Week2 Report

Zhengyi Wang
wangzhen17@mails.tsinghua.edu.cn

Abstract

This week, we talked about the fundemental of machine learning. Our talk is basically on support vector machine (SVM).

1 Optimization Goal

What is the goal of SVM?
SVM is used for classification problem. (Two classes, precisely.) Given a training set $S = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$ and $x_i \in R^m$, $i = 1, 2, ..., n$. The goal of SVM is to learn a function f(x) to help with the classification process. If f(x)>0 then the item will be classified as class A and will be classified as class B when f(x)<0.

2 Basic Terminology

2.1 Functional Margin

The functional margin of a item is defined as followed,

$$\hat{\gamma}^{(i)} = y^{(i)}(w^T x^{(i)} + b). \tag{1}$$

In which, $y^{(i)} \in \{-1, 1\}$ means the data label of the ith item.

2.2 Geometric Margin

Given geometric margin, we can easily derive the geometric margin. It's a normalized format of functional margin. Geometric margin is commonly used because it described the Euclid's distance from the data point to the hyper-plain.

$$\gamma^{(i)} = y^{(i)}(\frac{w}{||w||})^T x^{(i)} + \frac{b}{||w||}). \tag{2}$$

And the **margin of the training set** is defined as followed,

$$\gamma = \min_{i=1,2,...,n} \gamma^{(i)}. \tag{3}$$

To make the SVM more accurate in predicting the label, the margin of the training set has to be as big as possible, which means our optimization goal is to find out

$$\max_{\gamma,w,b} \gamma \quad \text{s.t.}\, y^{(i)}(w^T x^{(i)} + b) \geq \gamma\, (i = 1, 2, ...) \,\text{and}\, ||w|| = 1 \tag{4}$$

Which can be easily transformed into,

$$\max_{w,b} \frac{1}{||w||} \quad \text{s.t.}\, y^{(i)}(w^T x^{(i)} + b) \geq 1\, (i = 1, 2, ...) \tag{5}$$

Reading Group on Machine Learning (Apr 14th, 2019).

2.3 Kernel Function

The SVM mentioned above can only classify the items that are divided by a hyper-plain. However, in most of the cases, items do not always distribute like that. So we introduce the **feature space mapping**, to map the items into another space in which they can be divided by a hyper-plain. 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.

The feature space mapping maps x into another space,

$$x \to \Psi(x), \tag{6}$$

And the **kernel function** is the inner product of to vector x_1, x_2 in the higher-dimension space,

$$K(x_1, x_2) = \Psi(x_1)^T \Psi(x_2), \tag{7}$$

which describe the distance between x_1, x_2 in the mapped space.

In practice, we can directly use the kernel function to calculate the distance of two items in the feature space without even know the format of feature space mapping function. Just replace all the inner product in the above function with kernel function.

One of the most commonly used kernel function is listed below,

$$K(x, z) = e^{-\frac{||x-z||^2}{2\sigma^2}}. \tag{8}$$