Pattern Recognition and Machine Intelligence

James L. Crowley

ENSIMAG 3 MMIS First Semester 2010/2011

Lesson 11 18 January 2011

Support Vector Machines

Contents

Notation 2

Support Vector Machines 3

Hard-Margin SVMs - Separable Training Data 4

Soft Margin SVM's - Non-separable training data. 8

Source:
"Pattern Recognition and Machine Learning", C. M. Bishop, Springer Verlag, 2006.

Notation

x	a variable
X	a random variable (unpredictable value)
$\vec{x}$	A vector of D variables.
$\vec{X}$	A vector of D random variables.
D	The number of dimensions for the vector $\vec{x}$ or $\vec{X}$
$\{\vec{X}_m\}$	A set of M Training samples.
$\{t_m\}$	A set of class labels (indicators) for the samples. For a 2 Class problem, t_m is -1 or +1
M	Total number of training samples.
$\{a_m\}$	Lagrange Multipliers
$\{S_m\}$	a set of M slack variables for $\{\vec{X}_m\}$
C	trade-off between slack variables penalty and margin.
$S = \{\vec{X}_n\}$	a set of N support vectors (supporting sample points).
$\mathcal{T} \subseteq S$	a subset of S where $0 < a_m < C$

$k(\vec{X}_1, \vec{X}_2) = \phi(\vec{X}_1)^T \phi(\vec{X}_2)$ Kernel Function is Gram of non-linear function.

$k(\vec{X}, \vec{Z}) = (1 + \vec{X}^T \vec{Z})^2$ Popular Quadratic Kernel

$\phi(\vec{X}) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, \sqrt{2}x_1x_2, x_2^2)$ Nonlinear function for Quadratic Kernel for D=2 (Note N=6, for D=2)

Linear Hyper-plane : $y(\vec{X}) = \vec{w}^T \phi(\vec{X}) + b$ in nonlinear feature space.

Support Vector Machines

One significant limitations for Linear Leaning Machines seen last week is that the kernel function must be evaluated for every training point during learning.

An alternative is to use a Learning Algorithm with sparse support - that is a uses only a small number of points to learn the separation boundary.

Support Vector Machines (SVM) are such an algorithm.

SVM's are popular for problems of classifiaction, regression and novelty detection.
The solution of the model parameters corresponds to a convex optimisation problem.
Any local solution is a global solution.

We will use the two class problem, K=2, to illustrate the principle. Multiclass solutions are possible.

Our linear model is for the decision surface is

$$y(\vec{X}) = \vec{w}^T \phi(\vec{X}) + b$$

Where $\phi(\vec{X})$ is a feature space transformation that maps a hyper-plane in F dimensions into a non-linear decision surfaces in D dimensions. F >> D

Training data is a set of M training samples $\{\vec{X}_m\}$ and their indicator variable, $\{t_m\}$. For a 2 Class problem, t_m is -1 or +1.

A new, observed point (not in the training data) will be classified using the function $sign(y(\vec{X}))$, so that a classification of a training sample is correct if

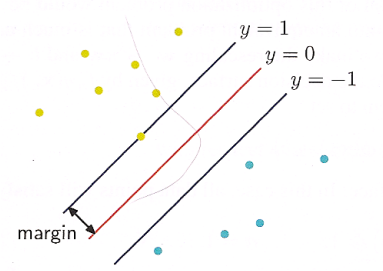
$$t_m y(\vec{X}_m) > 0$$

Hard-Margin SVMs - Separable Training Data

Let us assume, for the moment, that the data is linearly separable.
There exists a hyper-plane $y(\vec{X}) = \vec{w}^T \phi(\vec{X}) + b$ such that $t_m y(\vec{X}_m) > 0$ for all m.

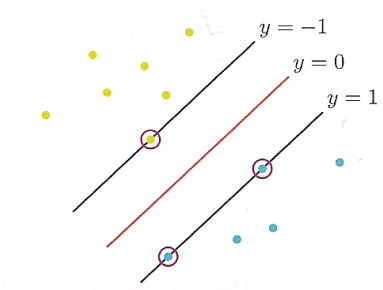
Generally there will exist many solutions for separable data.
For a support Vector Machine, the decision boundary is chosen to maximize the margin, γ.

Recall that the margin, γ is the minimum distance of any sample from the hyper-plane



Bishop p 327 (fig 7.1)

What we are going to do is design the decision boundary to that it has a equal distance from a small number of support points.



Bishop p 327 (fig 7.1)

The distance for a point from the hyper-plane is $\frac{|y(\vec{X})|}{\|\vec{w}\|}$

since we are only interested in points where $t_m y(\vec{X}_m) > 0$

The distance for the point $\vec{X}_m$ to the decision surface is:

$$\frac{t_m y(\vec{X}_m)}{\|\vec{w}\|} = \frac{t_m(\vec{w}^T \phi(\vec{X}_m) + b)}{\|\vec{w}\|}$$

We will seek to maximize the margin by solving

$$\arg\max_{w,b}\left\{\frac{1}{\|\vec{w}\|}\min_{m}\left\{t_m(\vec{w}^T\phi(\vec{X}_m)+b)\right\}\right\}$$

The factor $\frac{1}{\|\vec{w}\|}$ can be removed from the optimization because $\|\vec{w}\|$ does not depend on m.

Direct solution would be very difficult. We will convert this to an equivalent problem.

Note that rescaling the problem changes nothing. Thus we will scale the equation such for the sample that is closest to the decision surface (smallest margin):

$$t_m(\vec{w}^T\phi(\vec{X}_m)+b)=1 \quad \text{that is:} \quad t_m y(\vec{X}_m)=1$$

For all other sample points:

$$t_m(\vec{w}^T\phi(\vec{X}_m)+b)\geq 1$$

This is known as the Canonical Representation for the decision hyperplane.

The training sample where $t_m(\vec{w}^T\phi(\vec{X}_m)+b)=1$ are said to be the "active" constraint. All other training samples are "inactive".

By definition there is always at least one active constraint.

When the margin is maximized, there will be two active constraints.

Thus the optimization problem is to maximize $\arg\min_{w,b}\left\{\frac{1}{2}\|\vec{w}\|^2\right\}$ subject to the active constraints.

The factor of ½ is a convenience for later analysis.

To solve this problem, we will use Lagrange Multipliers, $a_m \geq 0$, with one multiplier for each constraint. This gives a Lagrangian function:

$$L(\vec{w},b,\vec{a})=\frac{1}{2}\|\vec{w}\|^2-\sum_{m=1}^{M}a_m\left\{t_m(\vec{w}^T\phi(\vec{X}_m)+b)-1\right\}$$

Setting the derivatives to zero, we obtain:

$$\frac{\partial L}{\partial w} = 0 \Rightarrow \qquad \vec{w} = \sum_{m=1}^{M} a_m t_m \phi(\vec{X}_m)$$

$$\frac{\partial L}{\partial b} = 0 \Rightarrow \qquad \sum_{m=1}^{M} a_m t_m = 0$$

Eliminating $\vec{w}, b$ from $L(\vec{w}, b, \vec{a})$ we obtain :

$$L(\boldsymbol{a}) = \sum_{m=1}^{M} a_m - \frac{1}{2} \sum_{m=1}^{M} \sum_{n=1}^{M} a_m a_n t_m t_n k(\vec{X}_m, \vec{X}_n)$$

with constratins:

$a_m \geq 0$ for m=1, ..., M

$$\sum_{m-1}^{M} a_m t_m = 0$$

where the kernel function is : $k(\vec{X}_1, \vec{X}_2) = \vec{\phi}(\vec{X}_1)^T \vec{\phi}(\vec{X}_2)$

The solution takes the form of a quadratic programming problem in D variables (the Kernel space). This would normally take $O(D^3)$ computations.

In going to the dual formulation, we have converted this to a dual problem over M data points, requiring $O(M^3)$ computations.
This can appear to be a problem, but the solution only depends on a small number of points!

To classify a new observed point, we evaluate:

$$y(\vec{X}) = \sum_{m=1}^{M} a_m t_m k(\vec{X}, \vec{X}_m) + b$$

The solution to optimization problems of this form satisfy the "Karush-Kuhn-Tucker" condition, requiring:

$$a_m \geq 0$$
$$t_m y(\vec{X}_m) - 1 \geq 0$$
$$a_m \left\{ t_m y(\vec{X}_m) - 1 \right\} \geq 0$$

For every data point in the training samples, $\left\{ \vec{X}_m \right\}$, either

$$a_m = 0 \qquad \text{or} \qquad t_m y(\vec{X}_m) = 1$$

Any point for which $a_m = 0$ does not contribute to $y(\vec{X}) = \sum_{m=1}^{M} a_m t_m k(\vec{X}, \vec{X}_m) + b$ and thus is not used! (is not active) .

The remaining points, for which $a_m \neq 0$ are called the "Support vectors".
These points lie on the margin at $t_m y(\vec{X}_m) = 1$ of the maximum margin hyperplane. Once the model is trained, all other points can be discarded!

Let us define the support vectors as the set *S*.

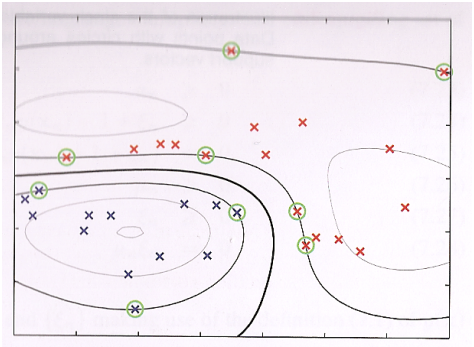
Now that we have solved for *S* and **a**, we can solve for b:

we note that : $$t_m \left(\sum_{n \in S} a_n t_n k(\vec{X}_m, \vec{X}_n) + b \right) = 1$$

averaging over all support vectors in S gives:

$$b = \frac{1}{N_S} \sum_{m \in S} \left(t_m - \sum_{n \in S} a_n t_n k(\vec{X}_m, \vec{X}_n) \right)$$

This can be expressed as minimization of an error function, $E_\infty(z)$ such that the error function is zero if $z \geq 0$ and ∞ otherwise.



From Bishop p 331.

Soft Margin SVM's - Non-separable training data.

So far we have assumed that the data are linearly separable in $\phi(\vec{X})$.
For many problems some training data may overlap.

The problem is that the error function goes to ∞ for any point on the wrong side of the decision surface. This is called a "hard margin" SVM.

We will relax this by adding a "slack" variable, S_m for each training sample:

$$S_m \geq 1$$

We will define

$S_m = 0$ for samples on the correct side of the margin, and

$S_m = \left| t_m - y(\vec{X}_m) \right|$ for other samples.

For a sample inside the margin, but on the correct side of the decision surface:

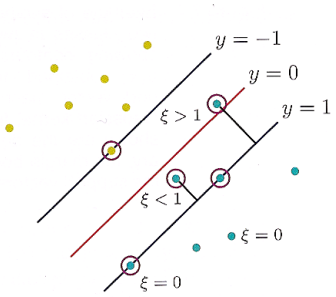
$$0 < S_m \leq 1$$

For a sample on the decision surface:

$$S_m = 1$$

For a sample on the wrong side of the decision surface:

$$S_m > 1$$



Soft margin SVM: Bishop p 332 (note use of ξ_m in place S_m)

This is sometimes called a soft margin. To softly penalize points on the wrong side, we minimize :

$$C\sum_{m=1}^{M} S_m + \frac{1}{2}\|\vec{w}\|^2$$

where C > 0 controls the tradeoff between slack variables and the margin.

because any misclassified point $S_m > 1$, the upper bound on the number of misclassified points is is $\sum_{m=1}^{M} S_m$.

C is an inverse factor. (note that C=∞) is the earlier SVM with hard margins.

To solve for the SVM we write the Lagrangian:

$$L(\vec{w},b,\vec{a}) = \frac{1}{2}\|\vec{w}\|^2 + C\sum_{m=1}^{M} S_m - \sum_{m=1}^{M} a_m\left\{t_m y(\vec{X}_m) - 1 + S_m\right\} - \sum_{m=1}^{M} \mu_m S_m$$

The KKT conditions are

$$\begin{aligned}
&a_m \geq 0\\
&t_m y(\vec{X}_m) - 1 + S_m \geq 0\\
&a_m\left\{t_m y(\vec{X}_m) - 1 + S_m\right\} \geq 0\\
&\mu_m \geq 0\\
&S_m \geq 1\\
&\mu_m S_m = 0
\end{aligned}$$

Solving the derivatives of $L(\vec{w},b,\vec{a})$ for zero gives

$$\frac{\partial L}{\partial w} = 0 \Rightarrow \qquad \vec{w} = \sum_{m=1}^{M} a_m t_m \phi(\vec{X}_m)$$

$$\frac{\partial L}{\partial b} = 0 \Rightarrow \qquad \sum_{m=1}^{M} a_m t_m = 0$$

$$\frac{\partial L}{\partial S} = 0 \Rightarrow \qquad a_m = C - \mu_m$$

using these to eliminate w, b and $\{S_m\}$ from *L(w, b, a)* we obtain

$$L(\boldsymbol{a}) = \sum_{m=1}^{M} a_m - \frac{1}{2}\sum_{m=1}^{M}\sum_{n=1}^{M} a_m a_n t_m t_n k(\vec{X}_m, \vec{X}_n)$$

This appears to be the same as before, except that the constraints are different.

$0 \le a_m \le C$

$$\sum_{m=1}^{M} a_m t_m = 0$$

(referred to as a "box" constraint). Solution is a quadratic programming problem, with complexity $O(M^3)$. However, as before, a large subset of training samples have $a_m = 0$, and thus do not contribute to the optimization.

For the remaining points $t_m y(\vec{X}_m) = 1 - S_m$

For samples ON the margin $a_m < C$ hence $\mu_m > 0$ requiring that $S_m = 0$

For samples INSIDE the margin: $a_m = C$ and $S_m \le 1$ if correctly classified and $S_m > 1$ if misclassified.

as before to solve for b we note that :

$$t_m\left(\sum_{n\in S} a_n t_n k(\vec{X}_m, \vec{X}_n) + b\right) = 1$$

averaging over all support vectors in S gives:

$$b = \frac{1}{N_S}\sum_{m\in T}\left(t_m - \sum_{n\in S} a_n t_n k(\vec{X}_m, \vec{X}_n)\right)$$

where $\mathcal{T}$ denotes the set of support vectors such that $0 < a_m < C$.