Multicategory Support Vector Machines

A dissertation submitted in partial fulfillment of the requirements for the M.Tech (Computer Science) degree of the Indian Statistical Institute, Kolkata.

By

Rajesh Kumar Singh

under the supervision of

Dr. Srimanta Pal
Electronics and Communication Science Unit

INDIAN STATISTICAL INSTITUTE
203, B.T. Road, Kolkata - 700035
21 July' 2004

Indian Statistical Institute,
203, Barrackpore Trunk Road,
Kolkata - 700 035

Certificate Of Approval

This is to certify that this thesis titled "*Multicategory Support Vector Machines*" submitted by Rajesh Kumar Singh towards partial fulfillment of requirements for the degree of M.Tech in Computer Science at Indian Statistical Institute, Kolkata embodies the work done under my supervision.

[illegible]
Dr. Srimanta Pal,
Electronics and Communication Sciences Unit,
Indian Statistical Institute,
Kolkata - 700 108.

(External Expert)

Acknowledgement

I take pleasure in thanking Dr.Srimanta Pal for his friendly guidance throughout the dissertation period.

I would also like to express my sincere gratitude to Dr. N.R. Pal for enjoyable discussions and valuable suggestions. His pleasant and encouraging words have always kept my spirits up.

I would like to thank Durga Prasad Muni and other Research Scholars of ECSU for their cooperation throughout my dissertation..

Finally, I take the opportunity to thank my classmates, friends and family members for their encouragement to finish this work.

Rajesh Kumar Singh

Abstract

In this dissertation first we critically review the Support Vector Machine for both two-class and multiclass problems. There have been a few attempts to deal with multiclass problems. Our analysis of a number of such methods revealed several problems associated with them. We then proposed six new methods. Three of proposed methods are modification of some existing methods, while three methods involve reformulation of Support Vector Machine . In this context we introduce a novel concept of utility of training points, which make Support Vector Machine less sensitive to outliers. We also introduce a new concept of optimal hyper-sphere to design a Support Vector Machine type of classifiers. All methods are theoretically formulated and illustrated using the suitable datasets.

Contents

Chapter 1

Basic SVM

1.1 Introduction

Support vector machine (SVM) devloped by Vapnik [1] is gaining popularity due to many attractive features and promising empirical performance compared to traditional neural networks. The problem which drove the initial devlopment of SVMs occurs in several guises- the bias variance trade off, capacity control, overfitting - but basic idea is same. Roughly speaking, for a given learning task. with given finite amount of training data, the best generalization peroformance will be achieved if the right balance is struck between the accuracy attained on the particular training set, and the capacity of machine,that is, ability of the machine to learn any training set without error. SVM generalization performance either matches or significantly better than that of competing methods. SVM possess following prominent advantages:
Advantages .

1. Strong theoretical backgroud provides SVM with high generalization capability and can avoid local minima.

2. SVM does not determine the network topology in advance which can be automatically obtained when training process ends.

3. SVM can solve high dimensional problem and therefore avoids the curse of dimensionality.

The root cause the SVM attain the more and more attention is that SVM adopts the structural risk minimization (SRM) ,which has been shown to be better than empirical risk minimization (ERM) employed by conventional neural networks. SRM minimizes the upper bound of test error by minimizing the VC dimension , as opposed ERM minimizes the training error. It is the difference that eqips SVM with good generalization performance which is the goal of learning problem.

As we know generalization performance is an important index to measure the goodness of a learning machine and it can be predicted by well known bound (Vapnik,1995)

$$R(\alpha) \leq R_{emp}(\alpha) + \sqrt{\left(\frac{h(\log(2l/h)+1) - \log(\eta/4)}{l}\right)} \tag{1.1}$$

where $R(\alpha)$ is the actual risk. R_{emp} is the empirical risk (training error). h is a non negative integer called Vapnik Chervonenkis (VC) dimension, and is a measure of the notion of the capacity mentioned .
The second term on the right hand side is called **VC confidence**.

The goal of learning is to minimize $R(\alpha)$,the actual risk.
Generally there is a trade off between the two terms of the right side of the inequality (1.1). When *VC confidence* is fixed, which means structure of learning machine is given, only R_{emp} need to be minimized. This case is well known neural network learning. On the other hand when R_{emp}(say, 0) is fixed and we need to need to minimize the VC confidence. This case is called support vector machine learning. The main idea of support vector machine is to map a non-linear problem from an input space to higher dimension via. non-linear mapping ϕ, where ϕ is unknown in general. Fortunately computation related to ϕ could be induced to the computation of inner product in the feature space.

1.1.1 Organization of the Report

This project report is organized in six chapters. In the first chapter we have described the advantages of SVM over the others competing methods and some basic idea of SVM. In chapter 2 we explain the construction of two-class support vector machine for linearly seprable classes. This is followed by considering more difficult case of non seprable data. In chapter 3 we describe four popular existing methods for multiclass support vector machine. In chapter 4 we explain the three proposed solution for multiclass SVM followed by a disscussion on problem of optimal boundary. This will followed by two proposed solutions for shifting the optimal boudary for getting the better result. Chapter 5 consists of experimental results of different types of SVM (existing and proposed) over artificial datasets, followed by observations and remarks. Chapter 6 describes the problem with vapnik's optimal hyperplane over certain type of datasets, followed by proposed concept of optimal hypersphere with complete theorectical foundation. Finally we discuss some possible future work.

Chapter 2

Two Class Support Vector Machine

2.1 Optimal Hyperplane for Linearly Seprable Patterns

Consider the training sample $(x_i, d_i)_{i=1}^{N}$, where x_i is the input pattern for the i th example and d_i is corresponding desired response (target output). we assume that the pattern (class) represented by the subset, $d_i = +1$ and the pattern represented by the subset $d_i = -1$ are *linearly seprable.* The equation of a decision surface in form of the a hyperplane that does the sepration is

$$w^T x + b = 0 \tag{2.1}$$

where x is input vector, w is an adjustable weight vector, and b is a bias. we may thus write

$$\begin{aligned} w^T x_i + b \geq 0 \quad & for \ \ d_i = +1 \\ w^T x_i + b < 0 \quad & for \ \ d_i = -1 \end{aligned} \tag{2.2}$$

For a given weight vector w and the bias b, the sepration between the hyperplane defined in Eqn. 2.1 and the closest data point is called the *margin of sepration*, denoted by ρ. The goal of support vector machine is to find the particular hyperplane for which margin of sepration ρ is to be maximized. Under this condition the decision surface

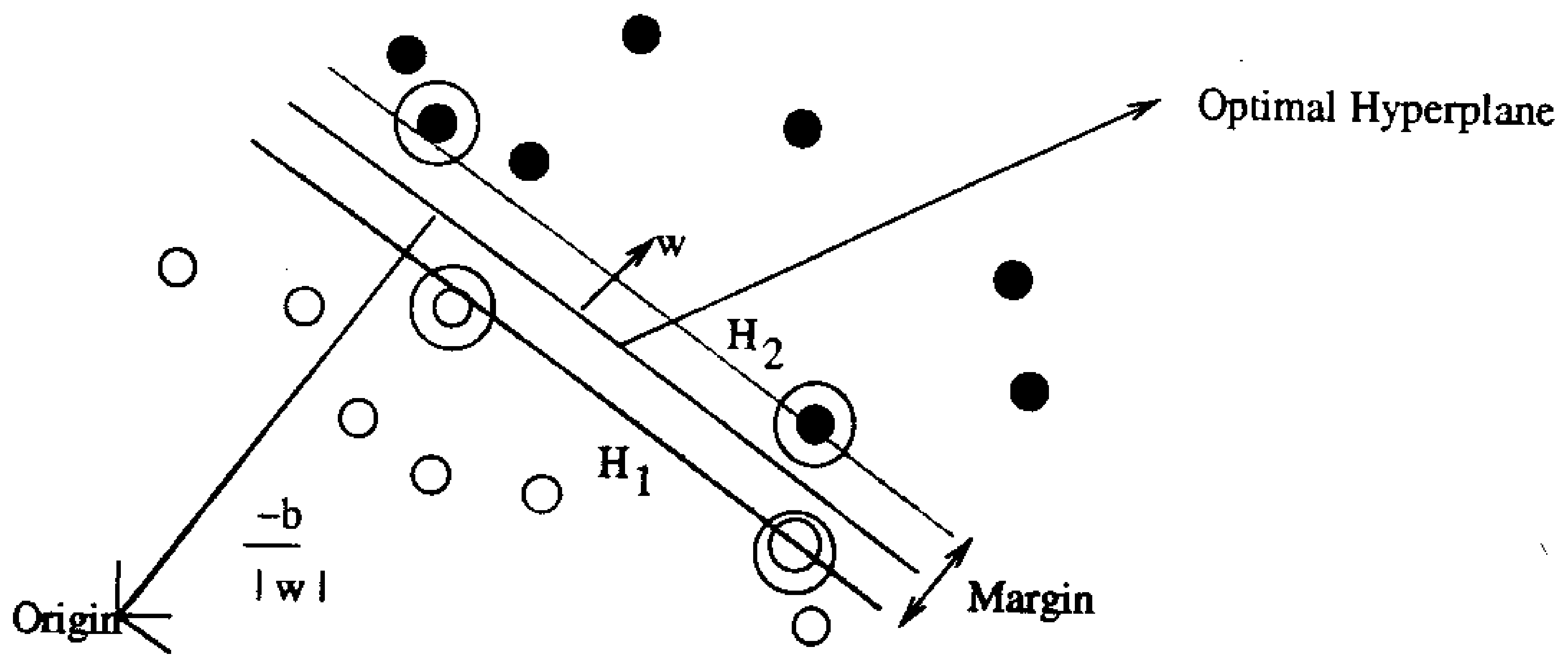


Figure 2.1: Linear separating hyperplanes for the separable case. The support vectors are circled

is reffered as *optimal hyperplane.* Figure 2.1 illustrates the geometric construction of an optimal hyperplane for a two dimensional input space. The issue at hand is to find the parameters w and b for the optimal hyperplane, given the training set $\Im = \{(x_i, d_i)\}_{i=1}^{N}$. In light of the results portrayed in Fig. 2.1, we see that the pair (w_0, b_0) must satisfy the constraint:

$$w^T x_i + b \geq 1 \qquad for \quad d_i = +1$$
$$w^T x_i + b \leq -1 \qquad for \quad d_i = -1 \tag{2.3}$$

Note that if the Eq.(2.2) holds, that is, the patterns are linearly seprable, we can always rescale w and b such that Eq.(2.2) holds; this scaling operation leaves the optimal hyperplane unaffected.

The particular data points (x_i, d_i) for which first or second line of Eq. (2.3) is satisfied is called *Support vector.* These vectors play a prominent role in this class of learning machines. In conceptual terms, support vectors are those training points that lie closest to decision surface and are most difficult to classify. Consider the support vector x^s for which $d^s = +1$. Then by definition, we have

$$w^T x_i \mp b = \mp 1 \qquad for \quad d_s = \mp 1 \tag{2.4}$$

The distance of the support vector $x^{(s)}$, $dist(x^s)$ (say) to the optimal plane is

$$dist(x^(s)) = \begin{cases} \frac{1}{\|w\|} & \text{if } d^{(s)} = +1 \\ \frac{-1}{\|w\|} & \text{if } d^{(s)} = -1 \end{cases} \tag{2.5}$$

where the plus indicates that $x^{(s)}$ lies on the positive side of the optimal plane and the negative sign indicate that $x^{(s)}$lies on the negative side of the optimal hyperplane. Let ρ denote the optimun value of *margin of sepration* between the two classes that constitutes the training set $\Im$.Then from the above equation it follows that

$$\rho = \frac{2}{\|w\|} \tag{2.6}$$

Equation (2.6) states that maximizing the margin of the sepration between the classes is equivalent to minimizing the Euclidean norm of the weight vector w .

2.2 Quadratic Optimization for Finding the Optimal Hyperplane

Our goal is to devlop a computationally efficient procedure for using the training sample $\Im = \{(x_i, d_i)\}_{i=1}^{N}$ to find the optimal plane subject to constraints

$$d_i(w^T x_i + b) \geq 1 \quad for \;\; i = 1, 2 \ldots, N. \tag{2.7}$$

Now we have to solve the constrained optimization problem. This may be stated as: *Given the training sample* $\Im = \{(x_i, d_i)\}_{i=1}^{N}$ *, find the optimum value of* w *and* b *that satisfy the constraints*

$$d_i(w^T x_i + b) \geq 1 \quad for \;\; i = 1, 2 \ldots, N. \tag{2.8}$$

and the weight vector w *minimizes the cost function*

$$\phi(w) = \frac{1}{2} w^T w$$

2.2.1 Lagrangian Formulation

We will switch to lagrangian formulation of the problem.There are two reasons for doing this.

1. The constraint, in Eqn. 2.8 will be replaced by constraints on themselves, which is much easier to handle.
2. This reformalisation results in, the training data will only appear in form of dot products between vectors. This is a crucial property which will allow us to generalize the procedure for nonlinear case.

We construct the *Lagrangian function:*

$$J(w, b, \alpha) = \frac{1}{2}w^T w - \sum_{i=1}^{N} \alpha_i[d_i(w^T x_i + b) - 1] \tag{2.9}$$

where the auxiliary nonnegative variables α_i are called *Lagrange multipliers*. The solution of the constrained optimization problem is determined by the *saddle point* of the Lagrangian function $J(w, b, \alpha)$, which has to be *minimized* with respect to w and b. Thus, differentiating $J(w, b, \alpha)$ with respect to w and b and the result to zero, we get the following two conditions for *optimality*:

Condition 1 :

$$\frac{\partial J(w, b. \alpha)}{\partial w} = 0$$

Condition 2 :

$$\frac{\partial J(w, b. \alpha)}{\partial b} = 0$$

condition 1 to lagrangian function in Eqn (2.9)yields

$$w = \sum_{i=1}^{N} \alpha_i d_i x_i = 0. \tag{2.10}$$

condition 2 to lagrangian function in Eqn(2.9)yields

$$\sum_{i=1}^{N} \alpha_i d_i = 0. \tag{2.11}$$

It is also important to note that at the saddle point, for each Lagrange multiplier α_i , the product of that multiplier with its corresponding constraint vanishes, as shown by

$$\alpha_i[d_i(w^T x_i + b) - 1] = 0 \qquad for \;\; i = 1, 2, \ldots, N \tag{2.12}$$

Therefore only those multiplier exactly meeting the Eqn.(2.12) can assume non zero value.This property follows from the *Kuhn-Tucker conditions* of optimization theory.Primal problem deals with convex cost function and linear constraints.Given such Primal problem,it is possible to construct another problem called the *dual problem.*This second problem has the same optimal value as the primal problem.This is a basically *duality theorem.* We state the dual problem:

Given the training sample $\{(x_i, d_i)\}_{i=1}^{N}$ *,find the Lagrange multipliers* $\{\alpha_i\}_{i=1}^{N}$ *that maximize the objective function*

$$(Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j d_i d_j {x_i}^T x_j$$

subject to the constraints

1.
$$\sum_{i=1}^{N} \alpha_i d_i = 0$$

2.
$$\alpha_i \geq 0 \quad for \;\; i = 1, 2, \ldots, N$$

Note that the dual problem is cast entirely in terms of training data.
Having datermined the optimum Lagrange multipliers.denoted by $\alpha_{o,i}$ we can compute optimum weight vector w_o,using the Eqn (2.10) so we can write

$$w_o = \sum_{i=1}^{N} \alpha_{o,i} d_i x_i \tag{2.13}$$

To compute the optimum bias b_o use the equation

$$b_o = 1 - w_o^T x^{(s)} \qquad for d^{(s)} = 1 \tag{2.14}$$

2.3 Optimal Hyperplane For Nonseprable Patterns

Now we consider the nonseprable pattern.Given such a set of training data, it is not possible to construct a hyperplane without encountering classification errors.so we would like to relax the constraints in Eqn (2.3) ,but only when necessary,that is,we introduce a further cost (i.e, an increase in the primal objective function) for doing so. This can be done introducing positive slack variables ξ_i, $i = 1\ldots,n$ in the constraints, which then become:

$$w^T x_i + b \geq +1 - \xi_i \quad for \;\; d_i = +1$$

$$w^T x_i + b \leq -1 + \xi_i \quad for \;\; d_i = -1 \tag{2.15}$$

$$for \;\; \xi_i \geq 0 \;\; \forall i \tag{2.16}$$

thus for an error to occur, the corresponding ξ_i must exceed unity, so $\sum_i \xi_i$ is an

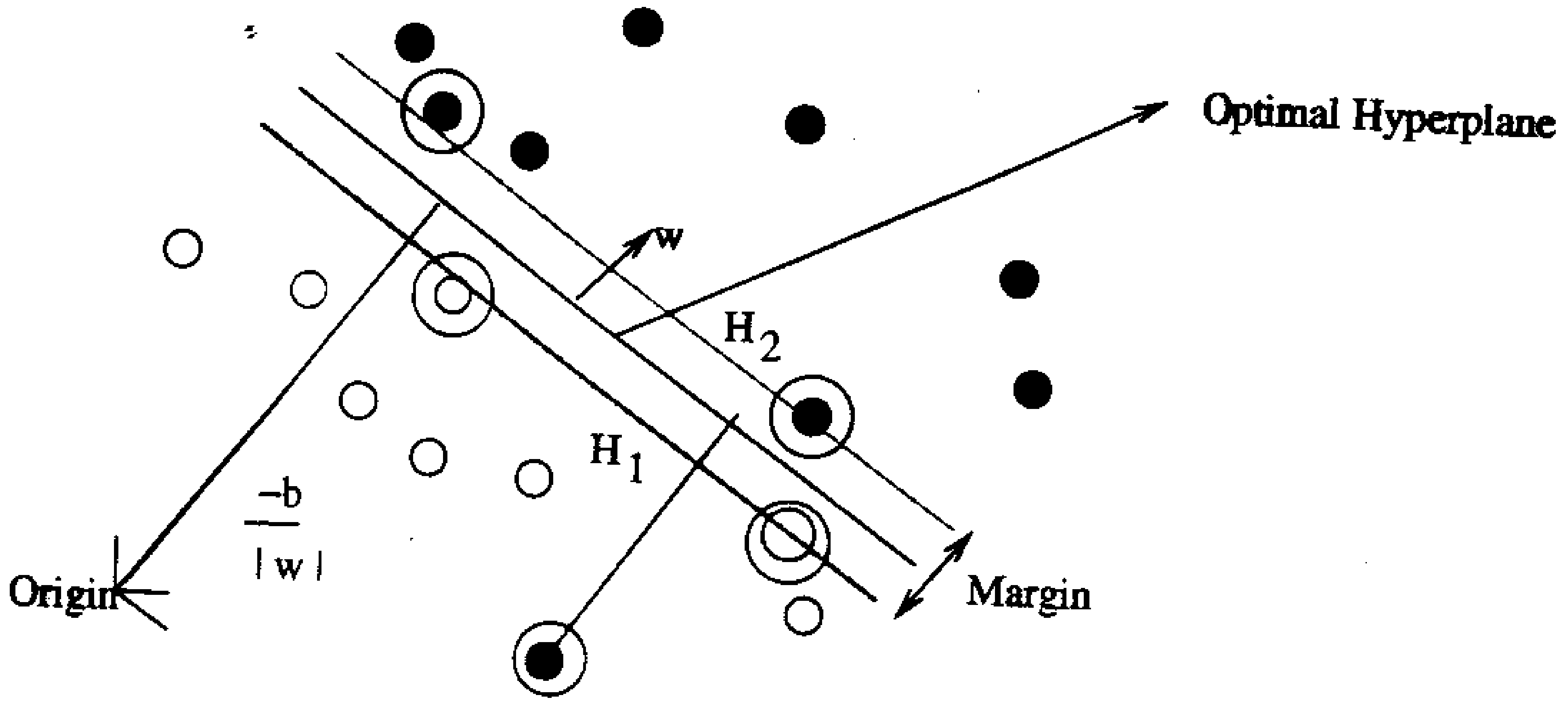


Figure 2.2: Linear separating hyperplanes for non-separable case. The support vectors are circled.

upper bound on the number of training errors. Hence a natural way to assign an extra cost for an errors is to change the objective function to be minimized from $\|w\|^2$ to $\|w\|^2 + c(\sum_i \xi_i)^k$, where c is a parameter to be chosen by the user, a large c corresponding to higher penalty to errors. As it stands, this is convex programming problem for any positive integer k. for $k = 2$ and $k = 1$ it is also quadratic programming problem.

and the choice $k = 1$ has further advandatage that neither the ξ_i, nor their Lagrange multipliers,appear in the Wolf dual problem, which becomes:

$$Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i\alpha_j d_i d_j {x_i}^T x_j \tag{2.17}$$

subject to the constraints

1.
$$\sum_{i=1}^{N} \alpha_i d_i = 0$$

2.
$$0 \leq \alpha_i \leq c \quad for \quad i = 1, 2, \ldots, N.$$

The solution is again given by

$$w = \sum_{i=1}^{N_s} \alpha_{o.i} d_i x_i \tag{2.18}$$

where N_s is the number of support vectors.The Kuhn-Tucker condition is now defined by

$$\alpha_i[d_i(w^T x_i + b) - 1 + \xi_i] = 0, \quad i = 1, 2, \ldots, N \tag{2.19}$$

$$and$$

$$\mu_i \xi_i = 0, \quad i = 1, 2, \ldots\ldots, N. \tag{2.20}$$

At the saddle point the derivative of lagrangian function for the primal problem with respect to the slack variable ξ_i is zero, the evalution of which yields

$$\alpha_i + \mu_i = c. \tag{2.21}$$

By combining the Eqn.(2.20) and Eqn.(2.21),we get

$$\xi_i = 0 \quad if \quad \alpha_i < c \tag{2.22}$$

We may determine the value of b_o by taking the data point (x_i, d_i) in the training set for which $0 < \alpha_{o,i} < c$ and therefore $\xi_i = 0$,and using that data point in Eqn.(2.19). However, from a numerical perspective it is better to take mean value of b_o resulting from all such data points in the training sample.

2.3.1 Decison function

For a given test pattern x, we calculate the class label by

$$D(x) = sign(w_o \cdot x + b_o)$$

2.4 Nonlinear Support Vector Machine

Now, We generalize to case where decision function is not the linear function. Basically, the idea hinges on two mathematical operations summerized here:

1. Nonlinear mapping of an input vector into higher dimensional *feature space.*
2. Construction of an optimal hyperplane for seprating features discovered in step 1.

Fortunately data appeared in the training problem Equation (2.17) is in form of dot products, $x_i \bullet x_j$. Now suppose we use the nonliear map ϕ to map the data from input space to feature space (H).

$$\phi : \mathbf{R^d} \mapsto H.$$

Then training algorithm only depend on the data through dot products in H i.e, on the fuction of the form $\phi(x_i) \cdot \phi(x_j)$. Now if there is *kernel function* K such that

$$K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j) \tag{2.23}$$

we only need to use K in the training algorithm, and never need to explicitly know the ϕ .

2.4.1 Optimum Design of a Support Vector Machine

We may now state the most general dual form for the constrained optimization of a support vector machine as follows:

$$Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j d_i d_j \phi(x_i)^T \phi(x_j) \tag{2.24}$$

subject to the constraints

1.
$$\sum_{i=1}^{N}\alpha_i d_i = 0$$

2.
$$0 \leq \alpha_i \leq c \quad for \quad i = 1,2,\ldots,N$$

where c is user specified positive parameter.
Adapting Eq.(2.10) to our present situation involving a feature space,we may write

$$w = \sum_{i=1}^{N}\alpha_i d_i \phi(x_i).$$

Here again we use Karush-Kuhn-Tucker(KKT) condition to calculate b as follows:

$$\alpha_i[d_i(w^T\phi(x_i) + b) - 1 + \xi_i] = 0, \qquad i = 1,2,\ldots,N$$

and

$$\mu_i\xi_i = 0$$

$$\alpha_i + \alpha_j = c$$

Consequenly we see that $\xi_i = 0$ if $\alpha_i < c$.So In this case the explicit formula for b is

$$b = d_i - \sum_{j=1}^{N} d_j\alpha_j\phi(x_j)\phi(x_i)$$

$$\Rightarrow b = d_i - \sum_{j=1}^{N} d_j\alpha_j K(x_j, x_i)$$

So we don't need to know ϕ explicitely for calculating b

Decision function

In the test phase. x will be classified by computing the sign of

$$f(x) = \sum_{i=1}^{N_s}\alpha_i d_i\phi(s_i)\cdot\phi(x) + b = \sum_{i=1}^{N_s}\alpha_i d_i K(s_i.x) + b$$

*where s_i are support vectors.*Also we don't need to calculate $\phi(x)$ explicitly.

2.5 Mercer's Condition

Mercer's condition tells us whether or not a perspective kernel is actually a dot product in some space ,In other word it tells for which kernel there exist a pair $\{H, \phi\}$,which satisfies the Eq.(2.23). Formally the Mercer's condition can be stated as follows:

There exists a mapping ϕ and an expansion

$$K(x, y) = \sum_i \phi(x)_i \phi(y)_i$$

if and only if, for any $g(x)$ such that

$$\int g(x)^2 dx$$

is finite then

$$\int K(x, y) g(x) g(y) dx dy \geq 0$$

Some example of kernels which satisfy the mercer's condition are as follows :

1. Polynomial kernel $K(x, y) = (x \cdot y + 1)^p$.
2. Gaussian kernel $K(x, y) = e^{-\|x-y\|/2\sigma^2}$.
3. Hypebolic tangen kernel $K(x, y) = tanh(kx \cdot y - \delta)$.

Chapter 3

Multiclass Support Vector Machine

In this chapter we explain the extension of SVM for multiclass problem. As discussed in the chapter [2], support vector machine directly determine the decision function for two-class problem. so that generalization ability is maximized while minimizing the training error. Because of this formulation, extension to multiclass problem is not unique. Here we explain the four most popular existing methods for multiclass problem which are as follows :

1. One-Against-All SVM proposed by Vapnik [1]
2. Pairwise SVM by Krebel [4]
3. MSVM proposed by K.P. Bennett [3]
4. Fuzzy Support Vector Machine (FSVM) proposed by Huang and Liu [2]

3.1 One-Against-All SVM

This concept of multiclass support vector machine proposed by Vapnik [1]. In this method n class problem is converted into n two class problem. In ith two class problem, we determine the optimal decision function $D_i(x)$ so that class i is seprated from remaining classes with maximum margin. So using Binary SVM discussed in chapter [2], we get the following n decision boundaries

$$D_i(x) = w_i^t x + b_i, \quad i = 1, 2, \ldots, n \tag{3.1}$$

where w_i is m-dimensional vector and b_i is a scalar.

The hyperplane, $D_i(x) = 0$, forms the optimal seprating hyperplane, and ,if the training data are linearly seprable, the support vectors belonging to class i satisfy $D_i(x) = 1$ and those belonging remaining classes satisfy $D_i(x) = -1$,if for the input vector x

$$D_i(x) > 0 \tag{3.2}$$

satisfies for one i. x is classified into class i. Since only the sign of the decision function is used, the decision is discrete.

If(4.8) is satisfied for plural i's or there is no i that satisfies(4.8),x is unclassifiable(the shaded region in Fig. 3.1 are unclassifiable regions).To avoid this. x is classified into the class:

$$arg \max_i D_i(x) \tag{3.3}$$

since the continuous value of the decision function determines classification the decision is continuous.

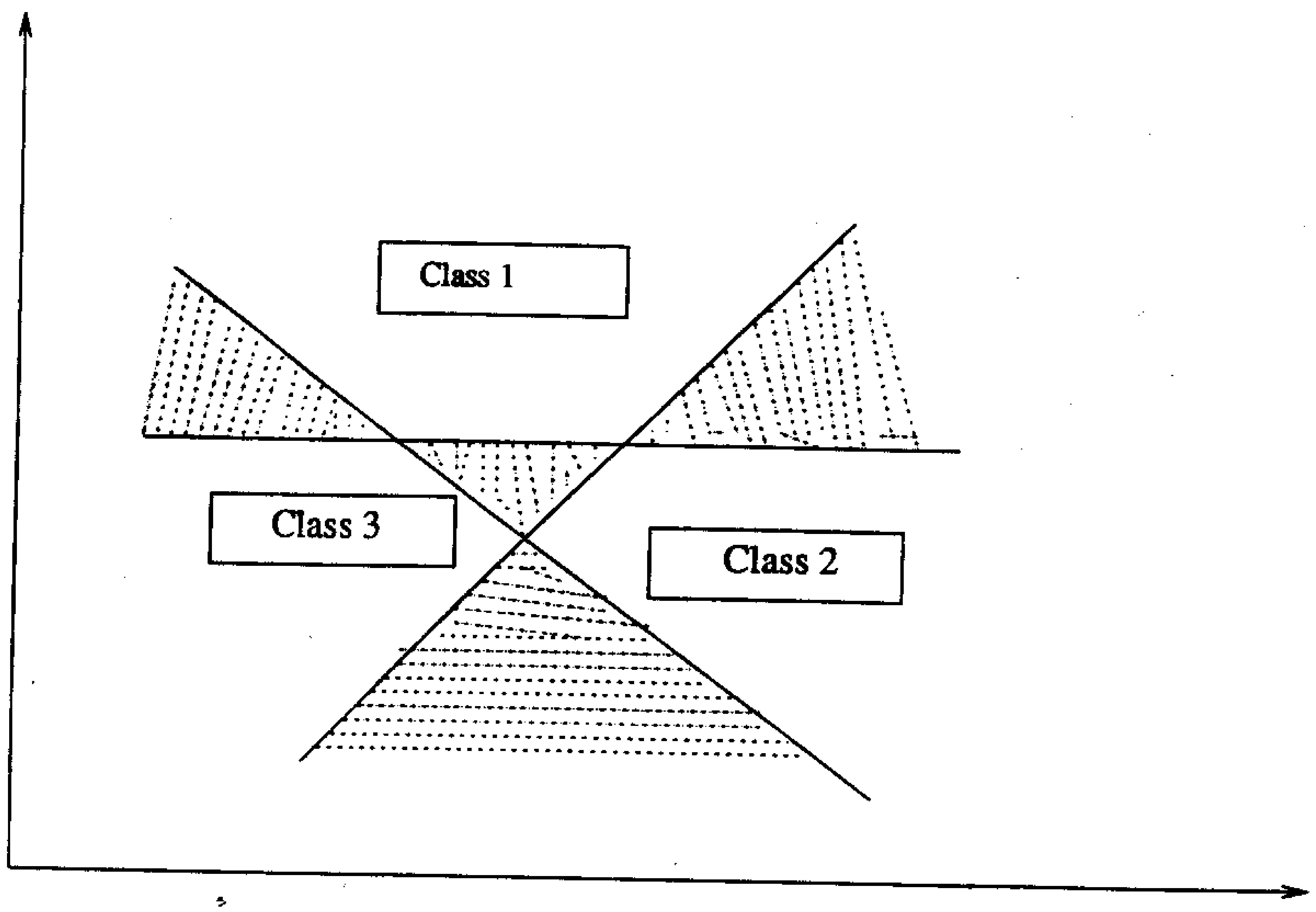


Figure 3.1: Unclassifiable region by the one-agaist-all formulation

3.2 Drawbacks of One-Against-All SVM

This formulation for muliclass svm has following demerits:

1. we need to solve n-quadratic program to detemine the decision boundary for n-class problem and Each Quadratic program has number of constraints equal to the number of Training points.So this makes SVM a computationally expensive classifier.
2. This contains unclassifiable region inside the convex hull of training datas ,which can effect the Generalization performance of SVM

3.3 Pairwise Support Vector Machine

In Pairwise classification, The n-class problem is converted into $\frac{n(n-1)}{2}$ two class problems which covers all pairs of classes. Let the decision function for class i against class j,with maximum margin, be

$$D_{ij}(x) = w_{ij}^t x + b_{ij}. \tag{3.4}$$

where w_{ij} is an m-dimensional vector, b_{ij} is a scalar, and $D_{ij} = -D_{ji}$.for input vector x we calculate

$$D_i(x) = \sum_{j \neq i, j=1}^{n} sign(D_{ij}(x)) \tag{3.5}$$

and classify x into the class

$$arg \max_{i=1,\dots,n} D_i(x). \tag{3.6}$$

But if Eq[3.6] is satisfied for plural i's, x is unclassifiable. In the shaded region in Fig.[3.2], $D_i(x) = 1, \forall i$.Thus the shaded region is unclassifiable. To resolve the unclassifiable region for the pairwise classification, Platt,Cristianini,and J.Shawe-Taylor [5]proposed Decision tree based pairwise classification Called Decision *Directed Acyclic Graph*(DDAG). Fig [4.1] shows the decision tree for three classes shown in Fig 3.2 At the top level, we can choose any pair of classes. And except for leaf node if $D_{ij}(x)$, we consider that x does not belong class j.If $D_{12} > 0$, x belong to class 1 or 3 and the next classification pair is Classes 1 and 3. The generalization region becomes as shown in Fig.7.Unclassifiable region resolved but clearly the generalization region depends on the tree formation.

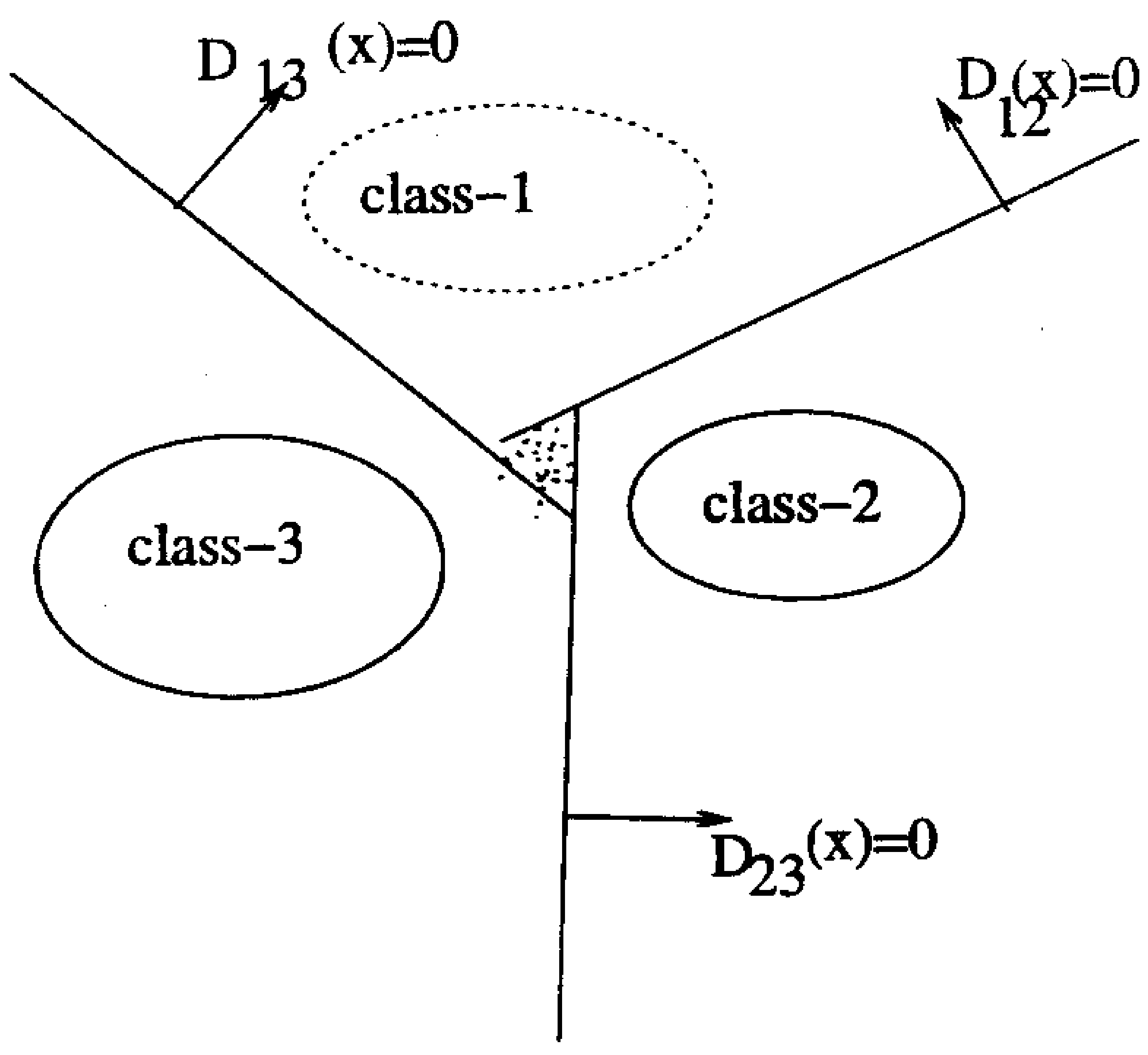


Figure 3.2: Unclassifiable regions by the pairwise formulation.

3.3.1 Drawbacks of Pairwise SVM

1. Number of quadratic programming problem is $\frac{n(n-1)}{2}$ for n-class problem which will make the training process very slow.
2. Although unclassifiable region reduced compared to *One-Against-All SVM* but still it remain.
3. Its performance is too bad for 3-class data set having two class overlap.

3.4 MSVM

Unlike *One-Against-All* and *pairwise svm* . MSVM rquires the solution of single quadratic program for determining the decision function. Assume

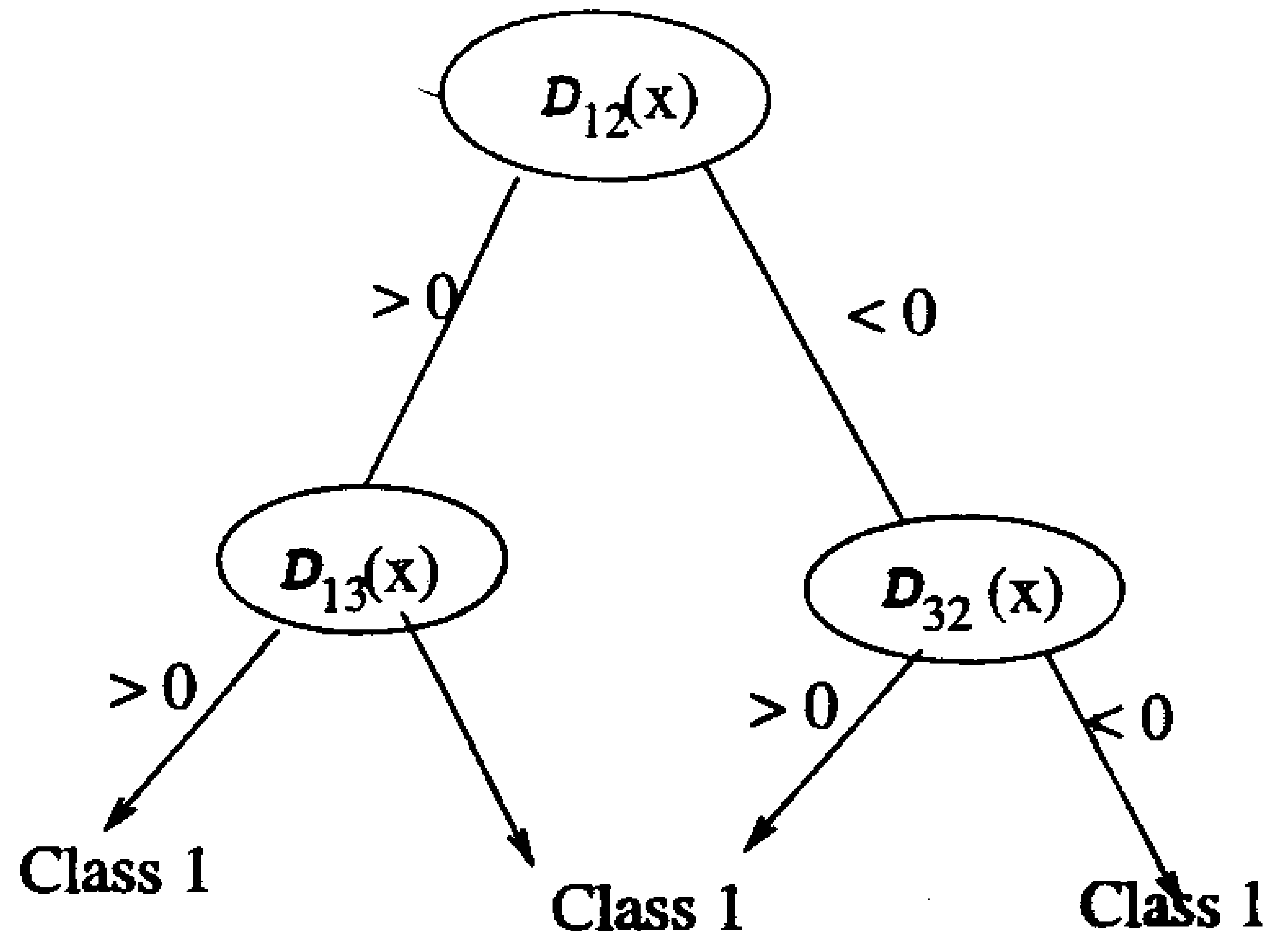


Figure 3.3: DDAG

that the k sets of points are piecewise-linearly seprable. i.e. there exist $w^i \in R^n$ $\gamma^i \in R, i = 1, \ldots, k$,such that For $x \in classi$

$$xw^i - \gamma^i e > xw^j - \gamma^j e \qquad i, = 1, \ldots, k, i \neq j. \tag{3.7}$$

The class of a point x is determined from $(w^i, \gamma^i), i = 1, \ldots, k$ by finding i such that

$$f_i(x) = x^T w^i - \gamma^i$$

is maximized.

For this piecewise-linearly seprable problem, infinitely many (w^i, γ^i)exist that satisfy Eq.[3.7]. intuitively,the "optimal" (w^i, γ^i)provides the largest margin of classification.The margins for each piece $(w^i - w^j, \gamma^i - \gamma^j)$of the piecewise linear seprating function. The margin of sepration between the classes i and j, i.e. the distance between

$$x(w^i - w^j) \geq (\gamma^i - \gamma^j)e + e \qquad for \quad x \in class \quad i$$

and

$$x(w^i - w^j) \geq (\gamma^i - \gamma^j)e - e \qquad for \quad x \in class \ \ j$$

is $\frac{2}{\|w^i - w^j\|}$.so we would like to minimize $\|w^i - w^j\| \ \ \forall i, j = 1.\dots.k. i \neq j$.Also,we add the regularization term $\frac{1}{2}\sum_{i=1}^{k} \|w^i\|^2$to the objective function.For convenience Let A^i be matrix whose rows denote the n-dimensional training points from the class i, For the piecewise linearly seprable problem we get the following Primal problem:

$$\begin{array}{lll} \min\limits_{w^i, \gamma^i} & \frac{1}{2}\sum\limits_{j=1}^{k}\sum\limits_{j=1}^{i-1} \|w^i - w^j\|^2 + \frac{1}{2}\sum\limits_{i=1}^{k} \|w^i\|^2 & \\ s.t & A^i(w^i - w^j) - e(\gamma^i - \gamma^j) - e \geq 0 & (3.8) \\ & i, j = 1, \dots, k \quad i \neq j & \end{array}$$

To simplify the notation for formulation of the piecewise-linear SVM, we rewrite this in matrix notation. For three class following matrix will be obtained: Let

$$\begin{bmatrix} I & -I & 0 \\ I & 0 & -I \\ 0 & I & -I \end{bmatrix}$$

where $I \in R^{n \times n}$ is the identity matrix.

$$A = \begin{bmatrix} A^1 & -A^1 & 0 \\ A^1 & 0 & -A^1 \\ -A^2 & A^2 & 0 \\ 0 & A^2 & -A^2 \\ -A^3 & 0 & A^3 \\ 0 & -A^3 & A^3 \end{bmatrix}$$

$$E = \begin{bmatrix} -e^1 & e^1 & 0 \\ -e^1 & 0 & e^1 \\ e^2 & -e^2 & 0 \\ 0 & -e^2 & e^2 \\ e^3 & 0 & A^3 \\ 0 & e^3 & -e^3 \end{bmatrix}$$

where $A^i \in R^{m_i \times n}, i = 1, \ldots, 3$, and $e^i \in R^{m_i \times l}, 1 = 1, \ldots, 3$, is vector of ones. Here m_i denotes number of point in class i. Using this notation the Primal problem becomes:

$$\begin{aligned} \min_{w,\gamma} \quad & \frac{1}{2}\|Cw\|^2 + \frac{1}{2}\|w\|^2 \\ s.t. \quad & Aw + E\gamma - e \geq 0 \end{aligned} \tag{3.9}$$

where $w = [w^1, \ldots, w^k]$ and $\gamma = [\gamma^1, \ldots, \gamma^k]^T$. The dual of this problem can be written as:

$$\begin{aligned} \min_{w,\gamma} \quad \frac{1}{2}\|Cw\|^2 + \frac{1}{2}\|w\|^2 - u^T(Aw + E\gamma - e) \\ s.t \qquad (I + C^TC)w = A^Tu \\ -E^Tu = 0 \\ u \geq 0. \end{aligned} \tag{3.10}$$

here u is column vector of lagrange multiplier. From Eq.(3.10) we get

$$w = (I + C^TC)^{-1}A^Tu = \frac{1}{k+1}A^Tu \tag{3.11}$$

Using this relationship we can eliminate w from the dual problem.Additionally γ is removed because $-E^Tu = 0$.

After some simplification the new dual problem becomes:

$$\max_{u} \quad e^Tu - \frac{1}{2(k+1)}u^TAA^Tu$$

$$s.t. \quad E^T u = 0 \qquad (3.12)$$
$$u \geq 0$$

So from Eq.(3.13) we will get the optimum value of u and then from Eq. 3.11 value of w can be determined.
To determine the value of γ^i having fixed w we use the inequality in Eq. 3.9.

3.4.1 Formulation of M-SVM:Piecewise Inseprable case

To construct a classification for a piecewise-linearly *inseprable* dataset. we need to add error term in the objective function.
Using the same matrix notation as for linearly seprable case MSVM. The resulting primal problem will be as follows:

$$\begin{aligned} \min_{w,\gamma} \quad & \frac{1}{2}\|Cw\|^2 + \frac{1}{2}\|w\|^2 + c\sum_i^N \xi_i \\ s.t. \quad & Aw + E\gamma - e + \xi \geq 0 \\ & \xi \geq 0 \end{aligned} \qquad (3.13)$$

where ξ is error vector and c is user specified parameter.
Solving for the dual. substituting $w = \frac{1}{k+1}A^T u$, and simplifying produce the following problem:

$$\begin{aligned} \max_u \quad & e^T u - \frac{1}{2(k+1)} u^T A A^T u \\ s.t. \quad & E^T u = 0 \\ & 0 \leq u \leq c \end{aligned} \qquad (3.14)$$

On solving the 3.15 we get the u which results in determination of w. To determine the threshold values $\gamma^i, i, \ldots, k$,we solve the primal problem (3.13) with w fixed. This problem is as follows:

$$\begin{aligned} & \min_{\gamma,\xi} \sum_{i}^{N} \xi_i \\ s.t. \qquad & Aw + E\gamma - e + \xi \geq 0 \\ & \xi \geq 0 \end{aligned} \tag{3.15}$$

3.5 Fuzzy Support Vector Machine

SVM is a powerful tool for solving classification problems, but there are still some limitations of this theory.From the formulation discussed in chapter(2), each training point belongs to either one classs or other.For each class,we can esily check that all training points of this class are treated uniformly in theory of SVM.

In many real -world applications, the effect of the training points are different.It is often that some training points are more important than others in the classification problem. We would require that the meaningful training points must be classified correctly and would not care about some training points like noises whether or not they are misclassified. That is,each training point no more exactly belongs to one of the two classes.It may be 90% belong to one class and 10% be meaningless .In other words,there is fuzzy memebership $0 < s_i \leq$ associated with each training point x_i .This fuzzy membeship $s_i$can be regarded as the attitude of the corresponding training point toward one class in the classification problem and the value $(1 - s_i)$ can be regarded as the attitude of meaningless.So the concept of SVM is extended with fuzzy membership and make it an FSVM.

3.5.1 Formulatisation of Fuzzy Support Vector Machine

Since the Fuzzy membership s_i is the attitude of corresponding point $x_i$toward one class and the parameter ξ_i is measure of error in the SVM, the term $s_i\xi_i$is a measure of error with different weighting. The optimal hyperplane problem is then regarded as solution to following primal quadratic program

$$\min_{w,\xi} \frac{1}{2} w.w + c \sum_{i=1}^{N} s_i \xi_i$$

$$s.t$$

$$d_i(w.x_i + b) \geq 1 - \xi_i, i = 1, \ldots, N$$

$$\xi_i \geq 0 \quad i = 1, \ldots, N \tag{3.16}$$

where c is constant.It is noted that a smaller s_i reduces the effect of the parameter ξ_i in problem (4.9) such that corresponding point x_i is treated as less important.

To solve this optimization problem we construct the Lagrangian

$$L(w, b, \xi, \alpha, \beta) = \frac{1}{2} w.w + c \sum_{i=1}^{N} s_i \xi_i - \sum_{i=1}^{N} \alpha_i (d_i(w \cdot x_i + b) - 1 + \xi_i) - \sum_{i=1}^{N} \beta_i \xi_i \tag{3.17}$$

and the saddle point of $L(w, b, \xi, \alpha, \beta)$. The parameters must satisfy the following conditions:

$$\frac{\partial L(w, b, \xi, \alpha, \beta)}{\partial w} = w - \sum_{i=1}^{N} \alpha_i d_i x_i = 0 \tag{3.18}$$

$$\frac{\partial L(w, b, \xi, \alpha, \beta)}{\partial b} = w - \sum_{i=1}^{N} \alpha_i d_i = 0 \tag{3.19}$$

$$\frac{\partial L(w, b, \xi, \alpha, \beta)}{\partial \xi_i} = s_i c - \alpha_i - \beta_i = 0 \tag{3.20}$$

Apply these codition into the 3.17,the problem 4.9.transformed into

$$Q(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j d_i d_j {x_i}^T x_j \tag{3.21}$$

subject to the constraints

1.
$$\sum_{i=1}^{N} \alpha_i d_i = 0$$

2.
$$0 \leq \alpha_i \leq s_i c \quad for \quad i = 1, 2, \ldots, N$$

and the KKT conditions are defined as

$$\alpha_i(d_i(w \ldots x_i + b) - 1 + \xi_i) = 0, \quad i = 1, \ldots, N \tag{3.22}$$

$$(s_i c - \alpha_i)\xi_i = 0, \quad i = 1, \ldots, N \tag{3.23}$$

The point x_i with the corresponding $\alpha_i > 0$ is called a support vector. There are two types of support vectors.The one with corresponding $0 < \alpha_i < s_i c$ lies on the margin of the hyperplane.The one with corresponding $\alpha_i = s_i c$ is misclassified. An important difference between SVM and FSVM is that the points with the same value of α_i may indicate a different type of support vectors in FSVM due to the factor s_i

Free Parameter

The only free parameter c in SVM controls the tradeoff between the maximization of margin and amount of misclassifications. A larger c makes the training of SVM less misclassification and narrower margin.The decrease of c makes SVM ignore more training points and get the wider margin.

In FSVM, we can set c to be sufficient large value.It is the same as SVM that the system will get narrower margin and allow less misclassification if we set all $s_i = 1$.With different value of s_i,we can control the tradeoff of the respective training point x_i in the sysytem.A smaller value of s_i makes the corresponding point x_i less important in the training.
There is only one free parameter in SVM while the number of free parameters equal to number of training points.

Chapter 4

Proposed Methods for Multiclass SVM

As we have seen there are number of drawbacks with the existing methods for the multiclass SVM. In this chapter we have proposed three new methods for multiclass SVM. we have also proposed two new reformulation of Basic SVM to enhance its performance.

4.1 MMSVM-1(modified multicategory support vector machine-1)

MMSVM is basically modification of MSVM. In MSVM we find n-hyperplane$(w^1, \gamma^1), \ldots .(w^n . \gamma^n)$ where w^i, γ^i represent the hyperplane seprating the class i from the remaining classes. As we know MSVM objective function contains two terms which are $\|w^i - w^j\|^2$ and $\|w^i\|^2$. Here first term is used to maximize the margin between the class i and class j which is piecewise seprated by the plane $(w^i - w^j)x + \gamma^i - \gamma^j)$ and the second term is maximizing the *margin of sepration* between the class i and remaining classes. If we see the decision function. which is $\max_i(x^T w^i - \gamma^i)$. used in MSVM infer that piecewise seprating hyperplane is not the actual decision boundary instead second term of objective function is defining real decision boundary.

Therefore In this modification we have removed $\|w^i - w^j\|^2$ from the objective of MSVM keeping the constraints unchanged.

4.1.1 Formulation of MMSVM1

In this formulation we have used the following notations.

n : number of classes,

N_{ij}^k :kth point in combinally taken training points of class i and class j.

ξ_{ij}^k :Error in kth point in combinally taken training points of class i and class j,

A^i :Training points of class i

A,E has defined in section 3.3

Other notation has usual meaning.

Primal problem:

$$\begin{array}{lc} & \min\limits_{w^i,\gamma^i} \quad \frac{1}{2}\sum_{i=1}^{n}\|w^i\|^2 + c\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{k=1}^{N_{ij}}\xi_{ij}^k \\ s.t & A^i(w^i - w^j) - e(\gamma^i - \gamma^j) - e + \xi_{ij}^k \geq 0 \\ & i,j = 1,\ldots,n \quad i \neq j \qquad \xi_{ij}^k \geq 0 \\ & i,j = 1,\ldots,n, \quad k = 1,\ldots,N_{ij} \end{array} \tag{4.1}$$

where c is user specified non negative number.

Dual problem :

$$\max_u \quad u^T e - \frac{1}{2}u^T AA^T u$$

$$s.t. \tag{4.2}$$
$$E^T u = 0$$
$$0 \leq u \leq c$$

From Dual problem we can obtain the value of u.
where w can be determined by:

$$w = u^T A$$

To determine γ, we need to solve primal problem for fixed w

4.1.2 Decision function

The class of x is determined by finding i such that

$$f_i(x) = x^T w^i - \gamma^i is \ \ maximized.$$

4.2 MMSVM2

In case of pairwise support vector machine discussed in section 3.2, we need to solve $\frac{n}{n-1}$ quadratic programming promblem and it has poor performance over dataset having two class overlap. So we have proposed the support vector machine whose objective function is combinally taking all objective function of pairwise support vector machine and also all the constraints at the same time. which will detemine the *pairwise decision boundary* between the n class and we need to solve one quadratic programming and has improved performance over above mentioned dataset.This gives better performace since here total error is reduced instead of error due to decision boundary between pair of classes. Here Although we need to solve only one quadratic program but it is very big compared to quadratic program in pairwise support vector machine.

4.2.1 Formulation of MMSVM-2

The following notations are used for this formulations *Notations* :
D_{ij}: Decision boundary between class i and class$j$
w_{ij} :Weight vector for D_{ij},
b_{ij} :Scalar for D_{ij},
x^k_{ij} :kth point when training point for class i and class j taken together

$$d^k_{ij} = \begin{cases} 1 & \text{if } x^k_{ij} \in \text{class } i \\ -1 & \text{if } x^k_{ij} \in \text{class } j \end{cases}$$

N_{ij} : Number of training points when class i and class j taken together,
ξ^k_{ij} : Slack variable for x^k_{ij},
α^k_{ij} : kth lagrange multiplier for the first constraints of (4.3)
μ^k_{ij} : kth lagrange multiplier for the second constraints of (4.3)
other notation has usual meaning **Primal problem** :

$$\min_{w_{ij},\xi^k_{ij}} \frac{1}{2}\sum_{i=1}^{n}\sum_{i<j,j=1}^{n} \|w_{ij}\|^2 + c\sum_{i=1}^{n}\sum_{i<j,j=1}^{n} \xi^k_{ij} \tag{4.3}$$

$s.t.$

$$d^k_{ij}(w^t_{ij}x^k_{ij} + b_{ij}) \geq 1 - \xi^k_{ij}$$
$$\xi^k_{ij} \geq 0$$

Here c is user specified parameter. To solve above primal problem we costruct the lagrangian

Lagrangian function:

$$L_p = \frac{1}{2}\sum_{i=1}^{n}\sum_{i<j,j=1}^{n} \|w_{ij}\|^2 + c\sum_{i=1}^{n}\sum_{i<j,j=1}^{n} \xi_{ij}^k - \sum_{i=1}^{n}\sum_{i<j,j=1}^{n}\sum_{k=1}^{N_{ij}} \alpha_{ij}^k\{d_{ij}^k(u_{ij}^t x_{ij}^k + b_{ij})$$
$$-1 + \xi_{ij}^k\}\sum_{i=1}^{n}\sum_{i<j,j=1}^{n}\sum_{k=1}^{N_{ij}} \mu_{ij}^k \xi_{ij}^k \quad (4.4)$$

and find the saddle point. The parameter must satisfy the condition

$$\frac{\partial L_p}{\partial w_{ij}} = 0, \frac{\partial L_p}{\partial b} = 0, \frac{\partial L_p}{\partial \xi_{ij}^k} = 0$$

so we get,

$$w_{ij} = \sum_{k=1}^{N_{ij}} \alpha_{ij}^k d_{ij}^k x_{ij}^k. \quad (4.5)$$

$$\sum_{k=1}^{N_{ij}} \alpha_{ij}^k d_{ij}^k = 0. \quad (4.6)$$

$$c - \alpha_{ij}^k - \xi_{ij}^k = 0. \quad (4.7)$$

Dual problem

$$Q(\alpha) = \max_{\alpha_{ij}{}^k} \sum_{i=1}^{n}\sum_{i<j,j=1}^{n}\sum_{k=1}^{N_{ij}} \alpha_{ij}^k$$
$$-\frac{1}{2}\sum_{i=1}^{n}\sum_{i<j,j=1}^{n}\sum_{k=1}^{N_{ij}}\sum_{l=1}^{N_{ij}} \alpha_{ij}^k \alpha_{ij}^l d_{ij}^k d_{ij}^l x_{ij}^k x_{ij}^l \quad (4.8)$$
$$\sum_{k=1}^{N_{ij}} \alpha_{ij}^k d_{ij}^k = 0$$
$$i, j = 1, \ldots, n$$
$$0 \le \alpha_{ij}^k \le c$$

solving the Eq.(4.8) we will get the optimal value of lagrange coefficients then by using the Eq.(4.5)we will get the value of weight vector w_{ij}.Using the KKT condition we will determine the value of b_{ij}So

$$b_{ij}^k = d_{ij}^k - w_{ij}^t x_{ij}^k$$

Decision boundary:
Here decision function will be same as used in pairwise support vector machine.

4.3 MMSVM-3

MMSVM-3 is modified form of One-Against-All svm. Here instead of solving seprate quadratic program for every hypeplane, we will solve one quadratic programs whose objective function is combination of all n objective functions of one-against-all svm and all costraints are taken together. Hence total number of constraints is n times the number of training points.It is also based on the criterion that minimizing the total error, instead of minimizing the error due to individual decision boundary, will enhance the performace of SVM.

4.3.1 Formulation of MMSVM-3

Following notations are used in this formulation:
D_i: Decision boundary between class i and remaining classes
w_i: Weight vector for D_i
b_i: Scalar for D_i

x_i^k :kth point in training point for class i +remaining classes

$$d_i^k = \begin{cases} 1 & \text{if } x_i^k \in \text{class i} \\ -1 & \text{Otherwise} \end{cases}$$

ξ_i^k : Slack variable for x_i^k,
α_i^k: kth lagrange multiplier for the first constraints in Eq.(4.9),
μ_i^k :k th lagrange multiplier for the second constraints Eq.(4.9).
Primal problem :

$$\min_{w_i,\xi_i^k} \frac{1}{2}\sum_{i=1}^{n} \|w_i\|^2 + c\sum_{i=1}^{n}\sum_{k=1}^{N} \xi_i^k \tag{4.9}$$

$s.t.$

$$d_i^k(w_i^t x_i^k + b_i) \geq 1 - \xi_i^k$$
$$\xi_i^k \geq 0$$

where c is user specified paramter. To solve this problem we costruct the lagrangian.
Lagrangian function:

$$L_p = \frac{1}{2}\sum_{i=1}^{n} \|w_i\|^2 + c\sum_{i=1}^{n} \xi_i^k - \sum_{i=1}^{n}\sum_{k=1}^{N} \alpha_i^k\{d_i^k(w_i^t x_i^k + b_i) - 1 + \xi_i^k\}$$
$$+\sum_{i=1}^{n}\sum_{i<j,j=1}^{n}\sum_{k=1}^{N_i} \mu_i^k \xi_i^k \tag{4.10}$$

and find the saddle point. The parameter must satisfy the condition

$$\frac{\partial L_p}{\partial w_i} = 0, \frac{\partial L_p}{\partial b_i} = 0, \frac{\partial L_p}{\partial \xi_i^k} = 0$$

So we get,

$$w_i = \sum_{k=1}^{N} \alpha_i^k d_i^k x_i^k \tag{4.11}$$

$$\sum_{k=1}^{N} \alpha_i^k d_i^k = 0 \tag{4.12}$$

$$c - \alpha_i^k - \xi_i^k = 0 \tag{4.13}$$

Dual problem

$$\begin{aligned} Q(\alpha) &= \max_{\alpha_i{}^k} \sum_{i=1}^{n} \sum_{k=1}^{N} \alpha_i^k \\ -\frac{1}{2} \sum_{i=1}^{n} \sum_{k=1}^{N} \sum_{l=1}^{N} \alpha_i^k \alpha_i^l d_i^k d_i^l x_i^k x_i^l & \\ \sum_{k=1}^{N} \alpha_i^k d_i^k &= 0 \\ i, j &= 1, \ldots, n \\ 0 \leq \alpha_i^k &\leq c \end{aligned} \tag{4.14}$$

After solving the Eq.(4.14) we will get the optimal value of lagrange coefficients then by using the Eq.(4.11) we will get the value of weight vector w_i. Using the KKT condition, we will determine the value of b_i.So

$$b_i^k = d_i^k - w_i^t x_i^k$$

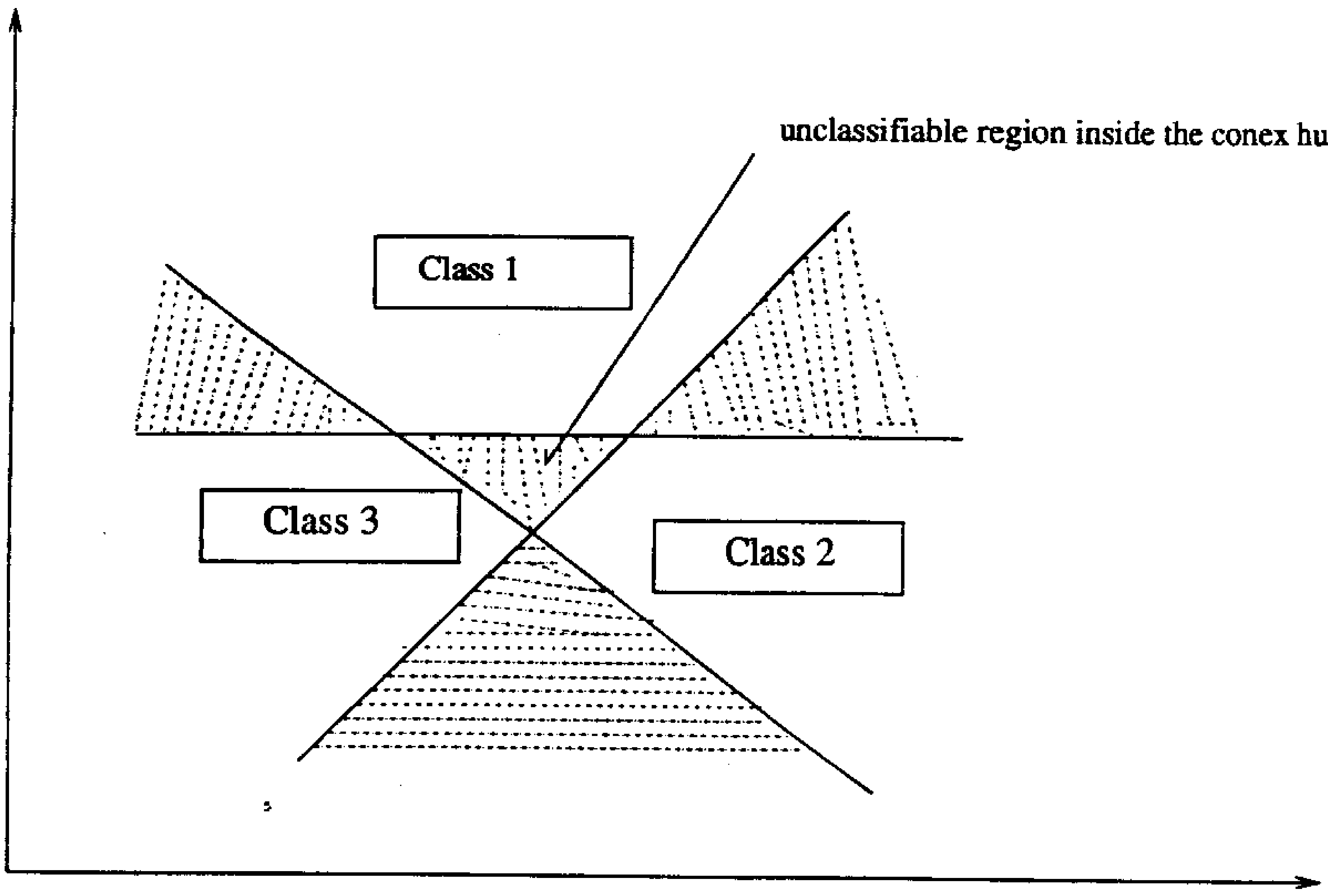
Decision Function:

Here decision function will be same as used in one-against-all support vector machine.

4.4 Reformulation of Basic SVM

Although SVM is a very powerful tool for solving the classification problem but it has two major drawbacks:

1. Unclassifiable region exist inside the convex hull of data as shown in Fig.(1.



2. SVM is highly sensitive to *outliers* and *noise* in the training set as shown in the figure 4.1

So to reduce the effect of above mentioned drawbacks we have proposed two types of SVMs :

1. SBSVM (Shift Boundary SVM).
2. USVM (utility based SVM).

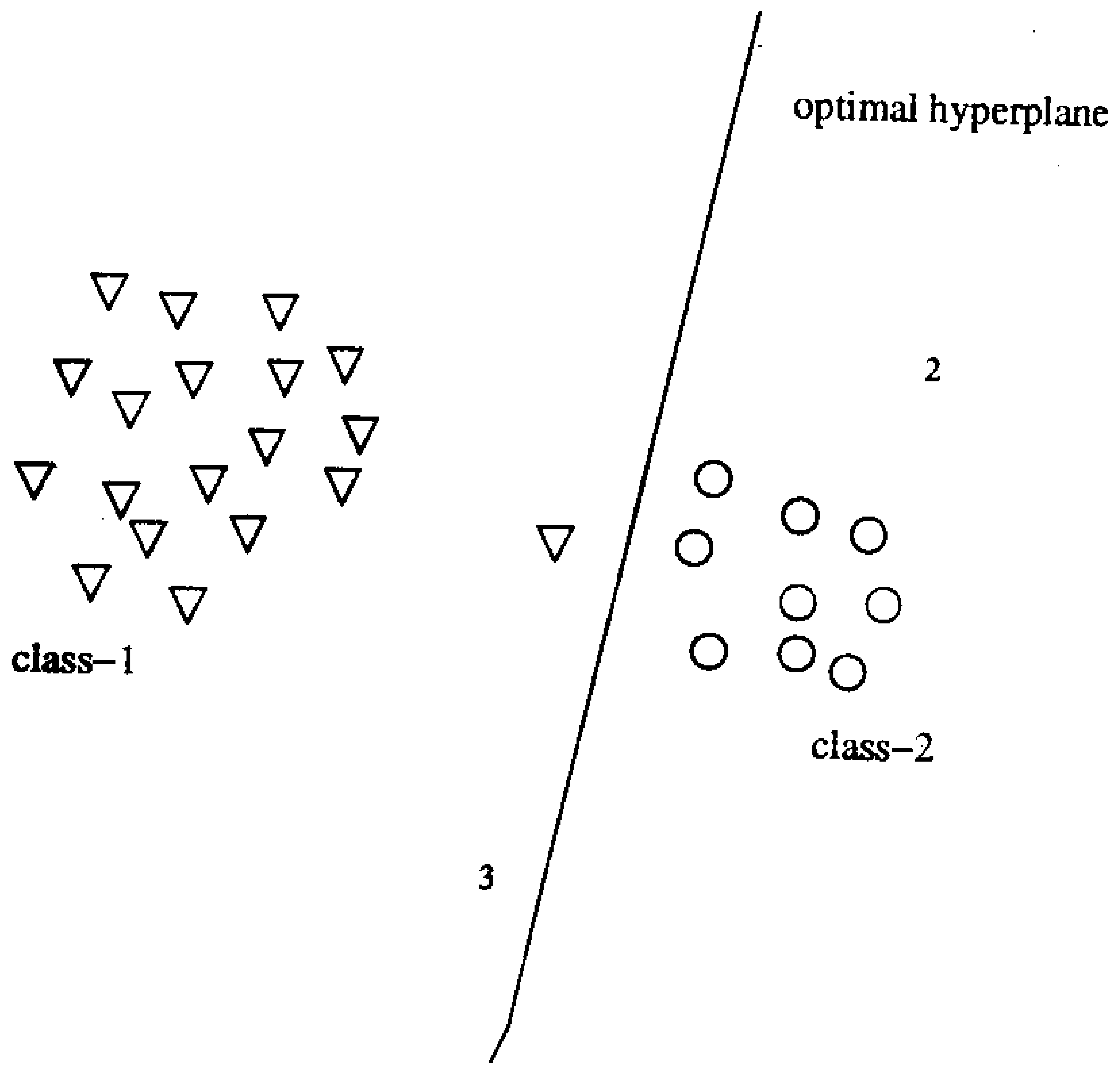


Figure 4.1: SVM hyperplane sensitive to outliers

4.4.1 SBSVM

Here we have proposed an algorithm to reduce the area of unclassifiable region inside the convex hull of the data in optimal way. This method is based on the fact(Vapnik [1]) that *if the dataset is nicely seprable then number of support vectors are less compare to the case when dataset is nonseprable.* consider a three class problem having decision boundary

$$w_i x + b_i, i = 1, 2, 3$$

here we will replace the plane by the parallel plane.So we only change the scalar b_i and w_i will be unaffected.Let the updation in the b_i is ub_i .so our new decision plane will be

$$w_i x + b_i + ub_i, i = 1, 2, 3.$$

Here we are replacing the optimal hypeplane by other parallel hyperplane such that sign of none of training points should change in other words sign of support vector with respect to new plane should be same as the previous optimal plane. Here we also want that shifting of the plane should be inversly proposnal to number support vectors. So to get the value of updation in scalar part we need solution of following LPP:

$$\max_{ub_1,ub_2,ub_3} \frac{1}{SV_1}ub_1 + \frac{1}{SV_2}ub_2 + \frac{1}{SV_2}ub_2$$
$$s.t.$$
$$D_i(x)(w_i x + b_i + ub_i) \geq 1 \tag{4.15}$$

Here SV_idenote the number of support vectors for the ith optimal hyprplane

4.4.2 Drawbacks of SBSVM

We observe the following drawbacks of SBSVM

1. This scheme can effect the generalisation of SVM.
2. For Diamond shape data .it will give poor performance.

4.4.3 USVM

We have proposed a method for getting the hyperplane which is less sensitive to outliers and noises in the dataset.Here we are constructing the hperplane based on the density of the training point in the respective class.we are giving the more inportance to the point which is deep inside the dataset and less importance to training which is far away from their respective class.To obtain this we are measuring the potential of training point.Let p_i denote the potential of the ith point in the class.Then *potential of ith training point is defined by*

$$p_i = e^{-\frac{\|x_v - x_i\|^2}{\sigma^2}}$$

where σ is determined by average edge length of minimum spaning tree (mst) of respective class and then the utility of the ith point is denoted by μ_i

$$\mu_i = \frac{p_i}{\max_i p_i}.$$

So then we need to solve following primal problem

$$\min_{w,\xi} \frac{1}{2} w.w + c \sum_{i=1}^{N} \mu_i \xi_i s.t d_i(w.x_i + b) \geq 1 - \xi_i, i = 1, \ldots, N \xi_i \geq 0 \quad i = 1, \ldots, N \tag{4.16}$$

where c is constant.

Chapter 5

Experimental Results

We have taken Three data sets generated randomly under the normal distribution using the MATLAB Toolbox.Each dataset containing the datas from three classes and each class has 70 data points in which 50 data points is for training and 20 data point for testing

1. : In first data set(say ,$F1$), datas of three classes are nicely seprable.
2. : In second data set(say ,$F2$), there is 2-class overlap.
3. : In third data set(say ,$F3$), there is 3-class overlap.

Linear Decision Boundary

Observations

From Table 5.1-5.3. we observed:

1. Modified MMSVM1 performance is either matches or better than MSVM.

Table 5.1: MSVM and MMSVM1

	$F1$	$F2$	$F3$	$F1$	$F2$	$F3$
Training error	0%	2%	18.66%	0%	2%	14.66%
Test error	0%	3.33%	23.33%	0%	1.66%	18.33%
No.of SV	5	14	75	5	13	57
Value of C	30	15	5	30	30	5

Table 5.2: Pairwise and MMSVM3

	$F1$	$F2$	$F3$	$F1$	$F2$	$F3$
Training error	0%	4.66%	18.66%	0%	3.33%	18.66%
Test error	0%	11.66%	25%	0%	8.33%	23.33%
Num SV	7	32	75	9	33	76
Value of C	30	30	30	30	30	5

2. Pairwise SVM giving poor performance for the data set having two class overlap. MMSVM3 giving slightly better performance.

3. MMSVM2 perform better than Pairwise SVM.

Non Linear Decision Boundary

Here we have used Gaussian kernel $K(x,y) = e^{-||x-y||/2\sigma^2}$ in all the experiments σ value mentioned in the table

Observations:

From Table 5.4-5.9, we observed:

1. MSVM performance is worst.

2. Value of σ is inversly propotional to number of support vectors.

Table 5.3: One against all and MMSVM2

	$F1$	$F2$	$F3$	$F1$	$F2$	$F3$
Training error	0%	2.66%	17.33%	0%	2.66%	17.33%
Test error	0%	3.33 %	20%	0%	3.33%	1.66%
Num SV	9	33	77	9	33	34
Value of C	30	30	10	30	30	10

Table 5.4: MSVM

Class name	F2	F2	F2	F3	F3	F3
Sigma	0.5	1	2	0.5	1	2
Training error	6%	6%	6%	18.66%	21.33%	26.66%
Test error	6.66%	5%	6.66%	33.33%	31.66%	33.33%
Num SV	118	63	45	175	109	66
Value of C	30	30	30	30	30	30

Table 5.5: MMSVM1

Class name	F2	F2	F2	F3	F3	F3
Sigma	0.5	1	2	0.5	1	2
Training error	6%	6%	6%	17.33%	23.33%	30%
Test error	6.66%	5%	6.66%	31.66%	35%	31.66%
Num SV	118	63	45	174	109	66
Value of C	30	30	30	30	30	30

Table 5.6: Pairwise SVM

Class name	F2	F2	F2	F3	F3	F3
Sigma	0.5	1	2	0.5	1	2
Training error	0.66%	2%	2%	6.66%	10%	12.66%
Test error	5%	5%	3.33%	33.33%	23.33%	25%
Num SV	115	64	45	143	91	93
Value of C	30	30	30	30	30	30

Table 5.7: MMSVM3

Class name	F2	F2	F2	F3
Sigma	0.5	1	2	2
Training error	0.66%	2%	2%	12%
Test error	5%	5%	3.33%	25%
Num SV	115	64	45	94
Value of C	30	30	30	30

Table 5.8: One against All

Class name	F2	F3
Sigma	0.5	0.5
Training error	2%	6.66%
Test error	3.33%	31.66%
Num SV	115	115
Value of C	30	30

Table 5.9: MMSVM2

Class name	F2	F2	F2	F3
Sigma	0.5	1	2	0.5
Training error	0.66%	2%	2%	6.66%
Test error	5%	5%	3.33%	31.66%
Num SV	100	97	65	100
Value of C	30	30	30	30

Chapter 6

Optimal Hypersphere

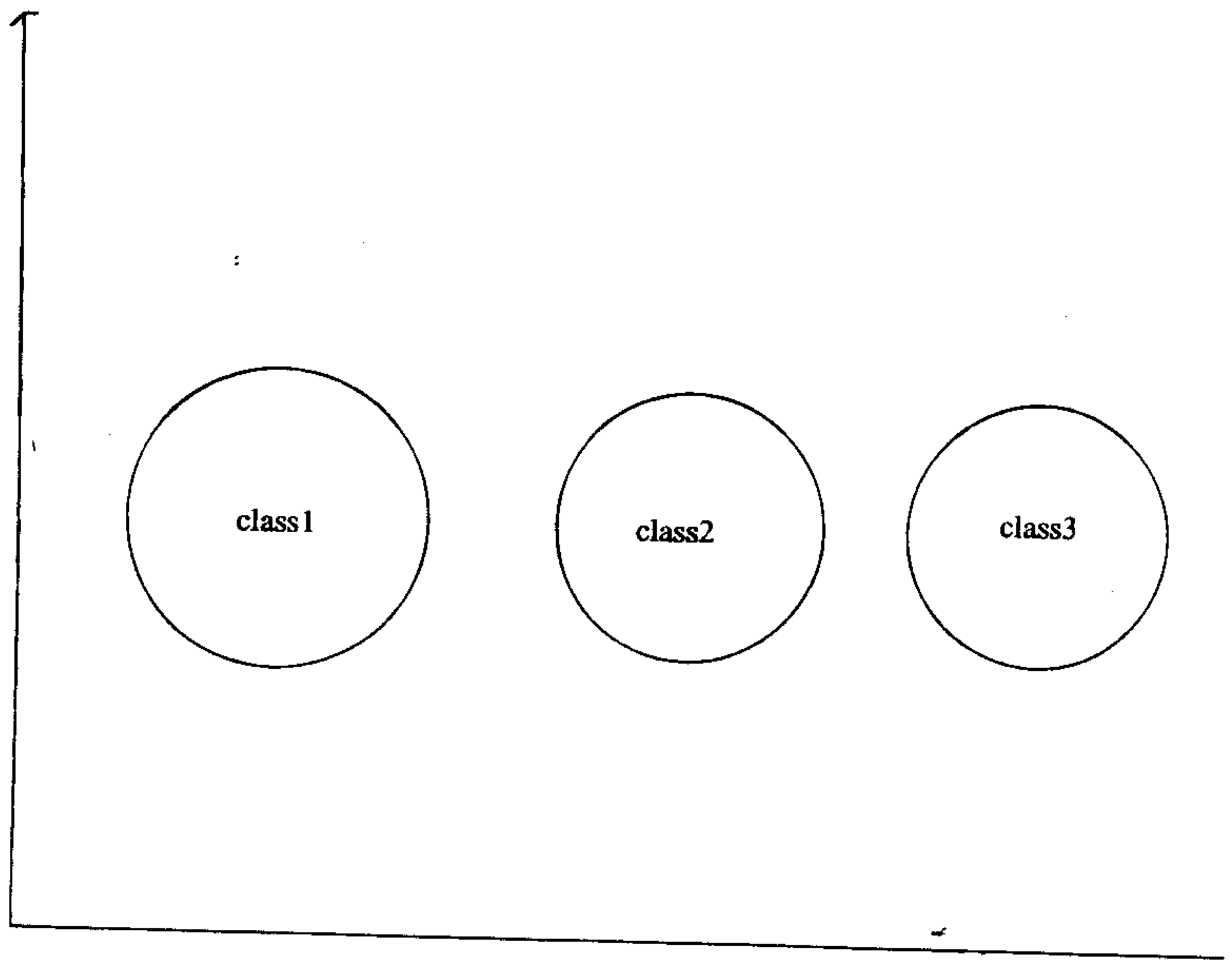


Figure 6.1: Three aligned class

If we have dataset as shown in figure(6.1)Then by One-Against-All

SVM(*without kernel*) for multiclass will not give any solution since it will not able to find the optimal hyperplane seprating class 2 from remaining.So we inspired after the work by David M.J.Tax and Duin [6] regarding data domain description using the hypersphere. We extended their concept to multiclass problem, so Here instead of finding optimal hyperplane, we will find sphere having minimun volume corresponding to each class containing all object of the class.

6.1 Theory

Now we formulate the concept of classification by hypersphere for two class problem. We try to find spheres for respective class with minimum volume containing all(or most of)dataobjects.This is very sensitive to most outlying object in the target data.Sphere with high volume will also increase the misclassification. Therefore we allow some for data points outside the sphere and introduce slack variables ξ_i

suppose R_1,R_2 and a_1,a_2 denotes the radius and centres of two spheres. To get the optimal value of R_1,R_2 and a_1,a_2, we need to solve following primal problem:

$$F(R_1, R_2, a_1, a_2, \xi_1^i, \xi_2^i) = \min_{R_1, R_2, a_1, a_2} R_1^2 + R_2^2 + c\sum {\xi_1}^i + c\sum {\xi_2}^i \tag{6.1}$$

$$\begin{aligned} s.t. \quad (x_1^i - a_1)^T(x_1^i - a_1) &\leq R_1^2 + \xi_1^i, for \quad x_1^i \in class1 \\ (x_2^i - a_2)^T(x_2^i - a_2) &\leq R_2^2 + \xi_2^i, for \quad x_2^i \in class2 \\ &\xi_1^i \geq 0, \xi_2^i \geq 0, \end{aligned}$$

Langrangian for above optimiztion problem will be

$F(R_1, R_2, a_1, a_2, \xi_1^i, \xi_2^i, \alpha_1^i + \alpha_2^i) = R_1^2 + R_2^2 + c\sum {\xi_1}^i + c\sum {\xi_2}^i$

$$-\Sigma_{i=1}^{N_1}\alpha_1^i\{R_1^2+\xi_1^i-({x_1^i}^2-2a_1x_1^i+a_1^2)\}$$

$$-\sum\gamma_1^i\xi_1^i\sum_{i=1}^{N_1}\alpha_2^i\{R_2^2+\xi_2^i-({x_2^i}^2-2a_2x_2^i+a_2^2)\}-\sum\gamma_2^i\xi_2^i \qquad (6.2)$$

L is minimized with respect to R_1, R_2, ξ_1, ξ_2 and maximized w.r.t α_1^i, α_2^i

$$\frac{\partial L}{\partial R_1}=0\Rightarrow 2R_1-2\sum_i\alpha_1^iR_1=0\Rightarrow\sum_i=\alpha_1^i=1 \qquad (6.3)$$

$$\frac{\partial L}{\partial R_2}=0\Rightarrow 2R_2-2\sum_i\alpha_2^iR_2=0\Rightarrow\sum_i=\alpha_2^i=1 \qquad (6.4)$$

$$\frac{\partial L}{\partial\xi_1^i}=0\Rightarrow c-\alpha_1^i-\gamma_1^i=0 \qquad (6.5)$$

$$\frac{\partial L}{\partial\xi_2^i}=0\Rightarrow c-\alpha_2^i-\gamma_2^i=0 \qquad (6.6)$$

$$\frac{\partial L}{\partial a_1}=0\Rightarrow a_1=\sum\alpha_1^ix_1^i \qquad (6.7)$$

$$\frac{\partial L}{\partial a_2}=0\Rightarrow a_2=\sum\alpha_2^ix_2^i \qquad (6.8)$$

$$\alpha_1^i\ge 0 \qquad (6.9)$$

$$\gamma_1^i\ge 0 \qquad (6.10)$$

take $0\le\alpha_1^i\le c$ and $0\le\alpha_2^i\le c$ this implies that $\xi_1^i=0,\xi_2^i=0$; Rewriting the Eq.(6.2) and using the equation we get the *Dual problem*, stated as follows:

$$\max_\alpha L=\sum\alpha_1^i(x_1^i,x_1^i)+\sum\alpha_1^i(x_1^i,x_1^i)-\sum_i\sum_j\alpha_1^i\alpha_1^j(x_1^i,x_1^j)$$
$$-\sum_i\sum_j\alpha_1^i\alpha_1^j(x_1^i,x_1^j)$$

$$\sum_i\alpha_1^i=1$$
$$\sum_i\alpha_2^i=1$$
$$0\le\alpha_1^i\le c$$
$$0\le\alpha_1^i\le c \qquad (6.11)$$

From the Eq. 6.7 we can see that the center of the sphere is a linear combination of dat with the weight factor α_1^i and α_2^i obtained by solving the dual problem.Only for small set of training points equality constraint in (6.2) satisfied .For those data point α_1^i and α_2^i are non zero and are called support vector.

6.1.1 Decision function

For a given Test point x
If x is not in ambiguous region then

$$f(x) = \min_i\{(x - a_i)^T(x - a_i) - R_i\} \tag{6.13}$$

value of $f(x)$ will be class label for x.
If x is in ambiguous region then

$$f(x) = \min_i\{(x - a_i)^T(x - a_i)\} \tag{6.14}$$

value of $f(x)$ will be class label for x.

6.2 Conclusion

In this report we have discussed various types of existing methods for multicategory support vector machine and point out their drawbacks and proposed six new formulation for multicategory SVM which are as follows :

1. MMSVM1 (Modified Multicategory Support Vector Machine 1)
2. MMSVM2 (Modified Multicategory Support Vector Machine 2)

3. MMSVM3 (Modified Multicategory Support Vector Machine 3)
4. SBSVM (Shift Boundary Support Vector Machine)
5. USVM (Utility based Support Vector Machine)
6. Optimal Hyper Sphere Support Vector Machine

We have given theoretical justification and followed by experimental results for each of them. . There are following thing which can be possible extensions of reported work.

1. : All the methods for muliclass SVM using optimal hyperplane can be extended for optimal Hyper sphere.
2. : Devlopment of multiclass SVM using using optimal ellipsoid.

Bibliography

[1] V.N Vapnik.*The nature of statistical learning theory* john Willy son,1995

[2] Han Pang Huang and Yi-Hung Liu.Fuzzy support vector machine for pattern recognition and data Mining*International journal of fuzzy support vector machine*vol.4,No.3,September 2002

[3] Kristin, E. J. Bredensteiner,Multicategory classification by support vector machine ,*Computational Optimizations and Applications*, 12, 1999, pp 53-79.

[4] U.H.G Krebel.pairwise classification and support vector machine.*www.colorado.edu/ling/courses/Fall2003/7800/nips03.pdf*

[5] S.A.Solla,T.K.Leen,and K.-R. Miiller,editors*Advances in neural Information Processing Systems 12*pages 547-553.The MIT Press.2000

[6] Support vector domain description *Pattern recognition letters* vol 20(1999)page no.1191-1199