FE670 Algorithmic Trading Strategies

Lecture 11. Data Mining and Computational Intelligence

Steve Yang

Stevens Institute of Technology

11/21/2013

Outline

Data Mining

- Data mining is all about finding and confirming trends and/or relationships, some of which may be obvious while others may be much more subtle. Many of them are statistical in nature.

  There are several major types of data mining techniques:

  (1) Classification and clustering analysis
  (2) Time-series mining
  (3) Association rule mining

  As its name suggests, data mining is all about data. By applying analytical techniques, trends and/or relationships may be found within data.

Data Mining for Financial Markets

- **Classification and clustering analysis**: Both seek to identify common features in the data. Any commonalities may then be used for predictions, since if the results are known for one entity they are likely to be comparable for those with similar properties.

  Note that often such predictions will focus on the direction rather than the potential value. For example, specific conditions might be used to determine whether a stock index or exchange rate will increase /decrease in a certain time span.

  Classification may be strictly hierarchical, in which case properties such as country, currency, industry and sector are useful. Clustering analysis may also be applied to create immediate hierarchies, based on results such as price returns or volatility.

Data Mining for Financial Markets

- **Time-series mining**: In finance, time-series analysis is often used for forecasting, but it can also use factor models to try to identify any regular features, such as cyclical or seasonal effects.

  Thus data is often de-trended and normalized. Cure fitting techniques may be applied as well, using lagged data, simple moving averages or even Fourier transforms or wavelets to try to model the data.

  Transformations may also be used: shifting, scaling or warping the data to try to highlight any patterns. Due to the non-linear nature of many of these patterns, AI techniques, such neural networks and support vector machines, have proved to be effective forecasting tools.

Data Mining for Financial Markets

- **Associative rule mining**: In statistics, an association represents any relationship between two variables that makes them dependent. In terms of probability, this means the occurrence of one event makes it more likely that the other will occur.

  Note, it is important to differentiate between association and causality. A statistical association between two types of data is simply that; it does not imply anything more.

  When looking at associations, it is also important to consider the measure of confidence we have in them. R^2 is a statistical measure of the "goodness of fit" between different datasets. $p - value$ is a statistical measure of significance of an inference or model.

Computational Intelligence

- The term Artificial Intelligence (AI) was first coined by John McCarthy in 1956 during a workshop on computers and intelligence at Dartmouth. AI systems are designed to adapt and learn, and so effectively think for themselves. We can split AI into:

  Conventional AI is a top-down approach that uses logic and rules to make decisions. This generally relies on data that has been translated into known symbols, employing custom-made knowledge bases or statistical analysis.

  Computational Intelligence is a bottom-up approach that takes its inspiration from biological mechanisms, such as neural networks and evolutionary algorithms, etc.

Computational Intelligence

Computational intelligence, or soft computing, applies an array of different techniques, using elements of learning, optimization and adaptation/evolution.

Machine learning is used by computer systems to recognize complex patterns in data, and make intelligent decisions based on this. There are three main categories:

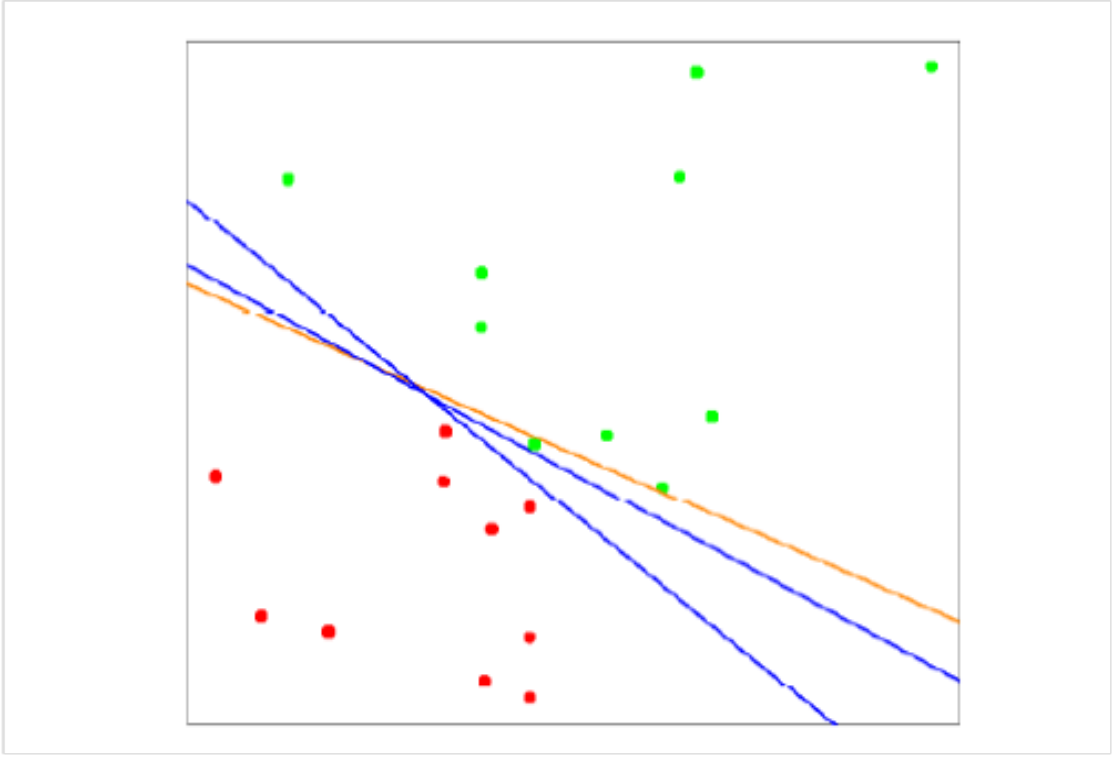
Supervised learning is adopted by many soft computing techniques, from the k-nearest neighbour algorithm to neural networks and support vector machines.

Unsupervised learning is learning without labels.

Reinforcement learning takes different approach: instead of using a dedicated training period, it relies on constant feedback from its environment.

Support Vector Machines

(1) Linear separable (2) Linear nonseparable (3) Nonlinear classification



(a) Separation: a decison boundary problem

Optimal Separating Hyperplane

- Suppose that the two classes can be linearly separated. Maximizes the distance to the closest point from either class. Find an unique solution
- **Training data**

  $(x_1, y_1), ..., (x_N, y_N)$
- **Response**

  $y_i \in -1, 1$
- **Hyperplane**

  $\{x : f(x) = x^T \beta + \beta_0, ||\beta|| = 1\}$

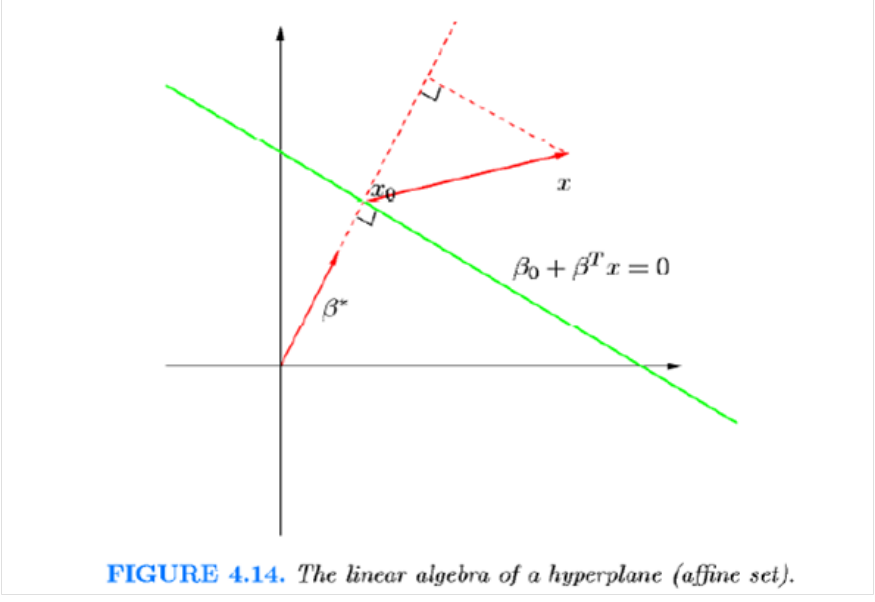
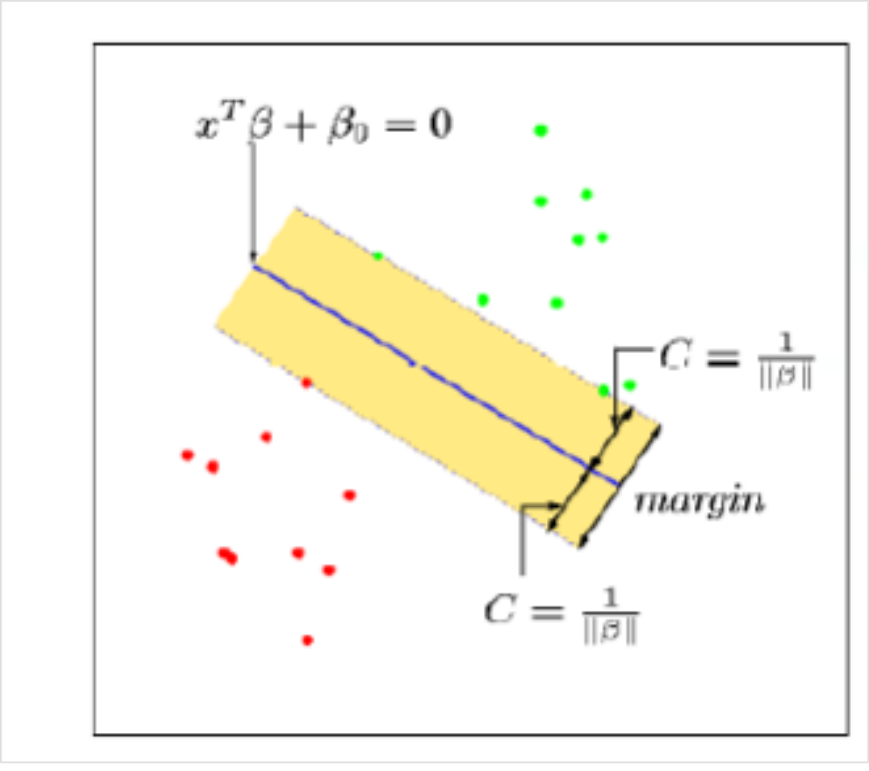


FIGURE 4.14. *The linear algebra of a hyperplane (affine set).*

Large Margin Decision Boundary

- The optimal hyperplane should be as far away from the data of both classes as possible
- To maximize the margin
- Note that $f(x)$ is signed the distance from a point x to the hyperplane:

$\{x : f(x) = x^T\beta + \beta_0, ||\beta|| = 1\}$ i.e.
$y_i f(x_i) = y_i(x_i^T\beta + \beta_0)$



Constrained Optimization

$$\max_{\beta_0,\beta,||\beta||} C$$

subject to:

$$y_i f(x_i) = y_i(x_i^T\beta + \beta_0) \geq C, \forall_i$$

$$\max_{\beta_0,\beta} ||\beta||$$

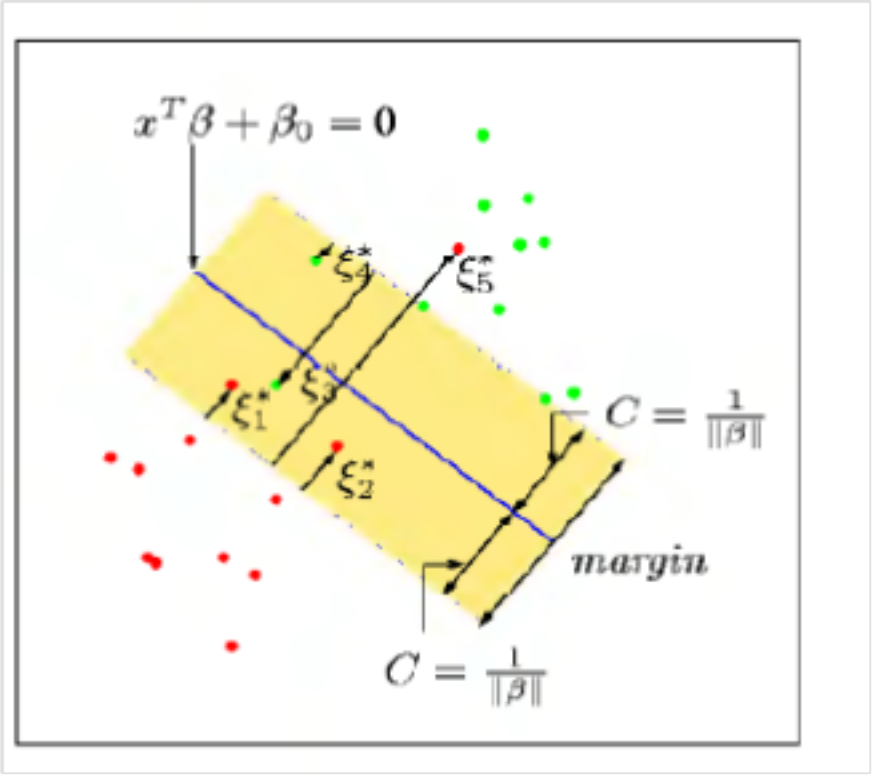
subject to:

$$y_i f(x_i) = y_i(x_i^T\beta + \beta_0) \geq 1, \forall_i$$

$$\max_{\beta_0,\beta,||\beta||} \frac{1}{2}||\beta||^2$$

subject to:

$$y_i f(x_i) = y_i(x_i^T\beta + \beta_0) \geq 1, \forall_i$$



The Primary Problem

- This is a convex optimization problem. (quadratic criterion with linear inequality constraints)
  This implies that child orders have acceptable market impact (if any), and execution time is not strictly defined.
  The Lagrange function

$$L_P = \frac{1}{2}\beta^T\beta - \sum_{i=1}^{N}\alpha_i[y_i(x_i^T\beta + \beta_0) - 1]$$

- Setting the derivatives to zeros

$$\beta + \sum_{i=1}^{N}\alpha_i(-y_i)x_i = 0 \Rightarrow \beta = \sum_{i=1}^{N}\alpha_i y_i x_i$$

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

The Dual Problem

- Substitution

$$L_D = \sum_{i=1}^{N} \alpha - \frac{1}{2} \sum_{i=1}^{N} \sum_{k=1}^{N} \alpha_i \alpha_k y_i y_k x_i^T x_k$$

subject to

$$\alpha_i \geq 0, \forall_i$$

$$\sum_{i=1}^{N} \alpha_i y_i = 0, \forall_i$$

- The original problem is known as the primal problem
- The new objective function is known as the duel problem, and is in terms of α_i only.
- It is known as the dual problem:
  if we know β, we know all α_i;
  if we know all α_i, we know β.

The Solution

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

  β can be recovered by

$$\beta = \sum_{i=1}^{N} \alpha_i y_i x_i$$

  Many of α_i are zeros; β is a linear combination of a small number of data points.

  x_i with nonzeros α_i are called support vectors (SV). The decision boundary is determined only by the support vectors.

- Karush-Kuhn-Tucker condition

$$\alpha_i[y_i(x_i^T \beta + \beta_0) - 1] = 0$$

Nonseparable Case

- **Separable**:

$$\max_{\beta_0,\beta,||\beta||} \frac{1}{2}||\beta||^2$$

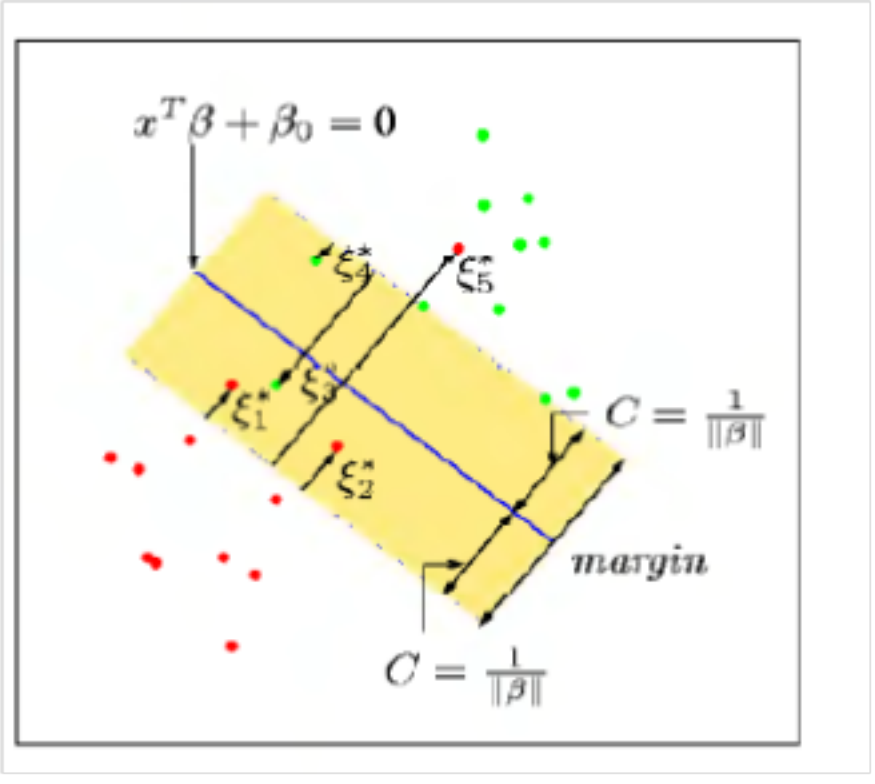
subject to: $y_i f(x_i) = y_i(x_i^T\beta + \beta_0) \geq 1, \forall_i$

- **Nonseparable**:

$$\max_{\beta_0,\beta,||\beta||} \frac{1}{2}||\beta||^2 + \gamma \cdot (\text{errors})$$

subject to certain constraints

* Maximize the margin, but allow for some errors



Constrained Optimization

- Modified constraints

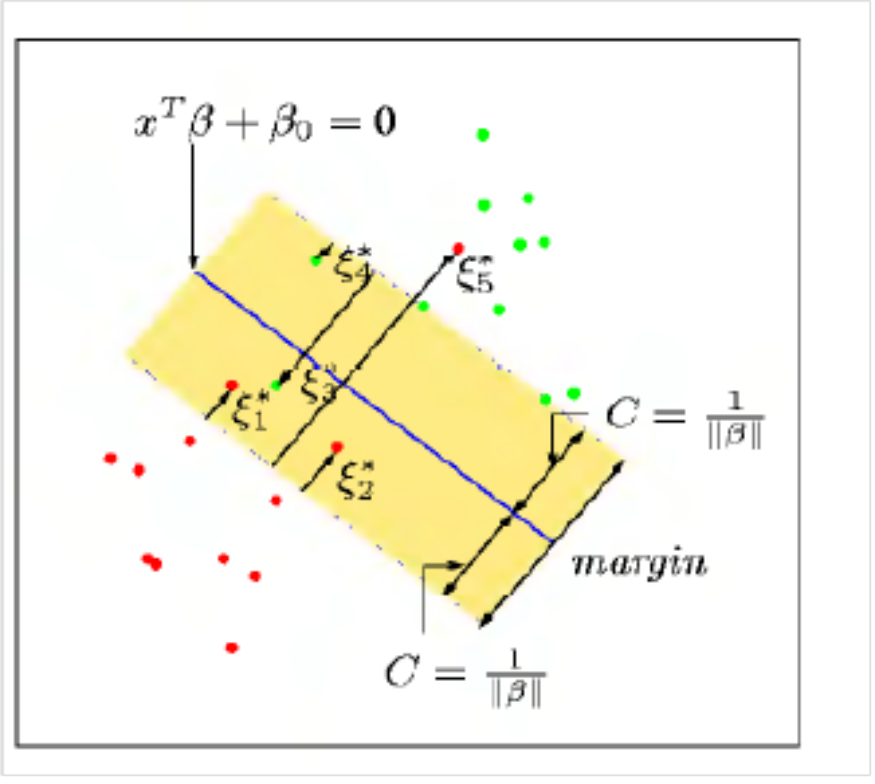
  $y_i(x_i^T\beta + \beta_0) \geq C(1-\xi), \forall_i$

  $\xi_i \geq 0$ the slack variable

  $\xi_i = 0$ no error

  The value ξ_i is the proportional amount by which the prediction is on the wrong side of its margin.

- $\sum_{i=1}^{N} \xi_i$ the total proportional amount by which predictions fall on the wrong side of their margin.



The Primary Problem

The primary problem

$$\min_{\beta_0,\beta} \frac{1}{2}||\beta||^2 - \gamma \sum_{i=1}^{N} \xi_i$$

subject to $\xi_i \geq 0, y_i(x_i^T\beta + \beta_0) \geq 1 - \xi_i, \forall_i$

gamma: trade off between error and margin.

Note that $\gamma = \infty$, is the separable case.

- The Lagrange function

$$L_P = \frac{1}{2}\beta^T\beta - +\gamma \sum_{i=1}^{N} \xi_i - \sum_{i=1}^{N} \alpha_i[y_i(x_i^T\beta + \beta_0) - 1]$$

The Dual Problem

- Setting the derivatives to zeros
  $\beta = \sum_{i=1}^{N} \alpha_i y_i x_i, 0 = \sum_{i=1}^{N} \alpha_i y_i, \alpha_i = \gamma - \mu_i$
- **The dual problem**:

$$L_D = \sum_{i=1}^{N} \alpha - \frac{1}{2} \sum_{i=1}^{N} \sum_{k=1}^{N} \alpha_i \alpha_k y_i y_k x_i^T x_k$$

subject to

$$\gamma \geq \alpha_i \geq 0, \forall_i$$

$$\sum_{i=1}^{N} \alpha_i y_i = 0, \forall_i$$

- β can be recovered $\beta = \sum_{i=1}^{N} \alpha_i y_i x_i$

The Solution

- Once again, a quadratic programming can solve and a global maximum of α_i can always be found
- This is very similar to the optimization problem in the linear separable case, except that there is an upper bound γ on α_i now

- Many of α_i are zeros;

- The observations with nonzeros α_i are called the support vectors.

  some lie on the edge of the margin, i.e. $\hat{\xi}_i = 0, 0 < \hat{\alpha}_i < \gamma$

  some lie on the wrong side of the margin, i.e. $\hat{\xi}_i > 0, \hat{\alpha}_i = \gamma$

  some lie on the wrong side of the decision boundary: $\hat{\xi}_i > 1, \hat{\alpha}_i = \gamma$

Nonlinear Decision Boundary

- Key idea: to transform the original (input) feature space into higher dimensional feature space
  $\mathbf{X} = (X_i, ..., X_p) \rightarrow h(\mathbf{X}) = (h(X_i), ..., h(X_p))$

- Transformation

* linear operation in the transformed feature space is equivalent to nonlinear operation in the original feature space

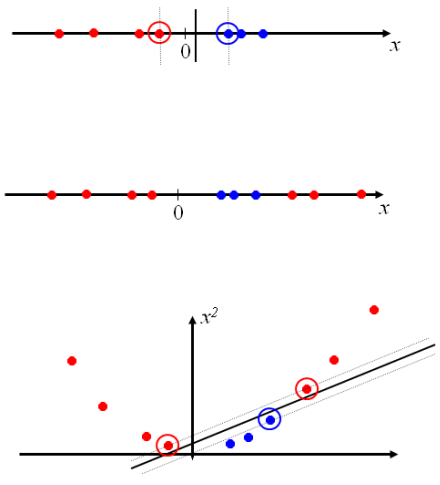
** classification can become easier with a proper transformation.

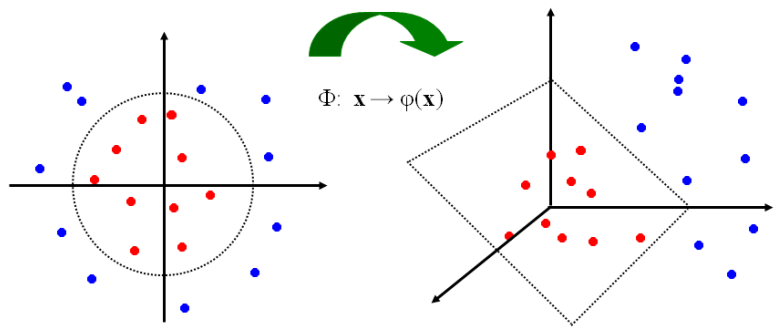
- Data points appear as the inner product $x_i^T x_k$

- As long as one can calculate the inner product in the feature space, one does not need to know the transformation explicitly.

- Replace $x_i^T x_k$ by $K(x_i, x_k) =< h(x_i), h(x_k) >$

Nonlinear Decision Boundary





Kernel Function- an Example

- Suppose

$$h(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}) = (1, \sqrt{2}x_1, \sqrt{2}x_2, x_1^2, x_2^2, \sqrt{2}x_1x_2) \quad (1)$$

- An inner product in the feature space

$$< h(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}), h(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}) >= (1 + x_1y_1 + x_2y_2)^2 \quad (2)$$

- If define the kernel function as follows, there is no need to carry out $h(\cdot)$ explicitly

$$K(X, Y) = (1 + x_1y_1 + x_2y_2)^2 \quad (3)$$

Kernel Function

- In practical use of SVM, the user specifies the kernel function; the transformation $h(\cdot)$ is not explicitly stated

- Given a kernel function $K(x, y)$, the transformation is $h(x)$ 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

- Kernel function, being an inner product, is really a similarity measure between the objects

Kernel Function

- Polynomial:

$$K(X, Y) = (1 + x^T y)^d \tag{4}$$

- Radial basis:

$$K(X, Y) = exp(\frac{-||x - y||^2}{2\sigma}) \tag{5}$$

* The feature space is infinite-dimensional.

- Neural network:

$$K(X, Y) = tanh(\kappa x^T y + \theta) \tag{6}$$

* It does not satisfy the Mercers condition on all κ and θ.

Kernel Function Issues

- Choice of kernel
  - Gaussian or polynomial kernel is default
  - if ineffective, more elaborate kernels are needed (similarity measures)
- Choice of kernel parameters
  - e.g. in Gaussian kernel
  - is the distance between closest points with different classifications
  - In the absence of reliable criteria, applications rely on the use of a validation set or cross-validation to set such parameters.
- Select the tuning parameter
  - use the value suggested by the SVM software, or determine the value of the parameter by the CV method.