Introduction to Support Vector Machines (SVM)

- History of SVM
- Large margin linear classifier
- Nonlinear classifiers: kernel trick
- A simple example
- Discussion on SVM

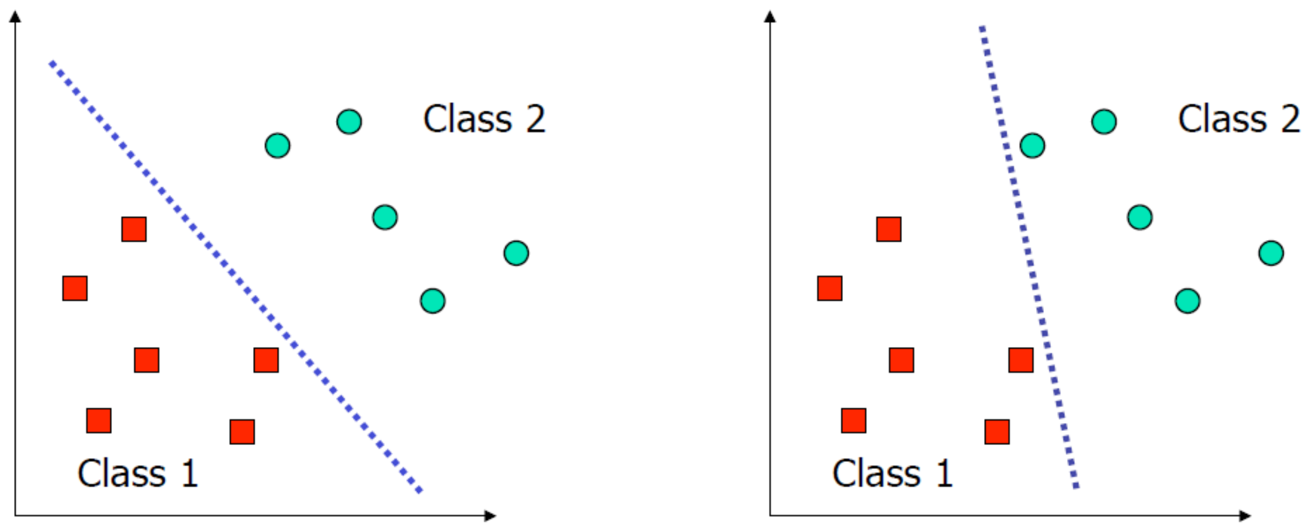
History of SVM

- SVM is related to statistical learning theory.
- SVM was first introduced in 1992.
- SVM becomes popular because of its success in handwritten digit recognition.
- SVM is now regarded as an important example of kernel methods. It is one of the key areas in machine learning.

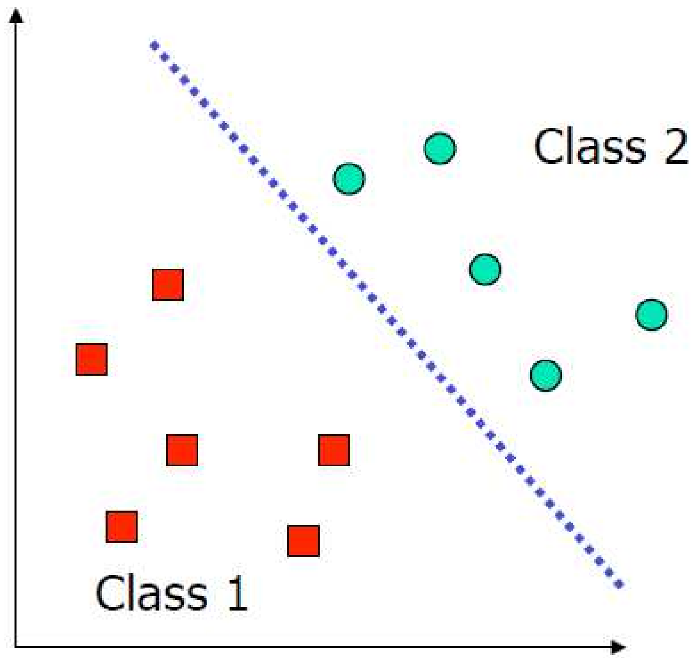
What is a Good Decision Boundary?

- Consider a two-class, linearly separable classification problem.
- Many decision boundaries.
  - The Perceptron algorithm can be used to find such a boundary.
  - Different algorithms have been proposed.
- Are all decision boundaries equally good?

Examples of Bad Decision Boundaries

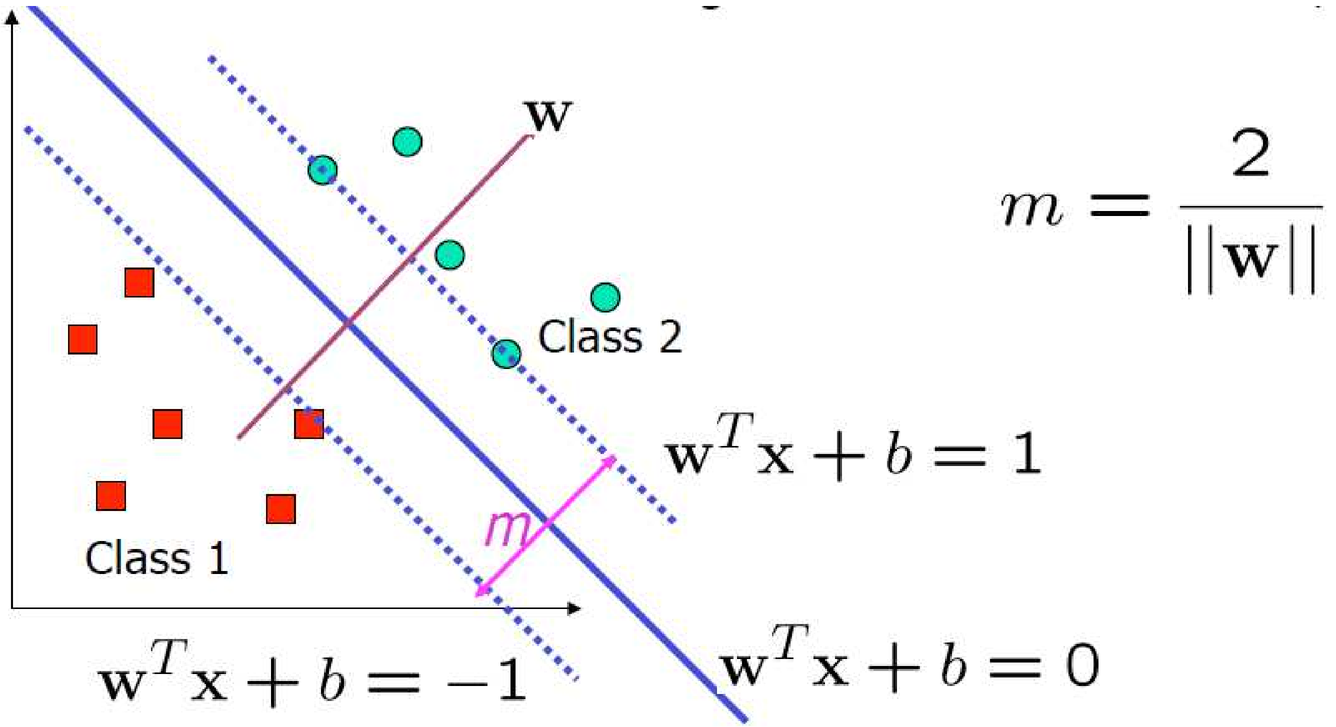


An Example of Good Decision Boundaries



Large margin Decision Boundary

- The decision boundary should be as far away from the data of both classes as possible. The margin m should be maximized.



Finding the Decision Boundary

- Let $\{x_1, ..., x_n\}$ be data set, and $y_i \in \{1, -1\}$ be the class label of x_i.
- The decision boundary should classify all points correctly: $y_i(w^T x_i + b) \geq 1$, for every i.
- The decision boundary can be found by solving the following constrained optimization problem:

$$Min. \frac{1}{2}||w||^2$$

subject to

$$y_i(w^T x_i + b) \geq 1$$

for each i.

- This is a constrained optimization problem.

Lagrangian Function

- The Lagrangian is

$$L = \frac{1}{2}w^T w + \sum_{i=1}^{n} \alpha_i (1 - y_i(w^T x_i + b))$$

where $||w||^2 = w^T w$.

- Setting the gradient of L with w to zero

$$w + \sum_{i=1}^{n} \alpha_i (-y_i) x_i = 0$$

$$w = \sum_{i=1}^{n} \alpha_i y_i x_i$$

- Setting the gradient of L with b to zero: $\sum_{i=1}^{n} \alpha_i y_i = 0$

The Dual Problem

If we substitute $w = \sum_{i=1}^{n} \alpha_i y_i x_i$ to L, we have

$$L = -\frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j + \sum_{i=1}^{n} \alpha_i$$

- The new objective function is in terms of α_i only.
- It is known as the dual problem: if we know w, we know all α_i; if we know all α_i, we know w.
- The original problem is known as the primal problem.
- The objective function of the dual problem needs to be maximized.

- The dual problem is

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

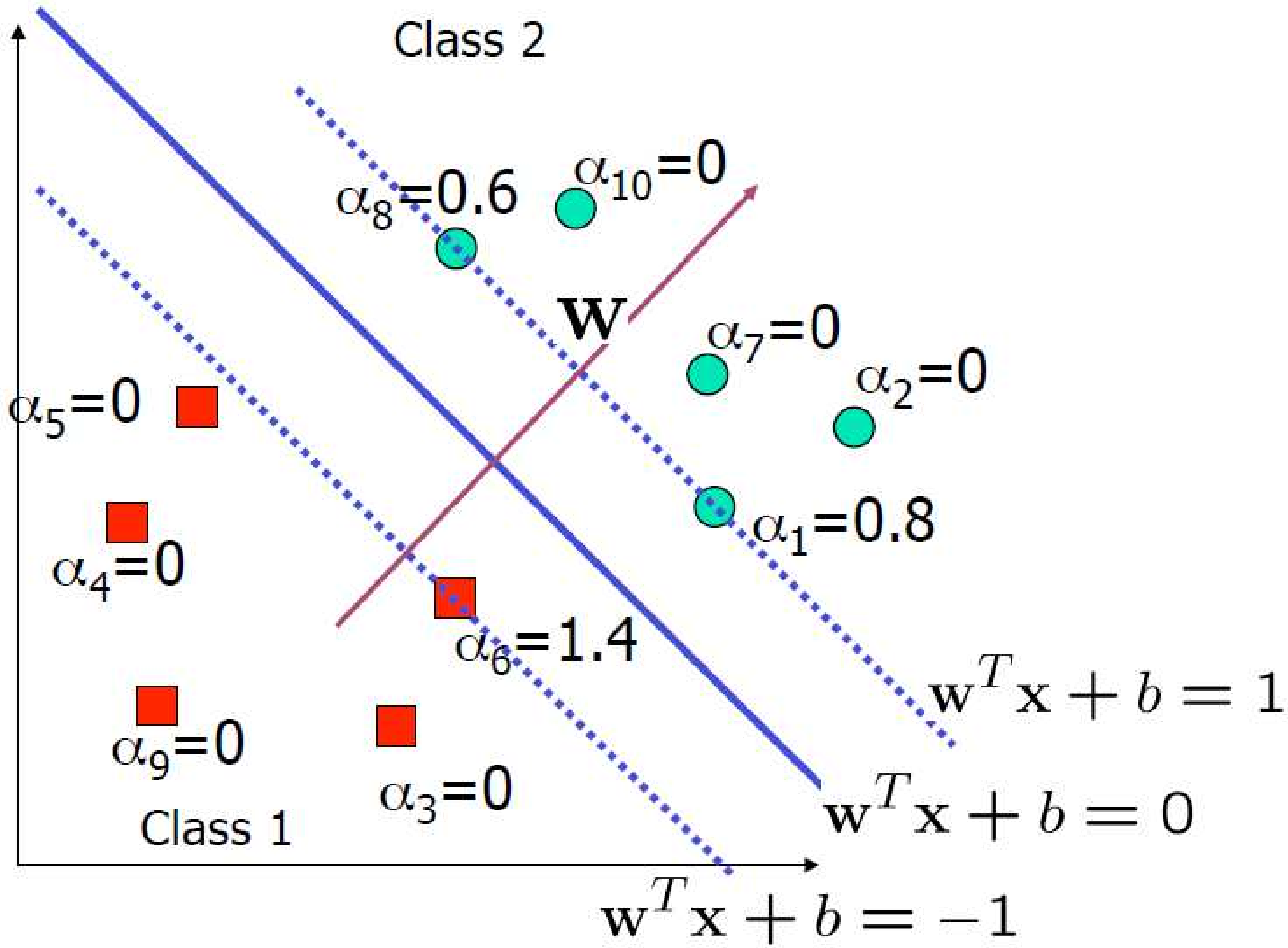
  subject to $\alpha_i \geq 0$, $\sum_{i=1}^{n} \alpha_i y_i = 0$.

- This is a quadratic programming (QP) problem. A global maximum of α_i can always be found.

Characteristics of the Solution

- Many of the α_i are zero. w is a linear combination of a small number of data points.
- x_i with non-zero α_i are called support vectors (SV).
- The decision boundary is determined only by the SV. Let t_j $(j = 1, \ldots, s)$ be the indices of the s support vectors. We have $w = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} x_{t_j}$.
- For testing with a new data z, Compute $w^T z + b = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} (x_{t_j}^T z) + b$.
- classify z as class 1 if the sum is positive, and class 2 otherwise.

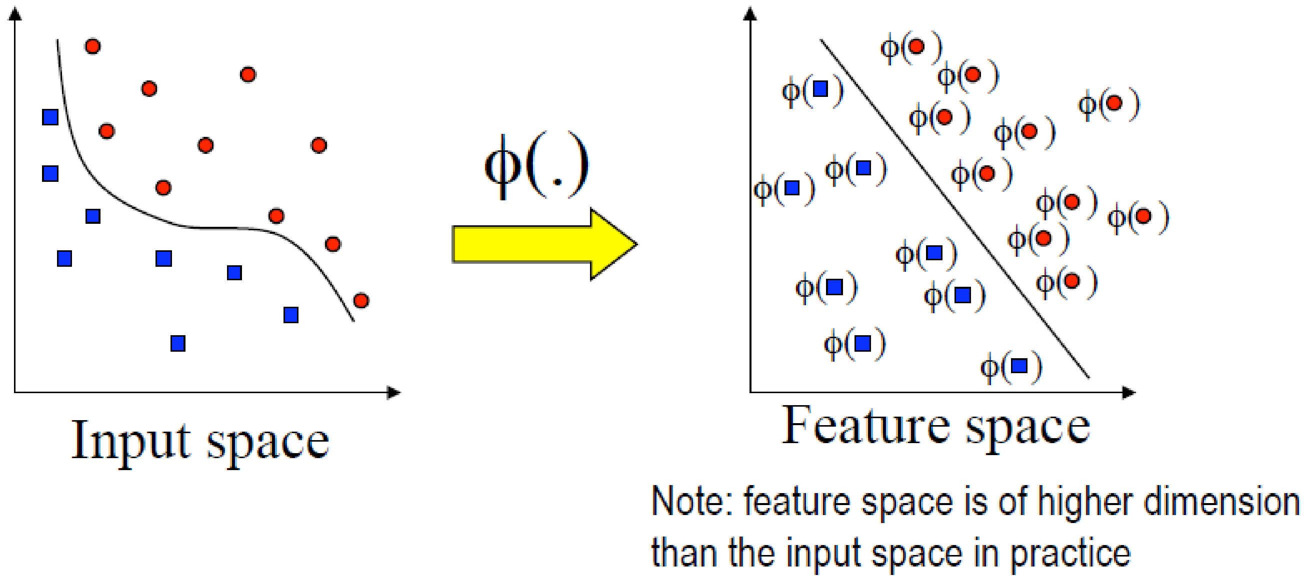
A Geometrical Interpretation



Extension to Non-linear Decision Boundary

- How to generalize it to become nonlinear?
- Key idea: transform x_i to a higher dimensional space to make classification easy.
- In the XOR problem, for example, adding a new feature of $x_1 x_2$ make the problem linearly separable.
- Input space: the space the point x_i are located.
- Feature space: the space of $\phi(x_i)$ after transformation.
- Why transform? Linear operation in the feature space is equivalent to nonlinear operation in input space. Classification can become easier with a proper transformation.

Transforming the Data



- Computation in the feature space can be costly because it is high dimensional.

- The kernel trick comes to rescue.

The Kernel Trick

- Recall the SVM optimization problem

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

  subject to $\alpha_i \geq 0$, $\sum_{i=1}^{n} \alpha_i y_i = 0$.

- The data points only appear as inner product.
- As long as we can calculate the inner product in the feature space, we do not need the mapping explicitly.
- Define the kernel function K by

$$K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$$

An Example for $\phi(.)$ and $K(.,.)$

- Suppose $\phi(.)$ is given as follows

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

- An inner product in the feature space is

$$< \phi([x_1, x_2]^T), \phi([y_1, y_2]^T) >= (1 + x_1y_1 + x_2y_2)^2$$

- if define the kernel function as follows, there is no need to carry out $\phi(.)$ explicitly:

$$K(x, y) = (1 + x_1y_1 + x_2y_2)^2$$

- This use of kernel function to avoid carrying out $\phi(.)$ explicitly is known as the kernel trick.

Kernel Functions

- In practical use of SVM, the user specifies the kernel function; the transformation $\phi(.)$ is not explicitly stated
- Given a kernel function $K(x_i, x_j)$, the transformation $\phi(.)$ is given by its eigenfunctions (a concept in functional analysis)
- Eigenfunctions can be difficult to construct explicitly.
- This is why people only specify the kernel function without worrying about the exact transformation.
- Another view: kernel function, being an inner product, is really a similarity measure between the objects.

Examples of Kernel Functions

- Polynomial kernel with degree d

$$K(x, y) = (x^T y + 1)^d$$

- Radial basis function kernel with width σ

$$K(x, y) = exp(-||x - y||^2 / (2\sigma^2))$$

Modification Due to Kernel Function in Training

- Change all inner products to kernel functions. The original is

$$max.W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{n} \alpha_i \alpha_j y_i y_j x_i^T x_j$$

subject to $\alpha_i \geq 0$, $\sum_{i=1}^{n} \alpha_i y_i = 0$.

- With kernel function

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

subject to $\alpha_i \geq 0$, $\sum_{i=1}^{n} \alpha_i y_i = 0$.

Modification Due to Kernel Function in Testing

- For testing, the new data z is classified as class 1 if $f \geq 0$, and as class 2 if $f < 0$.
- The original is $w = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} x_{t_j}$.

$$f = w^T z + b = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} (x_{t_j}^T z) + b$$

- With kernel function, $w = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} \phi(x_{t_j})$,

$$f = < w, \phi(z) > + b = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} K(x_{t_j}, z) + b$$

Example

- Suppose we have 5 1D data points.
  $x_1 = 1, x_2 = 2, x_3 = 4, x_4 = 5, x_5 = 6$, with x_1, x_2, x_5 as class 1 and x_3, x_4 as class 2, $y_1 = 1, y_2 = 1, y_3 = -1, y_4 = -1, y_5 = 1$
- Use the polynomial kernel of degree 2: $K(x, y) = (xy + 1)^2$, where C is set to 100
- Find α_i by

$$max \sum_{i=1}^{5} \alpha_i - \frac{1}{2} \sum_{i=1}^{5} \sum_{j=1}^{5} \alpha_i \alpha_j y_i y_j (x_i y_j + 1)^2$$

subject to $100 \geq \alpha_i \geq 0$, $\sum_{i=1}^{5} \alpha_i y_i = 0$.

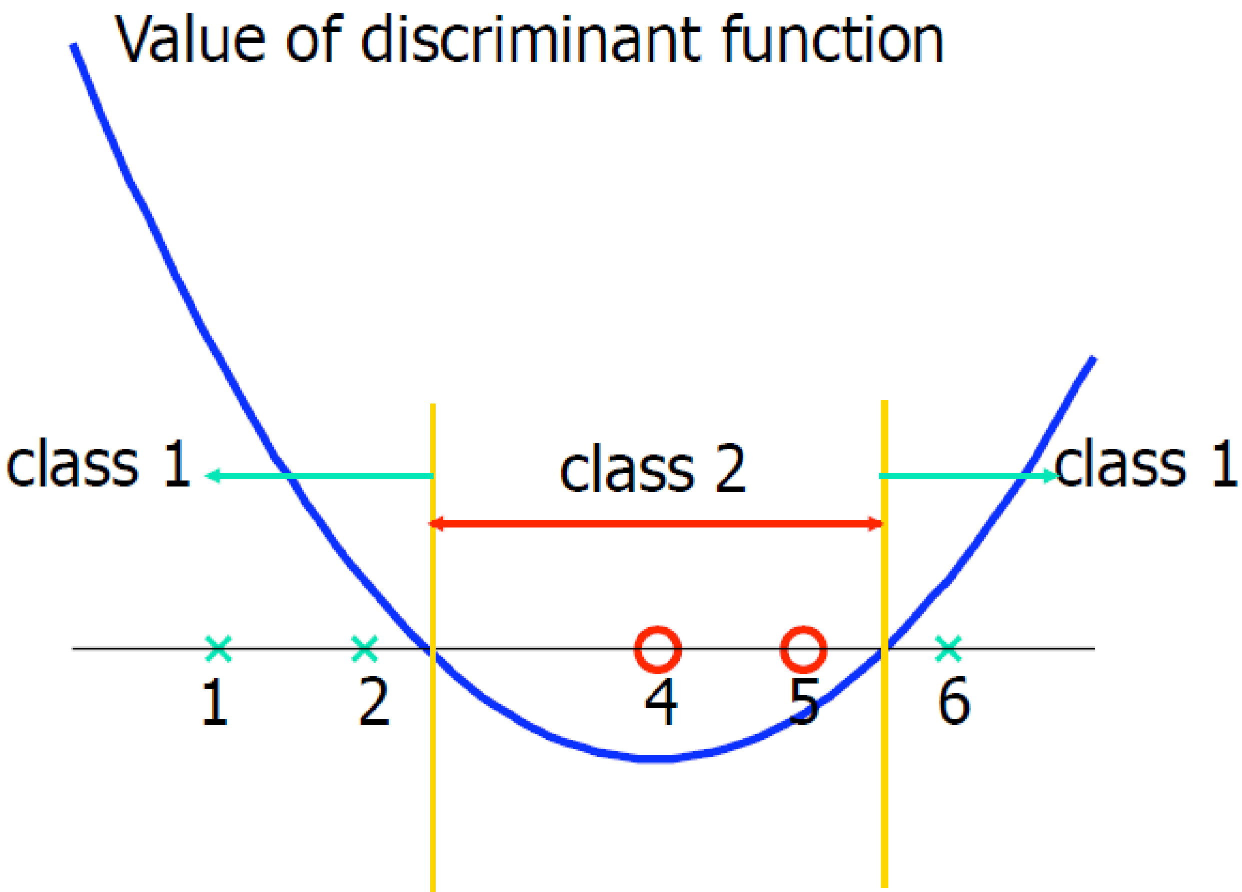
Solution

- Using a QP solver, we have $\alpha_1 = 0$,$\alpha_2 = 2.5$,$\alpha_3 = 0$, $\alpha_4 = 7.333$, $\alpha_5 = 4.833$.
- The support vectors are $\{x_2 = 2, x_4 = 5, x_5 = 6\}$
- The discriminant function is

$$f(z) = 2.5(1)(2z+1)^2+7.333(-1)(5z+1)^2+4.833(1)(6z+1)^2+b$$

- How to get b? b is recovered by solving $f(2) = 1$ or by $f(5) = -1$ or by $f(6) = 1$ since they lie on the decision line.
- $b = 9$, so $f(z) = 0.6667z^2 - 5.333z + 9$

Value of discriminant function



Strengths and Weaknesses of SVM

- Strengths
  - Training is relatively easy. No local optimal, unlike in neural networks.
  - It scales relatively well to high dimensional data.
  - Tradeoff between classifier complexity and error can be controlled explicitly.
- Weaknesses: Need to choose a "good" kernel function.

Steps for Classification

- Prepare the pattern matrix
- Select the kernel function to use
- Select the parameter of the kernel function and the value of C. You can use the values suggested by the SVM software, or you can set apart a validation set to determine the values of the parameter.
- Execute the training algorithm and obtain the α_i
- Unseen data can be classified using the α_i and the support vectors

Software

- A list of SVM implementation can be found at http://www.kernel-machines.org/software.html
- Some implementation (such as LIBSVM) can handle multi-class classification.
- SVMLight is among one of the earliest implementation of SVM.
- Several Matlab toolboxes for SVM are also available

Conclusion

- SVM is a useful alternative to neural networks.
- Two key concepts of SVM: maximize the margin and the kernel trick.
- Many SVM implementations are available on the web for you to try on your data set!

Resources

- http://www.kernel-machines.org/
- http://www.support-vector.net/
- http://www.support-vector.net/icml-tutorial.pdf
- http://www.kernel-machines.org/papers/tutorialnips.ps.gz
- http://www.clopinet.com/isabelle/Projects/SVM/applist.html