Tutorial: Kernel Methods

Chih-Wei Chang B00201037
Chung-Yen Hung B00201015

December 4, 2014

Abstract

The most common machine learning models are usually linear. Linear model possess some advantages such as simple, efficient, and computationally feasible. Real world data analysis problems, on the other hand, are rarely linear — they usually require nonlinear models to describe the sophisticated interdependencies behind the data. One way to solve this problem is mapping features to higher dimensional space. After the mapping, we simply apply linear models in that space, which in turn give nonlinear models in the original lower dimensional space. However, mapping features to higher dimensional space also increase the computational cost, the inner product, for example. The higher the dimension, the harder the computation. Kernel methods, luckily, provide a mathematical shortcut for us to prevent calculating the inner product directly. In other words, kernel methods give us the ability to fit sophisticated models while still have the computational feasibility at the same time.

1 Introduction to Kernel Method

1.1 Feature Transformation

Feature transformation provides us a way to tackle on sophisticated data. Take the quadratic transformation from $\mathbb{R}^2$ to $\mathbb{R}^6$ as an instance, which can be defined by:

$$\Phi : (x_1, x_2) \mapsto (x_1^2, x_2^2, \sqrt{2}x_1x_2, \sqrt{2}x_1, \sqrt{2}x_2, 1).$$

Note that a straight line in the special space actually corresponds to a quadratic curve in the original space $\mathbb{R}^2$. It implies that we can get a nonlinear model in the original space by fitting a linear model in $\mathbb{R}^6$. In this way, any linear model can be extended to a nonlinear model. To be specific, we fit linear model on transformed data $\Phi(\mathbf{x})$ and get a hypothesis $h : \mathbb{R}^6 \to \{0, 1\}$. After that, when new testing data $\mathbf{x}_{test}$ arriving, $h(\Phi(\mathbf{x}_{test}))$ gives the prediction.

1.2 Kernel Trick

While feature transformation seems promising — the model can be extremely sophisticated as long as we make the dimension of the transformed space higher — the price is the expensive computational cost. The computational difficulty also makes the infinity dimensional

transformation impossible because we can't do a infinity dimensional inner product on a computer. Fortunately, kernel methods come to rescue.

Kernel method is a mathematical shortcut to prevent computing inner product directly. For example, consider the quadratic transformation mentioned previously, the inner product of $\Phi(\mathbf{x})$ and $\Phi(\mathbf{y})$ is

$$\begin{aligned}\langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle &= \Phi(\mathbf{x})^T \Phi(\mathbf{y}) \\ &= x_1^2 y_1^2 + x_2^2 y_2^2 + 2x_1 x_2 y_1 y_2 + 2x_1 y_1 + 2x_2 y_2 + 1 \\ &= (\mathbf{x}^T \mathbf{y} + 1)^2 = (\langle \mathbf{x}, \mathbf{x} \rangle + 1)^2\end{aligned}$$

It implies that we can define the higher dimensional space's inner product by the simpler inner product in the original space, which successfully prevents the computational complexity. We hence define the kernel function, we will give a more rigorous definition later, by

$$K(\mathbf{x}_1, \mathbf{x_2}) = \langle \Phi(\mathbf{x_1}), \Phi(\mathbf{x_2}) \rangle$$

As long as we can represent the transformed space inner product by original space inner product, we can have both the sophisticated hypothesis and computational feasibility. We then call this the "Kernel Trick".

2 Common Kernel

2.1 Hilber Space and RKHS

Definition 2.1 (Inner Product Space)**.** An inner product space $\mathcal{X}$ is a vector space with an associated inner product $\langle \cdot, \cdot \rangle : \mathcal{X} \to \mathbb{R}$ that satisfies:

- Symmetry: $\langle \mathbf{x}, \mathbf{y} \rangle = \langle \mathbf{y}, \mathbf{x} \rangle$
- Linearity: $\langle a \cdot \mathbf{x}, \mathbf{y} \rangle = a \cdot \langle \mathbf{x}, \mathbf{y} \rangle$ and $\langle \mathbf{w} + \mathbf{x}, \mathbf{y} \rangle = \langle \mathbf{w}, \mathbf{y} \rangle + \langle \mathbf{x}, \mathbf{y} \rangle$
- Positive Semi-Definiteness: $\langle \mathbf{x}, \mathbf{x} \rangle \geq 0$

The inner product space is strict if $\langle \mathbf{x}, \mathbf{x} \rangle = 0$ iff $\mathbf{x} = 0$

Definition 2.2 (Hilbert Space)**.** A strict inner product space $\mathcal{F}$ is a Hilbert space if it is:

- Complete: Every Cauchy sequence $\{h_i \in \mathcal{F}\}_{i=1}^{\infty}$ converges to an element $h \in \mathcal{F}$
- Separable: There is a countable subset $\hat{\mathcal{F}} = \{h_i \in \mathcal{F}\}_{i=1}^{\infty}$ such that for all $h \in \mathcal{F}$ and $\epsilon > 0$, there exists $h_i \in \hat{\mathcal{F}}$ such that $\|h_i - h\| < \epsilon$.

The interval $[0, 1]$, the reals $\mathbb{R}$, the complex numbers $\mathbb{C}$ and Euclidean spaces $\mathbb{R}^D$ are all Hilber spaces.

2.2 Kernel Function

Just for temporal note: Maybe state and discuss Mercer's condition here.

2.3 Polynomial Kernel

The polynomial kernel is defined as

$$K_{\Phi_d}(x, y) = (\langle x, y \rangle + \alpha)^d$$

As a kernel, K corresponds to an inner product in a feature space based on some mapping:

$$K_{\Phi_d}(x, y) = \langle \Phi_d(x), \Phi_d(y) \rangle$$

Let $d = 2$, so we get the special case of the quadratic kernel

$$K(x, y) = (\sum_{i=1}^{n} x_i y_i + \alpha)^2 = \sum_{i=1}^{n} (x_i^2)(y_i^2) + \sum_{i=2}^{n} \sum_{j=1}^{i-1} (\sqrt{2} x_i x_j)(\sqrt{2} y_i y_j) + \sum_{i=1}^{n} (\sqrt{2c} x_i)(\sqrt{2c} y_i)$$

From this it follows that the feature map is given by:

$$\Phi_2(x) = (x_n^2, \ldots, x_1^2, \sqrt{2} x_n x_{n-1}, \ldots, \sqrt{2} x_n x_1, \sqrt{2} x_{n-1} x_{n-2}, \ldots, \sqrt{2c} x_n, \ldots, \sqrt{2c} x_1, c)$$

2.4 Gaussian Kernel

3 Kernel Machines

3.1 Kernel PCA

3.2 Kernel SVM

3.3 Kerenl Ridge Regression

3.4 Kernel Logistic Regression

4 Output Kernel

4.1 Kernel in Output Space

4.2 Principle Label Space Transform

Just for temporal reference: Feature-aware Label Space Dimension Reduction for Multi-label Classification (2012) YN Chen and HT Lin.

4.3 Conditional Principle Label Space Transform

4.4 Other Possible Output Kernel Techniques

5 Conclusion

Here comes the Conclusion