MATH 748: Weekly Report

Nov 29 - Dec 5, 2021

Nayeong Kim (nkim10@sfsu.edu)

16 Support Vector Machines

16.1 Introduction

Support Vector Machine(denoting SVM) is an approach for classification. It is widely used because its practical performance is quite nice. It comes from an idea to overcome a problem in linear models: (i) too sensitive to individual data (ii) not allowable for mislabeled data (iii) data distributed non-linearly.
Let us just consider the 2 classes problem. (We can apply this to multi-class problem with inductive application of two classes problem.) The key idea is that we will find a hyper plane that (almost) separates two classes in the feature space. By allowing some mistaken labeled data, we can soften the result to classify non-separable cases.
Hyper plane in n-dimension is an $n-1$ object which divide the whole space into two. For example, when $n=2$, a straight line can separate the 2-dimensional plane. A hyper plane in p dimension can be expressed as $\beta_0+\beta_1x_1+\cdots\beta_px_p=0$ which is an equation that a linear combination of $1,x_1,\cdots,x_p$ is equal to 0. We can put a point to the equation and classify it with the sign of the result. I.e. if $\beta_0+\beta_1x_1+\cdots\beta_px_p>0$, it is classified to class 1. Otherwise, if $\beta_0+\beta_1x_1+\cdots\beta_px_p<0$, it is classified to class 2. (We can ignore the case that $\beta_0+\beta_1x_1+\cdots\beta_px_p=0$ because it can be happen in a zero probability.

16.2 Maximal Margin Classifier

Maximal margin classifier is the simplest SVM. When we divide the feature space with training data, there are multiple choices for the boundaries which divide the classes. Maximal margin classifier maximizes the distance of the closes points to the boundary. We call the minimal distance between the observed points to the hyper plane as *margin* (hence maximal margin classifier maximizes margin). *Support vectors* are training observations on the margin plane while *margin plane* means that the plane crossing through the closest observation of the hyper plane which is also parallel with the hyper plane.
This can be expressed as an optimization problem that maximizes M for $\beta_0,\cdots,\beta_p$ having two constraints:

- $||\beta||_2=1$
- $y_i(\beta_0+\cdots\beta_px_{ip}\geq M$ for all $i=1,\cdots,n$

The first constraint is normalizing the model. When we think of the equation, it doesn't matter to multiply some constants. The second constraint implies that all the sample points are correctly classified even with M margin (when M is positive). With the two constraints, we have to choose $\beta_0,\cdots,\beta_p$ which maximize the margin.

16.3 Soft Margin Classifier

In practice, we have some non-separable data set with a hyper plane. It can happen because of some mistakes in labeling. Or it can be a non-linear cases. Maximal margin classifier doesn't work for the cases. Also, it has

a problem that it is too sensitive to individual points like outliers. Hence we need to soften the constraints to the optimization problem to allow some mis-classified points. Now, we will discuss the model considering two things:

- Robust to individual observations.
- Better classification of most (not all) training observations by allowing some violations.

There are two types of violations. One is that some points are on the incorrect side of margin. For example, some points are not mis-classified but they're between two margin planes. The other is misclassification. Considering both cases, we can define the new optimization problem which maximizes M for $\beta_0, \cdots, \beta_p$ having two constraints:

- $||\beta||_2 = 1$
- $y_i(\beta_0 + \cdots \beta_p x_{ip} \geq M(1-\varepsilon_i)$ for all $i = 1, \cdots, n$
- $\varepsilon_i \geq 0, \sum_{i=1}^{n} \varepsilon_i \leq C$

Let us call this *Soft Margin Classifier*. In soft margin classifier, C which is called as *violation budget* is a nonnegative parameter which can be tuned. For example, $C = 0$ implies that all the ε_i is zero. Hence there is no violation. For another example, $C = 10$ implies that at most 10 violations occur.
Also, $\varepsilon_1, \cdots, \varepsilon_n$ are called *slack variables* which allow violations. When ε_i is zero, there are no violation. If ε_i is in $(0,1)$, it violate margin but not mis-classified. If ε_i is greater than 1, it is mis-classified which is the most severe violation. With the optimized β, we can generate a hyper plane for classification.
In soft margin classifier, only the observations which lie on the margin plane or the observations violating the margin or hyperplane affect the position of the hyperplane. These are called *support vectors*. The tuning parameter C decides how many support vectors are included. Hence the classifier is robust to the behavior of observations which are far away from the hyper planes which is different from LDA.
Also, the parameter C controls the bias-variance trade-off. When C is large,it

- allows more violation of margin.
- has more support vectors.
- has wider margin.
- makes more observations be involved in determining the hyperplane.
- has small variance and high bias.

Therefore C acts like a regularization parameter.

16.4 Soft Vector Machine

When the dataset is distributed in non-linear way, linear classifiers fail. There are some technique to classify non-linearly distributed dataset. One idea is to *enlarge* the feature space. One way to do it is using *Kernel Trick*.

(a) Feature Extension
We can extend the feature space using existing features. For example, we can get new features with the

power of the original ones. I.e. $(x_1, \cdots, x_p) \mapsto (x_1, x_1^2, \cdots, x_p, x_p^2, x_p^3$. The extension of support vector classifier is called *support vector machine*.

(b) Kernel Trick
There are some advantages to use kernel. We can make a high dimensional transformation without extra computation for new features. Hence, it is efficient in computation.
Each kernel is corresponding to a feature space. For example, there is a linear kernel: $K(x_i, x_i') = \sum x_{i,j} x_{i'j}$. For another exmaple, there is a polynomial kernel: $K(x_i, x_i') = p(x_i, x_i')$ where p is a polynomial having degree d.