Introduction to Support Vector Machines

Colin Campbell,
Bristol University

Outline of talk.

Part 1. An Introduction to SVMs

1.1. SVMs for binary classification.

1.2. Soft margins and multi-class classification.

1.3. SVMs for regression.

Part 2. General kernel methods

2.1 Kernel methods based on linear programming and other approaches.

2.2 Training SVMs and other kernel machines.

2.3 Model Selection.

2.4 Different types of kernels.

2.5. SVMs in practice

Advantages of SVMs

- A principled approach to classification, regression or novelty detection tasks.
- SVMs exhibit good generalisation.
- Hypothesis has an explicit dependence on the data (via the support vectors). Hence can readily interpret the model.

- Learning involves optimisation of a convex function (no false minima, unlike a neural network).
- Few parameters required for tuning the learning machine (unlike neural network where the architecture and various parameters must be found).
- Can implement confidence measures, etc.

1.1 SVMs for Binary Classification.

Preliminaries: Consider a binary classification problem: input vectors are $\mathbf{x}_i$ and $y_i = \pm 1$ are the *targets* or *labels*. The index i labels the pattern pairs $(i = 1, \ldots, m)$.

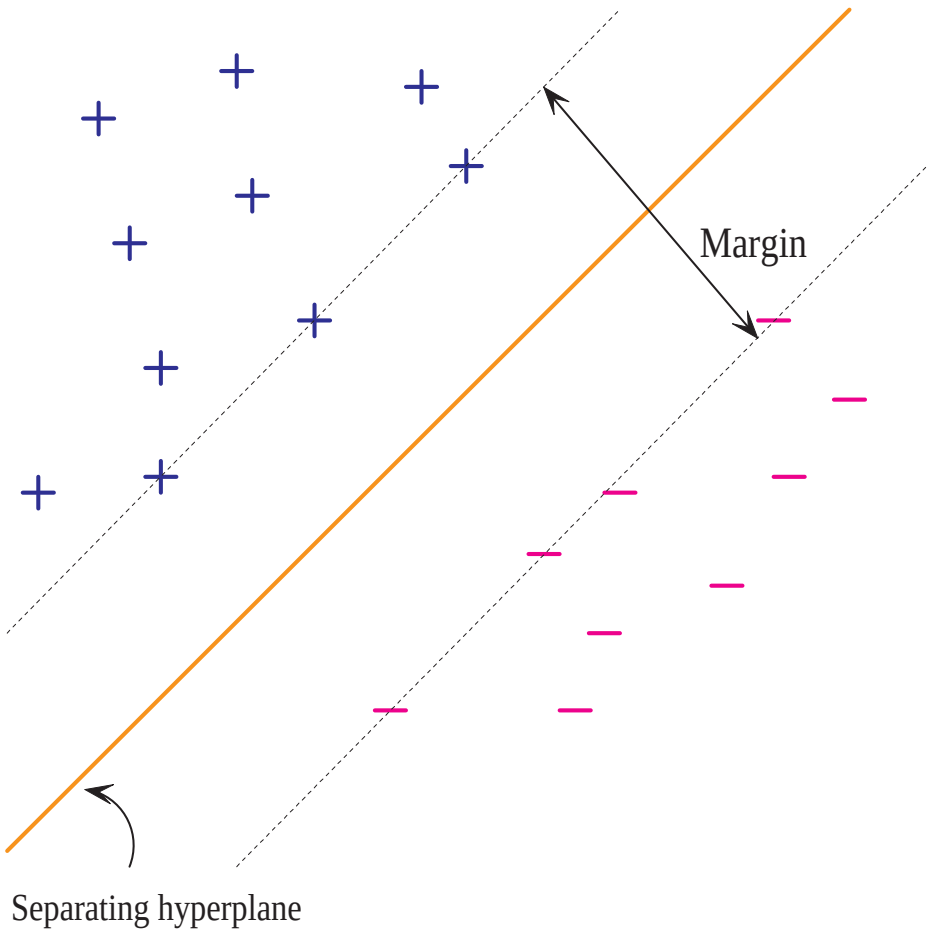
The $\mathbf{x}_i$ define a space of labelled points called *input space*.

From the perspective of *statistical learning theory* the motivation for considering binary classifier SVMs comes from theoretical bounds on the generalization error.

These generalization bounds have two important features:

1. the upper bound on the generalization error does not depend on the dimensionality of the space.

2. the bound is minimized by maximizing the *margin*, γ, i.e. the minimal distance between the hyperplane separating the two classes and the closest datapoints of each class.


Margin
Separating hyperplane

In an arbitrary-dimensional space a separating hyperplane can be written:

$$\mathbf{w} \cdot \mathbf{x} + \mathbf{b} = 0$$

where $\mathbf{b}$ is the *bias*, and $\mathbf{w}$ the weights, etc.

Thus we will consider a decision function of the form:

$$D(x) = \text{sign}\,(\mathbf{w} \cdot \mathbf{x} + \mathbf{b})$$

We note that the argument in $D(x)$ is invariant under a rescaling: $\mathbf{w} \to \lambda \mathbf{w}$, $b \to \lambda b$.

We will implicitly fix a scale with:

$$\mathbf{w} \cdot \mathbf{x} + b = 1$$

$$\mathbf{w} \cdot \mathbf{x} + b = -1$$

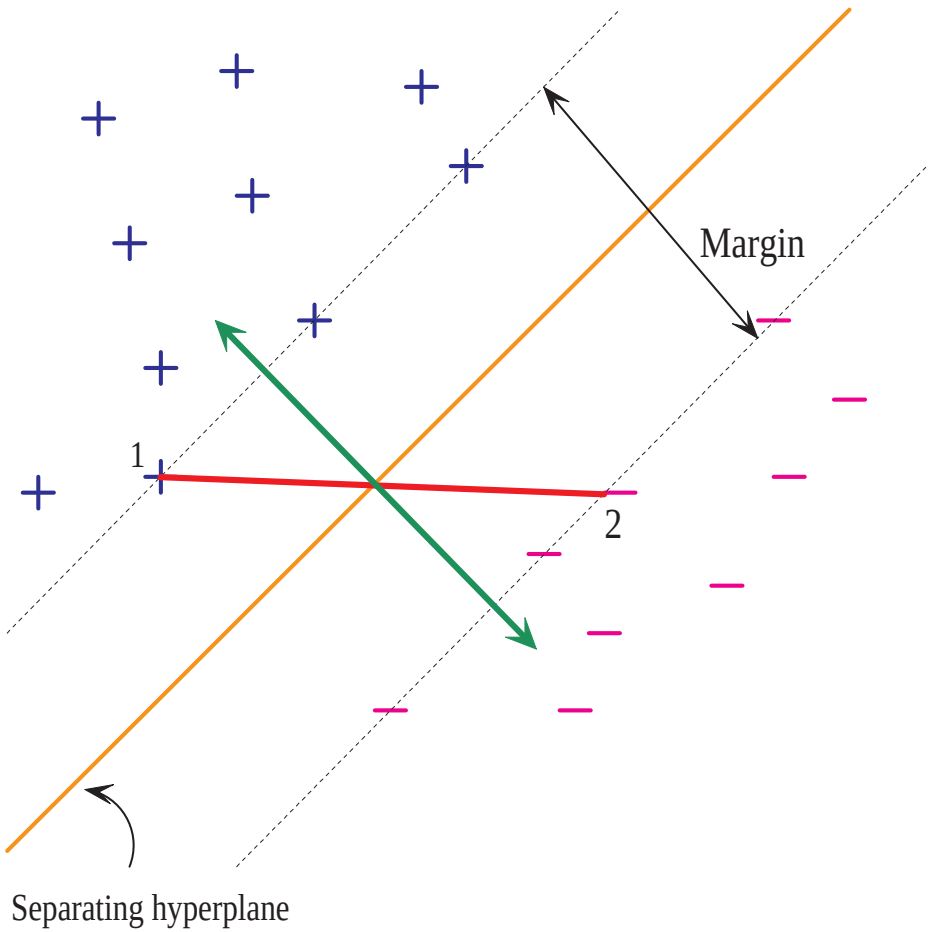
for the support vectors (*canonical hyperplanes*).

Thus:

$$\mathbf{w} \cdot (\mathbf{x}_1 - \mathbf{x}_2) = 2$$

For two support vectors on each side of the separating hyperplane.

The margin will be given by the projection of the vector $(\mathbf{x}_1 - \mathbf{x}_2)$ onto the normal vector to the hyperplane i.e. $\mathbf{w}/||\mathbf{w}||$ from which we deduce that the margin is given by $\gamma = 1/\left|\left|\mathbf{w}\right|\right|_2$.


Margin
1
2
Separating hyperplane

Maximisation of the margin is thus equivalent to minimisation of the functional:

$$\Phi(\mathbf{w}) = \frac{1}{2}(\mathbf{w} \cdot \mathbf{w})$$

subject to the constraints:

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

Thus the task is to find an optimum of the primal objective function:

$$L(\mathbf{w}, b) = \frac{1}{2}(\mathbf{w} \cdot \mathbf{w}) - \sum_{i=1}^{m} \alpha_i \left[y_i((\mathbf{w} \cdot \mathbf{x}_i) + b) - 1 \right]$$

Solving the saddle point equations $\partial L / \partial b = 0$ gives:

$$\sum_{i=1}^{m} \alpha_i y_i = 0$$

and $\partial L/\partial w = 0$ gives:

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i$$

which when substituted back in $L(\mathbf{w}, \alpha^\star, \alpha)$ tells us that we should maximise the functional (the Wolfe dual):

$$W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j \left(\mathbf{x}_i \cdot \mathbf{x}_j \right)$$

subject to the constraints:

$$\alpha_i \geq 0$$

(they are Lagrange multipliers)

and:

$$\sum_{i=1}^{m} \alpha_i y_i = 0$$

The decision function is then:

$$D(\mathbf{z}) = sign\left[\sum_{j=1}^{m} \alpha_j y_j \left(\mathbf{x}_j \cdot \mathbf{z}\right) + b\right]$$

So far we don't know how to handle non-separable datasets.

To answer this problem we will exploit the second point mentioned above: the theoretical generalisation bounds do not depend on the dimensionality of the space.

For the dual objective function we notice that the datapoints, $\mathbf{x}_i$, only appear inside an inner product.

To get a better representation of the data we can therefore map the datapoints into an alternative higher dimensional space through a replacement:

$$\mathbf{x}_i \cdot \mathbf{x}_j \longrightarrow \phi\left(\mathbf{x}_i\right) \cdot \phi(\mathbf{x}_j)$$

i.e. we have used a mapping $\mathbf{x}_i \rightarrow \phi(\ \mathbf{x}_i)$.

This higher dimensional space is called *Feature Space* and must be a *Hilbert Space* (the concept of an inner product applies).

The function $\phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$ will be called a *kernel*, so:

$$\phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j) = K(x_i, x_j)$$

The kernel is therefore the inner product between mapped pairs of points in Feature Space.

What is the mapping relation?

In fact we do not need to know the form of this mapping since it will be implicitly defined by the functional form of the mapped inner product in Feature Space.

Different choices for the kernel $K(x, x')$ define different Hilbert spaces to use. A large number of Hilbert spaces are possible with RBF kernels:

$$K(x, x') = e^{-||x-x'||^2/2\sigma^2}$$

and polynomial kernels:

$$K(x, x') = (< x, x' > +1)^d$$

being popular choices.

Legitimate kernels are not restricted to functions e.g. they may be defined by an algorithm, for example.

(1) String kernels

Consider the following text strings **car**, **cat**, **cart**, **chart**. They have certain similarities despite being of unequal length. **dug** is of equal length to **car** and **cat** but differs in all letters.

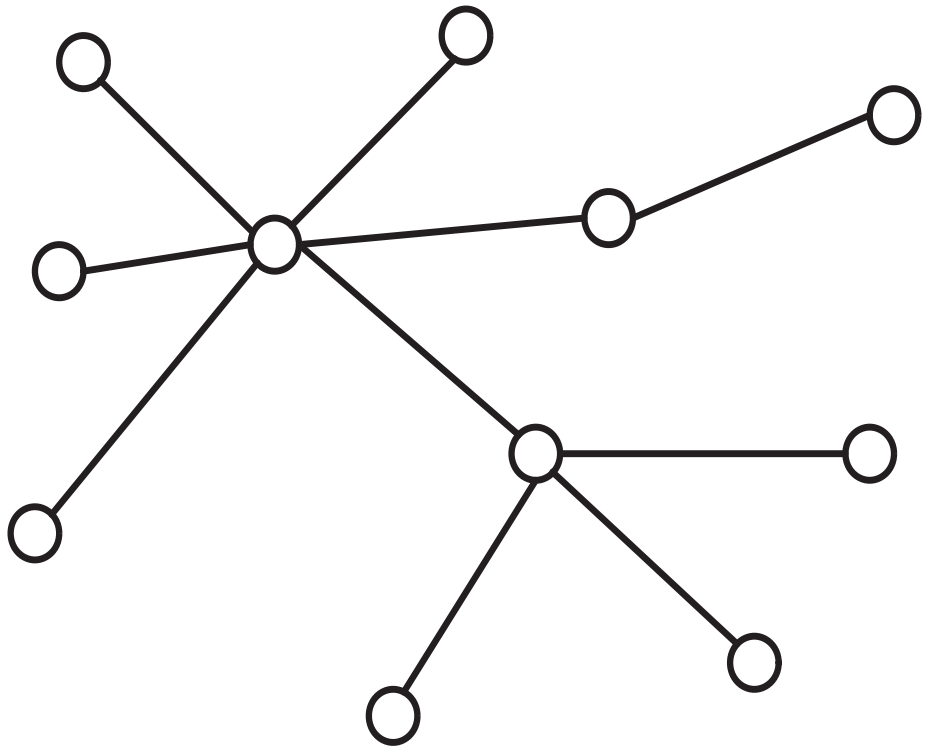
A measure of the degree of alignment of two text strings or sequences can be found using algorithmic techniques e.g. edit codes (dynamic programming).

Frequent applications in

- bioinformatics e.g. Smith-Waterman, Waterman-Eggert algorithm (4-digit strings ... ACCGTATGTAAA ... from genomes),
- text processing (e.g. WWW), etc.

(2) Kernels for graphs/networks

Diffusion kernel (Kondor and Lafferty)



Mathematically, valid kernel functions should satisfy *Mercer's condition*:

For any $g(x)$ for which:

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

it must be the case that:

$$\int K(x, x')g(x)g(x')dxdx' \geq 0$$

A simple criterion is that the kernel should be **positive semi-definite**.

Theorem: If a kernel is positive semi-definite i.e.:

$$\sum_{i,j} K(x_i, x_j) c_i c_j \geq 0$$

where $\{c_1, \ldots, c_n\}$ are real numbers, then there exists a function $\phi(x)$ defining an inner product of possibly higher dimension i.e.:

$$K(x, y) = \phi(x) \cdot \phi(y)$$

Thus the following steps are used to train an SVM:

1. Choose kernel function $K(x_i, x_j)$.

2. Maximise:

$$W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i=1}^{m} \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

subject to $\alpha_i \geq 0$ and $\sum_i \alpha_i y_i = 0$.

3. The bias b is found as follows:

$$\begin{aligned} b &= \frac{1}{2}\left[\min\left(\sum_{\{i|y_i=+1\}} \alpha_i y_i K(x_i, x_j)\right)\right. \\ &+ \left.\max\left(\sum_{\{i|y_i=-1\}} \alpha_i y_i K(x_i, x_j)\right)\right] \end{aligned}$$

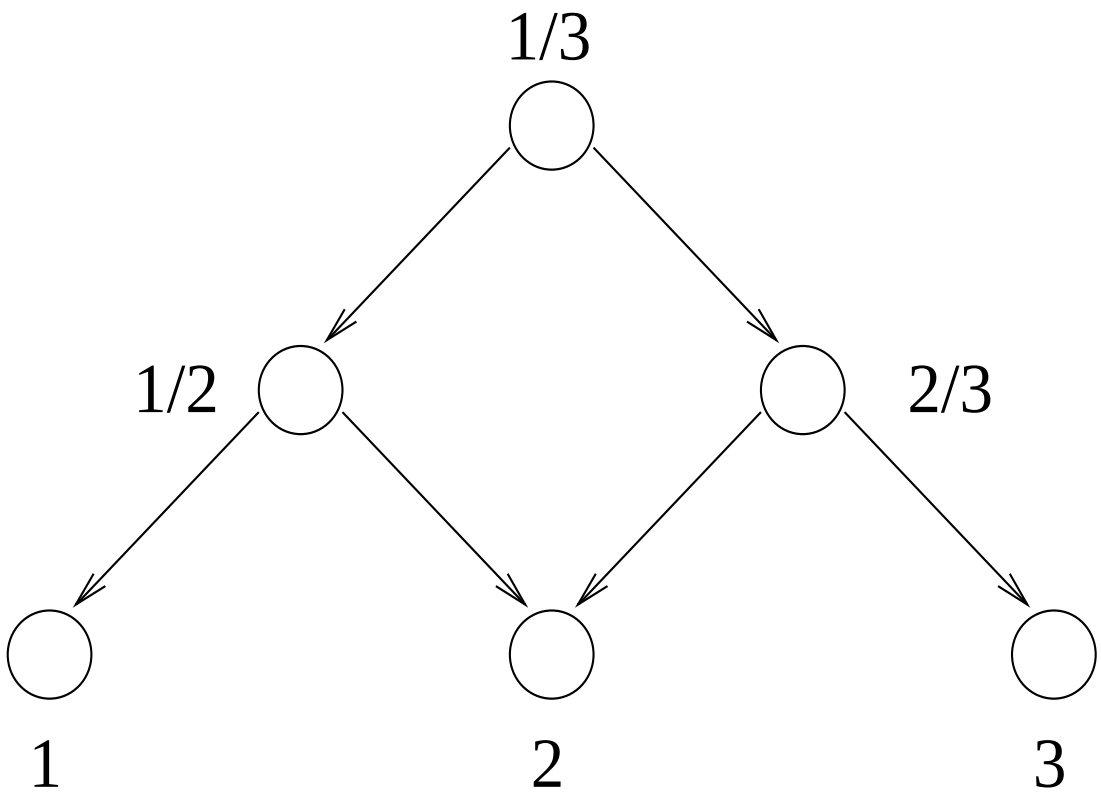
4. The optimal α_i go into the decision function:

$$D(\mathbf{z}) = sign\left(\sum_{i=1}^{m} \alpha_i y_i K(\mathbf{x}_i, \mathbf{z}) + b\right)$$

1.2. Soft Margins and Multi-class problems

Multi-class classification: Many problems involve multiclass classification and a number of schemes have been outlined.

One of the simplest schemes is to use a directed acyclic graph (DAG) with the learning task reduced to binary classification at each node:


1/3
1/2
2/3
1
2
3

Soft margins: Most real life datasets contain noise and an SVM can fit this noise leading to poor generalization.

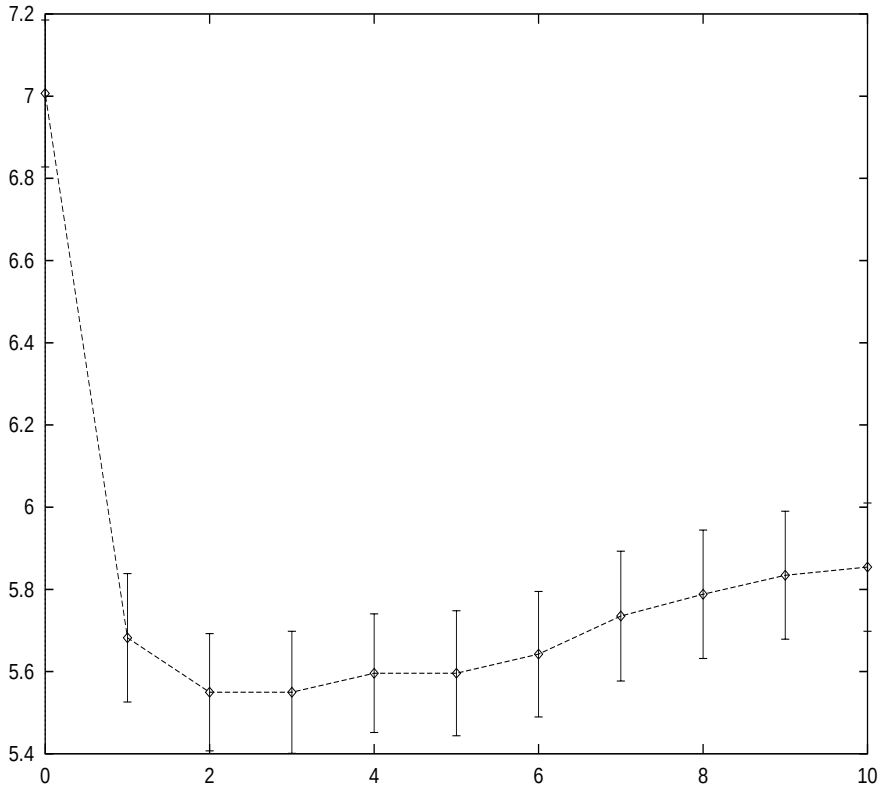
The effect of outliers and noise can be reduced by introducing a *soft margin*. Two schemes are commonly used:

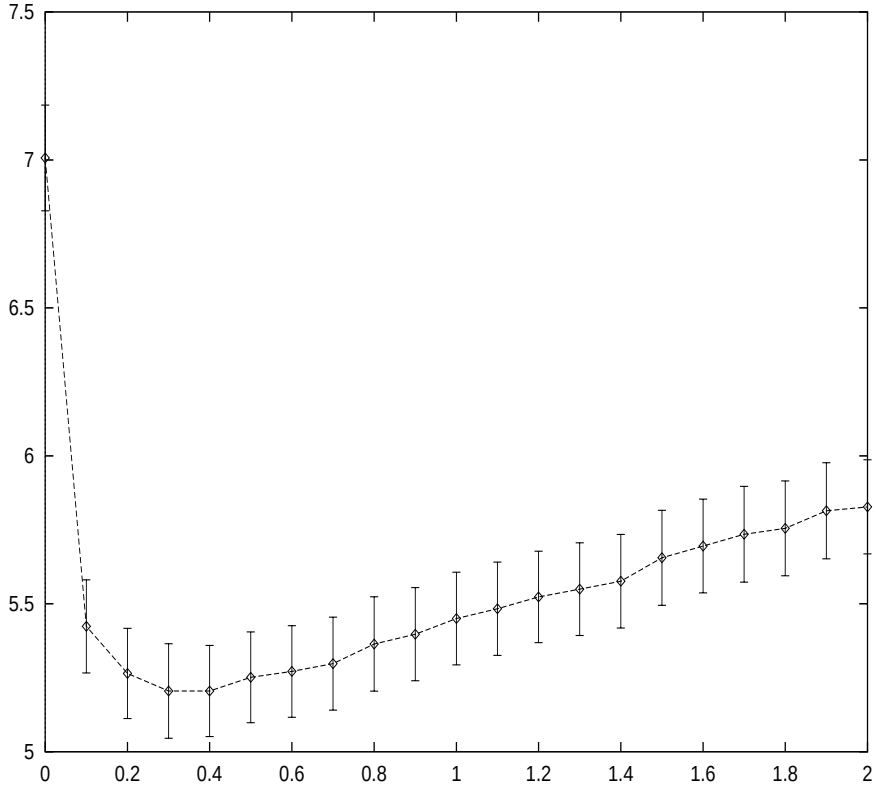
L_1 error norm:

$$0 \leq \alpha_i \leq C$$

L_2 error norm:

$$K(\mathbf{x}_i, \mathbf{x}_i) \leftarrow K(\mathbf{x}_i, \mathbf{x}_i) + \lambda$$


7.2
7
6.8
6.6
6.4
6.2
6
5.8
5.6
5.4
0
2
4
6
8
10


7.5
7
6.5
6
5.5
5
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6
1.8
2

The justification for these soft margin techniques comes from statistical learning theory.

Can be readily viewed as relaxation of the *hard margin* constraint.

For the L_1 error norm (prior to introducing kernels) we introduce a positive slack variable ξ_i:

$$y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b \right) \geq 1 - \xi_i$$

so minimize the sum of errors $\sum_{i=1}^{m} \xi_i$ in addition to $||\mathbf{w}||^2$:

$$\min \left[\frac{1}{2} \mathbf{w} \cdot \mathbf{w} + C \sum_{i=1}^{m} \xi_i \right]$$

This is readily formulated as a primal objective function :

$$\begin{aligned} L(\mathbf{w}, b, \alpha, \xi) &= \frac{1}{2}\mathbf{w} \cdot \mathbf{w} + C \sum_{i=1}^{m} \xi_i \\ &\quad - \sum_{i=1}^{m} \alpha_i \left[y_i \left(\mathbf{w} \cdot \mathbf{x}_i + b \right) - 1 + \xi_i \right] - \sum_{i=1}^{m} r_i \xi_i \end{aligned}$$

with Lagrange multipliers $\alpha_i \geq 0$ and $r_i \geq 0$.

The derivatives with respect to $\mathbf{w}$, b and ξ give:

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i = 0 \quad (1)$$

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{m} \alpha_i y_i = 0 \quad (2)$$

$$\frac{\partial L}{\partial \xi_i} = C - \alpha_i - r_i = 0 \quad (3)$$

Resubstituting back in the primal objective function obtain the same dual objective function as before.

However, $r_i \geq 0$ and $C - \alpha_i - r_i = 0$, hence $\alpha_i \leq C$ and the constraint $0 \leq \alpha_i$ is replaced by $0 \leq \alpha_i \leq C$.

Patterns with values $0 < \alpha_i < C$ will be referred to as *non-bound* and those with $\alpha_i = 0$ or $\alpha_i = C$ will be said to be *at bound*.

The optimal value of C must be found by experimentation using a validation set and it cannot be readily related to the characteristics of the dataset or model.

In an alternative approach, ν-SVM, it can be shown that solutions for an L_1-error norm are the same as those obtained from maximizing:

$$W(\alpha) = -\frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

subject to:

$$\sum_{i=1}^{m} y_i \alpha_i = 0 \qquad \sum_{i=1}^{m} \alpha_i \geq \nu \qquad 0 \leq \alpha_i \leq \frac{1}{m} \qquad (4)$$

where ν lies on the range 0 to 1.

The fraction of training errors is upper bounded by ν and ν also provides a lower bound on the fraction of points which are support vectors.

Hence in this formulation the conceptual meaning of the soft margin parameter is more transparent.

For the L_2 error norm the primal objective function is:

$$\begin{aligned} L(\mathbf{w}, b, \alpha, \xi) &= \frac{1}{2}\mathbf{w}\cdot\mathbf{w} + C\sum_{i=1}^{m}\xi_i^2 \\ &\quad - \sum_{i=1}^{m}\alpha_i\left[y_i\left(\mathbf{w}\cdot\mathbf{x}_i + b\right) - 1 + \xi_i\right] - \sum_{i=1}^{m} r_i\xi_i \end{aligned} \tag{57}$$

with $\alpha_i \geq 0$ and $r_i \geq 0$.

After obtaining the derivatives with respect to $\mathbf{w}$, b and ξ, substituting for $\mathbf{w}$ and ξ in the primal objective function and noting that the dual objective function is maximal when $r_i = 0$, we obtain the following dual objective function after kernel substitution:

$$W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} y_i y_j \alpha_i \alpha_j K(\mathbf{x}_i, \mathbf{x}_j) - \frac{1}{4C} \sum_{i=1}^{m} \alpha_i^2$$

With $\lambda = 1/2C$ this gives the same dual objective function as for hard margin learning except for the substitution:

$$K(\mathbf{x}_i, \mathbf{x}_i) \leftarrow K(\mathbf{x}_i, \mathbf{x}_i) + \lambda$$

For many real-life datasets there is an imbalance between the amount of data in different classes, or the significance of the data in the two classes can be quite different. The relative balance between the detection rate for different classes can be easily shifted by introducing *asymmetric soft margin parameters*.

Thus for binary classification with an L_1 error norm:

$$0 \leq \alpha_i \leq C_+$$

for $y_i = +1$), and

$$0 \leq \alpha_i \leq C_-,$$

for $y_i = -1$, etc.

1.3. Solving Regression Tasks with SVMs

ϵ-SV regression (Vapnik): not unique approach.

Statistical learning theory: we use constraints $y_i - \mathbf{w} \cdot \mathbf{x}_i - b \leq \epsilon$ and $\mathbf{w} \cdot \mathbf{x}_i + b - y_i \leq \epsilon$ to allow for some deviation ϵ between the eventual targets y_i and the function $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x} + b$, modelling the data.

As before we minimize $||\mathbf{w}||^2$ to penalise overcomplexity.

To account for training errors we also introduce slack variables $\xi_i, \widehat{\xi}_i$ for the two types of training error.

These slack variables are zero for points inside the tube and progressively increase for points outside the tube according to the loss function used.

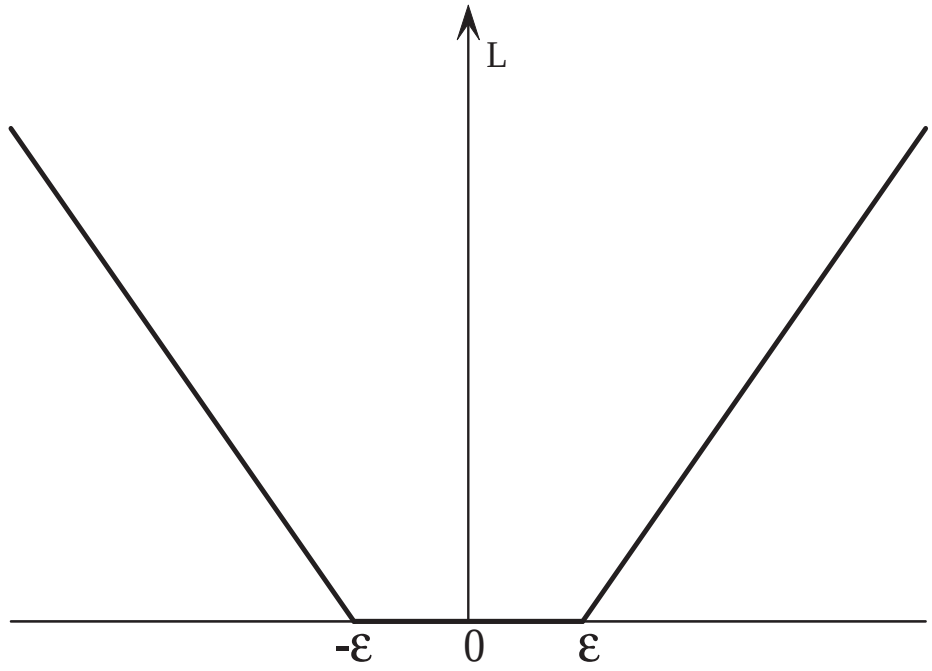


Figure 1: A linear ϵ-insensitive loss function

For a *linear* $\epsilon-$*insensitive loss function* the task is therefore to minimize:

$$\min \left[||\mathbf{w}||^2 + C \sum_{i=1}^{m} \left(\xi_i + \widehat{\xi}_i \right) \right]$$

subject to

$$\begin{aligned} y_i - \mathbf{w} \cdot \mathbf{x}_i - b &\leq \epsilon + \xi_i \\ (\mathbf{w} \cdot \mathbf{x}_i + b) - y_i &\leq \epsilon + \widehat{\xi}_i \end{aligned}$$

where the slack variables are both positive.

Deriving the dual objective function we get:

$$\begin{aligned} W(\alpha, \widehat{\alpha}) &= \sum_{i=1}^{m} y_i(\alpha_i - \widehat{\alpha}_i) - \epsilon \sum_{i=1}^{m} (\alpha_i + \widehat{\alpha}_i) \\ &- \frac{1}{2} \sum_{i,j=1}^{m} (\alpha_i - \widehat{\alpha}_i)(\alpha_j - \widehat{\alpha}_j) K(x_i, x_j) \end{aligned}$$

which is maximized subject to

$$\sum_{i=1}^{m} \widehat{\alpha}_i = \sum_{i=1}^{m} \alpha_i$$

and:

$$0 \leq \alpha_i \leq C \qquad 0 \leq \widehat{\alpha}_i \leq C$$

The function modelling the data is then:

$$f(\mathbf{z}) = \sum_{i=1}^{m} (\alpha_i - \widehat{\alpha}_i) K(\mathbf{x}_i, \mathbf{z}) + b$$

Similarly we can define many other types of loss functions.

We can define a linear programming approach to regression.

We can use many other approaches to regression (e.g. kernelising least squares).