Support Vector Machines

Professor S. Lambotharan

What is SVM?

A classifier based on Convex Optimisation Techniques.
Unlike many mathematical problems in which some form of explicit formula based on a number of inputs resulting in an output, in classification of data there will be no model or formula of this kind. In such cases the system should be trained to be able to recognise the inputs.

Why SVM?

Many classification algorithms do not perform efficiently when

a) The number of features is high,

b) There is a limited time for performing classification,

c) There is a non-uniform weighting among the features,

d) There is a nonlinear map between the inputs and the outputs,

e) The distribution of the data is not known,

f) The convergence is not convex (monotonic), so it may fall into a local minima.

Among the supervised classifiers support vector machines (SVM) is the one, which performs well in the above situations. SVM was initiated in 1979 by Vapnik.

It was used for isolated handwritten digit recognition (Cortes and Vapnik, 1995; Scholkopf, Burges, and Vapnik in 1995, 1996, and 997), object recognition (Blantz et al., 1996), speaker identification (Schmit, 1996), channel quark detection, face detection (Osuna, Freund, and Girosi), text categorization (Joachims, 1997)

Linear Discriminant Functions

$S = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \cdots, (\mathbf{x}_m, y_m)\}$

$\mathbf{x} \in R^d$,

y$\in$ {-1,1} is the class label.

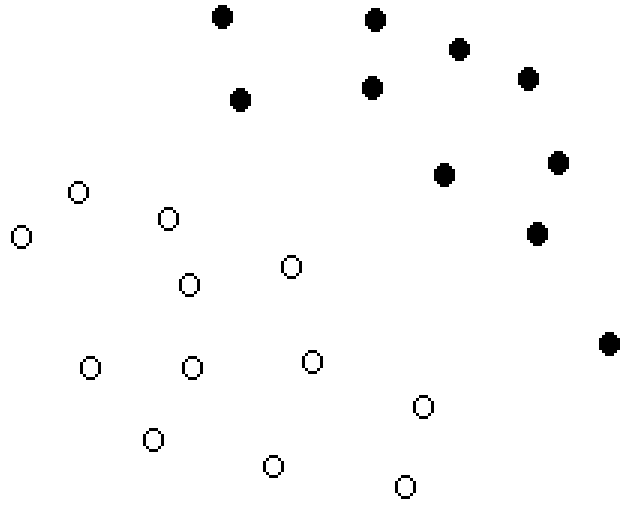


Figure 1 A two dimensional separable dataset

A discriminating function could be defined as:

$$f(\mathbf{x}) = \text{sgn}(\langle \mathbf{w}.\mathbf{x} \rangle + b) = \begin{cases} +1 & \text{if } x \text{ belongs to the first class } \bullet \\ -1 & \text{if } x \text{ belongs to the second class } \circ \end{cases} \quad (1)$$

Infinite number of possible planes !

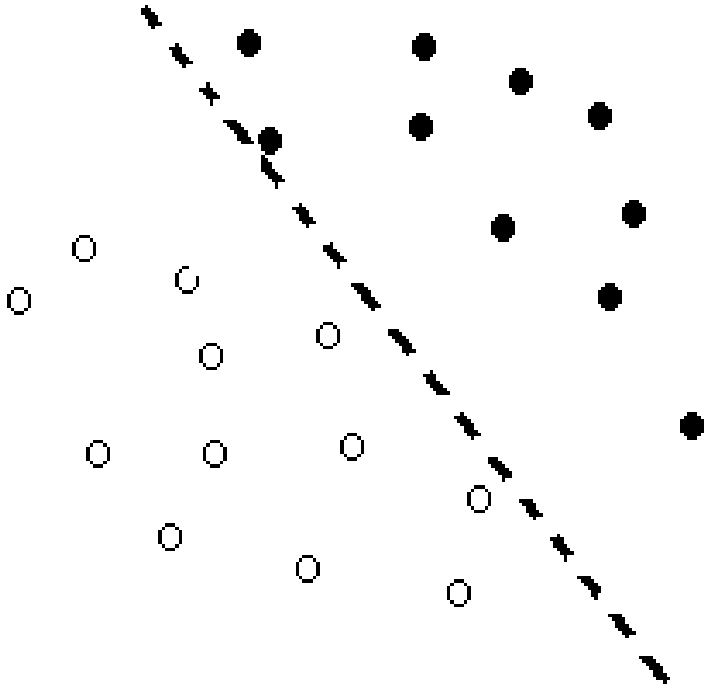


Figure 2 A separating hyperplane

The set of input vectors is said to be optimally separated by the hyperplane if they are separated without error and the distance between the closest vector and the hyperplane is maximal.

So only one hyperplane !

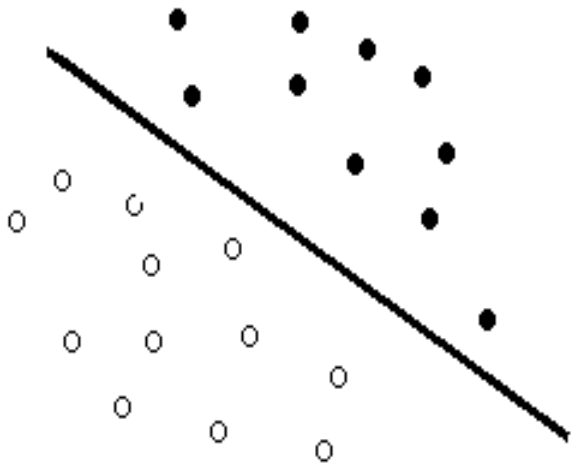


Figure 3 An optimal separating hyperplane

Using Convex Hulls

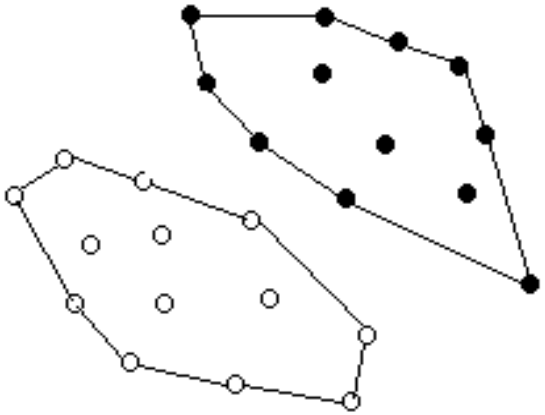


Figure 4 Convex hulls

By examining the hulls one can then determine the closest two points lying on the hulls of each class By constructing a plane that is perpendicular and equivalent to these two points should result in an optimal hyperplane and a robust classifier.

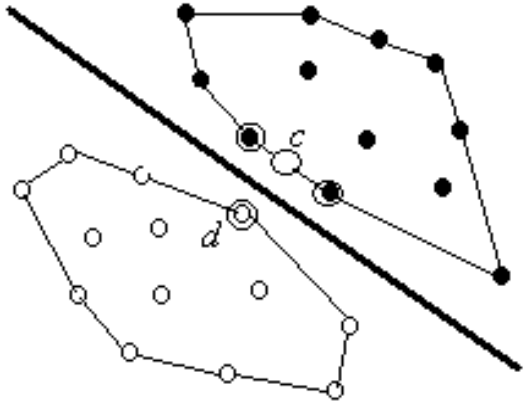
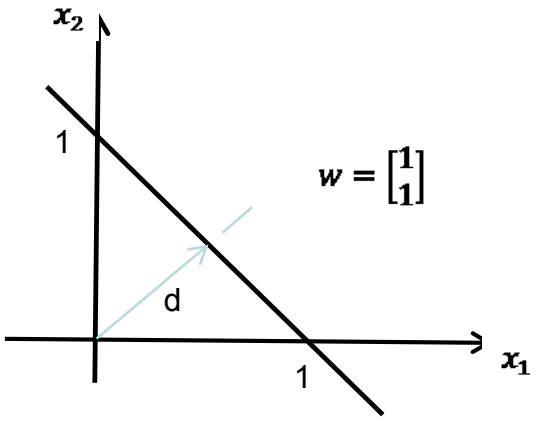


Figure 5 Graphical determination of the convex hulls

The three data points identified with circles are the *Support Vectors* (SVs).

Hyperplanes



$$x_2 = -x_1 + 1$$

$$\Rightarrow \quad x_1 + x_2 - 1 = 0$$

$$\Rightarrow \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} - 1 = 0$$

$$\Rightarrow \quad \mathbf{w}^T\mathbf{x} - c = 0$$

$\|\mathbf{w}\| = \sqrt{2}$

$\frac{-b}{\|\mathbf{w}\|}$ is the perpendicular distance from the hyperplane to the origin. In our case, $d = 1.\cos\left(\frac{\pi}{4}\right) = 1/\sqrt{2}$, which is equal to $\frac{-b}{\|\mathbf{w}\|}$.

SVM ; A mathematical solution

Separable Data

We start with the simplest case: linear machines trained on separable data (as we shall see, the analysis for the general case; nonlinear machines trained on non-separable data results in a very similar quadratic programming problem).

$\{\mathbf{x}_i, y_i\}, i = 1, \cdots, m,\ y_i \in \{-1, 1\},\ \mathbf{x}_i \in R^d.$

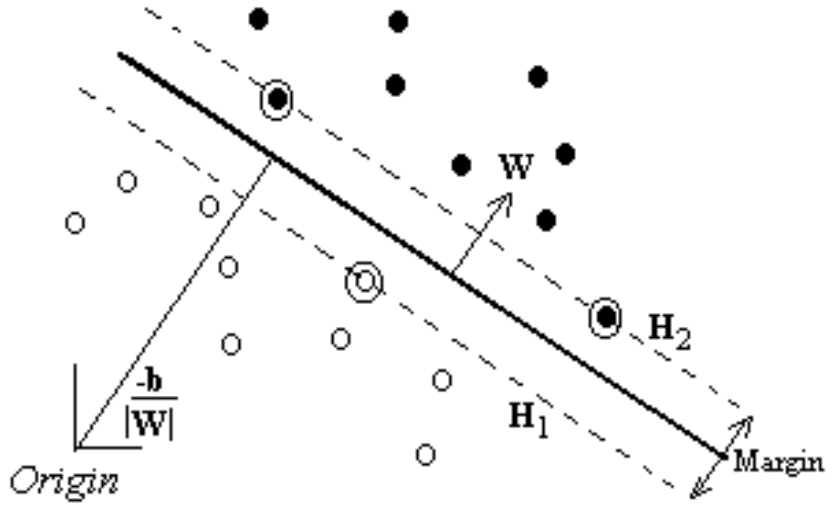
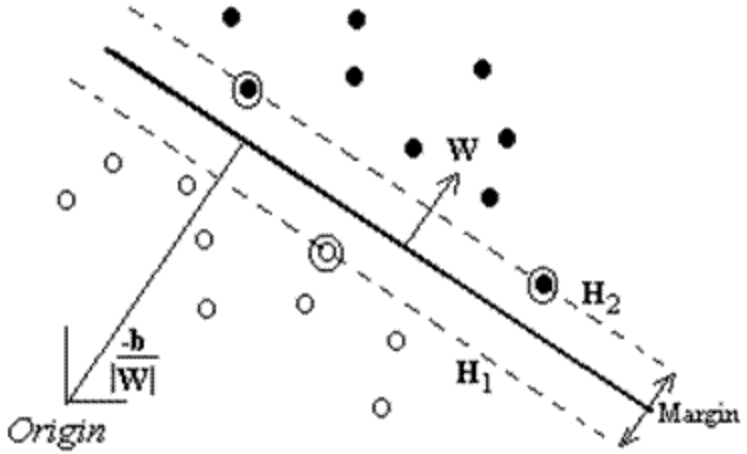


Figure 6: Linear separating hyperplane for the separable case. The support vectors are circled

SVM ; A mathematical solution



Suppose we have a hyperplane which separates the positive from the negative examples.

The points **x** which lie on the hyperplane satisfy $\langle \mathbf{w}.\mathbf{x} \rangle + b = 0$, where **w** is normal to the hyperplane,

$|b|/\|\mathbf{w}\|$ is the perpendicular distance from the hyperplane to the origin,

$\|\mathbf{w}\|$ is the Euclidean norm of **w**.

"Margin" as in the figure

The support vector algorithm simply looks for the separating hyperplane with largest margin.

The Approach

Reduce the problem into convex optimisation by minimising a quadratic function under linear inequality constraints.
Inherent degree of freedom for the features' scales to allow the margins to be set to be equal to 1 for simplicity and subsequently minimise the norm of the weight vector.

To find the plane farthest from both classes of data, we simply maximise the margin between the supporting canonical hyperplanes for each class.

The support planes are pushed apart until they meet the closest data points, which are then deemed to be the support vectors (circled in Figure 6).

So

$$\begin{aligned} \langle \mathbf{x}_i.\mathbf{w} \rangle + b \geq +1 \quad & for\ y_i = +1 \\ \langle \mathbf{x}_i.\mathbf{w} \rangle + b \leq -1 \quad & for\ y_i = -1 \end{aligned} \qquad (2)$$

combined into $y_i\left(\langle \mathbf{x}_i.\mathbf{w} \rangle + b\right) - 1 \geq 0 \quad \forall i$,

The margin between these supporting planes (H_1 and H_2) can be shown to be $\gamma = 2/\|\mathbf{w}\|$.
To maximise this margin we therefore need to

$$\begin{aligned} & minimise\ \langle \mathbf{w}.\mathbf{w} \rangle \\ & subject\ to\ y_i\left(\langle \mathbf{x}_i.\mathbf{w} \rangle + b\right) - 1 \geq 0 \quad i = 1, \cdots, m. \end{aligned} \qquad (3)$$

Solve using Lagrange Multipliers

This leads to minimisation of an unconstrained empirical risk function (Lagrangian) which consequently results in a set of conditions called Karush-Kuhn-Tucker (KKT) conditions as follows.

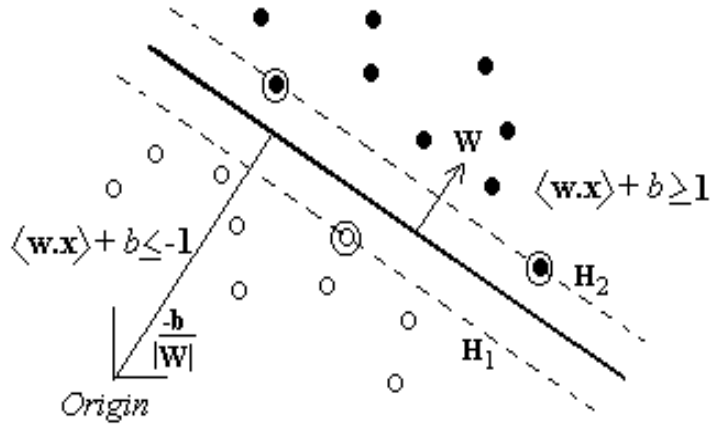


Figure 6: The constraints for the SVM

To perform Lagrangian optimisation we must construct the so-called *primal form*;

$$L(\mathbf{w},b,\boldsymbol{\alpha}) = \frac{1}{2}\langle \mathbf{w}.\mathbf{w} \rangle - \sum_{i=1}^{m} \alpha_i \left[y_i \left(\langle \mathbf{x}_i.\mathbf{w} \rangle + b \right) - 1 \right] \qquad (4)$$

where α_i are the Lagrange multipliers.

Minimise with respect to w, b and maximise with respect to $\alpha_i \geq 0$;

$$\frac{\partial L(\mathbf{w},b,\boldsymbol{\alpha})}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{m} y_i \alpha_i \mathbf{x}_i = 0 \tag{5}$$

thus

$$\mathbf{w} = \sum_{i=1}^{m} y_i \alpha_i \mathbf{x}_i \tag{6}$$

and

$$\frac{\partial L(\mathbf{w},b,\boldsymbol{\alpha})}{\partial b} = \sum_{i=1}^{m} y_i \alpha_i = 0 \tag{7}$$

By replacing into the primal form we get the dual form as

$$L(\mathbf{w},b,\boldsymbol{\alpha}) = \frac{1}{2}\sum_{j=1}^{m}\sum_{i=1}^{m} y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i . \mathbf{x}_j \rangle - \sum_{i=1}^{m}\sum_{i=1}^{m} y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i . \mathbf{x}_j \rangle + \sum_{i=1}^{m} \alpha_i \tag{8}$$

which is reduced to the following optimization:

Maximise $L(\mathbf{w},b,\boldsymbol{\alpha}) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i=1}^{m} y_i y_j \alpha_i \alpha_j \langle \mathbf{x}_i . \mathbf{x}_j \rangle$ (9)

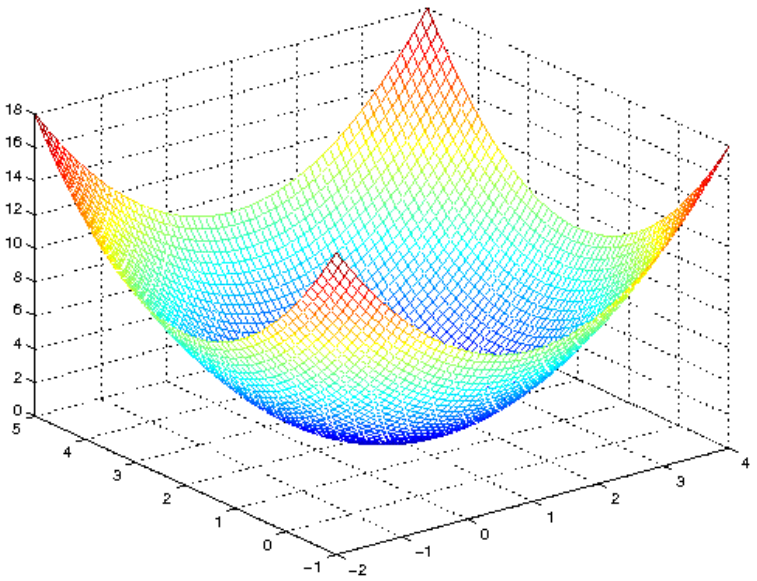
Subject to $\sum_{i=1}^{m} y_i \alpha_i = 0$ and $\alpha_i \geq 0$.

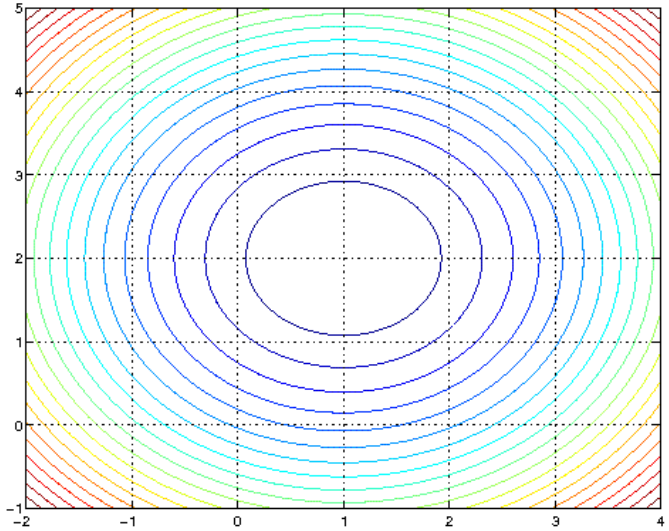
Solvable using QP

Next few slides provide an overview of convex optimization techniques, but we are interested only on the linearly constrained quadratic programming and KKT conditions.

Example of an unconstrained optimization problem

Minimise $J = (x_1 - 1)^2 + (x_2 - 2)^2 = (\mathbf{x} - \mathbf{x}_c)^T (\mathbf{x} - \mathbf{x}_c)$



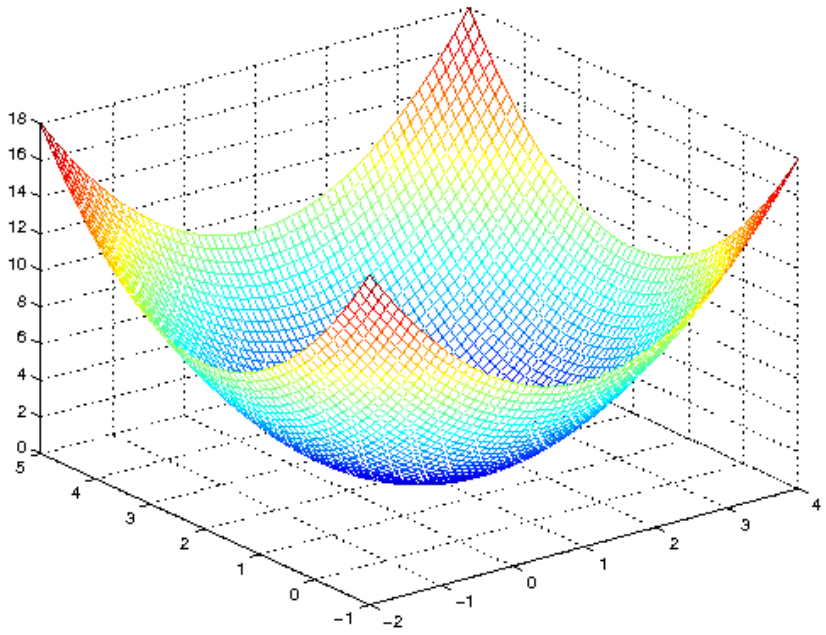


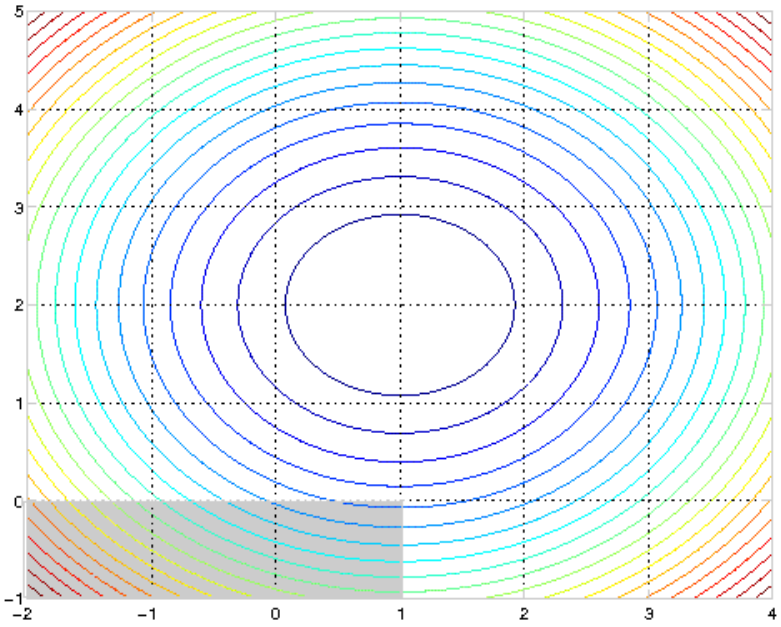
Examples of constrained optimization problems

Minimise $J = (x_1 - 1)^2 + (x_2 - 2)^2 = (\mathbf{x} - \mathbf{x}_c)^T (\mathbf{x} - \mathbf{x}_c)$

subject to $x_1 \leq 1$

$x_2 \leq 0$

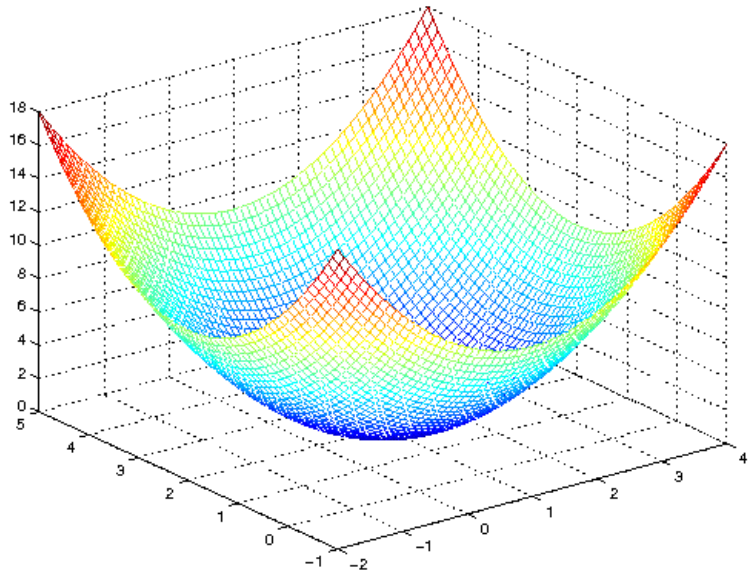


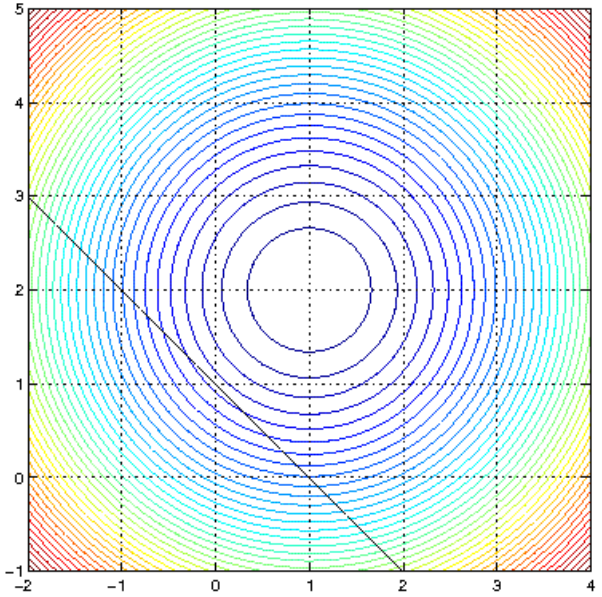


Examples of constrained optimization problems

Minimise $J = (x_1 - 1)^2 + (x_2 - 2)^2 = (\mathbf{x} - \mathbf{x}_c)^T (\mathbf{x} - \mathbf{x}_c)$

subject to $a^T (x - x_0) \le 0$ where $a = [1 \quad 1]^T$ and $x_0 = [1 \quad 0]^T$





$$L(\mathbf{x}, \lambda) = (\mathbf{x} - \mathbf{x}_c)^T (\mathbf{x} - \mathbf{x}_c) + \lambda \, \mathbf{a}^T (\mathbf{x} - \mathbf{x}_0)$$

$$\frac{\partial L(\mathbf{x}, \lambda)}{\partial \mathbf{x}} = 2\mathbf{x} - 2\mathbf{x}_c + \lambda \mathbf{a} = 0 \quad \Rightarrow \mathbf{x}_{opt} = \mathbf{x}_c - \frac{\lambda}{2}\mathbf{a}$$

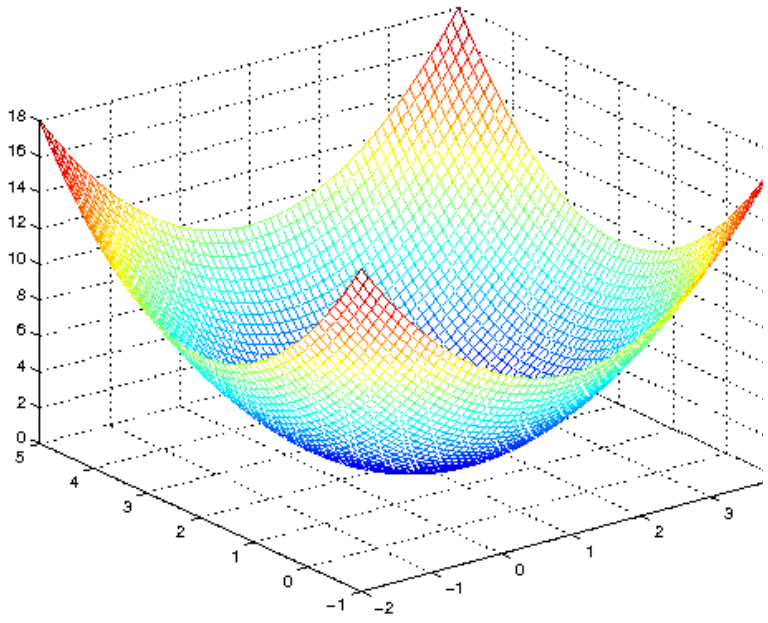
$$\mathbf{a}^T (\mathbf{x}_{opt} - \mathbf{x}_0) = 0 \quad \Rightarrow \quad \mathbf{a}^T (\mathbf{x}_c - \frac{\lambda}{2}\mathbf{a} - \mathbf{x}_0) = 0 \quad \Rightarrow \quad \lambda = 2$$

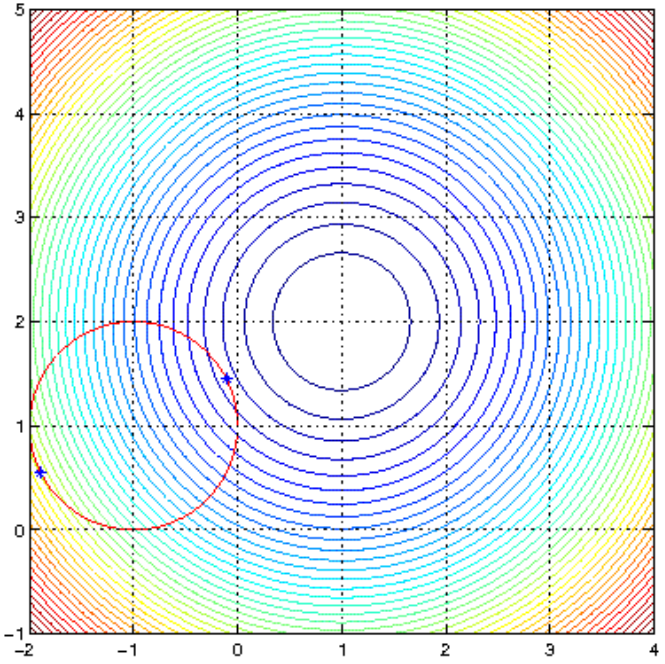
Hence $\mathbf{x}_{opt} = \mathbf{x}_c - \frac{\lambda}{2}\mathbf{a} = \mathbf{x}_c - \mathbf{a} = \begin{bmatrix} 1 \\ 2 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

Examples of constrained optimization problems

Minimise $J = (x_1 - 1)^2 + (x_2 - 2)^2 = (\mathbf{x} - \mathbf{x}_c)^T (\mathbf{x} - \mathbf{x}_c)$

subject to $\left\| (x - x_\mu) \right\|_2^2 \leq 1$ where $x_\mu = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$





Convex Optimization Problem

$$\begin{aligned} \text{minimise} \quad & f_0(x) \\ \text{subject to} \quad & f_i(x) \le 0 \quad i = 1,2,\cdots,m \\ & h_i(x) = b_i \quad i = 1,2,\cdots,p \end{aligned}$$

The objective function must be convex.
Inequality constraint functions must be convex.
The equality constraint function must be affine.

The feasible set of a convex optimization problem is convex.

$$D = \bigcap_{i=0}^{m} dom\ f_i$$

i.e. A convex objective function is minimised over a convex set.

- Linear Programming (LP)
- Linear Fractional Programming
- Quadratic Programming (QP)
- Quadratically Constrained Quadratic Programming (QCQP)
- Second Order Cone Programming (SCP)
- Geometric Programming (GP)
- Semidefinite Programming (SDP)

Linear Programming (LP)

$$\begin{aligned} \text{minimise} \quad & c^T x \\ \text{subject to} \quad & G x \leq h \\ & A x = b \end{aligned}$$

Linear Fractional Programming (LP)

$$\begin{aligned} &\text{minimise} && f_0(x) \\ &\text{subject to} && G\,x \le h \\ &&& A\,x = b \end{aligned}$$

$$\text{where} \quad f_0(x) = \frac{c^T x + d}{e^T x + f} \qquad dom\ f_0 = \{x \,|\, e^T x + f > 0\}$$

Quadratic Programming

$$\begin{aligned} \text{minimise} \quad & x^T P x + q^T x + r \\ \text{subject to} \quad & G x \leq h \\ & A x = b \end{aligned}$$

where $P \in S_+^n, G \in R^{m \text{x} n}, A \in R^{p \text{x} n}$

Quadratically Constrained Quadratic Programming

$$\text{minimise} \quad x^T P_0 x + {q_0}^T x + r_0$$

$$\text{subject to} \quad x^T P_i x + {q_i}^T x + r_i \le 0 \quad i = 1,2,\cdots,m$$

$$A x = b$$

$$\text{where } P \in S_+^n, \quad A \in R^{p \text{x} n}$$

Second Order Cone Programming (SP)

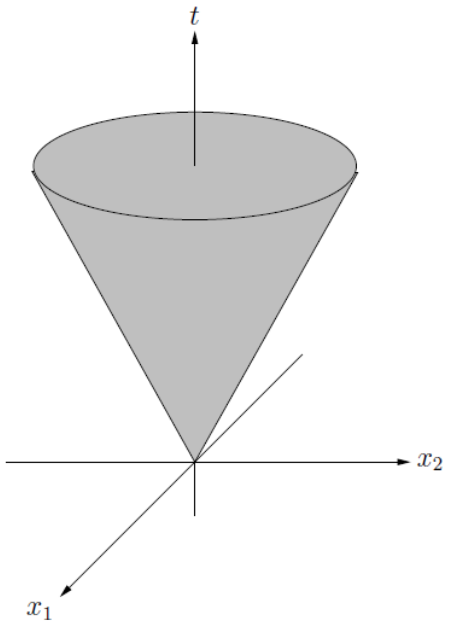
minimise $f^T x$

subject to $\|A_i x + b_i\|_2 \le c_i^T x + d_i \quad i = 1,2,\cdots,m$

$F x = g$

where $x \in R^n$, $A_i \in R^{n_i x n}$, $F \in R^{p x n}$

The constraint $\|A_i x + b_i\|_2 \le c_i^T x + d_i$ is called a second order cone constraint.



Second-order cone in $\mathbf{R}^3$, $\left\{(x_1, x_2, t) \;\middle|\; (x_1^2 + x_2^2)^{1/2} \le t\right\}$

Source: Stephen Boyd- Convex Optimization

Geometric Programming

$$\text{minimise} \quad f_0(x)$$

$$\text{subject to} \quad \begin{aligned} f_i(x) &\le 1 \quad i = 1,2,\cdots,m \\ h_i(x) &= 1 \quad i = 1,2,\cdots,p \end{aligned}$$

where $f_i(x), \quad i = 0,1,2,\cdots,m$ are posynomials and

$h_i(x) = 1 \quad i = 1,2,\cdots,p$ are monomials

Domain of the problem $D = R^n_{++}$

Monomial function: $h(x) = c\, x_1^{a1}\, x_2^{a2} \cdots x_n^{an}, \quad c > 0, a_i \in R$

Posynomial function: $f(x) = \sum_{k=1}^{K} c_k\, x_1^{a_{1k}}\, x_2^{a_{2k}} \cdots x_n^{a_{nk}}, \quad c_i > 0, a_{ik} \in R$

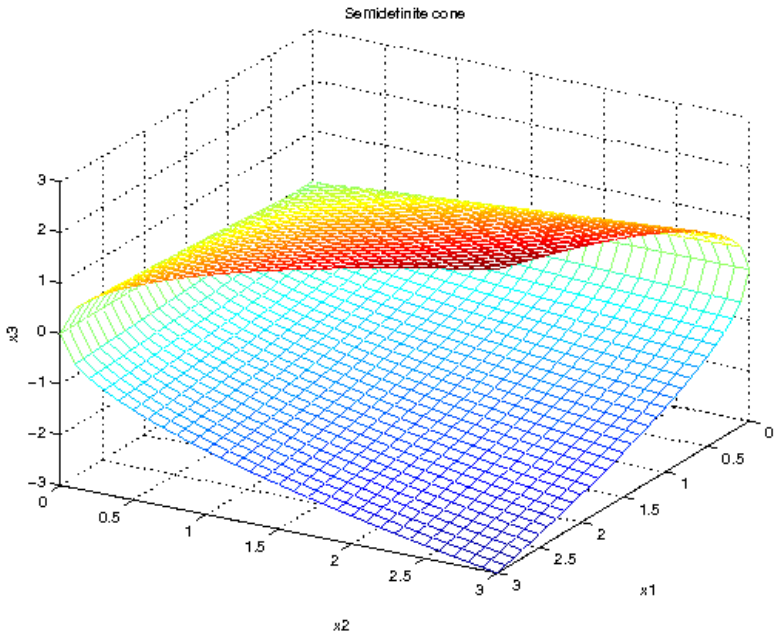
Semidefinite Programming

$$\text{minimise} \quad c^T x$$

$$\text{subject to} \quad x_1 F_1 + x_2 F_2 + \cdots + x_n F_n + G \le 0$$
$$A\,x = b$$

where $F_1, F_2, F_n, G \in S^k, A \in R^{p\text{x}n}$

An example of semidefinite cone $\mathbf{X} = \begin{bmatrix} x1 & x3 \\ x3 & x2 \end{bmatrix} \ge 0$



KKT conditions

Consider the following convex optimization problem [13],

$$\begin{aligned} &\text{minimise} && f_0(x) \\ &\text{subject to} && f_i(x) \le 0, \quad i = 1, \cdots, m \\ & && h_i(x) = 0, \quad i = 1, \cdots, p \end{aligned}$$

We can define Lagrangian associated with the above optimization problem as

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x)$$

The Lagrange dual function is defined as

$$g(\lambda, \nu) = \inf_x L(x, \lambda, \nu) = \inf_x \left(f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x) \right)$$

Assume $f_i(x)$ and $h_i(x)$ are differentiable. Let x^* and (λ_i^*, ν_i^*) be any primal and dual optimal points with zero duality gap. For a problem in which strong duality obtains, the following KKT conditions will be satisfied.

$$\nabla f_0(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla f_i(x^*) + \sum_{i=1}^{p} \nu_i^* \nabla h_i(x^*) = 0$$

$$f_i(x^*) \le 0$$

$$h_i(x^*) = 0$$

$$\lambda_i^* \ge 0$$

$$\lambda_i^* f_i(x^*) = 0$$

The last condition is known as complementary slackness.

Non Separable Case

Can we use a complicated nonlinear hyperplane to perfectly separate the datasets?

overfitting problem !

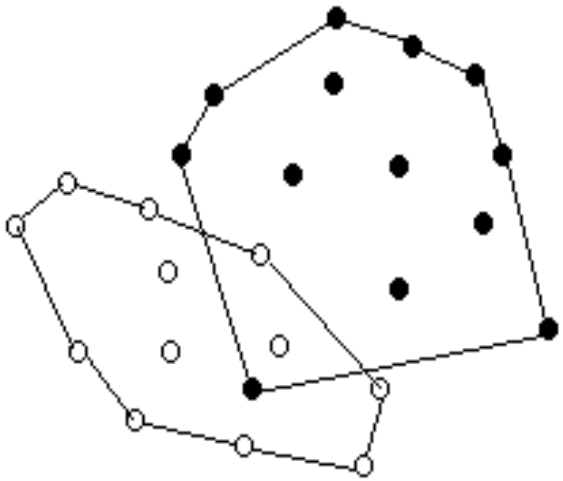


Figure 7. Encompassed regions for non-separable case

The datasets are no longer linearly separable.
The ideal solution where no points are misclassified and no points lie within the margin is no longer feasible.
So relax the constraints to allow for the minimum amount of misclassification.
Use a *soft margin classifier*.

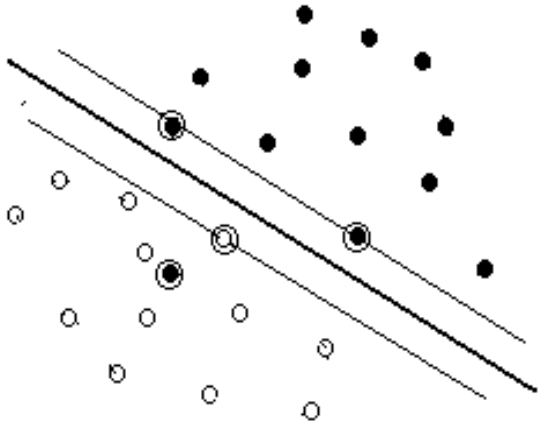


Figure 8. Support vectors in a non-separable case with a linear hyperplane.

$$\begin{aligned} &\langle \mathbf{x}_i.\mathbf{w} \rangle + b \geq +1 - \xi_i \quad for\ y_i = +1 \\ &\langle \mathbf{x}_i.\mathbf{w} \rangle + b \leq -1 + \xi_i \quad for\ y_i = -1 \\ &and\ \xi_i \geq 0\ \forall i \end{aligned} \tag{10}$$

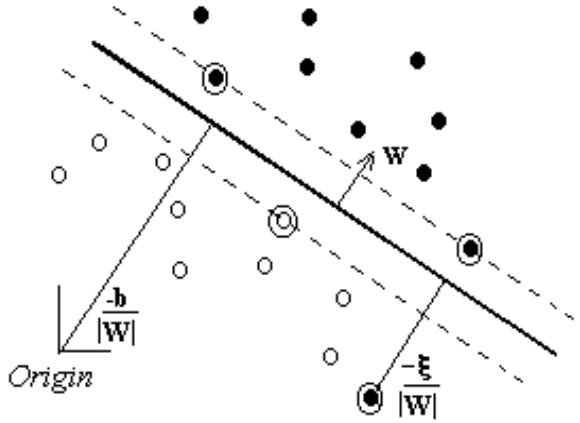


Figure 9. Soft margin and the concept of the slack parameter

For an error to occur, the corresponding ξ_i must exceed unity. $\sum_i \xi_i$ is an upper bound on the number of training errors.

The objective function changes to

$$\begin{aligned} &\textit{minimise } \langle \mathbf{w}.\mathbf{w} \rangle + C\sum_{i=1}^{m}\xi_i \\ &\textit{subject to } y_i\left(\langle \mathbf{x}_i.\mathbf{w} \rangle + b\right) \geq 1-\xi_i \quad \textit{and } \xi_i \geq 0 \quad i=1,\cdots,m\,. \end{aligned} \tag{11}$$

The primal form will then be:

$$L(\mathbf{w},b,\xi,\boldsymbol{\alpha},\mathbf{r}) = \frac{1}{2}\langle \mathbf{w}.\mathbf{w} \rangle + C\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{m}\alpha_i\left[y_i\left(\langle \mathbf{w}.\mathbf{x}_i \rangle + b\right)-1+\xi_i\right]-\sum_{i=1}^{m}r_i\xi_i \tag{12}$$

Hence,

$$\frac{\partial L(\mathbf{w},b,\xi,\boldsymbol{\alpha},\mathbf{r})}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{m} y_i\alpha_i\mathbf{x}_i = 0 \tag{13}$$

Thus again

$$\mathbf{w} = \sum_{i=1}^{m} y_i\alpha_i\mathbf{x}_i \tag{14}$$

and

$$\frac{\partial L(\mathbf{w},b,\xi,\boldsymbol{\alpha},\mathbf{r})}{\partial \xi} = C - \alpha_i - r_i = 0 \tag{15}$$

Thus

$$\alpha_i + r_i = C \tag{16}$$

and

$$\frac{\partial L(\mathbf{w},b,\boldsymbol{\alpha})}{\partial b} = \sum_{i=1}^{m} y_i\alpha_i = 0 \tag{17}$$

So the dual form will be

$$L(\mathbf{w},b,\xi_i,\boldsymbol{\alpha},\mathbf{r}) = \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} y_iy_j\alpha_i\alpha_j\langle \mathbf{x}_i.\mathbf{x}_j \rangle \tag{18}$$

Again, by considering that $\sum_{i=1}^{m} y_i \alpha_i = 0$ and $\alpha_i \geq 0$.

This is similar to the maximal marginal classifier.

The only difference is the new constraints of $\alpha_i + r_i = C$, where $r_i \geq 0$, hence $0 \leq \alpha_i \leq C$.

This implies that the value *C,* sets an upper limit on the Lagrangian optimisation variables α_i.

The value of *C* offers a trade-off between accuracy of data fit and regularization. A small value of *C* (i.e. <1) significantly limits the influence of error points (or outliers), whereas if *C* is chosen to be very large (or infinite) then the soft margin approach becomes identical to the maximal margin classifier.

The choice of the value of *C* will depend heavily on the data. Appropriate selection of *C* is of great importance and is an area of research.
One way to set *C* is gradually increasing *C* from max (α_i) for $\forall i$, and find the value for which the error (outliers, cross validation, or number of misclassified points) is minimum. Finally, *C* can be found empirically.

Multi-Dimensional Feature Space

There will be no change in formulation of the SVM !

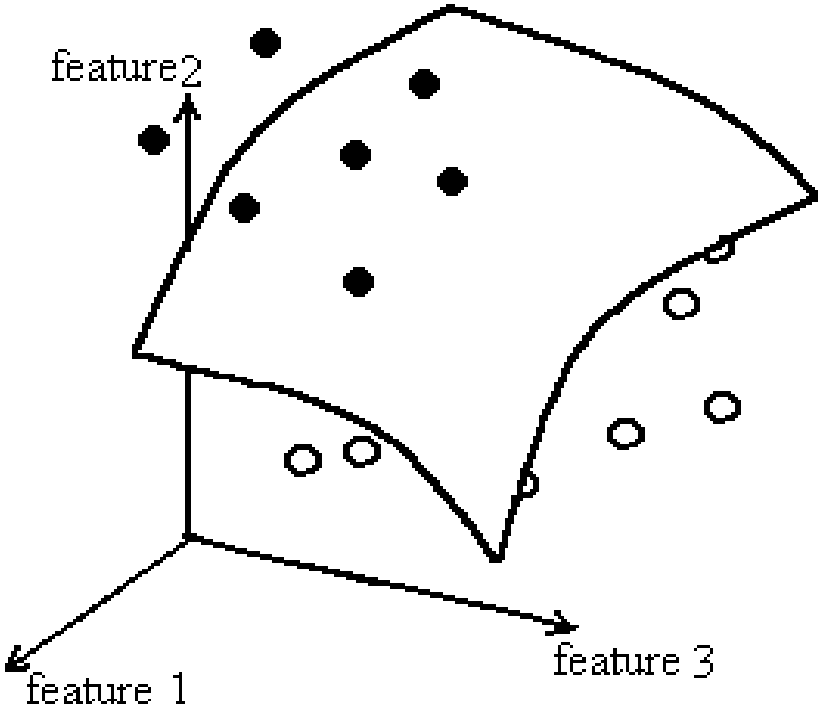


Figure 10. A three-dimensional hyperplane

Nonlinear Discriminant Functions

As can be seen in Figure 11, the datasets are separable if a nonlinear hypeplane is used.

Kernel mapping offers an alternative solution by non-linearly projecting the data into a (usually) higher dimensional feature space to allow the separation of such cases.

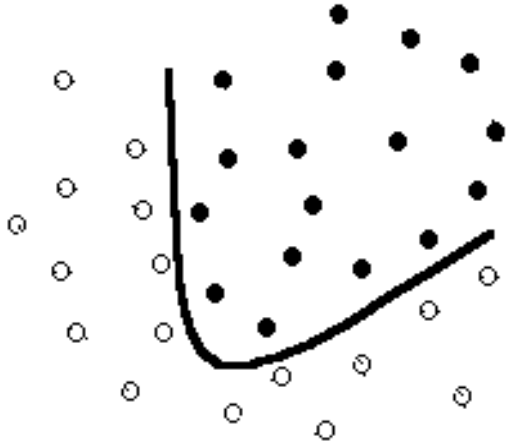


Figure 11. Nonlinear discriminant hyperplane

The key success of Kernel mapping is that special types of mapping that obey Mercer's theorem, sometimes called Reproducing Kernel Hilbert Spaces (RKHS), offers an implicit mapping into feature space.

$$K(\mathbf{x},\mathbf{z}) = \langle \varphi(\mathbf{x}).\varphi(\mathbf{z}) \rangle \qquad (19)$$

This means that the explicit mapping need not be known or calculated, rather the inner-product itself is sufficient to provide the mapping.

This simplifies the computational burden dramatically and in combination with SVM's inherent generality largely mitigates the so-called *curse of dimensionality*. Further this means that the input feature inner-product can simply be substituted with the appropriate Kernel function to obtain the mapping whilst having no effect on the Lagrangian optimisation theory.

Hence,

$$L(\mathbf{w},b,\xi_i,\boldsymbol{\alpha},\mathbf{r}) = \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{m} y_i y_j \alpha_i \alpha_j K\left(\mathbf{x}_i.\mathbf{x}_j\right) \qquad (20)$$

The relevant classifier function then becomes

$$f(\mathbf{x}) = \operatorname{sgn}\left(\sum_{i=1}^{nSVs} y_i \alpha_i K(\mathbf{x}_i,\mathbf{x}_j) + b\right) \qquad (21)$$

All the benefits of the original linear SVM method are maintained.

We can train a highly non-linear classification function such as a polynomial or a radial basis function or even a sigmoidal neural network using robust and efficient algorithm that do not suffer from local minima.

The use of Kernel functions transforms a simple linear classifier into a powerful and general non-linear classifier.

Some examples of popular reproducing Kernel Hilbert Space functions used in SVM:

Polynomial (degree d) $K(u,v) = (u.v + c)^d$

Gaussian Radial Basis Function $K(u,v) = \exp\left(-\frac{\|u-v\|^2}{2\sigma^2}\right)$

Exponential Radial Basis Function $K(u,v) = \exp\left(-\frac{\|u-v\|}{2\sigma^2}\right)$

Multi-Layer Perceptron $K(u,v) = \tanh\left(\rho(\langle u.v\rangle + c\right)$

Over fitting problem

Potentially it is possible to fit a hyperplane using an appropriate Kernel to the data in order to avoid overlapping the sets (or non-separable cases) and therefore produce a classifier with no error on the training set. In a two-dimensional case it may look like Figure 12.

This however, is unlikely to generalize well.

Not robust any more since a testing or new input can be misclassified easily.

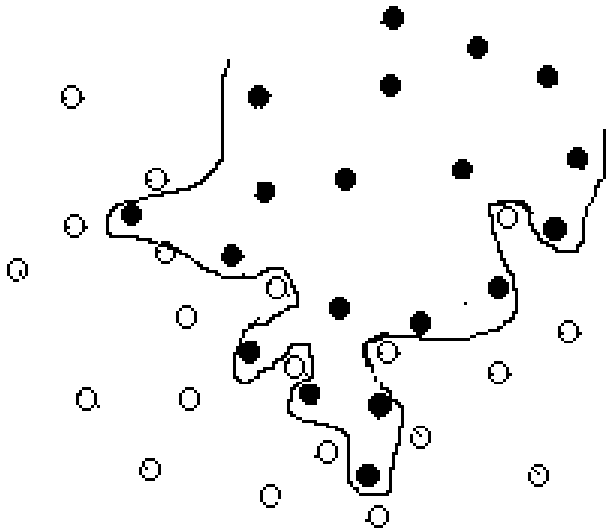


Figure 12. Overfitting problem

Conclussion

In summary, SV classifiers (SVCs) in the form of maximal margin classifier and in particular, the soft margin classifier with Kernel functions, bring many advantageous properties to the problem of pattern classification.

- SVM is a classifier based on conventional risk minimisation ($\min(y-f(x,\alpha))$ using convex optimisation.
- SVCs find the optimal separating hyperplane by maximising the margin between classes and thereby minimise complexity and risk of overfitting.
- They are simple to implement, are computationally efficient (trained fast) and in particular new algorithms based on sequential minimal optimisation (SMO) [6] scale to extremely large datasets.
- Kernel functions allow efficient mapping into high dimensional feature space implicitly, without burdensome explicit calculation and circumvent the so-called "Curse of Dimensionaity".
- Nonlinear Kernels facilitate the optimal classification of all types of datasets with little or no prior information on the underlying characteristics of the dataset.
- Multi-class SVCs [5] have been developed for *m*-ary classification applications.
- SVCs have proven to be highly effective in a wide variety of real-world applications often offering considerable improvement over competing methods.

References

[1] K. P. Bennet and C. Campbell, “Support Vector Machines: Hype or Hallelujah?”, SIGKDD Explorations, 2, 2, 2000, 1-13 {http://www.rpi.edu/~bennek}.

[2] N. Christianini and J. Shawe-Taylor, An Introduction to Support Vector Machines, Cambridge university Press, 2000

[3] V. Vapnik, Statistical Learning Theory, Wiley Press, 1998

[4] D. DeCoste and B. Scholkopf, “Training Invariant Support Vector Machines,” Machine Learning, Kluwer Press, 2001

[5] J. Weston and C. Wstkins, Support vector machines for multi-class pattern recognition,” Technical Report, CSD-TR-98-04, Royal Holloway, 1998

[6] J. Platt, “Sequential minimal optimisation: A fast algorithm for training support vector machines,” Technical Report, MSR-TR-98-14, Microsoft research, 1998.

[7] C. Burges, “ A tutorial on support vector machines for pattern recognition,” Data Mining and knowledge Discovery, 2, pp. 121-167, 1998

[8] S. Gunn, “Support vector machines for classification and regression,” Technical Reports, Dept. of Electronics and Computer Science, Southampton University, 1998.

[9] V. Vapnik, The nature of statistical learning theory, Springer, New York, 1995.

[10] http://www.support-vector.net

[11] http://www.kernel-machines.org

[12] O. Chapelle, and V. Vapnik, “Choosing Multiple Parameters for Support Vector Machines,” Machine Learning, 46, 131-159, 2002.

KKT conditions

Consider the following convex optimization problem [13],

$$\begin{aligned} &\text{minimise} && f_0(x) \\ &\text{subject to} && f_i(x) \le 0, \quad i = 1, \cdots, m \\ & && h_i(x) = 0, \quad i = 1, \cdots, p \end{aligned}$$

We can define Lagrangian associated with the above optimization problem as

$$L(x, \lambda, \nu) = f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x)$$

The Lagrange dual function is defined as

$$g(\lambda, \nu) = \inf_x L(x, \lambda, \nu) = \inf_x \left(f_0(x) + \sum_{i=1}^{m} \lambda_i f_i(x) + \sum_{i=1}^{p} \nu_i h_i(x) \right)$$

Assume $f_i(x)$ and $h_i(x)$ are differentiable. Let x^* and (λ_i^*, ν_i^*) be any primal and dual optimal points with zero duality gap. For a problem in which strong duality obtains, the following KKT conditions will be satisfied.

$$\nabla f_0(x^*) + \sum_{i=1}^{m} \lambda_i^* \nabla f_i(x^*) + \sum_{i=1}^{p} \nu_i^* \nabla h_i(x^*) = 0$$

$$f_i(x^*) \le 0$$

$$h_i(x^*) = 0$$

$$\lambda_i^* \ge 0$$

$$\lambda_i^* f_i(x^*) = 0$$

The last condition is known as complementary slackness.