

ECE595 / STAT598: Machine Learning I

Lecture 21.2: Support Vector Machine - Kernel Trick

Spring 2020

Stanley Chan

School of Electrical and Computer Engineering
Purdue University



Outline

Support Vector Machine

- Lecture 19 SVM 1: The Concept of Max-Margin
- Lecture 20 SVM 2: Dual SVM
- **Lecture 21 SVM 3: Soft SVM and Kernel SVM**

This lecture: Support Vector Machine: Soft and Kernel

- Soft SVM
 - Motivation
 - Formulation
 - Interpretation
- **Kernel Trick**
 - Nonlinearity
 - Dual Form
 - Kernel SVM

The Kernel Trick

- A trick to turn linear classifier to nonlinear classifier.
- Dual SVM

$$\begin{aligned} & \underset{\lambda \geq 0}{\text{maximize}} && -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \lambda_i \lambda_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j + \sum_{j=1}^n \lambda_j \\ & \text{subject to} && \sum_{j=1}^n \lambda_j y_j = 0. \end{aligned}$$

- Kernel Trick

$$\begin{aligned} & \underset{\lambda \geq 0}{\text{maximize}} && -\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \lambda_i \lambda_j y_i y_j \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j) + \sum_{j=1}^n \lambda_j \\ & \text{subject to} && \sum_{j=1}^n \lambda_j y_j = 0. \end{aligned}$$

- You have to do this in dual. Primal is hard. See next slide.

The Kernel Trick

- Define

$$K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j).$$

- The matrix \mathbf{Q} is

$$\mathbf{Q} = \begin{bmatrix} y_1 y_1 \mathbf{x}_1^T \mathbf{x}_1 & \dots & y_1 y_N \mathbf{x}_1^T \mathbf{x}_N \\ y_2 y_1 \mathbf{x}_2^T \mathbf{x}_1 & \dots & y_2 y_N \mathbf{x}_2^T \mathbf{x}_N \\ \vdots & \vdots & \vdots \\ y_N y_1 \mathbf{x}_N^T \mathbf{x}_1 & \dots & y_N y_N \mathbf{x}_N^T \mathbf{x}_N \end{bmatrix}$$

- By Kernel Trick:

$$\mathbf{Q} = \begin{bmatrix} y_1 y_1 K(\mathbf{x}_1, \mathbf{x}_1) & \dots & y_1 y_N K(\mathbf{x}_1, \mathbf{x}_N) \\ y_2 y_1 K(\mathbf{x}_2, \mathbf{x}_1) & \dots & y_2 y_N K(\mathbf{x}_2, \mathbf{x}_N) \\ \vdots & \vdots & \vdots \\ y_N y_1 K(\mathbf{x}_N, \mathbf{x}_1) & \dots & y_N y_N K(\mathbf{x}_N, \mathbf{x}_N) \end{bmatrix}$$

Kernel

- The inner product $\Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j)$ is called a **kernel**

$$K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)^T \Phi(\mathbf{x}_j).$$

- Second-Order Polynomial kernel

$$K(\mathbf{u}, \mathbf{v}) = (\mathbf{u}^T \mathbf{v})^2.$$

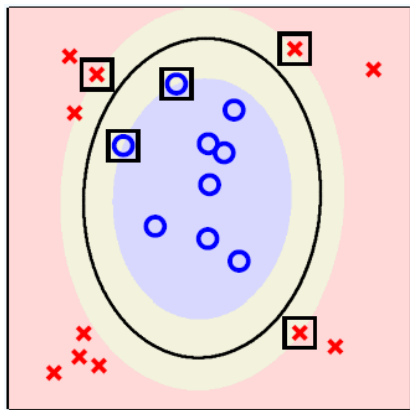
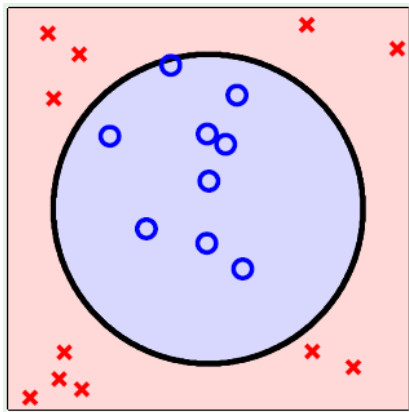
- Degree-Q Polynomial kernel

$$K(\mathbf{u}, \mathbf{v}) = (\gamma \mathbf{u}^T \mathbf{v} + c)^Q.$$

- Gaussian Radial Basis Function (RBF) Kernel

$$K(\mathbf{u}, \mathbf{v}) = \exp \left\{ -\frac{\|\mathbf{u} - \mathbf{v}\|^2}{2\sigma^2} \right\}.$$

SVM with Second Order Kernel

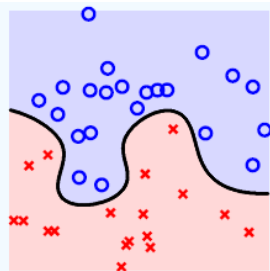


Boxed samples = Support vectors.

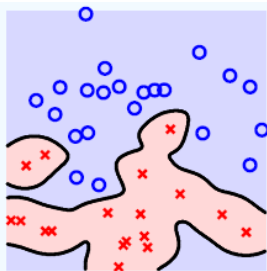
Radial Basis Function

Radial Basis Function takes the form of

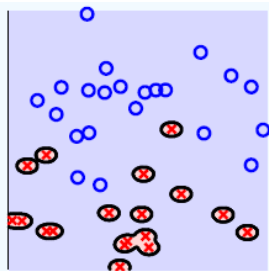
$$K(\mathbf{u}, \mathbf{v}) = \exp \{ -\gamma \|\mathbf{u} - \mathbf{v}\|^2 \}.$$



$$\exp(-1\|\mathbf{x} - \mathbf{x}'\|^2)$$



$$\exp(-10\|\mathbf{x} - \mathbf{x}'\|^2)$$



$$\exp(-100\|\mathbf{x} - \mathbf{x}'\|^2)$$

- Typical $\gamma \in [0, 1]$.
- γ too big: Over-fit.

Non-Linear Transform for RBF?

- Let us consider scalar $u \in \mathbb{R}$.

$$\begin{aligned}K(u, v) &= \exp\{-(u - v)^2\} \\&= \exp\{-u^2\} \exp\{2uv\} \exp\{-v^2\} \\&= \exp\{-u^2\} \left(\sum_{k=0}^{\infty} \frac{2^k u^k v^k}{k!} \right) \exp\{-v^2\} \\&= \exp\{-u^2\} \left(1, \sqrt{\frac{2^1}{1!}} u, \sqrt{\frac{2^2}{2!}} u^2, \sqrt{\frac{2^3}{3!}} u^3, \dots, \right)^T \\&\quad \times \left(1, \sqrt{\frac{2^1}{1!}} v, \sqrt{\frac{2^2}{2!}} v^2, \sqrt{\frac{2^3}{3!}} v^3, \dots, \right) \exp\{-v^2\}\end{aligned}$$

- So Φ is

$$\Phi(x) = \exp\{-x^2\} \left(1, \sqrt{\frac{2^1}{1!}} x, \sqrt{\frac{2^2}{2!}} x^2, \sqrt{\frac{2^3}{3!}} x^3, \dots, \right)$$

So You Need

Example. Radial Basis Function

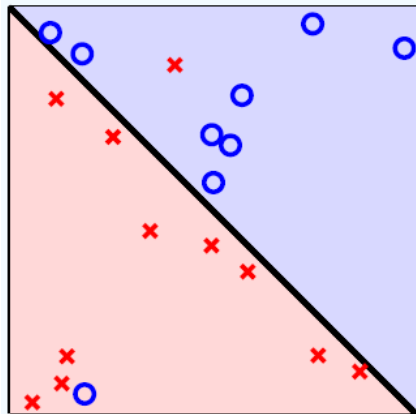
$$K(\mathbf{u}, \mathbf{v}) = \exp \{ -\gamma \|\mathbf{u} - \mathbf{v}\|^2 \}.$$

The non-linear transform is:

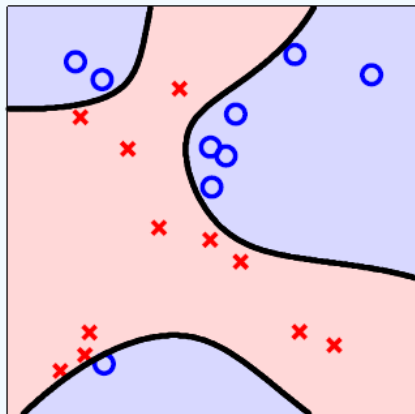
$$\Phi(x) = \exp\{-x^2\} \left(1, \sqrt{\frac{2^1}{1!}}x, \sqrt{\frac{2^2}{2!}}x^2, \sqrt{\frac{2^3}{3!}}x^3, \dots, \right)$$

- You need infinite dimensional non-linear transform!
- But to compute the kernel $K(\mathbf{u}, \mathbf{v})$ you do not need Φ .
- Another Good thing about Dual SVM: You can do infinite dimensional non-linear transform.
- Cost of computing $K(\mathbf{u}, \mathbf{v})$ is bottleneck by $\|\mathbf{u} - \mathbf{v}\|^2$.

Is RBF Always Better than Linear?



(a) linear classifier



(b) Gaussian-RBF kernel

- Noisy dataset: Linear works well.
- RBF: Over fit.

Testing with Kernels

- Recall:

$$\mathbf{w}^* = \sum_{n=1}^N \lambda_n^* y_n \mathbf{x}_n.$$

- The hypothesis function is

$$\begin{aligned} h(\mathbf{x}) &= \text{sign} \left(\mathbf{w}^{*T} \mathbf{x} + w_0^* \right) \\ &= \text{sign} \left(\left(\sum_{n=1}^N \lambda_n^* y_n \mathbf{x}_n \right)^T \mathbf{x} + w_0^* \right) \\ &= \text{sign} \left(\sum_{n=1}^N \lambda_n^* y_n \mathbf{x}_n^T \mathbf{x} + w_0^* \right). \end{aligned}$$

- Now you can replace $\mathbf{x}_n^T \mathbf{x}$ by $K(\mathbf{x}_n, \mathbf{x})$.

Reading List

Support Vector Machine

- Mustafa, *Learning from Data*, e-Chapter
- Duda-Hart-Stork, *Pattern Classification*, Chapter 5.5
- Chris Bishop, *Pattern Recognition*, Chapter 7.1
- UCSD Statistical Learning
<http://www.svcl.ucsd.edu/courses/ece271B-F09/>