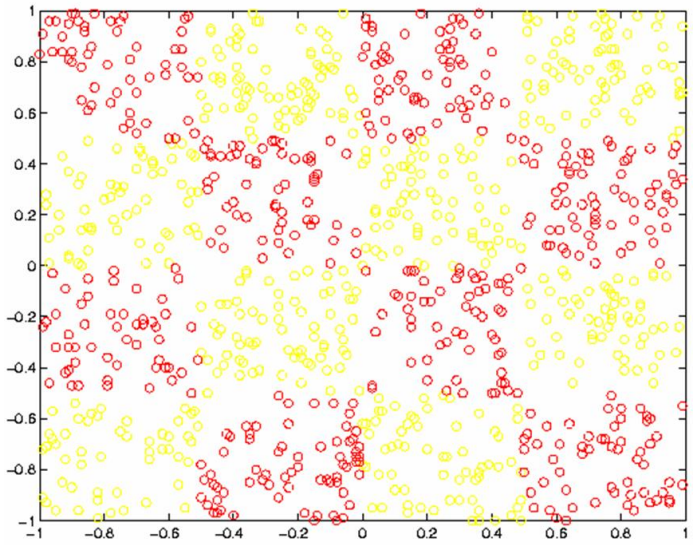
Support Vector Machines II--Kernel

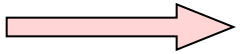
Jie Tang

Department of Computer Science and Technology
Tsinghua University

Kernel

1. Why kernel?





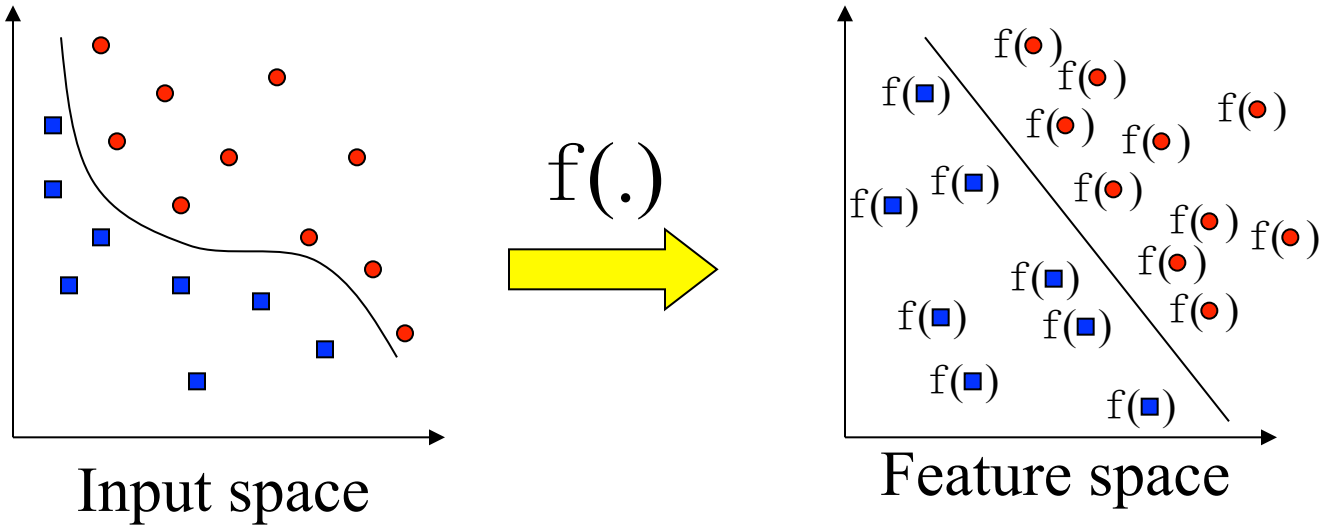
- **Key idea:** transform $\mathbf{x}_i$ to a higher dimensional space to "make life easier"
  - **Input space:** the space the point $\mathbf{x}_i$ are located
  - **Feature space:** the space of f(**x**i) after transformation
- Why transform?
  - Linear operation in the feature space is equivalent to non-linear operation in input space
  - Classification can become easier with a proper transformation. In the XOR problem, for example, adding a new feature of x_1x_2 make the problem linearly separable

2. What is feature space mapping?

$$x \Rightarrow \Phi(x)$$

$$K(x,z) = \Phi(x)^T \Phi(z)$$ kernel function

Feature Space Mapping



Note: feature space is of higher dimension than the input space in practice

Replace all $\langle x^{(i)}, x \rangle$ by $\langle \Phi(x^{(i)}), \Phi(x) \rangle$

$$\max_{\alpha} L(\alpha) = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{N} y^{(i)} y^{(j)} \alpha_i \alpha_j \langle \Phi(x^{(i)}), \Phi(x^{(j)}) \rangle$$

$$s.t. \qquad \alpha_i \geq 0, i \in [1, N]$$

$$\sum_{i=1}^{N} \alpha_i y^{(i)} = 0$$

$$w^T x + b = (\sum_{i=1}^{N} \alpha_i y^{(i)} x^{(i)})^T x + b$$

$$= \sum_{i=1}^{N} \alpha_i y^{(i)} \langle \Phi(x^{(i)}), \Phi(x) \rangle + b$$

With kernel, what's more interesting to us?

We can compute kernel without calculating mapping

- Computation in the feature space can be costly because it is high dimensional
  - The feature space is even infinite-dimensional!
- The kernel trick comes to rescue
- Both in the training and test, The data points only appear as inner product <x, z>.
- As long as we can calculate the inner product in the feature space, we do not need the mapping explicitly

$$K(x,z) = \Phi(x)^T \Phi(z)$$

For example: $$K(x,z) = \exp(-\frac{\|x-z\|^2}{2\sigma^2})$$

An Example for f(.) and K(.,.)

- Suppose f(.) is given as follows

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

- An inner product in the feature space is

$$\langle \phi(\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}), \phi(\begin{bmatrix} y_1 \\ y_2 \end{bmatrix}) \rangle = (1 + x_1y_1 + x_2y_2)^2$$

- So, if we define the kernel function as follows, there is no need to carry out f(.) explicitly

$$K(\mathbf{x}, \mathbf{y}) = (1 + x_1y_1 + x_2y_2)^2$$

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

Kernel Functions

- In practical use of SVM, the user specifies the kernel function; the transformation $\mathtt{f}(.)$ is not explicitly stated
- Given a kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$, the transformation $\mathtt{f}(.)$ 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 actually a similarity measure between the objects

Examples of Kernel Functions

- Polynomial kernel with degree *d*

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x}^T\mathbf{y} + 1)^d$$

- Radial basis function kernel with width s

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

  - Closely related to radial basis function neural networks
  - The feature space is infinite-dimensional

- Sigmoid with parameter k and q

$$K(\mathbf{x}, \mathbf{y}) = \tanh(\kappa\mathbf{x}^T\mathbf{y} + \theta)$$

  - It does not satisfy the Mercer condition on all k and q

Modification Due to Kernel Function

- Change all inner products to kernel functions
- For training,

Original

$$\text{max. } W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{n} \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j$$

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

With kernel function

$$\text{max. } W(\boldsymbol{\alpha}) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1,j=1}^{n} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$$

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

Modification Due to Kernel Function

- For testing, the new data **z** is classified as class +1 if $f \geq 0$, and as class -1 if $f < 0$

Original

$$\mathbf{w} = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} \mathbf{x}_{t_j}$$

$$f = \mathbf{w}^T \mathbf{z} + b = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} \mathbf{x}_{t_j}^T \mathbf{z} + b$$

With kernel function

$$\mathbf{w} = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} \phi(\mathbf{x}_{t_j})$$

$$f = \langle \mathbf{w}, \phi(\mathbf{z}) \rangle + b = \sum_{j=1}^{s} \alpha_{t_j} y_{t_j} K(\mathbf{x}_{t_j}, \mathbf{z}) + b$$

Characteristics of Kernel Functions

- Since the training of SVM only requires the value of $K(\mathbf{x}_i, \mathbf{x}_j)$, there is no restriction of the form of $\mathbf{x}_i$ and $\mathbf{x}_j$
  - $\mathbf{x}_i$ can be a sequence or a tree, instead of a feature vector
- $K(\mathbf{x}_i, \mathbf{x}_j)$ is just a similarity measure comparing $\mathbf{x}_i$ and $\mathbf{x}_j$
- For a test object $\mathbf{z}$, the discriminat function essentially is a weighted sum of the similarity between z and a pre-selected set of objects (the support vectors)

$$f(\mathbf{z}) = \sum_{\mathbf{x}_i \in \mathcal{S}} \alpha_i y_i K(\mathbf{z}, \mathbf{x}_i) + b$$

$\mathcal{S}$: the set of support vectors

Characteristics of Kernel Functions

- Not all similarity measure can be used as kernel function, however
  - The kernel function needs to satisfy the Mercer function, i.e., the function is "positive-definite"
  - This implies that the *n* by *n* kernel matrix, in which the (i,j)-th entry is the $K(\mathbf{x}_i, \mathbf{x}_j)$, is always positive definite
  - This also means that the QP is convex and can be solved in polynomial time

Example

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

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

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

Example

- By using a QP solver, we get
  - a_1=0, a_2=2.5, a_3=0, a_4=7.333, a_5=4.833
  - Note that the constraints are indeed satisfied
  - The support vectors are {x_2=2, x_4=5, x_5=6}
- The discriminant function is

α_5 y_5 $K(z, x_5)$

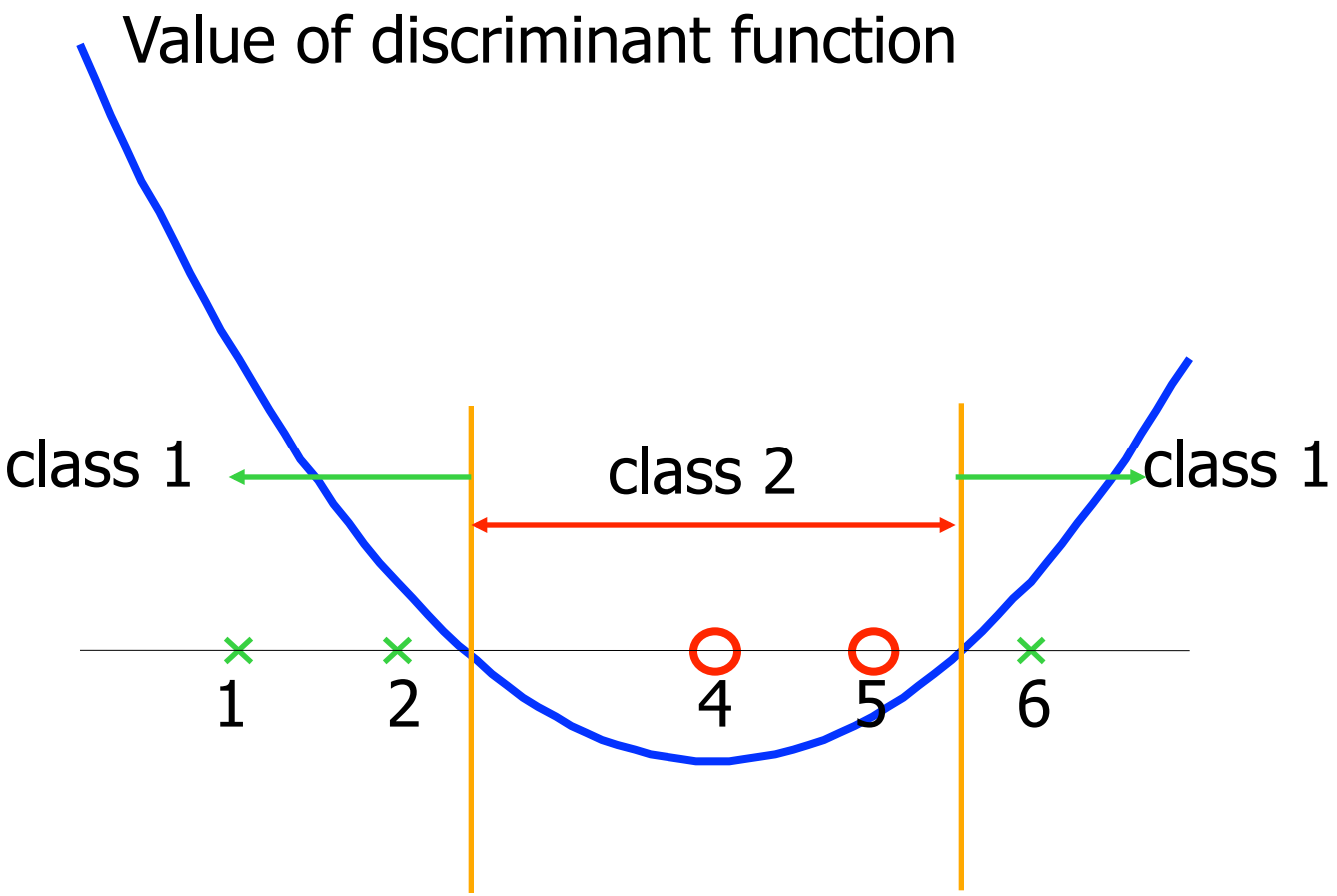
$$\begin{aligned} & f(z) \\ &= 2.5(1)(2z+1)^2 + 7.333(-1)(5z+1)^2 + 4.833(1)(6z+1)^2 + b \\ &= 0.6667z^2 - 5.333z + b \end{aligned}$$

- *b* is recovered by solving f(2)=1 or by f(5)=-1 or by f(6)=1, as x_2 and x_5 lie on the line $\phi(\mathbf{w})^T\phi(\mathbf{x}) + b = 1$ and x_4 lies on the line

$$\phi(\mathbf{w})^T\phi(\mathbf{x}) + b = -1$$

- All three give b=9 $\Longrightarrow$ $f(z) = 0.6667z^2 - 5.333z + 9$

Example


Value of discriminant function
class 1
class 2
class 1
1
2
4
5
6

Choosing the Kernel Function

- Probably the most tricky part of using SVM.
- The kernel function is important because it creates the kernel matrix, which summarizes all the data
- Many principles have been proposed (diffusion kernel, Fisher kernel, string kernel, ...)
- There is even research to estimate the kernel matrix from available information

- In practice, a low degree polynomial kernel or RBF kernel with a reasonable width is a good initial try
- Note that SVM with RBF kernel is closely related to RBF neural networks, with the centers of the radial basis functions automatically chosen for SVM

Summary: Steps for Classification

- Prepare the feature 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 a_i
- Unseen data can be classified using the a_i and the support vectors

Conclusions

- SVM is a useful alternative to neural networks
- Two key concepts of SVM:
  - Maximal margin
  - Kernel trick

Other Resources

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
- Svm-Light is among one of the earliest implementation of SVM
- Weka
- Several Matlab toolboxes for SVM are also available

Useful Link

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

References

- Vladimir N. Vapnik. The nature of statistical learning theory. Springer-Verlag New York. 1998.
- Andrew Ng. CS229 Lecture notes. Lectures from 10/19/03 to 10/26/03. Part V. Support Vector Machines
- CHRISTOPHER J.C. BURGES. A Tutorial on Support Vector Machines for Pattern Recognition. Data Mining and Knowledge Discovery, 2, 121–167 (1998). 1998 Kluwer Academic Publishers, Boston. Manufactured in The Netherlands.
- Cristianini, N., Shawe-Taylor, J., An Introduction to Support Vector Machines, Cambridge University Press, (2000).

People

- Vladimir Vapnik.
- J. Platt
- J. Platt, N. Cristianini, J. Shawe-Taylor
- Shawe-Taylor, J.
- Burges, C. J. C.
- Thorsten Joachims
- Chih-Jen Lin
- Find more by
  - http://aminer.org/search?q=support vector machine

Useful Slides

- Sergios Theodoridis. Support Vector Machines: A Geometric View Point. University of Athens.
- Olvi L. Mangasarian. Mathematical Programming for Support Vector Machines. University of Wisconsin-Madison.
- Ahmed Abbasi. Pattern Recognition using Support Vector Machine and Principal Component Analysis. University of Arizona.
- Martin Law. A Simple Introduction to Support Vector Machines. Michigan State University

Thanks!

HP: http://keg.cs.tsinghua.edu.cn/jietang/

Email: jietang@tsinghua.edu.cn