Support Vector Machines (SVMs)
Part 3: Kernels

Yingming Li
yingming@zju.edu.cn

Data Science & Engineering Research Center, ZJU

3rd April 2018

Kernels are generalizations of inner products

- A kernel is a function of two data points such that

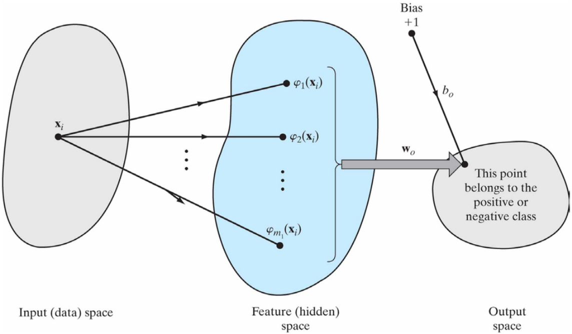
$$k(x, x') = \phi^T(x)\phi(x')$$

  For some function $\phi(x)$
- It is therefore symmetric: $k(x, x') = k(x', x)$
- Can compute $k(x, x')$ from an explicit $\phi(x)$
- Or prove that $k(x, x')$ corresponds to some $\phi(x)$
  - Never need to actually compute $\phi(x)$

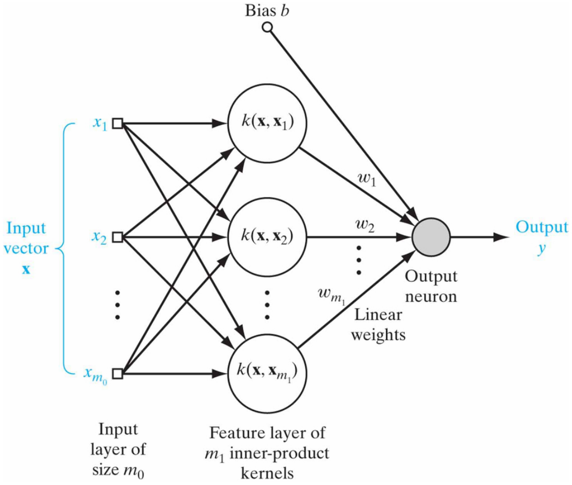
SVM as a kernel machine

- **Cover's theorem:** A complex classification problem, cast in a high-dimensional space nonlinearly, is more likely to be linearly separable than in the low-dimensional input space
- SVM for pattern classification
  - Nonlinear mapping of the input space onto a high-dimensional feature space
  - Constructing the optimal hyperplane for the feature space

Kernel machine illustration



Kernelized SVM looks a lot like an RBF net



Kernel matrix

- The matrix

$$\mathbf{K} = \begin{bmatrix} k(\mathbf{x}_1, \mathbf{x}_1) & \cdots & k(\mathbf{x}_1, \mathbf{x}_N) \\ & \vdots & \\ \cdots & k(\mathbf{x}_i, \mathbf{x}_j) & \cdots \\ & \vdots & \\ k(\mathbf{x}_N, \mathbf{x}_1) & \cdots & k(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix}$$

  is called the kernel matrix, or the Gram matrix.
- $\mathbf{K}$ is positive semidefinite

Mercer's theorem relates kernel functions and inner product spaces

- Suppose that for all finite sets of points $\{\mathbf{x}_p\}_{p=1}^{N}$ and real number $\{\mathbf{a}\}_{p=1}^{\infty}$

$$\sum_{i,j} a_j a_i k(\mathbf{x}_i, \mathbf{x}_j) \geq 0$$

- Then $\mathbf{K}$ is called a positive semidefinite kernel
- And can be written as

$$k(\mathbf{x}, \mathbf{x}') = \phi^T(\mathbf{x})\phi(\mathbf{x}')$$

- For some vector-valued function $\phi(\mathbf{x})$

Kernels can be applied in many situations

- Kernel trick: when predictions are based on inner products of data points, replace with kernel function
- Some algorithms where this is possible
  - Linear / ridge regression
  - Principal components analysis
  - Canonical correlation analysis
  - Perceptron classifier

Some popular kernels

- Polynomial kernel, parameters c and p

$$k(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T \mathbf{x}' + c)^p$$

  - Finite-dimensional $\phi(\mathbf{x})$ can be explicitly computed
- Gaussian or RBF kernel, parameter σ

$$k(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{1}{2\sigma}||\mathbf{x} - \mathbf{x}'||^2\right)$$

  - Infinite-dimensional $\phi(\mathbf{x})$
  - Equivalent to RBF network, but more principled way of finding centers

Some popular kernels

- Hypebolic tangent kernel, parameters β_1 and β_2

$$k(\mathbf{x}, \mathbf{x}') = \tanh(\beta_1 \mathbf{x}^T \mathbf{x}' + \beta_2)$$

  - Only positive semidefinite for some values of β_1 and β_2
  - Inspired by neural networks, but more principled way of selecting number of hidden units
- String kernels or other structure kernels
  - Can prove that they are positive definite
  - Computed between non-numeric items
  - Avoid converting to fixed-length feature vectors

Example: polynomial kernel

- Polynomial kernel in 2D, $c = 1$, $p = 2$
  $k(\mathbf{x}, \mathbf{x}') = (\mathbf{x}^T\mathbf{x}' + 1)^2 = (x_1 x_1' + x_2 x_2' + 1)^2 =$
  $x_1^2 x_1'^2 + x_2^2 x_2'^2 + 2x_1 x_1' x_2 x_2' + 2x_1 x_1' + 2x_2 x_2' + 1$
- If we define

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

- Then $k(\mathbf{x}, \mathbf{x}') = \phi^T(\mathbf{x})\phi(\mathbf{x}')$

Example: XOR problem again

- Consider (once again) the XOR problem
- The SVM can solve it using a polynomial kernel
  - With $p = 2$ and $c = 1$

TABLE 6.2 XOR Problem

Input vector $\mathbf{x}$	Desired response d
$(-1, -1)$	-1
$(-1, +1)$	$+1$
$(+1, -1)$	$+1$
$(+1, +1)$	-1

XOR: first compute the kernel matrix

- In general, $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) = (1 + \mathbf{x}_i^T \mathbf{x}_j)^2$
- For example,

$$K_{11} = k\left(\begin{bmatrix} -1 \\ -1 \end{bmatrix}, \begin{bmatrix} -1 \\ -1 \end{bmatrix} \right) = (1+2)^2 = 9$$

$$K_{12} = k\left(\begin{bmatrix} -1 \\ -1 \end{bmatrix}, \begin{bmatrix} -1 \\ +1 \end{bmatrix} \right) = (1+0)^2 = 1$$

- So

$$K = \begin{bmatrix} 9 & 1 & 1 & 1 \\ 1 & 9 & 1 & 1 \\ 1 & 1 & 9 & 1 \\ 1 & 1 & 1 & 9 \end{bmatrix}$$

XOR: first compute the kernel matrix

- Or compute $\phi(x_i)$ and their inner products, e.g.,
  - $\phi(\mathbf{x}) = [x_1^2, x_2^2, \sqrt{2}x_1x_2, \sqrt{2}x_1, \sqrt{2}x_2, 1]$, where 1 is added for b.
  - Since $\phi(\mathbf{x})$ includes 1, no need for separate b later

$$\phi(\mathbf{x}_1) = \phi\left(\begin{bmatrix}-1\\-1\end{bmatrix}\right) = [1, 1, \sqrt{2}, -\sqrt{2}, -\sqrt{2}, 1]^T$$

$$\phi(\mathbf{x}_2) = \phi\left(\begin{bmatrix}-1\\+1\end{bmatrix}\right) = [1, 1, -\sqrt{2}, -\sqrt{2}, \sqrt{2}, 1]^T$$

- Then

$$K_{11} = \phi^T(\mathbf{x}_1)\phi(\mathbf{x}_1) = 1 + 1 + 2 + 2 + 2 + 1 = 9$$
$$K_{12} = \phi^T(\mathbf{x}_1)\phi(\mathbf{x}_1) = 1 + 1 - 2 + 2 - 2 + 1 = 1$$

- Results in same K matrix, but more computation

XOR: Combine class labels into K

- Define matrix $\tilde{K}$ such that $\tilde{K}_{ij} = K_{ij} d_i d_j$
- Recall $\mathbf{d} = [-1, +1, +1, -1]^T$

$$\tilde{K} = \begin{bmatrix} +9 & -1 & -1 & +1 \\ -1 & +9 & +1 & -1 \\ -1 & +1 & +9 & -1 \\ +1 & -1 & -1 & +9 \end{bmatrix}$$

XOR: Solve dual Lagrangian for a

- Find fixed points of

$$\tilde{L}(\mathbf{a}) = \mathbf{1}^T\mathbf{a} - \frac{1}{2}\mathbf{a}^T\tilde{K}\mathbf{a}$$

- Set matrix gradient to 0

$$\nabla\tilde{L} = \mathbf{1} - \tilde{K}\mathbf{a} = \mathbf{0}$$

$$\Rightarrow \mathbf{a} = \tilde{K}^{-1}\mathbf{1} = \left[\frac{1}{8}, \frac{1}{8}, \frac{1}{8}, \frac{1}{8}\right]^T$$

- Satisfies all conditions: $a_p \geqslant 0 \forall p \qquad \sum_p a_p d_p = 0$
  - So this is the solution
- All points are support vectors

XOR: Compute w (including b) from a

$$\mathbf{w} = \sum_p a_p d_p \mathbf{x}_p$$

$$= -\frac{1}{8}\phi(\mathbf{x}_1) + \frac{1}{8}\phi(\mathbf{x}_2) + \frac{1}{8}\phi(\mathbf{x}_3) - \frac{1}{8}\phi(\mathbf{x}_4)$$

$$= \frac{1}{8}\left(- \begin{bmatrix} 1 \\ 1 \\ \sqrt{2} \\ -\sqrt{2} \\ -\sqrt{2} \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ -\sqrt{2} \\ -\sqrt{2} \\ \sqrt{2} \\ 1 \end{bmatrix} + \begin{bmatrix} 1 \\ 1 \\ -\sqrt{2} \\ \sqrt{2} \\ -\sqrt{2} \\ 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \\ \sqrt{2} \\ \sqrt{2} \\ \sqrt{2} \\ 1 \end{bmatrix} \right) = \begin{bmatrix} 0 \\ 0 \\ -\frac{1}{\sqrt{2}} \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

XOR: Examine prediction function

- Prediction function

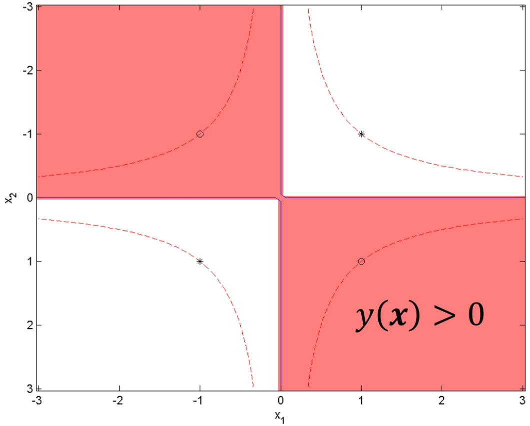
$$y(\mathbf{x}) = \mathbf{w}^T\phi(\mathbf{x})$$
$$= \left[0, 0, -\frac{1}{\sqrt{2}}0, 0, 0,\right]^T \left[x_1^2, x_2^2, \sqrt{2}x_1x_2, \sqrt{2}x_1, \sqrt{2}x_2, 1\right]$$
$$= -x_1x_2$$

- Predictions are based on product of the dimensions

$$\begin{aligned}
y(\mathbf{x}_1) &= -(-1)(-1) = -1 \\
y(\mathbf{x}_2) &= -(-1)(+1) = +1 \\
y(\mathbf{x}_3) &= -(+1)(-1) = +1 \\
y(\mathbf{x}_4) &= -(+1)(+1) = -1
\end{aligned}$$

XOR: Decision boundaries

- Decision boundary at $y(\mathbf{x}) = -x_1 x_2 = 0$
- Support vectors at $y(\mathbf{x}) = -x_1 x_2 = 1$



Thank you!