CSE/ECE-471: Statistical Methods in AI **Spring 2020**

Support Vector Machines

Prepared by: 2018122006; 2019201085;2019900077

Suppose we have to deal with linearly separable data, we all know that a plane which 'just' classifies the data would not suffice and intuitively there is a higher chance that a test data sample get miss-classified. So we tend to look forward for a separating hyper plane that is far from all the samples. In short, we want to find a separating hyper plane that maximizes the margin. That is what support vector machines(SVM) are.SVMs are very popular classifiers even today.They have good theoretical properties.

1 Theory

Let's take a quick look at the intuition the we developed in the previous lectures.We know that the distance from the origin to line ax+by+c=0 is $\frac{c}{\sqrt{a^2+b^2}}$ In a similar manner we can see that distance from origin to $W^T + b = 1$ is $\frac{c(1-b)}{\|w\|}$ Or the distance between the two side planes is $\frac{2}{\|w\|}$.Thus our objective is to maximize the margin or maximize $\frac{1/2}{\|w\|}$ or minimize $\frac{W^T W}{2}$. so minimization of this quantity under no constraint conditions lead the previously mentioned quantity to zero which we don't want. In order to classify the samples, we need to add some constraints that say whether a sample is correctly classified or not.

- For $y_i = 1$ we want the samples to be $W^T x + b = +1$;
- For $y_i = -1$ we want the samples to be $W^T x + b = -1$;
- We can combine these two constraints into one by multiplying y_i on both sides.(Note that when$y_i = 1$ the inequality sign also reverses.i.e., $y_i(W^T x_i + b) >= 1 \forall i$

1.1 Primal problem

The SVM problem is therefore

- Minimize $\frac{W^T W}{2}$
- such that $y_i(W^T x_i + b) >= 1 \quad \forall i; \quad y_i = [-1, 1]$

1.2 Solution

We can solve the above problem in many ways but however the dual of this problem is popular and also the above equation is convex so the duality gap will be zero, so we can arrive at the solutions to the dual problem instead of the primal problem.So let us the dual problem equation

$$\sum_{i=1}^{\mathbb{N}} \alpha_i - 1/2 \sum_{i=1}^{\mathbb{N}} \sum_{j=1}^{\mathbb{N}} \alpha_i \alpha_j y_i y_j X_i^T X_i \qquad (1)$$

$$\sum_{i=1}^{\mathbb{N}} \alpha_i y_i = 0$$

$$\alpha_i \geq 0$$

here α_is are the Lagrangian multipliers. Let us also have a quick look how did we arrive at the dual problem from the primal.Here it is more of a simple mathematical exercise of how to rewrite the objectives from primal to dual. We start with our primal objective of Converting the constrained problem to unconstrained problem. We have to minimise

$$J(\alpha, b, W) = \frac{W^T W}{2} - \sum_{i=1}^{\mathbb{N}} \alpha_i [y_i (W^T X_i + b) - 1] \tag{1}$$

here $\alpha_i >= 0$ are the Lagrangian multipliers. The optimal conditions are:

$$\frac{\partial J(\alpha, b, W)}{\partial W} = 0 \quad ; \quad \frac{\partial J(\alpha, b, W)}{\partial b} = 0$$

$$\text{by solving we get,} \quad W = \sum_{i=1}^{\mathbb{N}} \alpha_i y_i X_i$$

$$\sum_{i=1}^{\mathbb{N}} \alpha_i y_i = 0$$

find the optimal values of which can give the optimal values of $J(\alpha, b, W)$,

$$J(\alpha, b, W) = \frac{W^T W}{2} - \sum_{i=1}^{\mathbb{N}} \alpha_i [y_i (W^T X_i + b) - 1]$$

$$J(\alpha, b, W) = \frac{W^T W}{2} - \sum_{i=1}^{\mathbb{N}} \alpha_i y_i X_i W^T - b \sum_{i=1}^{\mathbb{N}} \alpha_i y_i + \sum_{i=1}^{\mathbb{N}} \alpha_i = 0$$

by substituting the optimal conditions we get,

$$W^T W = \sum_{i=1}^{\mathbb{N}} \alpha_i y_i X_i W^T = \sum_{i=1}^{\mathbb{N}} \sum_{j=1}^{\mathbb{N}} \alpha_i \alpha_j y_i y_j X_i^T X_j$$

we get the function that has to be maximized,

$$J(\alpha) = \sum_{i=1}^{\mathbb{N}} \alpha_i - 1/2 \sum_{i=1}^{\mathbb{N}} \sum_{j=1}^{\mathbb{N}} \alpha_i \alpha_j y_i y_j X_i^T X_j$$

So, the maxima of $J(\alpha)$ is equal to the minima of $J(\alpha, b, W)$. To understand this, let us consider a situation where we have to minimize f(x) such that $g(x) >= 0$
The corresponding Lagrangian equation in the above situation would be: $L(x, \lambda) = f(x) - \lambda^T g(x)$

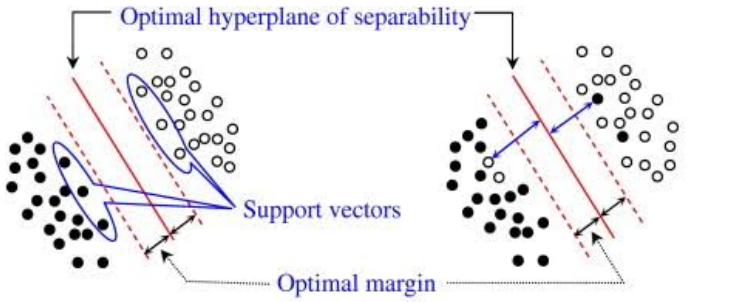
- $max_{\lambda \geq 0} L = \begin{cases} \inf & \text{if } \boldsymbol{g}(x) \leq 0 \\ \boldsymbol{f}(x) & \text{otherwise} \end{cases}$
- Primal Problem:$min_x max_{\lambda \geq 0} L(x, \lambda)$
- Dual Problem: $max_{\lambda \geq 0} min_x L(x, \lambda)$

Therefore the primal problem of minimization over x became a maximization problem over the Lagrangians λ

2 Adding tolerance or Soft margin

Before learning about tolerance or soft margin, If we had removed some of the training samples, those are away from the side planes, the solution will not change because the solution depends only on the samples that are hard

to classify i.e., the samples on the side planes.When we solve the dual problem,the αs are the only some samples have impact on the final solution. Support vectors are the ones where α_i is non zero. At the test time, we just need to test the sign of $W^T X + b$ and decide whether it is positive or negative class. Therefore we decide them as $sign(\sum_{i=1}^{N} \alpha_i y_i X_i^T X + b)$.

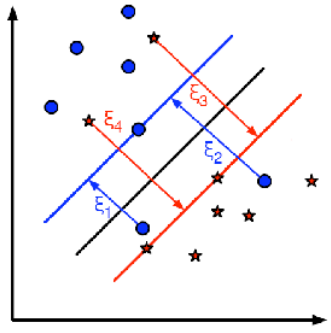


(a) Linear separable case (b) Non-linear separable case

In most of the real world applications, we often come across non separable data. In order to find a way, we relax the constraints a bit and introduce some tolerance.So the new conditions are,

- For $y_i = 1$ we want the samples to be $W^T x + b = +1 - \varepsilon_i \quad \forall i$;
- For $y_i = -1 + \varepsilon_i$
- We can combine these two constraints into one by multiplying y_i on both sides.(Note that when$y_i = 1$ the inequality sign also reverses.i.e $y_i(W^T x_i + b) >= 1 - \varepsilon_i \quad \forall i$

We can see that in the below figure



(1).png (1).bb

Indeed if $\varepsilon_i s$ are all zero, we will have our previous SVM equation that we saw already. Our new problem of interest is now to minimize both WW^T and $\sum_{i=1}^{N} \quad \varepsilon_i$. There are two quantities to simultaneously minimize. We balance the relative importance of these two terms with a non negative constant C. Our problem is now,

$$\begin{aligned} & minimize \quad C\sum_{i=1}^{\mathbb{N}} \varepsilon_i + \frac{W^T W}{2} \\ \text{such that} \quad & y_i(W^T x_i + b) >= 1 - \varepsilon_i \qquad \forall i \end{aligned} \tag{2}$$

If C is too small (say zero), we are easily allowing violations. i.e., the algorithm will look for large margin but discard the concern of violations in the separability. If C is too large, then violations are taken too seriously and not the margin. The parameter C is one that one may have to set in the SVM implementations.

2.1 Dual problem

We derive the dual problem for soft margin just like in the case of regular SVM that we did earlier

$$\mathbb{Q}(\alpha) = \sum_{i=1}^{\mathbb{N}} \alpha_i - 1/2 \sum_{i=1}^{\mathbb{N}} \sum_{j=1}^{\mathbb{N}} \alpha_i \alpha_j y_i y_j X_i^T X_i$$

$$\sum_{i=1}^{\mathbb{N}} \alpha_i y_i = 0$$

$$\alpha_i \geq 0 \quad ; \alpha_i \quad \leq C$$

3 Multi-class Classification

All this time, we have been looking at binary classifier. What about the data with multiple classes? we have to cultivate some strategies to deal with this problem. Here are some strategies that can be followed.
Let us consider K classes

- We can build K one Vs rest classifiers
- We can build $\binom{k}{2}$ pairwised (i vs j) classifiers

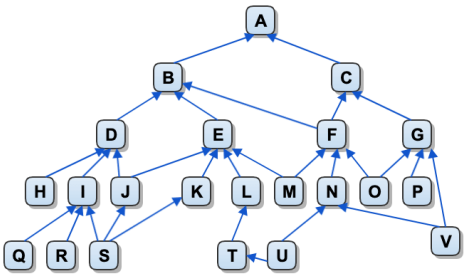
In fact we can many (find the combinatorics!) classifier that separates samples from a set or classes from that of another set of classes

3.1 One vs Rest

If the K one vs rest classifiers can provide probabilities,life is simple. Pick the class with highest probability.If probabilities are not available (and only class label is predicted), then we may have two cases to worry. All classifiers says rest. or multiple classifiers assign a class label. In either case, the final decision is difficult without some additional information

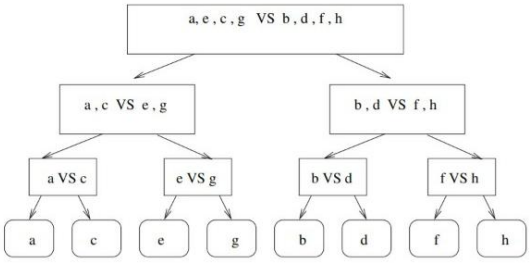
3.2 Pair wise classifiers

When there are $\binom{k}{2}$ classes, one can use simple majority voting to find the best classifier. One can also use DAG (directed acyclic graph) like structure as shown below.



3.3 Multi-class Classification

We achieve multi-class classification by hierarchically arranging a number of classifiers of type 3. The advantage is that we can achieve the final class in log() evaluations. However, the challenge is in designing. How many classes need to be trained? How they should be arranged? Popularly this type of classifiers are called Binary Hierarchical Classifier (BHC).



The above figure is an example of BHC.

4 SVM for Non linear Data

Consider a 2D pattern of two concentric circles.Inner circle is class 1 and outer circle from class 2. How to approach this problem?. The idea is that with an appropriate feature transformation the problem becomes simpler and the classifier/algorithm can do superior.

Consider a feature transformation (or feature map) as Let us start with a specific example: Let$p = [p1, p2]^T$ and $\theta(p)$ be $[p_1^2, p_2^2, \sqrt{2}p_1p_2]$. You may wonder how this helps us or why we do this.We will see the utility as we move forward. Note the notations. Here the sample p has two features/dimensions p1 and p2. A number of algorithms in machine learning use dot product as the basic operation. You already had seen W^TX. The beauty of dot product is that the result is scalar independent of the dimensionality of the vector. i.e., the dot product of two vectors in 2D and 3D will be still a scalar, independent of the dimension. Let us now compute the dot product of $\theta(p)$ and $\theta(q)$. Here q is also a 2D vector similar to p. i.e., $q = [q_1, q_2]^T$. We know that $p^Tq = p_1q_1 + p_2q_2$. Let us compute the dot/scalar/inner product in the new feature space.

$$\begin{aligned} \theta(p)^T\theta(q) &= [p_1^2, p_2^2, \sqrt{2}p_1p_2]^T[q_1^2, q_2^2, \sqrt{2}q_1q_2] \\ \theta(p)^T\theta(q) = \quad & p_1^2q_1^2 + p_2^2q_2^2 + 2p_1q_1p_2q_2 = (p_1q_1 + p_2q_2)^2 \\ \theta(p)^T\theta(q) = \quad & (p^Tq)^2 = \kappa(p, q) \end{aligned} \tag{3}$$

What it says is something simple. If we want to compute the dot products of $\theta(p)^T$ and $\theta(q)$, what we need to do is only computing dot product of p and q and then square it. (indeed, that is true only for this specific $\theta()$) Note that if we compute $\theta(p)^T\theta(q)$ like this, we do not have to really compute $\theta()$ explicitly. That could be a huge advantage in many situations. Later today, we also would like to map p into infinite dimension with a $\theta()$.The advantage of not requiring to compute $\theta()$ will be a big advantage then.The function $\kappa()$ is a kernel function. We saw the kernel function of $(p^Tq)^2$. This need not be square. It could be $(p^Tq)d$. This is a polynomial kernel. With some effort we can write the $\theta()$ corresponding to this kernel. There are many other potential kernels. A kernel functions allows us to compute an inner/dot product in a new feature space.
Let us consider another $\theta()$ for the same $p = [p1, p2]^T$.Let $\theta(p)$ as $[p_1^2, p_2^2, p_1p_2, p_2p_1]$. Here $\theta()$ maps from 2D to

4D. (instead of 2 to 3 as in the previous example).

$$\begin{aligned} \theta(p)^T\theta(q) &= [p_1^2, p_2^2, p_1p_2, p_1p_2]^T[q_1^2, q_2^2, q_1q_2, q_1q_2] \\ \theta(p)^T\theta(q) &= p_1^2q_1^2 + p_2^2q_2^2 + 2p_1q_1p_2q_2 = (p_1q_1 + p_2q_2)^2 \\ \theta(p)^T\theta(q) &= (p^Tq)^2 = \kappa(p,q) \end{aligned} \tag{4}$$

4.1 SVM

The notion of Kernels is very nicely related to SVMs.One may wonder whether SVMs are designed for Kernels or Kernels are designed for SVMs!!.
Let us now come back to our SVM problem (dual)

$$maximize J(\alpha) = \sum_{i=1}^{\mathbb{N}} \alpha_i - 1/2 \sum_{i=1}^{\mathbb{N}} \sum_{j=1}^{\mathbb{N}} \alpha_i\alpha_j y_i y_j X_i^T X_i \qquad (1)$$

$$\sum_{i=1}^{\mathbb{N}} \alpha_i y_i = 0$$

$$\alpha_i \geq 0$$

Assume the input data (x_i, y_i) was mapped by$\theta()$,leading to $\theta(x_i, y_i)$. We can find a linear boundary in the new feature space. And the corresponding decision boundary in the original space is going to be a nonlinear one.
This makes the SVM problem (with appropriate constraints) as

$$max_\alpha J(\alpha) = \sum_{i=1}^{\mathbb{N}} \alpha_i - 1/2 \sum_{i=1}^{\mathbb{N}} \sum_{j=1}^{\mathbb{N}} \alpha_i\alpha_j y_i y_j \theta(X_i^T)\theta(X_j)$$

$$max_\alpha J(\alpha) = \sum_{i=1}^{\mathbb{N}} \alpha_i - 1/2 \sum_{i=1}^{\mathbb{N}} \sum_{j=1}^{\mathbb{N}} \alpha_i\alpha_j y_i y_j \theta(X_i, X_j)$$

Many practical solvers use the precomputed NXN kernel matrix. This avoids repeated computation of kernel functions. But this necessitates the need to store and manipulate a NXN matrix while solving. Not very nice for big data sets.
When you solve this nonlinear SVM problem, we get α_i corresponding to the new nonlinear SVM. Or what we get is the nonlinear SVM in the original feature space. Typically the output of the training is a set of α_i (that are non zero). At the test time, we only need to evaluate the sign of $W^T\theta(x) + b$

4.2 Popular Kernels

- Linear Kernel : $\kappa(x, y) = (x \cdot y)$
- Polynomial Kernel : $\kappa(x, y) = (x \cdot y)^d$
- Sigmoid Kernel : $\kappa(x, y) = \tanh \gamma(x \cdot y)$
- Radial Basis Kernel : $\kappa(x, y) = \exp \gamma|(x - y)\|^d$

4.3 Valid kernels

We also saw a number of exponential kernels. Many of them also correspond to infinite dimensional feature maps $\theta()$.Will every kernels function have a corresponding feature map? or what is a valid kernel? When Kernel matrix K is PSD, the corresponding kernel is valid. This comes positive semi-definte from the Mercers theorem.