

Lecture 18:

Support Vector Machines

Many thanks to Carlos Thomaz who authored the original version of these slides

Introduction

Vapnik's philosophy, 1990's

“If you possess a restricted amount of information for solving some problem, try to **solve the problem directly** and never solve a more general one as an intermediate step. It is possible that the **available information is sufficient for a direct solution** but is insufficient for solving a more general intermediate problem.”

Introduction

Vapnik's philosophy, 1990's

“In contrast to classical methods of statistics where in order to control performance one decreases the dimensionality of a feature space, the Support Vector Machine (SVM) **dramatically increases dimensionality** and relies on the so-called large margin factor.”

Introduction

What is SVM?

SVM is primarily a two-class* classifier that **maximises the width of the margin between classes**, that is, the empty area around the decision boundary defined by the distance to the nearest training patterns.

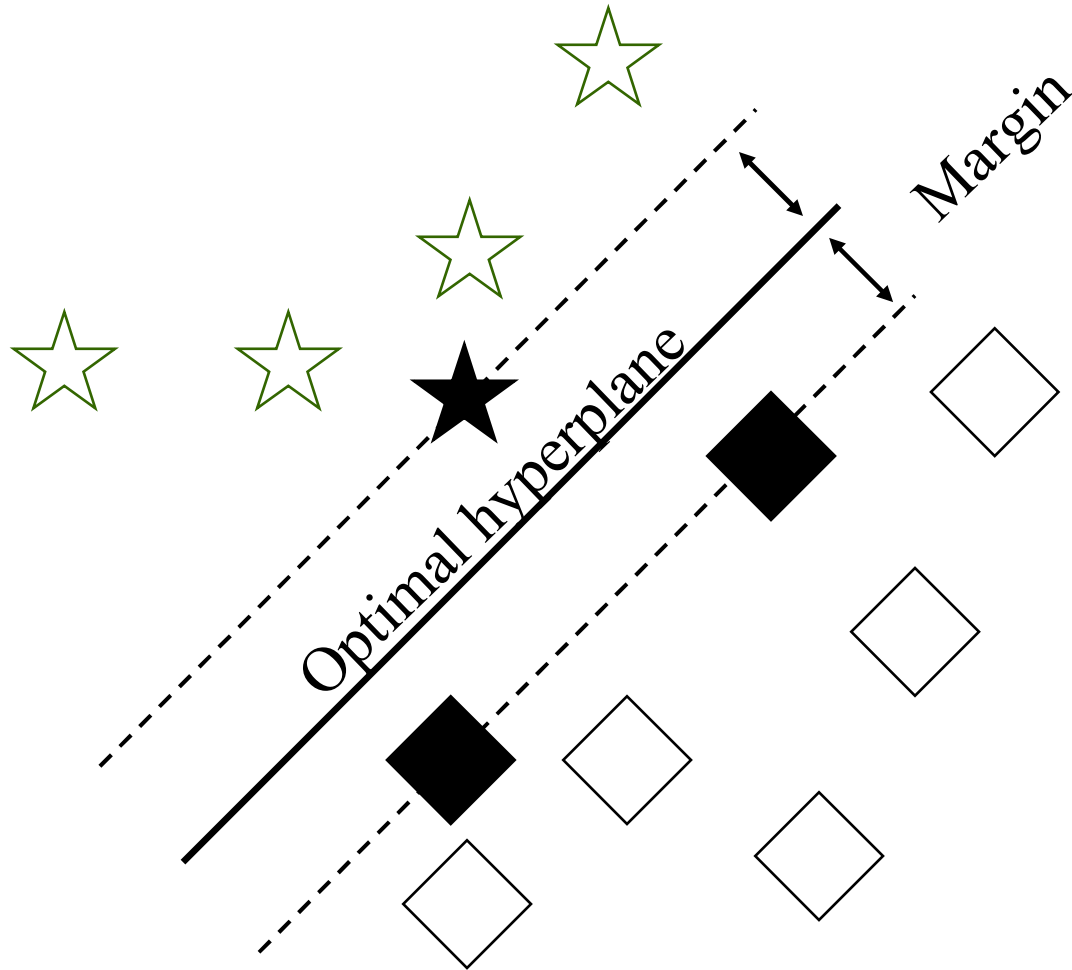
* It is feasible to extend the system to more than 2 classes

For example

If $\{x_i\}$ is a set of points in n -dimensional space with corresponding classes $\{y_i: y_i \in \{-1, 1\}\}$ then the training algorithm attempts to place an hyperplane between points where $y_i = 1$ and points where $y_i = -1$.

Once this has been achieved a new pattern x can then be classified by testing which side of the hyper-plane the point lies on.

Geometric Idea



Linear SVM

Let's start with the simplest case: linear machines trained on separable data. Suppose the training data

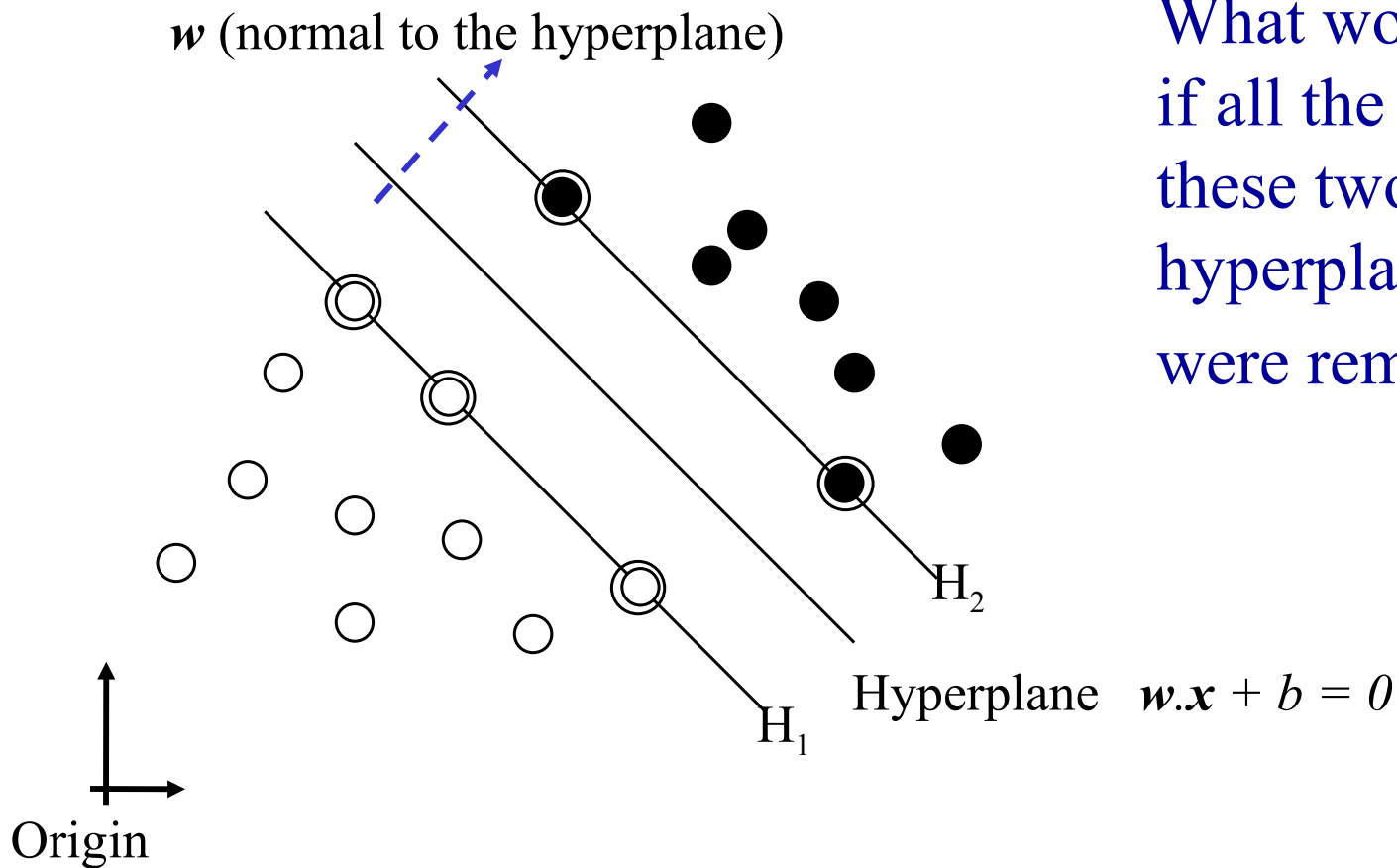
$$(x_1, y_1), \dots, (x_N, y_N), x \in \mathbb{R}^n, y \in \{-1, +1\}$$

can be separated by a hyperplane

$$(w \cdot x) + b = 0.$$

We say that this set is separated by the optimal hyperplane if it is separated without error and the distance between the closest point to the hyperplane is maximal.

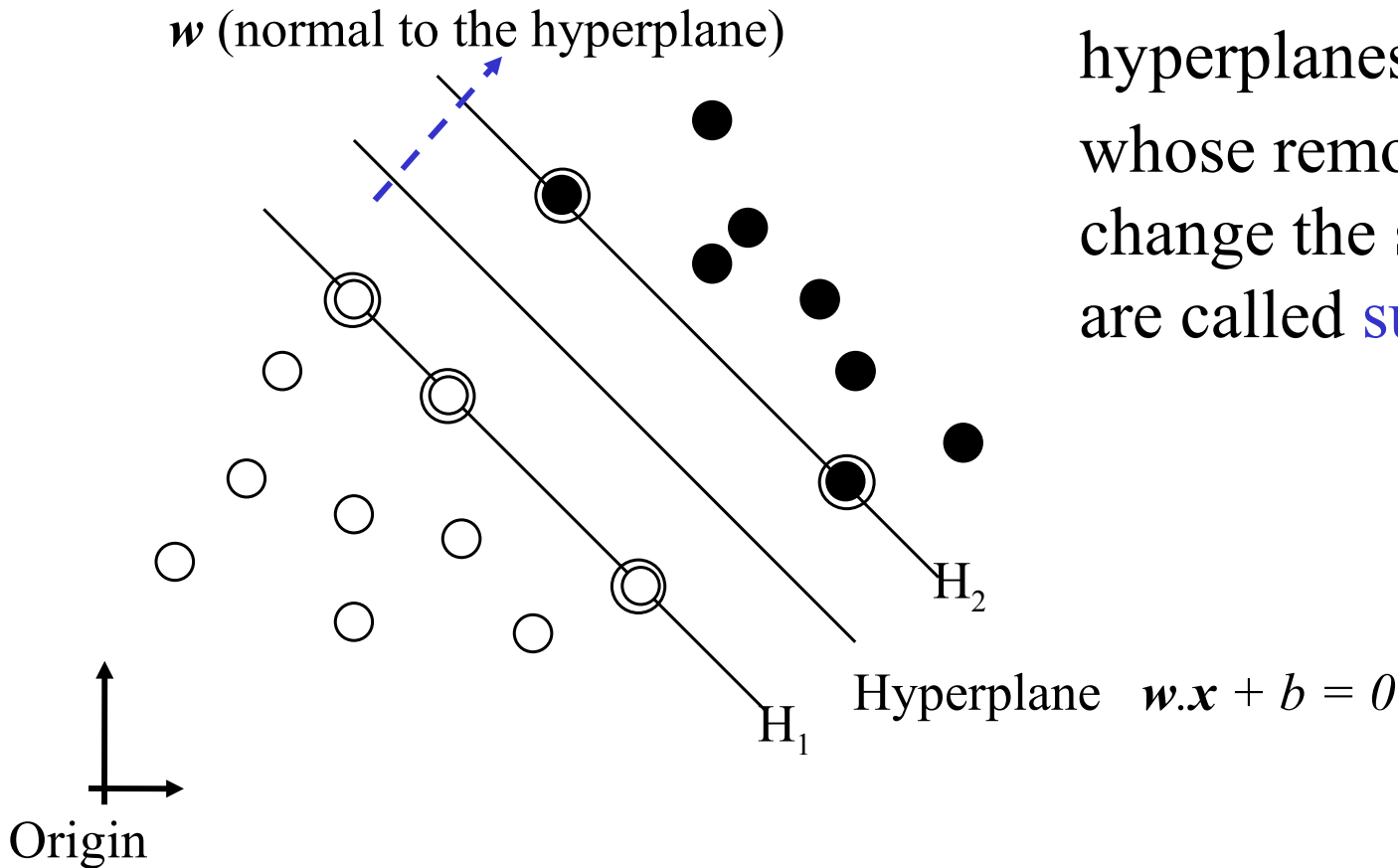
The Separating Hyperplane



What would happen if all the points not on these two boundary hyperplanes (H_1, H_2) were removed?

The Separating Hyperplane

Those training points which lie on one of the hyperplanes (H_1, H_2) and whose removal would change the solution found are called **support vectors**.



The Separating Hyperplane (cont.)

To describe the separating hyperplane let us suppose that all the training data satisfy the following constraints:

$$(w \cdot x_i) + b \geq +1 \quad \text{for } y_i = +1.$$

$$(w \cdot x_i) + b \leq -1 \quad \text{for } y_i = -1.$$

which can be combined into one set of inequalities:

$$y_i (w \cdot x_i + b) - 1 \geq 0 \quad \text{for } i = 1, \dots, N.$$

The Separating Hyperplane (cont.)

Now consider the points for which the equality in the previous equations holds. These points lie either on

$$H_1: (w \cdot x_i) + b = 1 \quad \text{or} \quad H_2: (w \cdot x_i) + b = -1$$

with normal w and perpendicular distance from the origin respectively $|1 - b|/|w|$ and $|-1 - b|/|w|$.

Therefore the shortest distance from the separating hyperplane to the closest positive (negative) example is $1/|w|$ and the margin is simply $2/|w|$.

Finding the separating hyperplane

The distance separating the classes is $2/|w|$, the optimal hyperplane therefore is the one for which $|w|$ is minimal.

Finding the hyperplane is formulated as the following optimisation problem:

Minimise $|w|^2 = w \cdot w$ subject to constraints

$y_i(w \cdot x_i + b) - 1 \geq 0$ for all points (x_i, y_i)

The solution uses Lagrange Multipliers

Observations about the Lagrange Formulation

There are two reasons for switching to a Lagrange formulation of the problem.

First, the **constraints** will be replaced by constraints on the **lagrangian multipliers** α_i which will be much easier to handle.

Secondly, in this formulation the training data will only appear in the form of **dot products** between vectors. This is a crucial property which will allow us to generalise the procedure to the non-linear case.

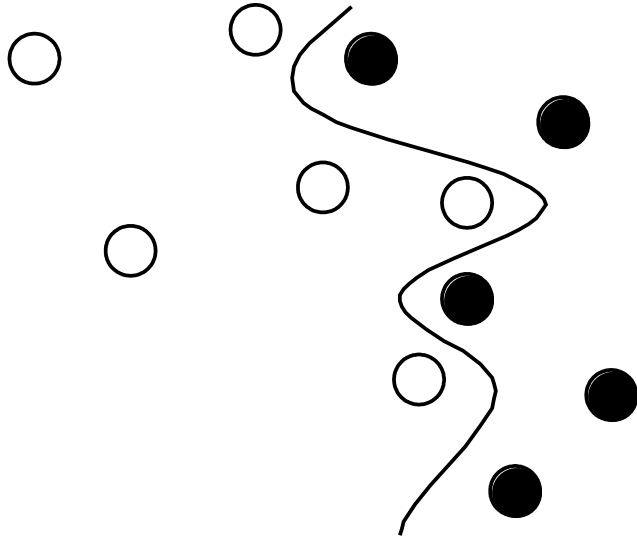
Nonlinear SVM

The nonlinear SVM implements the following idea.

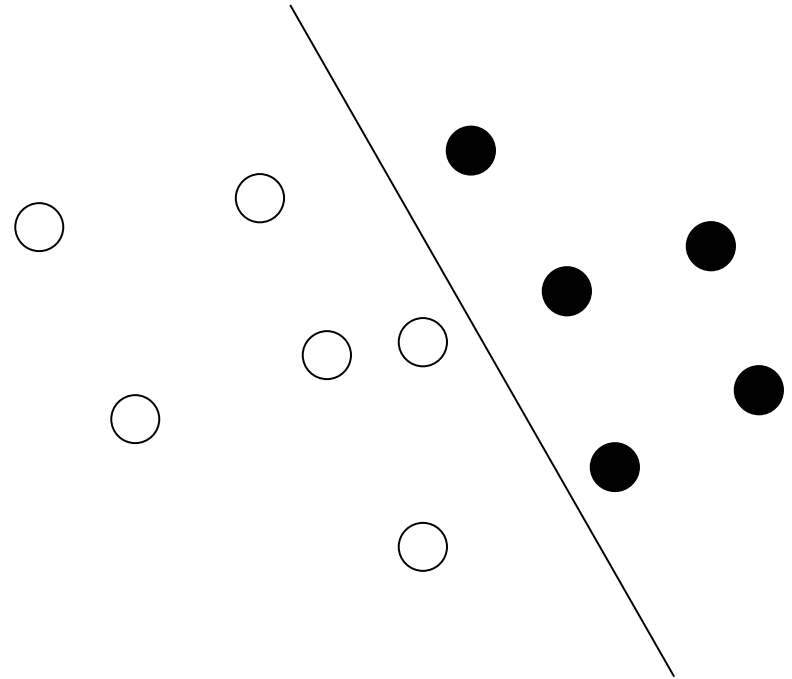
It maps the training points x_i into a high-dimensional feature space through some non-linear mapping chosen a priori.

In this higher dimensional space, an optimal separating hyperplane can be found.

Geometric Idea



Original Space



Feature Space
(higher dimension)

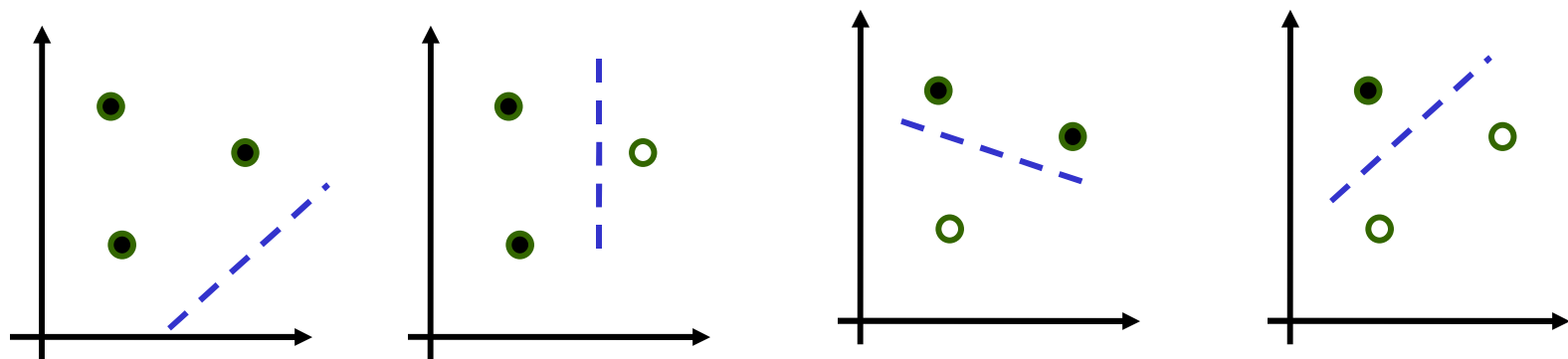
VC (Vapnik and Chervonenkis) Dimension

To show that the idea works we define the VC dimension:

The VC dimension of a class of functions $\{f_i\}$ is the maximum number of points that can be separated (shattered) into two classes in all possible ways.

VC Dimension (an example)

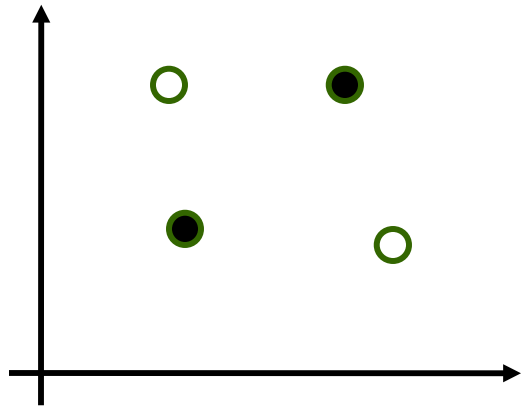
Suppose the class $\{f\}$ is straight lines on a 2D plane,
Any three (non collinear) points can be separated



Only these possibilities exist

VC Dimension Example

However for four points there is a case that cannot be separated by a single line:



Hence the VC dimension of a line is 3

VC Dimension (cont.)

It can be proved that:

The VC dimension of the set of oriented hyperplanes in R^n is $n+1$.

So, four points may be shattered (divided into two classes in all possible ways) in a three dimensional space.

In general a set of n data points can be shattered in an $n-1$ dimensional space.

VC Dimension (cont.)

To increase the dimension of the data points we can define a mapping from an n -dimensional space to the m -dimensional space where $m > n$:

$$\phi : \mathbb{R}^n \rightarrow \mathbb{R}^m$$

Then we can use the same training algorithm as the linearly separable case by replacing

$$x_i \cdot x_j \text{ with } \phi(x_i) \cdot \phi(x_j)$$

in all the equations. As a result the separating surface's normal vector \mathbf{w} will be in \mathbb{R}^m .

Kernels

However, mapping a point into a higher dimensional space might be too calculation- and storage-intensive to be practical.

“Kernel functions used in support vector machines are functions calculated in the original space that are equivalent to the dot product of the mapped vectors in the higher dimensional space”.

Kernels (cont.)

Therefore if there were a Kernel function K such that

$$K(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$$

we only need to use K in the training algorithm and would never need to explicitly do the mapping.

Thus if we replace $x_i \cdot x_j$ with $K(x_i, x_j)$ everywhere in the training algorithm, the algorithm will produce a SVM which lives in a high dimensional space.

All the linear considerations hold since we are still doing a linear separation but in a different space.

Kernels (cont.)

Having solved for \mathbf{w} using the kernel function, in place of the dot product, we can make our decision on an unknown point \mathbf{x}_u by finding the sign of:

$$K(\mathbf{w}, \mathbf{x}_u) + b$$

replacing the term

$$\mathbf{w} \cdot \mathbf{x}_u + b$$

That we used for the linear SVM.

So at no point do we explicitly need to use the lifting function ϕ

Kernels (cont.)

Suppose that our vectors are in \mathbb{R}^2 and we choose

$$K(u, v) = (u \cdot v)^2 = u_1^2 v_1^2 + 2u_1 v_1 u_2 v_2 + u_2^2 v_2^2$$

It's easy to find a mapping ϕ from \mathbb{R}^2 to \mathbb{R}^3 such that

$$(u \cdot v)^2 = \phi(u) \cdot \phi(v)$$

We can choose

$$\phi(x) = \begin{pmatrix} x_1^2 \\ \sqrt{2}x_1x_2 \\ x_2^2 \end{pmatrix}$$

Kernels (cont.)

Note that neither the mapping ϕ nor the higher dimensional space are unique. We could equally well have chosen the space to again be \mathbb{R}^3 and

$$\phi(x) = \frac{1}{\sqrt{2}} \begin{pmatrix} (x_1^2 - x_2^2) \\ 2x_1x_2 \\ (x_1^2 + x_2^2) \end{pmatrix}$$

or to be \mathbb{R}^4 and

$$\phi(x) = \begin{pmatrix} x_1^2 \\ x_1x_2 \\ x_1x_2 \\ x_2^2 \end{pmatrix}$$

Kernels (cont.)

Since we do not actually need to find the lifting function ϕ , but only use the kernel in the computation an interesting question is what functions can we use as kernels?

The answer is given by Mercer's condition.

Mercer's condition

If K is a function on vectors u and v then there exists a mapping ϕ into some vector space with an expansion:

$$K(u, v) = \phi(u) \cdot \phi(v)$$

if and only if for any $g(u)$ such that

$$\int g(u)^2 du$$

is finite then

$$\int K(u, v) g(u) g(v) dudv \geq 0$$

Some Examples of Non-linear Kernels

- Polynomial: $K(u, v) = (u \cdot v + 1)^p$

- Gaussian Radial Basis Function:

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

- Sigmoidal (hyperbolic separating surface):

$$K(u, v) = \tanh(\kappa u \cdot v - \delta)$$

The radial basis kernel

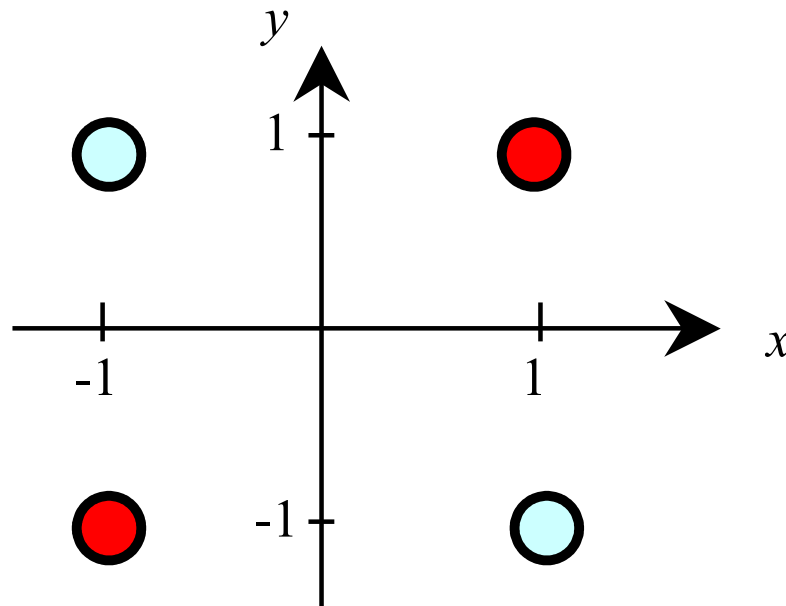
The radial basis kernel is interesting because it implies that the lifting function ϕ will raise the dimension of the data space infinitely.

This can be seen by that fact that there is an infinite series expansion of e^x which is required to gather the dot product terms.

Thus the radial basis function can solve any classification problem regardless of size.

SVM for the XOR Problem

The exclusive OR is the simplest problem that cannot be solved using a single 2D linear discriminant operating on the features.



SVM for the XOR Problem

We choose the quadratic polynomial kernel:

$$K(u, v) = (u \cdot v + 1)^2$$

For our two dimensional problem we can write

$$\begin{aligned} K(u, v) &= (u_1 v_1 + u_2 v_2 + 1)^2 \\ &= (u_1 v_1)^2 + (u_2 v_2)^2 + 2u_1 v_1 u_2 v_2 + 2u_1 v_1 + 2u_2 v_2 + 1 \end{aligned}$$

A possible lifting function is

$$\phi(x_1, x_2) = (x_1^2, x_2^2, \sqrt{2}x_1 x_2, \sqrt{2}x_1, \sqrt{2}x_2, 1)$$

It is easily seen that

$$K(u, v) = \phi(u) \cdot \phi(v)$$

SVM for the XOR Problem

Using

$$\phi(x_1, x_2) = (x_1^2, x_2^2, \sqrt{2}x_1x_2, \sqrt{2}x_1, \sqrt{2}x_2, 1)$$

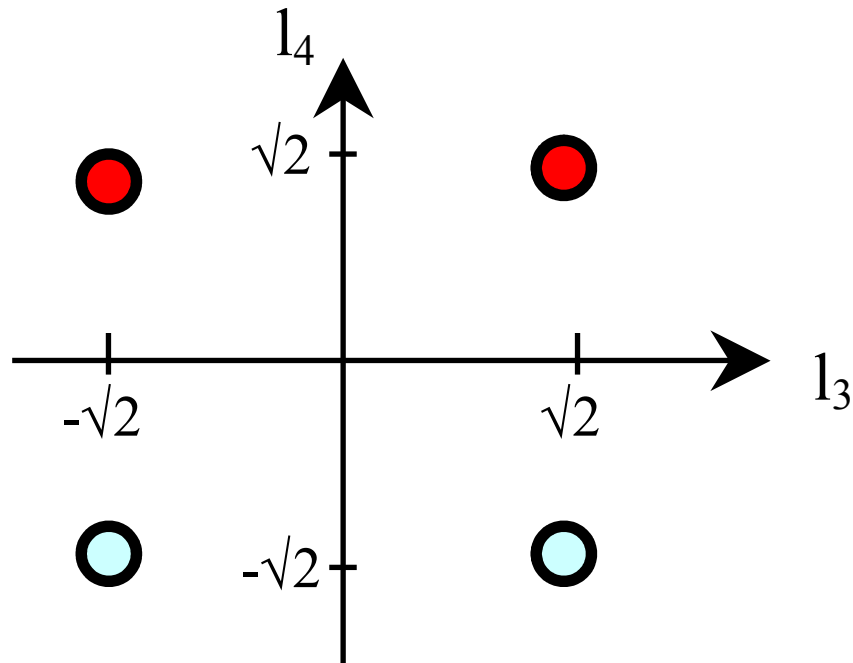
The classes are completely separable with

$w = \{0, 0, 1, 0, 0, 0\}$ in the high dimension space.

	Point		Lifted point coordinate					
	x_1	x_2	$l_1=x_1^2$	$l_2=x_2^2$	$l_3=\sqrt{2}x_1x_2$	$l_4=\sqrt{2}x_1$	$l_5=\sqrt{2}x_2$	$l_6=1$
Class 1	1	1	1	1	$\sqrt{2}$	$\sqrt{2}$	$\sqrt{2}$	1
	-1	-1	1	1	$\sqrt{2}$	$-\sqrt{2}$	$-\sqrt{2}$	1
Class 2	1	-1	1	1	$-\sqrt{2}$	$\sqrt{2}$	$-\sqrt{2}$	1
	-1	1	1	1	$-\sqrt{2}$	$-\sqrt{2}$	$\sqrt{2}$	1

SVM for the XOR Problem

Plotting the l_3 and l_4 axes we get:



Examples of the behaviour of kernels

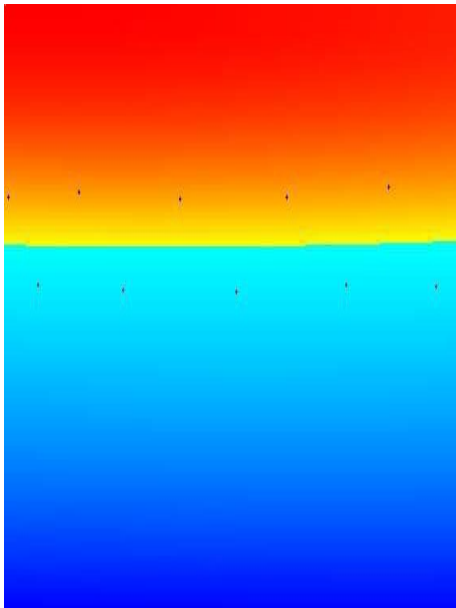
In the following examples the behaviour of different kernels is shown. The picture is coloured according to the value of the term $(K(\mathbf{w}, \mathbf{x}) + b)$.

Cyan to dark blue: 0, increasing negatively

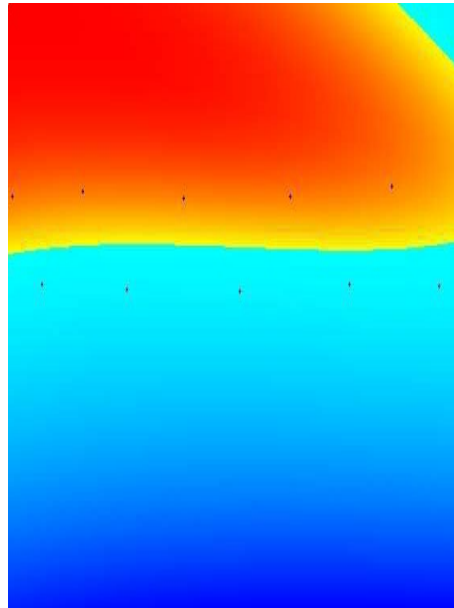
Yellow to dark red: 0, increasing positively

Polynomial Kernels

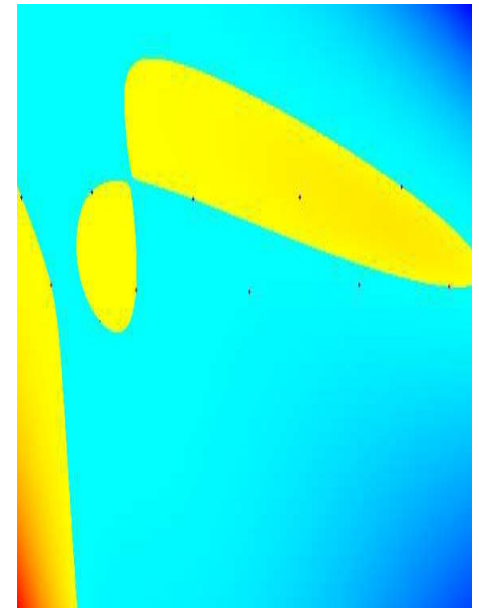
$p=2$



$p=3$



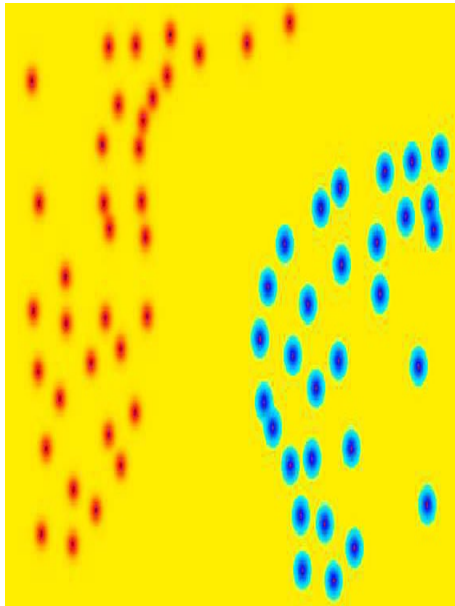
$p=5$



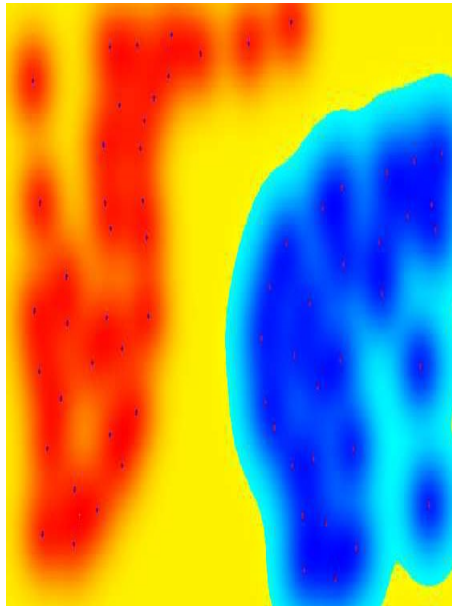
Images from Tom Weedon's report

Gaussian Kernels

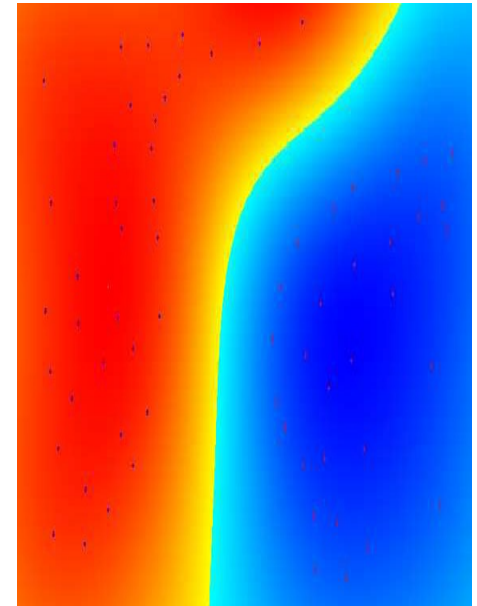
$\sigma=5$



$\sigma=20$



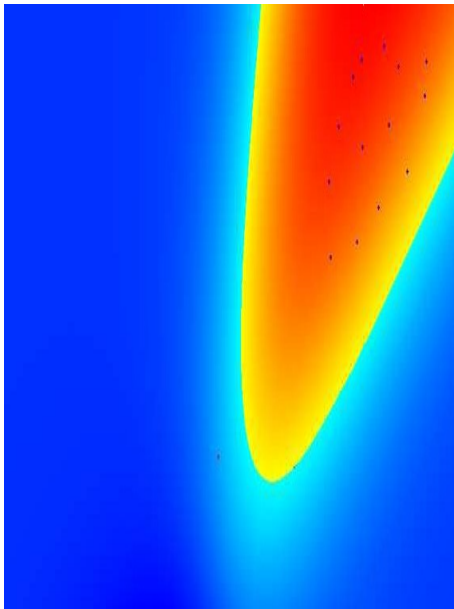
$\sigma=100$



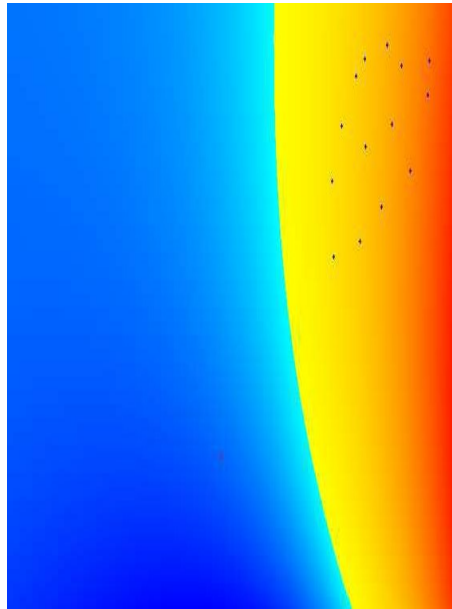
Images from Tom Weedon's report

Sigmoidal Kernels

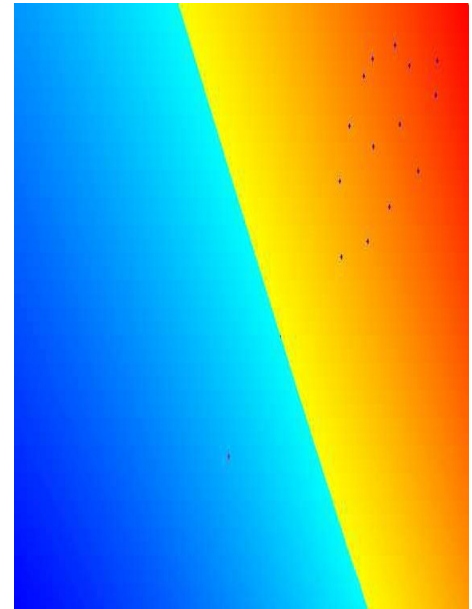
$[\kappa, \gamma] = [0.00003, -5]$



$[0.00001, -5]$



$[0.000001, -5]$



Images from Tom Weedon's report

Final Comment

An important benefit of the SVM approach is that the complexity of the resulting classifier is characterised by the number of support vectors rather than the dimensionality of the transformed space.

As a result, SVMs tend to be less prone to problems of overfitting than some other methods.

Have a great Christmas break!