymbol{z}_i = \phi(\boldsymbol{x}_i)$, the separating hyperplane with maximal margin can be given by solving the following problem with the normalization $\boldsymbol{w}^T\boldsymbol{z} + b = \pm 1$ at points with the minimum interior deviation:

$$\begin{array}{lll} \text{minimize} & ||\boldsymbol{w}|| & (\text{SVM}_{hard})_P \\ \text{subject to} & y_i\left(\boldsymbol{w}^T\boldsymbol{z}_i + b\right) \geqq 1,\ i = 1, \ldots, \ell. & \end{array}$$

Several kinds of norm are possible. When $||\boldsymbol{w}||_2$ is used, the problem is reduced to quadratic programming, while the problem with $||\boldsymbol{w}||_1$ or $||\boldsymbol{w}||_\infty$ is reduced to linear programming (see, e.g., [12]).

Dual problem of $(\text{SVM}_{hard})_P$ with $\frac{1}{2}||\boldsymbol{w}||_2^2$ is

$$\begin{array}{lll} \text{maximize} & \displaystyle\sum_{i=1}^{\ell} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{\ell} \alpha_i\alpha_j y_i y_j \phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}_j) & (\text{SVM}_{hard})_D \\ \text{subject to} & \displaystyle\sum_{i=1}^{\ell} \alpha_i y_i = 0, & \\ & \alpha_i \geqq 0,\ i = 1, \ldots, \ell. & \end{array}$$

Using the kernel function $K(\boldsymbol{x}, \boldsymbol{x}') = \phi(\boldsymbol{x})^T\phi(\boldsymbol{x}')$, the problem $(\text{SVM}_{hard})_D$ can be reformulated as follows:

$$\begin{array}{lll} \text{maximize} & \displaystyle\sum_{i=1}^{\ell} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{\ell} \alpha_i\alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) & (\text{SVM}_{hard}) \\ \text{subject to} & \displaystyle\sum_{i=1}^{\ell} \alpha_i y_i = 0, & \\ & \alpha_i \geqq 0,\ i = 1, \ldots, \ell. & \end{array}$$

Several kinds of kernel functions have been suggested: among them, q-polynomial

$$K(\boldsymbol{x}, \boldsymbol{x}') = (\boldsymbol{x}^T\boldsymbol{x}' + 1)^q$$

and Gaussian

$$K(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\frac{||\boldsymbol{x} - \boldsymbol{x}'||^2}{r^2}\right)$$

are most popularly used.

8.3 Review of Multi-objective Programming and Goal Programming

Multi-objective programming (MOP) problems are formulated as follows:

$$\text{(MOP)} \qquad \text{Maximize} \quad g(\boldsymbol{x}) \equiv (g_1(\boldsymbol{x}),\ g_2(\boldsymbol{x}), \ldots,\ g_p(\boldsymbol{x}))$$
$$\text{over} \ \ \boldsymbol{x} \in X.$$

The constraint set X may be given by

$$c_j(\boldsymbol{x}) \leqq 0, \qquad j = 1, \ldots, m,$$

and/or a subset of R^n itself. For the problem (MOP), Pareto solutions are candidates of final decision ($\hat{\boldsymbol{x}}$ is said *Pareto optimal*, if there is no better solution $\boldsymbol{x} \in X$ other than $\hat{\boldsymbol{x}}$).

In general, there may be many Pareto solutions. The final decision is made among them taking the total balance over all criteria into account. This is a problem of value judgment of decision maker (in abbreviation, DM). The totally balancing over criteria is usually called *trade-off.* It is important to help DM to trade-off easily in practical decisin making problems.

There have been developed several kinds of methods for multi-objective programming (see, e.g., Steuer [21], Chankong-Haims [4], Sawaragi-Nakayama-Tanino [18], Nakayama [15], Miettinen [14]). Among them, interactive multi-objective programming methods, which were developed remarkably in 1980's, have been observed to be effective in various fields of practial problems. Those methods search a solution in an interactive way with DM while eliciting information on his/her value judgment.

On the other hand, Goal Programming (GP) was developed by Charnes-Cooper [5] much earlier than interactive programming methods. The idea was originated from getting rid of no feasible solution in usual mathematical programming. Namely, many constraints should be regarded as "goal" to be attained, and we try to find a solution which attains those goals as much as possible.

For example, suppose that we want to make

$$g_i(\boldsymbol{x}) \geqq \overline{g}_i, \quad i = 1, \ldots, p.$$

Introducing the degree of overattainment (or surplus, or interior deviation) η_i and the degree of unattainment (or slackness, or exterior deviation) ξ_i, we have the following goal programming formulation:

$$\begin{aligned} &\text{minimize} && \sum_{i=1}^{p} h_i \xi_i && (\text{GP}_0) \\ &\text{subject to} && g_i(\boldsymbol{x}) - \overline{g}_i = \eta_i - \xi_i, \\ & && \xi_i,\ \eta_i \geqq 0,\ i = 1, \ldots, p \\ & && \boldsymbol{x} \in X. \end{aligned}$$

where h_i $(i = 1, \ldots, p)$ are positive weighting parameters which are given by DMs.

It should be noted that in order for η_i and ξ_i in the above formulation to have the meaning of the degree of overattainment and the degree of unattainment, respectively, the relation $\xi_i \cdot \eta_i = 0$ has to be satisfied. The above formulation assures this property due to the following lemma (Lemma 7.3.1 of [18]):

Lemma 1. *Let $\boldsymbol{\xi}$ and $\boldsymbol{\eta}$ be vectors of R^p. Then consider the following problem:*

$$\begin{array}{ll} \textit{minimize} & P(\boldsymbol{\xi}, \boldsymbol{\eta}) \\ \textit{subject to} & g_i(\boldsymbol{x}) - \overline{g}_i = \eta_i - \xi_i, \\ & \xi_i,\ \eta_i \geqq 0,\ i = 1, \ldots, p, \\ & \boldsymbol{x} \in X. \end{array}$$

Suppose that the function P is monotononically increasing with respect to elements of $\boldsymbol{\xi}$ and $\boldsymbol{\eta}$ and strictly monotonically increasing with respect to at least either ξ_i or η_i for each i $(i = 1, \ldots, p)$. Then, the solution $\hat{\boldsymbol{\xi}}$ and $\hat{\boldsymbol{\eta}}$ to the preceding problem satisfy

$$\hat{\xi}_i \hat{\eta}_i = 0, \quad i = 1, \ldots, p.$$

In the original formulation of goal programming, once a solution which attains every goal, no efforts are made for further improvement. Therefore, the obtained solution by goal programming is not necessarily Pareto optimal. This is due to the fact that the idea of goal programming is based on "satisficing" rather than "optimization".

In order to overcome this difficulty, we can put the degree of overattainment in the objective function in (GP$_0$) as follows:.

$$\begin{array}{ll} \text{minimize} & \displaystyle\sum_{i=1}^{p} h_i \xi_i - \sum_{i=1}^{p} k_i \eta_i \\ \text{subject to} & g_i(\boldsymbol{x}) - \overline{g}_i = \eta_i - \xi_i, \\ & \xi_i,\ \eta_i \geqq 0,\ i = 1, \ldots, p, \\ & \boldsymbol{x} \in X. \end{array} \tag{GP$_1$}$$

Note that if the relation $h_i > k_i$ for each $i = 1, \ldots, p$ holds, then the relation $\xi_i \eta_i = 0$ for each $i = 1, \ldots, p$ is satisfied at the solution. This follows in a similar fashion to Lemma 1 by considering

$$\sum_{i=1}^{p} h_i \xi_i - \sum_{i=1}^{p} k_i \eta_i = \sum_{i=1}^{p} k_i (\xi_i - \eta_i) + \sum_{i=1}^{p} (h_i - k_i) \xi_i.$$

Moreover, if $k_i = h_i$ for each $i = 1, \ldots, p$, then by substituting the right hand side of the equality constraints of (GP$_1$) into the objective function we have

$$\begin{array}{lll} \text{maximize} & \displaystyle\sum_{i=1}^{p} h_i(g_i(\boldsymbol{x}) - \overline{g}_i) & (\text{MOP/GP}_0) \\ \text{subject to} & \boldsymbol{x} \in X. & \end{array}$$

Since the term of $-\overline{g}_i$ does not affect to maximizing the objective function, it can be removed. Namely the formulation (MOP/GP$_0$) is reduced to the usual scalarization using the linearly weighted sum in multi-objective programming.

However, the scalarization of linearly weighted sum has another drawbacks: e.g., it can not yield solutions on nonconvex parts of the Pareto frontier. To overcome this, the formulation of improvement of the worst level of objective function as much as possibel is applied as follows:

$$\begin{array}{lll} \text{maximize} & \eta & (\text{MOP/GP}_1) \\ \text{subject to} & g_i(\boldsymbol{x}) - \overline{g}_i \geqq \eta, \quad i = 1, \ldots, p, & \\ & \boldsymbol{x} \in X. & \end{array}$$

The solution to (MOP/GP$_1$) is guaranteed to be weakly Pareto optimal. Further discussion on scalarization functions can be seen in the literatures ([21], [4], [18], [15], [14]).

8.4 MOP/GP Approaches to Pattern Classification

In 1981, Freed-Glover suggested to get just a hyperplane separating two classes with as few misclassified data as possible by using goal programming [9] (see also [8]). Let ξ_i denote the exterior deviation which is a deviation from the hyperplane of a point $\boldsymbol{x}_i$ improperly classified. Similarly, let η_i denote the interior deviation which is a deviation from the hyperplane of a point $\boldsymbol{x}_i$ properly classified. Some of main objectives in this approach are as follows:

i) Minimize the maximum exterior deviation (decrease errors as much as possible)

ii) Maximize the minimum interior deviation (i.e., maximize the margin)

iii) Maximize the weighted sum of interior deviation

iv) Minimize the weighted sum of exterior deviation

Although many models have been suggested, the one considering iii) and iv) above may be given by the following linear goal programming:

$$\begin{array}{lll} \text{minimize} & \displaystyle\sum_{i=1}^{\ell}(h_i\xi_i - k_i\eta_i) & \text{(GP)} \\ \text{subject to} & y_i(\boldsymbol{x}_i^T\boldsymbol{w} + b) = \eta_i - \xi_i, & \\ & \xi_i,\ \eta_i \geqq 0,\ i = 1, \dots, \ell, & \end{array}$$

where since $y_i = +1$ or -1 according to $\boldsymbol{x}_i \in \mathcal{A}$ or $\boldsymbol{x}_i \in \mathcal{B}$, two equations $\boldsymbol{x}_i^T\boldsymbol{w} + b = \eta_i - \xi_i$ for $\boldsymbol{x}_i \in \mathcal{A}$ and $\boldsymbol{x}_i^T\boldsymbol{w} + b = -\eta_i + \xi_i$ for $\boldsymbol{x}_i \in \mathcal{B}$ can be reduced to the following one equation

$$y_i(\boldsymbol{x}_i^T\boldsymbol{w} + b) = \eta_i - \xi_i.$$

Here, h_i and k_i are positive constants. As was stated in the preceding section, if $h_i > k_i$ for $i = 1, \dots, \ell$, then we have $\xi_i\eta_i = 0$ for every $i = 1, \dots, \ell$ at the solution to (GP). Hence then, ξ_i and η_i are assured to have the meaning of the exterior deviation and the interior deviation respectively at the solution.

It should be noted that the above formulation may yield some unacceptable solutions such as $\boldsymbol{w} = 0$ and unbounded solution. In the goal programming approach to linear classifiers, therefore, some appropriate normality condition must be imposed on $\boldsymbol{w}$ in order to provide a bounded nontrivial optimal solution. One of such normality conditions is $||\boldsymbol{w}|| = 1$.

If the classification problem is linearly separable, then using the normalization $||\boldsymbol{w}|| = 1$, the separating hyperplane $H : \boldsymbol{w}^T\boldsymbol{x} + b = 0$ with maximal margin can be given by solving the following problem [3]:

$$\begin{array}{lll} \text{maximize} & \eta & (\text{MOP/GP}_2) \\ \text{subject to} & y_i(\boldsymbol{x}_i^T\boldsymbol{w} + b) \geqq \eta,\ \ i = 1, \dots, \ell, & \\ & ||\boldsymbol{w}|| = 1. & \end{array}$$

However, this normality condition makes the problem to be of nonlinear optimization. Instead of maximizing the minimum interior deviation in (MOP/GP$_2$), we can use the following equivalent formulation with the normalization $\boldsymbol{x}^T\boldsymbol{w} + b = \pm 1$ at points with the minimum interior deviation [13]:

$$\begin{array}{lll} \text{minimize} & ||\boldsymbol{w}|| & (\text{MOP/GP}_2') \\ \text{subject to} & y_i\left(\boldsymbol{x}_i^T\boldsymbol{w} + b\right) \geqq \eta,\ \ i = 1, \dots, \ell, & \\ & \eta = 1. & \end{array}$$

This formulation is the same as the one used in SVM.

8.5 Soft Margin SVM

Separating two sets $\mathcal{A}$ and $\mathcal{B}$ completely is called the hard margin method, which tends to make overlearning. This implies the hard margin method is easily affected by noise. In order to overcome this difficulty, the soft margin method is introduced. The soft margin method allows some slight error which is represented by slack variables (exterior deviation) ξ_i $(i = 1, \dots, \ell)$. Using the trade-off parameter C between minimizing $||\boldsymbol{w}||$ and minimizing $\sum_{i=1}^{\ell} \xi_i$, we have the following formulation for the soft margin method:

$$\begin{array}{lll} \text{minimize} & \dfrac{1}{2}||\boldsymbol{w}||_2^2 + C\displaystyle\sum_{i=1}^{\ell} \xi_i & (\mathrm{SVM}_{soft})_P \\ \text{subject to} & y_i \left(\boldsymbol{w}^T \boldsymbol{z}_i + b\right) \geqq 1 - \xi_i, & \\ & \xi_i \geqq 0, \ \ i = 1, \dots, \ell. & \end{array}$$

Using a kernel function in the dual problem yields

$$\begin{array}{lll} \text{maximize} & \displaystyle\sum_{i=1}^{\ell} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) & (\mathrm{SVM}_{soft}) \\ \text{subject to} & \displaystyle\sum_{i=1}^{\ell} \alpha_i y_i = 0, & \\ & 0 \leqq \alpha_i \leqq C, \ i = 1, \dots, \ell. & \end{array}$$

It can be seen that the idea of soft margin method is the same as the goal programming approach to linear classifiers. This idea was used in an extension of MSM by Benett [2]. Not only exterior deviations but also interior deviations can be considered in SVM. Such MOP/GP approaches to SVM are discussed by the authors and their coresearchers [1], [16], [23]. When applying GP approaches, it was pointed out in Section 3 that we need some normality condition in order to avoid unacceptable solutions.

Glover suggested the following necessary and sufficient condition for avoiding unacceptable solutions [10]:

$$\left(-l_{\mathcal{A}} \sum_{i \in I_{\mathcal{B}}} \boldsymbol{x}_i + l_{\mathcal{B}} \sum_{i \in I_{\mathcal{A}}} \boldsymbol{x}_i\right)^T \boldsymbol{w} = 1, \tag{8.1}$$

where $l_{\mathcal{A}}$ and $l_{\mathcal{B}}$ denote the number of data for the category $\mathcal{A}$ and $\mathcal{B}$, respectively. Geometrically, the normalization (8.1) means that the distance between two hyperplanes passing through centers of data respectively for $\mathcal{A}$ and $\mathcal{B}$ is scaled by $l_{\mathcal{A}} l_{\mathcal{B}}$.

Lately, taking into account the objectives (ii) and (iv) of goal programming stated in the previous section, Schölkopf et al. [19] suggested ν-support vector algorithm:

$$\begin{array}{lll}\text{minimize} & \frac{1}{2}||\boldsymbol{w}||_2^2 - \nu\rho + \frac{1}{\ell}\sum_{i=1}^{\ell}\xi_i & (\nu\text{−SVM})_P \\ \text{subject to} & y_i\left(\boldsymbol{w}^T\boldsymbol{z}_i + b\right) \geqq \rho - \xi_i, & \\ & \rho \geqq 0,\ \ \xi_i \geqq 0,\ \ i = 1,\ldots,\ell. & \end{array}$$

where $0 \leqq \nu \leqq 1$ is a parameter.

Compared with the existing soft margin algorithm, one of the differences is that the parameter C for slack variables does not appear, and another difference is that the new variable ρ appears in the above formulation. The problem $(\nu\text{−SVM})_P$ maximizes the variable ρ which corresponds to the minimum interior deviation (i.e., the minimum distance between the separating hyperplane and correctly classified points).

The Lagrangian dual problem to the problem $(\nu\text{−SVM})_P$ is as follows:

$$\begin{array}{lll}\text{maximize} & -\frac{1}{2}\sum_{i,j=1}^{\ell} y_i y_j \alpha_i \alpha_j K\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) & (\nu\text{−SVM}) \\ \text{subject to} & \sum_{i=1}^{\ell} y_i\alpha_i = 0, & \\ & \sum_{i=1}^{\ell} \alpha_i \geqq \nu, & \\ & 0 \leqq \alpha_i \leqq \frac{1}{\ell},\ \ i = 1,\ldots,\ell. & \end{array}$$

8.6 Extensions of SVM by MOP/GP

In this section, we propose various algorithms of SVM considering both slack variables for misclassified data points (i.e., exterior deviations) and surplus variables for correctly classified data points (i.e., interior deviations).

8.6.1 Total Margin Algorithm

In order to minimize the slackness and to maximize the surplus, we have the following optimization problem:

$$\begin{array}{lll}\text{minimize} & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + C_1\sum_{i=1}^{\ell}\xi_i - C_2\sum_{i=1}^{\ell}\eta_i & (\text{SVM}_{total})_P \\ \text{subject to} & y_i\left(\boldsymbol{w}^T\boldsymbol{z}_i + b\right) \geqq 1 - \xi_i + \eta_i, & \\ & \xi_i \geqq 0,\ \ \eta_i \geqq 0,\ \ i = 1,\ldots,\ell, & \end{array}$$

where C_1 and C_2 are chosen in such a way that $C_1 > C_2$ which ensures that at least one of ξ_i and η_i becomes zero. The Lagrangian function for the problem $(\mathrm{SVM}_{total})_P$ is

$$\begin{aligned} L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\eta}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma}) = \frac{1}{2}\|\boldsymbol{w}\|_2^2 &+ C_1 \sum_{i=1}^{\ell} \xi_i - C_2 \sum_{i=1}^{\ell} \eta_i \\ &- \sum_{i=1}^{\ell} \alpha_i \left[y_i \left(\boldsymbol{w}^T \boldsymbol{z}_i + b \right) - 1 + \xi_i - \eta_i \right] \\ &- \sum_{i=1}^{\ell} \beta_i \xi_i - \sum_{i=1}^{\ell} \gamma_i \eta_i, \end{aligned}$$

where $\alpha_i \geqq 0, \ \ \beta_i \geqq 0$ and $\gamma_i \geqq 0$.

Differentiating the Lagrangian function with respect to $\boldsymbol{w}$, b, $\boldsymbol{\xi}$ and $\boldsymbol{\eta}$ yields the following conditions:

$$\begin{aligned} \frac{\partial L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\eta}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})}{\partial \boldsymbol{w}} &= \boldsymbol{w} - \sum_{i=1}^{\ell} \alpha_i y_i \boldsymbol{z}_i = \boldsymbol{0}, \\ \frac{\partial L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\eta}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})}{\partial \xi_i} &= C_1 - \alpha_i - \beta_i = 0, \\ \frac{\partial L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\eta}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})}{\partial \eta_i} &= -C_2 + \alpha_i - \gamma_i = 0, \\ \frac{\partial L(\boldsymbol{w}, b, \boldsymbol{\xi}, \boldsymbol{\eta}, \boldsymbol{\alpha}, \boldsymbol{\beta}, \boldsymbol{\gamma})}{\partial b} &= \sum_{i=1}^{\ell} \alpha_i y_i = 0. \end{aligned}$$

Substituting the above stationary conditions into the Lagrangian function L and using kernel representation, we obtain the following dual optimization problem:

$$\begin{aligned} &\text{maximize} && \sum_{i=1}^{\ell} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{\ell} y_i y_j \alpha_i \alpha_j K\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right) && (\mathrm{SVM}_{total}) \\ &\text{subject to} && \sum_{i=1}^{\ell} y_i \alpha_i = 0, \\ &&& C_2 \leqq \alpha_i \leqq C_1, \ \ i = 1, \ldots, \ell. \end{aligned}$$

Let $\boldsymbol{\alpha}^*$ be the optimal solution to the problem (SVM_{total}). Then, the discrimination function can be written by

$$f(\phi(\boldsymbol{x})) = \sum_{i=1}^{\ell} \alpha_i^* y_i K\left(\boldsymbol{x}, \boldsymbol{x}_i\right) + b.$$

The offset b is given as follows: Let n_+ be the number of $\boldsymbol{x}_j$ with $C_2 < \alpha_j^* < C_1$ and $y_j = +1$, and let n_- be the number of $\boldsymbol{x}_j$ with $C_2 < \alpha_j^* < C_1$ and $y_j = -1$, respectively. From the Karush-Kuhn-Tucker complementarity conditions, if $C_2 < \alpha_j^* < C_1$, then $\beta_j > 0$ and $\gamma_j > 0$. This implies that $\xi_j = \eta_j = 0$. Then,

$$b^* = \frac{1}{n_+ + n_-}\left((n_+ - n_-) - \sum_{j=1}^{n_+ + n_-}\sum_{i=1}^{\ell} y_i \alpha_i^* K\left(\boldsymbol{x}_i, \boldsymbol{x}_j\right)\right).$$

8.6.2 $\mu-$SVM

Minimizing the worst slackness and maximizing the sum of surplus, we have a reverse formulation of $\nu-$SVM. We introduce a new variable σ which represents the maximal distance between the separating hyperplane and misclassified data points. Thus, the following problem is obtained:

$$\begin{array}{lll} \text{minimize} & \dfrac{1}{2}\|\boldsymbol{w}\|_2^2 + \mu\sigma - \dfrac{1}{\ell}\displaystyle\sum_{i=1}^{\ell}\eta_i & (\mu\text{–SVM})_P \\ \text{subject to} & y_i\left(\boldsymbol{w}^T\boldsymbol{z}_i + b\right) \geqq \eta_i - \sigma, & \\ & \sigma \geqq 0, \ \ \eta_i \geqq 0, \ \ i = 1, \ldots, \ell, & \end{array}$$

where μ is a parameter which reflects the trade-off between σ and the sum of η_i.

The Lagrangian function for the problem $(\mu\text{–SVM})_P$ is

$$\begin{aligned} L(\boldsymbol{w}, b, \boldsymbol{\eta}, \sigma, \boldsymbol{\alpha}, \boldsymbol{\beta}, \gamma) = {} & \frac{1}{2}\|\boldsymbol{w}\|_2^2 + \mu\sigma - \frac{1}{\ell}\sum_{i=1}^{\ell}\eta_i \\ & - \sum_{i=1}^{\ell}\alpha_i\left[y_i\left(\boldsymbol{w}^T\boldsymbol{z}_i + b\right) - \eta_i + \sigma\right] - \sum_{i=1}^{\ell}\beta_i\eta_i - \gamma\sigma, \end{aligned}$$

where $\alpha_i \geqq 0$, $\beta_i \geqq 0$ and $\gamma \geqq 0$.

Differentiating the Lagrangian function with respect to $\boldsymbol{w}$, b, $\boldsymbol{\eta}$ and σ yields the following conditions:

$$\begin{aligned} \frac{\partial L(\boldsymbol{w}, b, \boldsymbol{\eta}, \sigma, \boldsymbol{\alpha}, \boldsymbol{\beta}, \gamma)}{\partial \boldsymbol{w}} &= \boldsymbol{w} - \sum_{i=1}^{\ell}\alpha_i y_i \boldsymbol{z}_i = \boldsymbol{0}, \\ \frac{\partial L(\boldsymbol{w}, b, \boldsymbol{\eta}, \sigma, \boldsymbol{\alpha}, \boldsymbol{\beta}, \gamma)}{\partial \eta_i} &= -\frac{1}{\ell} + \alpha_i - \beta_i = 0, \\ \frac{\partial L(\boldsymbol{w}, b, \boldsymbol{\eta}, \sigma, \boldsymbol{\alpha}, \boldsymbol{\beta}, \gamma)}{\partial \sigma} &= \mu - \sum_{i=1}^{\ell}\alpha_i - \gamma = 0, \\ \frac{\partial L(\boldsymbol{w}, b, \boldsymbol{\eta}, \sigma, \boldsymbol{\alpha}, \boldsymbol{\beta}, \gamma)}{\partial b} &= \sum_{i=1}^{\ell}\alpha_i y_i = 0. \end{aligned}$$

Substituting the above stationary conditions into the Lagrangian function L, we obtain the following dual optimization problem:

$$\begin{aligned}
&\text{maximize} && -\frac{1}{2}\sum_{i,j=1}^{\ell}\alpha_i\alpha_j y_i y_j K\left(\boldsymbol{x}_i,\boldsymbol{x}_j\right) && (\mu\text{–SVM})\\
&\text{subject to} && \sum_{i=1}^{\ell}\alpha_i y_i = 0,\\
& && \sum_{i=1}^{\ell}\alpha_i \leqq \mu,\\
& && \alpha_i \geqq \frac{1}{\ell},\ \ i=1,\ldots,\ell.
\end{aligned}$$

Let $\boldsymbol{\alpha}^*$ be the optimal solution to the problem (μ–SVM). To compute the offset b, we take the set $\mathcal{A}$ of $\boldsymbol{x}_j$ which is the same size n with $\frac{1}{\ell} < \alpha_j^*$. From the Karush-Kuhn-Tucker complementarity conditions, if $\frac{1}{\ell} < \alpha_j^*$, then $\beta_j > 0$ which implies $\eta_j = 0$. Thus,

$$b^* = -\frac{1}{2n}\sum_{\boldsymbol{x}_j\in\mathcal{A}}\sum_{i=1}^{\ell}\alpha_i^* y_i K\left(\boldsymbol{x}_i,\boldsymbol{x}_j\right).$$

8.6.3 μ–ν–SVM

Applying SVM$_{total}$ and μ–SVM, all training points become support vectors due to the second constraint of the problem (SVM$_{total}$) and the third constraint of the problem (μ–SVM). In other words, the algorithms (SVM$_{total}$) and (μ–SVM) lack in the sparsity of support vectors. In order to overcome this problem in (SVM$_{total}$) and (μ–SVM), we suggest the following formulation, which combines the ideas of ν–SVM and μ–SVM:

$$\begin{aligned}
&\text{minimize} && \frac{1}{2}\|\boldsymbol{w}\|_2^2 - \nu\rho + \mu\sigma && (\mu-\nu\text{–SVM})_P\\
&\text{subject to} && y_i\left(\boldsymbol{w}^T\boldsymbol{z}_i + b\right) \geqq \rho - \sigma,\ \ i=1,\ldots,\ell,\\
& && \rho \geqq 0,\ \ \sigma \geqq 0,
\end{aligned}$$

where ν and μ are parameters.

The Lagrangian function to the problem (μ–ν–SVM)$_P$ is

$$\begin{aligned}
L(\boldsymbol{w},b,\rho,\sigma,\boldsymbol{\alpha},\beta,\gamma) = {} & \frac{1}{2}\|\boldsymbol{w}\|_2^2 - \nu\rho + \mu\sigma\\
& -\sum_{i=1}^{\ell}\alpha_i\left[y_i\left(\boldsymbol{w}^T\boldsymbol{z}_i + b\right) - \rho + \sigma\right] - \beta\rho - \gamma\sigma,
\end{aligned}$$

where $\alpha_i \geqq 0$, $\beta \geqq 0$ and $\gamma \geqq 0$.

Differentiating Lagrangian function with respect to $\boldsymbol{w}$, b, ρ and σ yields the four conditions

$$\begin{aligned}
\frac{\partial L(\boldsymbol{w}, b, \rho, \sigma, \boldsymbol{\alpha}, \beta, \gamma)}{\partial \boldsymbol{w}} &= \boldsymbol{w} - \sum_{i=1}^{\ell} \alpha_i y_i \boldsymbol{z}_i = \boldsymbol{0}, \\
\frac{\partial L(\boldsymbol{w}, b, \rho, \sigma, \boldsymbol{\alpha}, \beta, \gamma)}{\partial \rho} &= -\nu + \sum_{i=1}^{\ell} \alpha_i - \beta = 0, \\
\frac{\partial L(\boldsymbol{w}, b, \rho, \sigma, \boldsymbol{\alpha}, \beta, \gamma)}{\partial \sigma} &= \mu - \sum_{i=1}^{\ell} \alpha_i - \gamma = 0, \\
\frac{\partial L(\boldsymbol{w}, b, \rho, \sigma, \boldsymbol{\alpha}, \beta, \gamma)}{\partial b} &= \sum_{i=1}^{\ell} \alpha_i y_i = 0.
\end{aligned}$$

Substituting the above stationary conditions into the Lagrangian function L and using kernel representation, we obtain the following dual optimization problem:

$$\begin{aligned}
&\text{maximize} && -\frac{1}{2} \sum_{i,j=1}^{\ell} \alpha_i \alpha_j y_i y_j K(\boldsymbol{x}_i, \boldsymbol{x}_j) && (\mu - \nu\text{–SVM}) \\
&\text{subject to} && \sum_{i=1}^{\ell} \alpha_i y_i = 0, \\
& && \nu \leqq \sum_{i=1}^{\ell} \alpha_i \leqq \mu, \\
& && \alpha_i \geqq 0, \ \ i = 1, \ldots, \ell.
\end{aligned}$$

Letting $\boldsymbol{\alpha}^*$ be the optimal solution to the problem (μ–ν–SVM), the offset b^* can be chosen easily for any i satisfying $\alpha_i^* > 0$. Otherwise, b^* can be obtained by the similar way with the decision of the b^* in the other algorithms.

8.7 Numerical Examples

In order to investigate the performance of our proposed method, we compare the results for four data sets in the following: (The data can be downloaded from `http://www.ics.uci.edu/˜ mlearn/MLSummary.html`)

I. MONK's Problem (all data sets with 7 attributes)
   a) case 1
      i. training : 124 instances ($\mathcal{A}$: 62 instances, $\mathcal{B}$: 62 instances)
      ii. test : 432 instances ($\mathcal{A}$: 216 instances, $\mathcal{B}$: 216 instances)

b) case 2
    i. training : 169 instances ($\mathcal{A}$: 64 instances, $\mathcal{B}$: 105 instances)
    ii. test : 432 instances ($\mathcal{A}$: 142 instances, $\mathcal{B}$: 290 instances)
c) case 3
    i. training : 122 instances ($\mathcal{A}$: 60 instances, $\mathcal{B}$: 62 instances)
    ii. test : 432 instances ($\mathcal{A}$: 228 instances, $\mathcal{B}$: 204 instances)

II. Cleveland heart-disease from Long Beach and Cleveland Clinic Foundation : 303 instances ($\mathcal{A}$: 164 instances, $\mathcal{B}$: 139 instances) with 14 attributes

III. BUPA liver disorders from BUPA Medical Research Ltd. : 345 instances ($\mathcal{A}$: 200 instances, $\mathcal{B}$: 145 instances) with 7 attributes

IV. PIMA Indians diabetes database : 768 instances ($\mathcal{A}$: 268 instances, $\mathcal{B}$: 500 instances) with 9 attributes

In the following numerical experiments, QP solver of MATLAB was used for solving QP problems in SVM formulations; Gaussian kernels with $r = 1.0$ were used with the data normalization for each sample $\boldsymbol{x}_i$

$$\tilde{x}_{ki} = \frac{x_{ki} - \mu_k}{\sigma_k}$$

where μ_k and σ_k are the mean value and the standard deviation of k-th component of given the sample data $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_p\}$, respectively. For parameters in applying GP model, we set $h_1 = h_2 = \cdots = h_\ell = C_1$ and $k_1 = k_2 = \cdots = k_\ell = C_2$.

For the dataset I, we followed both the training data and the test data as in the benchmark of the WEB site. Tables 8.1–8.6 compare the classification rates by using the existing algorithms (GP), (SVM$_{soft}$) and (ν−SVM) with the proposed algorithms (SVM$_{total}$), (μ−SVM) and (μ−ν−SVM), respectively.

For the datasets II and III, we adopt the 'cross validation test' method which makes 10 trials for randomly selected training data of 70% from the original data set and the test data of the rest 30%. Tables 8.7–8.18 compare the average (AVE) and the standard deviation (STDV) of classification rates by using the existing algorithms (GP), (SVM$_{soft}$) and (ν−SVM) with the proposed algorithms (SVM$_{total}$), (μ−SVM) and (μ−ν−SVM), respectively.

For the dataset IV, there is an unbalance between the number of elements of two classes: $\mathcal{A}$ (tested positive for diabetes) has 268 elements, while $\mathcal{B}$ (tested non-positive for diabetes) 500 elements. We selected randomly 70% from the whole data set as the training samples, and set the rest 30% as the test samples. We compared the results by (GP), (SVM$_{soft}$) and (SVM$_{total}$ ν−SVM) with the proposed algorithms (SVM$_{total}$), (μ−SVM) and (μ−ν−SVM) as seen in Tables 8.19–8.24, respectively.

Table 8.25 shows the rate of support vectors in terms of percentage for each problem and each method.

Throughout our numerical experiments, it has been observed that even though the result depends on the value of parameters, the family of SVM using MOP/GP such as ν−SVM, SVM$_{total}$, μ−SVM and $\mu - \nu$−SVM show a relatively good performance in comparison with the simple SVM$_{soft}$. Sometimes unbalanced data sets cause a difficulty in predicting the category with fewer samples. In our experiments, MONK (case2) and PIMA diabetes are of this kind. It can be seen in those problems that the classification ability for the class with fewer samples is much sensitive to the value of C in SVM$_{soft}$. In other words, we have to select the appropriate value of C in SVM$_{soft}$ carefully in order to attain some reasonable classification rate for unbalanced data sets. SVM$_{total}$ and $\mu-\nu-$SVM, however, have advantage over SVM$_{soft}$ in classification rate of the class with fewer elements. In addition, the data set of MONK seems not to be linearly separated. In this example, therefore, SVMs using MOP/GP show much better performance than the mere GP.

Table 8.1. Classification Rate by GP for MONK's Problem

	C_1	1			10			100			
	C_2	0.001	0.01	0.1	0.01	0.1	1	0.1	1	10	average
Training	case 1	73.39	73.39	73.39	73.39	73.39	71.77	73.39	73.39	73.39	73.21
	case 2	63.31	63.31	63.31	63.31	63.31	63.91	63.31	63.31	65.09	63.57
	$\mathcal{A}$	53.13	53.13	45.31	51.56	51.56	48.44	51.56	51.56	45.31	50.17
	$\mathcal{B}$	69.52	69.52	74.29	70.48	70.48	73.33	70.48	70.48	77.14	71.75
	case 3	88.52	88.52	88.52	88.52	88.52	88.52	88.52	88.52	88.52	88.52
Test	case 1	66.67	66.67	66.67	66.67	66.67	65.97	66.67	66.67	66.67	66.59
	case 2	58.33	58.33	58.33	59.03	59.03	59.26	59.03	59.03	61.11	59.05
	$\mathcal{A}$	39.44	39.44	35.92	40.14	40.14	37.32	40.14	40.14	35.92	38.73
	$\mathcal{B}$	67.59	67.59	70.69	68.28	68.28	70.00	68.28	68.28	73.45	69.16
	case 3	88.89	88.89	88.89	88.89	88.89	88.89	88.89	88.89	88.89	88.89

Table 8.2. Classification Rate by SVM$_{soft}$ for MONK's Problem

	C	0.1	1	10	100	average
Training	case 1	87.90	95.16	100	100	95.77
	case 2	62.13	85.80	100	100	86.98
	$\mathcal{A}$	0.00	64.06	100	100	66.02
	$\mathcal{B}$	100	99.05	100	100	40.84
	case 3	81.15	99.18	100	100	95.08
Test	case 1	78.94	83.80	92.36	92.36	86.86
	case 2	67.13	70.14	79.63	80.09	74.25
	$\mathcal{A}$	0.00	40.14	82.39	83.10	51.41
	$\mathcal{B}$	100	84.83	78.28	78.62	85.43
	case 3	69.44	95.83	91.67	91.67	87.15

Table 8.3. Classification Rate by $\nu-$SVM for MONK's Problem

	ν	0.1	0.2	0.3	0.4	0.5	0.6	0.7	average
Training	case 1	100	100	100	99.19	98.39	94.35	91.94	97.70
	case 2	100	100	100	98.82	98.82	95.27	88.17	97.30
	$\mathcal{A}$	100	100	100	96.88	96.88	89.06	70.31	93.30
	$\mathcal{B}$	100	100	100	100	100	99.05	99.05	99.73
	case 3	100	99.18	99.18	99.18	97.54	95.90	94.26	97.89
Test	case 1	92.36	92.13	91.20	88.43	87.04	84.03	80.56	87.96
	case 2	80.09	80.09	79.40	78.70	77.78	74.31	71.06	77.35
	$\mathcal{A}$	83.10	83.10	82.39	80.28	73.94	60.56	45.07	72.64
	$\mathcal{B}$	78.62	78.62	77.93	77.93	79.66	81.03	83.79	79.66
	case 3	91.67	94.44	95.14	96.06	95.60	93.52	92.13	94.08

Table 8.4. Classification Rate by SVM$_{total}$ for MONK's Problem

	C_1	1			10			100			average
	C_2	0.001	0.01	0.1	0.01	0.1	1	0.1	1	10	
Training	case 1	95.16	95.16	95.97	100	100	100	100	100	90.38	97.40
	case 2	86.98	87.57	88.76	100	100	100	100	100	80.47	93.75
	$\mathcal{A}$	70.31	71.88	76.56	100	100	100	100	100	100	90.97
	$\mathcal{B}$	97.14	97.14	96.19	100	100	100	100	100	68.57	95.45
	case 3	99.18	99.18	99.18	100	100	100	100	100	95.0	99.18
Test	case 1	84.49	84.26	84.03	92.59	92.59	86.57	92.59	86.57	79.40	87.01
	case 2	69.68	69.91	70.83	77.78	78.01	78.01	77.78	78.01	69.91	74.43
	$\mathcal{A}$	47.18	47.89	50.70	86.62	87.32	89.44	87.32	89.44	85.92	74.65
	$\mathcal{B}$	80.69	80.69	80.69	73.45	73.45	72.41	73.10	72.41	62.07	74.33
	case 3	95.83	95.83	96.06	91.90	91.90	91.90	91.90	91.90	90.51	93.08

Table 8.5. Classification Rate by $\mu-$SVM for MONK's Problem

	μ	1.3	1.4	1.5	1.6	1.7	1.8	1.9	2	average
Training	case 1	90.32	90.32	90.32	90.32	90.32	90.32	90.32	90.32	90.32
	case 2	71.01	71.01	71.01	71.01	71.01	71.01	71.01	71.01	71.01
	$\mathcal{A}$	100	100	100	100	100	100	100	100	100
	$\mathcal{B}$	53.33	53.33	53.33	53.33	53.33	53.33	53.33	53.33	53.33
	case 3	100	100	100	100	100	100	100	100	100
Test	case 1	77.46	77.46	77.46	77.46	77.46	77.46	77.46	77.46	77.46
	case 2	62.73	62.73	62.73	62.73	62.73	62.73	62.73	62.73	62.73
	$\mathcal{A}$	97.18	97.18	97.18	97.18	97.18	97.18	97.18	97.18	97.18
	$\mathcal{B}$	45.86	45.86	45.86	45.86	45.86	45.86	45.86	45.86	45.86
	case 3	93.52	93.52	93.52	93.52	93.52	93.52	93.52	93.52	93.52

Table 8.6. Classification Rate by $\mu - \nu -$SVM for MONK's Problem

	μ	1			10			100			
	ν	0.001	0.01	0.1	0.01	0.1	1	0.1	1	10	average
Training	case 1	100	100	100	100	100	100	100	100	100	100
	case 2	100	100	100	100	100	100	100	100	100	100
	$\mathcal{A}$	100	100	100	100	100	100	100	100	100	100
	$\mathcal{B}$	100	100	100	100	100	100	100	100	100	100
	case 3	100	100	100	100	100	100	100	100	100	100
Test	case 1	95.37	93.06	92.59	92.59	92.59	92.36	92.59	92.36	92.36	92.88
	case 2	80.56	75.69	75.46	75.46	75.46	80.09	75.46	80.09	80.09	77.60
	$\mathcal{A}$	95.77	92.96	92.96	92.96	92.96	83.10	92.96	83.10	83.10	89.98
	$\mathcal{B}$	73.10	67.24	66.90	66.90	66.90	78.62	66.90	78.62	78.62	71.53
	case 3	93.98	93.52	93.52	93.52	93.52	91.67	93.52	91.67	91.67	92.95

Table 8.7. Classification Rate by GP for Cleveland Heart-disease

C_1	1																	
C_2	0.001						0.01						0.1					
	training			test			training			test			training			test		
	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$
AVE	87.93	88.80	86.88	79.11	80.05	78.07	88.03	88.98	86.88	79.11	80.05	78.07	88.22	89.50	86.68	78.89	80.30	77.35
STD	1.07	1.21	2.05	3.61	5.73	5.14	1.03	1.22	2.05	3.61	5.73	5.14	0.93	1.47	1.94	3.02	5.24	4.48
C_1	10																	
C_2	0.01						0.1						1					
AVE	88.12	89.16	86.88	78.56	78.86	78.44	88.12	89.16	86.88	78.56	78.86	78.44	88.26	89.77	86.47	79.22	80.29	78.20
STD	1.41	1.56	2.17	2.77	5.32	5.45	1.41	1.56	2.17	2.77	5.32	5.45	1.07	1.44	2.23	3.26	5.77	5.28
C_1	100																	
C_2	0.1						1						10					
AVE	88.12	89.16	86.88	78.56	78.86	78.44	88.12	89.16	86.88	78.56	78.86	78.44	88.22	89.77	86.38	79.00	80.12	77.92
STD	1.41	1.56	2.17	2.77	5.32	5.45	1.41	1.56	2.17	2.77	5.32	5.45	1.08	1.30	1.95	3.08	5.54	5.84

Table 8.8. Classification Rate by SVM$_{soft}$ for Cleveland Heart-disease

C	0.01						0.1						1.0					
	training			test			training			test			training			test		
	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$
AVE	53.05	90.00	10.00	53.89	90.00	10.00	53.05	90.00	10.00	53.89	90.00	10.00	99.72	100	99.40	73.89	75.19	74.30
STD	1.67	30.00	30.00	7.36	30.00	30.00	1.67	30.00	30.00	7.36	30.00	30.00	0.23	0.00	0.49	4.31	11.86	16.65
C	10						100											
AVE	100	100	100	74.56	74.49	76.28	100	100	100	74.56	74.49	76.28						
STD	0.00	0.00	0.00	3.93	10.80	14.14	0.00	0.00	0.00	3.93	10.80	14.14						

Table 8.9. Classification Rate by ν−SVM for Cleveland Heart-disease

ν	0.1						0.2						0.3					
	training			test			training			test			training			test		
	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$
AVE	100	100	100	74.56	74.49	76.28	100	100	100	74.56	74.49	76.28	100	100	100	74.56	74.49	76.28
STD	0.00	0.00	0.00	3.93	10.80	14.14	0.00	0.00	0.00	3.93	10.80	14.14	0.00	0.00	0.00	3.93	10.80	14.14
ν	0.4						0.5						0.6					
AVE	100	100	100	74.67	74.87	76.04	100	100	100	74.33	74.45	75.72	99.91	100	99.80	74.33	74.64	75.48
STD	0.00	0.00	0.00	4.18	10.74	14.59	0.00	0.00	0.00	3.83	10.66	14.52	0.19	0.00	0.40	3.99	10.75	14.65
ν	0.7						0.8											
AVE	99.86	100	99.70	74.67	75.05	75.76	99.72	100	99.40	73.78	75.02	74.30						
STD	0.22	0.00	0.46	4.38	10.75	15.40	0.23	0.00	0.49	4.59	12.21	16.65						

Table 8.10. Classification Rate by SVM$_{total}$ for Cleveland Heart-disease

C_1	1																	
C_2	0.0001						0.001						0.01					
	training			test			training			test			training			test		
	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$
AVE	99.72	100	99.40	74.44	74.80	75.97	99.72	100	99.40	74.11	74.23	75.97	99.72	100	99.40	74.11	74.23	75.97
STD	0.23	0.00	0.49	4.99	11.80	17.23	0.23	0.00	0.49	4.87	11.99	17.23	0.23	0.00	0.49	4.87	11.99	17.23
C_1	10																	
C_2	0.001						0.01						0.1					
AVE	100	100	100	74.22	73.26	77.00	100	100	100	74.22	73.26	77.00	100	100	100	74.22	73.05	77.25
STD	0.00	0.00	0.00	3.47	10.62	13.93	0.00	0.00	0.00	3.47	10.62	13.93	0.00	0.00	0.00	3.47	10.56	14.14
C_1	100																	
C_2	0.01						0.1						1					
AVE	100	100	100	74.22	73.26	77.00	100	100	100	74.22	73.05	77.25	100	100	100	72.56	57.91	91.63
STD	0.00	0.00	0.00	3.47	10.62	13.93	0.00	0.00	0.00	3.47	10.56	14.14	0.00	0.00	0.00	3.37	5.67	5.76

Table 8.11. Classification Rate by μ−SVM for Cleveland Heart-disease

μ	1.2						· · ·						1.5					
	training			test			training			test			training			test		
	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$
AVE	99.81	99.67	100.00	81.00	82.25	79.72	· · ·			· · ·			99.81	99.67	100.00	81.00	82.25	79.72
STD	0.33	0.59	0.00	2.19	3.33	4.02	· · ·			· · ·			0.33	0.59	0.00	2.19	3.33	4.02
μ	1.6						· · ·						2.0					
AVE	99.81	99.67	100.00	81.00	82.25	79.72	· · ·			· · ·			99.81	99.67	100.00	81.00	82.25	79.72
STD	0.33	0.59	0.00	2.19	3.33	4.02	· · ·			· · ·			0.33	0.59	0.00	2.19	3.33	4.02

Table 8.12. Classification Rate by $\mu - \nu-$SVM for Cleveland Heart-disease

μ	1																	
ν	0.0001						0.001						0.01					
	training			test			training			test			training			test		
	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$
AVE	100	100	100	94.67	95.50	93.30	100	100	100	85.00	86.66	82.69	100	100	100	80.44	82.56	77.61
STD	0.00	0.00	0.00	1.71	2.74	3.49	0.00	0.00	0.00	2.91	4.30	4.90	0.00	0.00	0.00	2.73	5.00	4.14
μ	10																	
ν	0.001						0.01						0.1					
AVE	100	100	100	85.00	86.66	82.69	100	100	100	80.44	82.56	77.61	100	100	100	79.11	80.93	76.59
STD	0.00	0.00	0.00	2.91	4.30	4.90	0.00	0.00	0.00	2.73	5.00	4.14	0.00	0.00	0.00	2.80	4.82	4.22
μ	100																	
ν	0.01						0.1						1					
AVE	100	100	100	80.44	82.56	77.61	100	100	100	79.11	80.93	76.59	100	100	100	74.56	74.49	76.28
STD	0.00	0.00	0.00	2.73	5.00	4.14	0.00	0.00	0.00	2.80	4.82	4.22	0.00	0.00	0.00	3.93	10.80	14.14

Table 8.13. Classification Rate by GP for Liver Disorders

C_1	1																	
C_2	0.001						0.01						0.1					
	training			test			training			test			training			test		
	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$
AVE	71.32	75.48	65.57	69.71	73.17	64.92	71.32	75.62	65.38	69.71	73.50	64.39	72.31	77.61	64.97	70.10	74.89	63.51
STD	1.42	1.45	1.64	2.67	5.38	3.38	1.46	1.67	1.44	2.70	5.26	3.61	1.83	2.23	1.92	2.81	5.02	4.27
C_1	10																	
C_2	0.01						0.1						1					
AVE	71.36	75.48	65.67	69.71	73.17	64.92	71.45	75.62	65.67	69.81	73.34	64.92	72.31	77.75	64.78	70.10	75.06	63.30
STD	1.37	1.45	1.58	2.67	5.38	3.38	1.44	1.61	1.58	2.62	5.36	3.38	1.80	2.23	1.87	2.97	5.17	4.56
C_1	100																	
C_2	0.1						1						10					
AVE	71.36	75.48	65.67	69.71	73.17	64.92	71.45	75.62	65.67	69.81	73.34	64.92	72.31	77.75	64.78	70.10	75.06	63.30
STD	1.37	1.45	1.58	2.67	5.38	3.38	1.44	1.61	1.58	2.62	5.36	3.38	1.80	2.23	1.87	2.97	5.17	4.56

Table 8.14. Classification Rate by SVM$_{soft}$ for Liver Disorders

C	0.01						0.1						1					
	training			test			training			test			training			test		
	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$
AVE	58.02	100	0.00	57.86	100	0.00	58.02	100	0.00	57.86	100	0.00	86.69	93.79	76.89	70.10	85.05	49.79
STD	1.32	0.00	0.00	3.11	0.00	0.00	1.32	0.00	0.00	3.11	0.00	0.00	1.42	1.16	3.16	3.93	4.83	5.89
C	10						100											
AVE	95.29	96.29	93.91	66.12	73.70	56.22	99.46	99.36	99.61	63.20	69.54	54.92						
STD	1.34	0.80	2.81	3.72	2.39	8.67	0.32	0.49	0.48	4.20	5.31	7.59						

Table 8.15. Classification Rate by $\nu-$SVM for Liver Disorders

ν	0.1						0.2						0.3					
	training			test			training			test			training			test		
	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$
AVE	99.46	99.36	99.61	63.59	70.04	55.12	98.26	98.29	98.22	65.34	72.72	55.73	96.07	96.50	95.46	65.05	71.90	56.21
STD	0.32	0.49	0.48	3.34	5.20	6.55	0.40	0.49	1.15	3.01	4.61	8.44	0.89	1.01	1.75	3.07	2.95	8.74
ν	0.4						0.5						0.6					
AVE	93.47	95.51	90.64	67.86	76.06	57.12	91.98	95.15	87.63	68.64	78.52	55.44	90.12	94.45	84.15	69.22	80.86	53.54
STD	0.80	0.68	1.73	4.17	3.11	8.25	0.95	0.95	1.69	3.61	3.64	7.69	1.22	1.62	2.30	3.86	3.82	8.07
ν	0.7						0.8											
AVE	87.81	93.80	79.48	69.42	83.09	50.94	99.72	100	99.40	73.78	75.02	74.30						
STD	1.19	1.23	2.97	3.08	4.17	6.41	0.23	0.00	0.49	4.59	12.21	16.65						

Table 8.16. Classification Rate by SVM$_{total}$ for Liver Disorders

C_1	1																	
C_2	0.0001						0.001						0.01					
	training			test			training			test			training			test		
	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$
AVE	82.73	90.66	71.66	65.53	74.71	53.00	86.74	92.94	78.17	69.51	82.72	51.56	86.74	92.94	78.17	69.42	82.55	51.56
STD	1.81	2.71	4.91	5.11	6.74	7.08	1.33	1.20	2.89	3.17	3.99	5.34	1.29	1.20	2.72	3.17	4.18	5.34
C_1	10																	
C_2	0.001						0.01						0.1					
AVE	95.25	95.86	94.40	65.83	71.85	57.97	95.25	95.86	94.40	65.83	71.85	57.97	95.29	95.86	94.50	65.53	71.35	57.97
STD	1.17	0.91	2.51	3.09	2.99	7.16	1.17	0.91	2.51	3.09	2.99	7.16	1.19	0.91	2.50	3.02	3.18	7.16
C_1	100																	
C_2	0.01						0.1						1					
AVE	99.42	99.29	99.61	61.84	64.54	58.20	99.42	99.29	99.61	61.84	64.76	58.00	99.46	99.29	99.70	63.11	66.45	58.76
STD	0.27	0.44	0.48	4.43	6.13	6.65	0.27	0.44	0.48	4.47	5.96	6.50	0.26	0.44	0.46	3.53	5.86	6.85

Table 8.17. Classification Rate by $\mu-$SVM for Liver Disorders

μ	1.2						1.3						1.4					
	training			test			training			test			training			test		
	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$
AVE	69.75	93.11	37.37	65.34	92.87	27.54	73.93	90.99	50.32	69.03	90.26	39.75	74.92	90.64	53.14	68.83	88.96	41.10
STD	1.79	3.19	7.83	4.27	4.75	6.06	1.04	1.70	3.22	4.61	3.71	5.73	1.83	1.85	5.37	3.65	3.39	4.35
μ	1.5						1.6						1.7					
AVE	78.22	91.71	59.57	68.64	88.59	41.11	87.89	95.09	77.87	68.45	87.78	41.77	92.19	97.00	85.49	69.13	88.12	42.93
STD	7.85	3.13	15.24	3.63	2.46	4.27	12.85	5.18	23.69	3.58	2.93	5.17	12.62	4.84	23.61	3.29	3.27	5.21
μ	1.8						$\cdots$						2.0					
AVE	100	100	100	69.22	87.64	43.87	$\cdots$			$\cdots$			100	100	100	69.22	87.64	43.87
STD	0.00	0.00	0.00	3.58	3.30	5.06	$\cdots$			$\cdots$			0.00	0.00	0.00	3.58	3.30	5.06

Table 8.18. Classification Rate by $\mu-\nu-$SVM for Liver Disorders

μ	1																	
ν	0.0001						0.001						0.01					
	training			test			training			test			training			test		
	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$
AVE	97.81	97.25	98.58	93.69	93.58	93.85	100	100	100	72.52	76.50	66.84	100	100	100	63.20	66.74	58.24
STD	3.15	5.43	3.33	2.50	5.75	6.19	0.00	0.00	0.00	4.45	5.64	5.31	0.00	0.00	0.00	4.71	5.39	5.30
μ	10																	
ν	0.001						0.01						0.1					
AVE	100	100	100	72.52	76.50	66.84	100	100	100	63.20	66.74	58.24	100	100	100	62.14	67.87	54.53
STD	0.00	0.00	0.00	4.45	5.64	5.31	0.00	0.00	0.00	4.71	5.39	5.30	0.00	0.00	0.00	4.23	4.48	8.78
μ	100																	
ν	0.01						0.1						1					
AVE	100	100	100	63.20	66.74	58.24	100	100	100	62.14	67.87	54.53	100	100	100	62.14	67.87	54.53
STD	0.00	0.00	0.00	4.71	5.39	5.30	0.00	0.00	0.00	4.23	4.48	8.78	0.00	0.00	0.00	4.23	4.48	8.78

Table 8.19. Classification Rate by GP for PIMA

C_1	1																	
C_2	0.001						0.01						0.1					
	training			test			training			test			training			test		
	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$
AVE	78.10	67.75	83.49	77.35	66.40	83.67	78.18	67.24	83.85	77.57	65.82	84.36	78.72	63.34	86.70	76.87	60.93	86.06
STD	1.19	1.58	1.02	2.92	5.37	2.05	1.14	1.79	1.02	2.87	5.34	1.96	1.37	2.32	1.07	3.00	5.92	1.74
C_1	10																	
C_2	0.01						0.1						1					
AVE	78.14	68.02	83.40	77.43	66.08	84.01	78.14	67.41	83.71	77.74	66.18	84.43	78.74	63.77	86.51	77.00	61.55	85.93
STD	1.29	1.83	1.07	2.86	5.15	1.92	1.13	1.55	1.08	2.71	5.09	1.91	1.38	2.00	1.25	3.41	6.82	1.94
C_1	100																	
C_2	0.1						1						10					
AVE	78.12	67.97	83.40	77.48	66.19	84.01	78.12	67.41	83.68	77.78	66.18	84.49	78.79	63.94	86.51	77.00	61.55	85.93
STD	1.27	1.76	1.07	2.82	5.00	1.92	1.15	1.55	1.11	2.75	5.09	1.92	1.39	1.86	1.31	3.51	7.07	1.94

Table 8.20. Classification Rate by SVM$_{soft}$ for PIMA

C	0.01						0.1						1					
	training			test			training			test			training			test		
	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$
AVE	65.61	0.00	100	63.91	0.00	100	65.61	0.00	100	63.91	0.00	100	91.47	80.71	97.08	74.17	52.51	86.40
STD	1.03	0.00	0.00	2.41	0.00	0.00	1.03	0.00	0.00	2.41	0.00	0.00	0.70	2.26	0.57	1.82	3.02	2.25
C	10						100											
AVE	99.44	98.43	99.97	69.83	54.29	78.60	100	100	100	68.91	54.45	77.09						
STD	0.20	0.63	0.08	1.63	4.24	2.05	0.00	0.00	0.00	2.07	3.90	2.71						

Table 8.21. Classification Rate by $\nu-$SVM for PIMA

ν	0.1						0.2						0.3					
	training			test			training			test			training			test		
	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$
AVE	99.91	99.73	100	69.04	54.33	77.34	99.33	98.16	99.94	70.04	54.62	78.74	97.66	94.69	99.20	71.61	54.51	81.24
STD	0.12	0.36	0.00	1.88	3.81	2.37	0.25	0.66	0.11	1.80	4.25	2.23	0.46	0.69	0.46	1.42	4.24	1.59
ν	0.4						0.5						0.6					
AVE	94.93	88.52	98.27	72.65	53.40	83.57	93.03	84.18	97.64	73.43	52.66	85.18	90.65	78.75	96.85	74.09	51.75	86.76
STD	0.58	1.82	0.53	2.30	5.27	2.29	0.46	1.92	0.65	2.25	3.92	2.56	0.86	3.12	0.79	2.08	3.49	2.47

Table 8.22. Classification Rate by SVM$_{total}$ for PIMA

C_1	1																	
C_2	0.0001						0.001						0.01					
	training			test			training			test			training			test		
	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$
AVE	68.77	87.51	58.95	65.22	85.36	53.85	91.62	82.29	96.48	73.91	57.85	82.99	91.67	82.51	96.45	73.57	58.19	82.23
STD	1.29	1.31	1.84	2.01	2.97	3.02	0.72	2.23	0.92	2.39	2.78	2.45	0.74	2.14	0.94	2.44	2.75	2.51
C_1	10																	
C_2	0.001						0.01						0.1					
AVE	99.48	98.70	99.89	69.17	63.45	72.36	99.48	98.70	99.89	69.17	63.69	72.22	99.44	98.86	99.75	68.65	66.35	69.89
STD	0.20	0.37	0.19	2.40	2.84	3.14	0.20	0.37	0.19	2.47	2.78	3.18	0.28	0.52	0.23	2.64	3.41	3.31
C_1	100																	
C_2	0.01						0.1						1					
AVE	100	100	100	67.35	64.60	68.89	100	100	100	67.35	67.23	67.37	70.13	100	54.52	62.22	91.43	45.59
STD	0.00	0.00	0.00	2.51	4.05	3.14	0.00	0.00	0.00	2.54	3.57	2.94	3.36	0.00	4.68	2.98	2.62	5.69

Table 8.23. Classification Rate by $\mu-$SVM for PIMA

μ	1.4						1.5						1.6					
	training			test			training			test			training			test		
	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$
AVE	78.01	46.68	94.62	66.91	26.41	89.14	93.92	85.70	97.86	67.87	30.92	88.14	100	1000	100	68.00	31.32	88.06
STD	2.13	7.25	1.20	3.13	5.78	2.84	9.81	23.91	3.69	4.70	6.78	2.47	0.00	0.00	0.00	4.41	6.13	2.40
μ	1.7						$\cdots$						2.0					
AVE	100	100	100	68.00	31.32	88.06	$\cdots$			$\cdots$			100	100	100	68.00	31.32	88.06
STD	0.00	0.00	0.00	4.41	6.13	2.40	$\cdots$			$\cdots$			0.00	0.00	0.00	4.41	6.13	2.40

Table 8.24. Classification Rate by $\mu-\nu-$SVM for PIMA

μ	1																	
ν	0.0001						0.001						0.01					
	training			test			training			test			training			test		
	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$	rate	$\mathcal{A}$	$\mathcal{B}$
AVE	98.05	99.95	97.02	89.57	94.29	87.13	100	100	100	73.00	64.93	77.64	100	100	100	69.17	60.46	74.17
STD	2.01	0.16	3.12	3.20	3.99	5.43	0.00	0.00	0.00	2.14	4.26	2.77	0.00	0.00	0.00	2.25	4.46	3.44
μ	10																	
ν	0.001						0.01						0.1					
AVE	100	100	100	73.00	64.93	77.64	100	100	100	69.17	60.46	74.17	100	100	100	69.09	56.07	76.49
STD	0.00	0.00	0.00	2.14	4.26	2.77	0.00	0.00	0.00	2.25	4.46	3.44	0.00	0.00	0.00	1.72	5.06	3.63
μ	100																	
ν	0.01						0.1						1					
AVE	100	100	100	69.17	60.46	74.17	100	100	100	69.09	56.07	76.49	100	100	100	68.91	54.45	77.09
STD	0.00	0.00	0.00	2.25	4.46	3.44	0.00	0.00	0.00	1.72	5.06	3.63	0.00	0.00	0.00	2.07	3.90	2.71

Table 8.25. Rates of Support Vectors (unit : %)

		SVM_{soft}	$\nu-$SVM	SVM_{total}	$\mu-$SVM	$\mu-\nu-$SVM
MONK	AVE	74.60	76.69	100	100	62.83
(case 1)	STD	15.19	13.65	0	0	0.23
MONK	AVE	76.90	73.10	100	100	69.00
(case 2)	STD	7.43	4.78	0	0	0.85
MONK	AVE	70.70	74.23	100	100	56.11
(case 3)	STD	17.40	12.53	0	0	1.41
Cleveland	AVE	97.40	96.97	100	100	96.83
Heart-disease	STD	0.75	0.70	0	0	0.70
Liver Disorders	AVE	81.59	75.37	100	100	59.02
	STD	2.13	2.18	0	0	3.54
PIMA	AVE	72.53	71.35	100	100	64.26
	STD	1.74	1.71	0	0	1.98

8.8 Concluding Remarks

In this chapter, we introduced various SVM algorithms using MOP/GP. The authors have given a generalization error bound, and proved that the error bound can be decreased by minimizing slack variables and maximizing surplus variables [23]. As a total, $\mu-\nu-$SVM shows relatively good performance in our experiences. However, SVM_{total} and $\mu-$SVM among the proposed algorithms are inferior to the standard SVM algorithms in terms of sparsity of support vectors. This means that those methods cause some difficulty in computation for large scale data sets. It is observed in our experience, moreover, that some values of μ yield unacceptable solutions in $\mu-$SVM algorithm. However, $\mu-\nu-$SVM overcomes the lack of sparsity of support vectors, and does not cause so much difficulty in computation even for large scale data sets. For regression problems, moreover, $\mu-\nu-$SVM minimizing the exterior deviation is akin to function approximation using the Tchebyshev error, which is widely applied to many real problems. This is another point for which $\mu-\nu-$SVM is promising. The details on regression by $\mu-\nu-$SVM will be discussed elsewhere.

Acknowledgement

This research was supported by JSPS.KAKENHI13680540.

References

[1] Asada, T. and Nakayama, H. (2003) SVM using Multi Objective Linear Programming and Goal Programming, in T. Tanino, T. Tanaka and M. Inuiguchi (eds.), *Multi-objective Programming and Goal Programming*, 93-98

[2] Bennett, K.P. and Mangasarian, O.L. (1992) Robust Linear Programming Discrimination of Two Linearly Inseparable Sets, *Optimization Methods and Software*, **1**, 23-34

[3] Cavalier, T.M., Ignizio, J.P. and Soyster, A.L., (1989) Discriminant Analysis via Mathematical Programming: Certain Problems and their Causes, *Computers and Operations Research*, **16**, 353-362

[4] Chankong, V. and Haimes, Y.Y., (1983) *Multiobjective Decision Making Theory and Methodlogy* , Elsevier Science Publsihing

[5] Charnes, A. and Cooper W.W., (1961) *Management Models and Industrial Applications of Linear Programming* , vol. 1, Wiley

[6] Cortes, C. and Vapnik, V., (1995) Support Vector Networks, *Machine Learning*, **20**, pp. 273–297

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

[8] Erenguc, S.S. and Koehler, G.J., (1990) Survey of Mathematical Programming Models and Experimental Results for Linear Discriminant Analysis, *Managerial and Decision Economics*, **11**, 215-225

[9] Freed, N. and Glover, F., (1981) Simple but Powerful Goal Programming Models for Discriminant Problems, *European J. of Operational Research*, **7**, 44-60

[10] Glover, F. (1990) Improved Linear Programming Models for Discriminant Analysis, *Decision Sciences*, **21**, 771-785

[11] Mangasarian, O.L., (1968) Multisurface Method of Pattern Separation, *IEEE Transact. on Information Theory*, **IT-14**, 801-807

[12] Mangasarian, O.L., (1999) *Arbitrary-Norm Separating Plane*, Operations Research Letters **23**

[13] Marcotte, P. and Savard, G., (1992) Novel Approaches to the Discrimination Problem, *ZOR–Methods and Models of Operations Research*, **36**, pp.517-545

[14] Miettinen, K. M., (1999) *Nonlinear Multiobjective Optimization* , Kluwer Academic Publishers

[15] Nakayama, H., (1995) Aspiration Level Approach to Interactive Multi-objective Programming and its Applications, *Advances in Multicriteria Analysis*, ed. by P.M. Pardalos, Y. Siskos and C. Zopounidis, Kluwer Academic Publishers, pp. 147-174

[16] Nakayama, H. and Asada, T., (2001) Support Vector Machines formulated as Multi Objective Linear Programming, *Proc. of ICOTA2001*, **3**, pp.1171-1178

[17] Novikoff, A.B., (1962) On the Convergence Proofs on Perceptrons, In*Symposium on the Mathematical Theory of Automata*, vol. 12, pp. 615–622, Polytechnic Institute of Brooklyn

[18] Sawaragi, Y., Nakayama, H. and Tanino, T., (1994) *Theory of Multiobjective Optimization*, Academic Press

[19] Schölkopf, B. and Smola, A.J., (1998) New Support Vector Algorithms, *NeuroCOLT2 Technical report Series*, NC2-TR-1998-031

[20] B.Schölkopf, and A.J.Smola, (2002) *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond-*, MIT Press

[21] Steuer, R.E., (1986) *Multiple Criteria Optimization: Theory, Computation, and Application* , Wiley

[22] Vapnik, V.N., (1998) *Statistical Learning Theory*, John Wiley & Sons, New York

[23] Yoon, M., Yun, Y.B. and Nakayama, H., (2003) A Role of Total Margin in Support Vector Machines, *Proc. IJCNN'03*, 2049-2053