CSE 475: Statistical Methods in AI **Monsoon 2019**

SMAI-M-2019 14: Support Vector Machines

Lecturer: C. V. Jawahar *Date: 26, Sep 2019*

14.86 Maximization of Margin

We now know linear classifiers such as perceptron and logistic regression. We know that if the data is linearly separable, perceptron algorithm will converge with a feasible solution. i..e, a separating hyper plane. However, we know that not all separating hyper planes are equally useful. For example, a plane that "just" classifies a training sample is not the best. Intuitively this leaves higher chance for a test sample (even if it is somewhat similar to the training one) to get misclassified. Conceptually we prefer a separating hyper plane that is far from all the samples.

In short, we want to find a separating hyperplane that maximimzes the margin. That is what support vector machines (SVM) are. SVMs are very popular classifiers even today. They have many nice theoretical properties. The optimization problem is convex and that is a special advantage.

(A figure missing) We know from the mid school mathematics that the distance from origin to the 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 $\mathbf{w}^T\mathbf{x} + b = 1$ is $\frac{1-b}{||\mathbf{w}||}$. Similarly the distance form origin to the $\mathbf{w}^T\mathbf{x} + b = -1$ is $\frac{-1-b}{||\mathbf{w}||}$. Or the distace between the two side planes is $\frac{2}{||\mathbf{w}||}$.

Thus our objective is to maximize the margin or maximize $\frac{1}{||\mathbf{w}||}$ or minimize $\frac{1}{2}\mathbf{w}^T\mathbf{w}$.

Indeed the unconstrained minimization of this could lead to $\mathbf{w}$ becoming zero. That is not useful. Also this problem does not say anything about the samples correctly classified. We need to add constraints that says that the samples are correctly classified.

- When $y_i = +1$ we would like the samples to be $\mathbf{w}^T\mathbf{x} + b \geq +1$
- When $y_i = -1$ we would like the samples to be $\mathbf{w}^T\mathbf{x} + b \leq -1$.
- We can combine these two constraints into one by multiplying y_i on both side. (Note that when y_i is -1 the inequality sign also reverses. i.e.,

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \quad \forall i$$

14.86.1 Primal Problem

The SVM problem is therefore

$$minimize \ \frac{1}{2}\mathbf{w}^T\mathbf{w}$$

such that

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \quad \forall i$$
$$y_i \in \{-1, +1\}$$

14.87 Solution

The primal problem of interest is

$$minimize \frac{1}{2}\mathbf{w}^T\mathbf{w}$$

such that

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \quad \forall i$$
$$y_i \in \{-1, +1\}$$

This problem can be solved in many ways. We could even use gradient descent to solve this. Being convex in problem, we will obtain the optimal solutions with this. (You could read the paper: Shai Shalev-Shwartz "Pegasos: Primal Estimated sub-GrAdient SOlver for SVM" (though analysis could be hard, initial sections could be OK to read/follow, with some background in optimization. More over, you can write (or download) 20 line matlab or similar code and implement svms!!)

14.87.1 Dual Problem

However, the popular problem is a dual version of the same. (since problem is convex, the optima of the primal and dual will be the same or duality gap will be zero.). With no derivation, let us write the dual problem as

$$maximize \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j \qquad (14.28)$$

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

with $\alpha_i \geq 0$.

Here α_i are the Lagrangian multipliers. (though popular notation for Lagrangian multipliers is λ, SVMs use α.)

How did we get this dual problem? A brief explanation is given at the end of this lecture.

Dual problem is a classical quadratic programming problem. Many optimization libraries could help in this regard. Let us not aim for writing our own code for this at this stage.

The related to $\mathbf{w}$ is:

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i$$

Q: How do you find b?

14.88 Interpretations

If we had removed some of the training samples, those are away from the side planes, the solution will not change. (why?) 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 sparse. i.e., only some samples have impact on the final solution.

Support Vectors Support vectors are the ones where α_i is non zero.

At the test time, we just need to test the sign of $\mathbf{w}^T\mathbf{x}+b$ and decide whether it is positive or negative class. i.e., decide as

$$sign(\sum_{i=1}^{N} \alpha_i y_i \mathbf{x}_i^T \mathbf{x} + b) \tag{14.29}$$

Dot Products Everywhere Another important thing to note is that the samples appear only as dot product in both training (the optimization problem equation 14.28) and the testing (equation 14.29). This is very important and we exploit this smartly when we extend the linear SVMs to nonlinear SVMs using Kernels.

Number of SVs . Assume we do a leave one out testing (LOO). When we leave a non-SV sample and test on it, they all will be correctly classified. Zero error. At the same time if we leave a SVs, and train the solution (i.e., $\mathbf{w}, b$) could change leading to an error. Therefore an upper bound on the error is

$$\frac{\#SV}{N}$$

14.89 Soft Margin SVMs

We made a strong assumption that the samples are linearly separable. That is too restrictive in practice. Let us relax that by allowing a penalty ξ_i is the constraint is violated.

- When $y_i = +1$ we would like the sameples to be $\mathbf{w}^T\mathbf{x} \geq +1 - \xi_i$
- When $y_i = -1$ we would like the sameples to be $\mathbf{w}^T\mathbf{x} \leq -1 + \xi_i$.
- We can combine these two constraints into one by multiplying y_i on both side. (like for hard margin SVM) i.e.,

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i \quad \forall i$$

Indeed if ξ_is are all zero, we will have our hard margin SVM that we saw already. Our new problem of interest is now to minimize both $\mathbf{w}^T\mathbf{w}$ and $\sum_{i=1}^{N} \xi_i$. 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

$$minimize \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{N} \xi_i$$

such that

$$y_i(\mathbf{w}^T\mathbf{x}_i) \geq 1 - \xi_i \quad \forall i$$

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.

Implementation We know how to implement many of the algorithms that we studied. However, SVMs are not that easy in practice. There are nice implementations like libsvm, liblinear etc. and most of the popular libraries have very good implementations.

Roughly this is what happens:

- Input $(\mathbf{x}_i, y_i)$ for $i = 1, \ldots, N$.
- Solve a quadratic optimization problem, i.e., the dual problem of SVM. Return non-zero α_i or the lagrangians corresponding to the support vectors.
- Given a test sample, compute the class label as $sign(\sum_{i \in SV} \alpha_i y_i \mathbf{x_i}^T \mathbf{x}) + b$.

There are also nice gradient descent solvers for SVMs (read pegasos algorithm, which is also extended for nonliner and softmargin).

14.90 Variations

We already saw the hard margin SVMs and soft margin SVMs. In the first case, we insisted that the sameples should be linearly separable. While in the second we allowed some violations (eg. some outliers or erroneous labels). There are other variations also.

For example the penalty term is L1 or L2 norm of ξ_i.

$$L1 : C\sum_{i=1}^{N}\xi_i$$

$$L2 : C\sum_{i=1}^{N}\xi_i^2$$

(With no technical explanations), minimization of specific norms leads to a solution that is sparse. i.e., we allow some violations but only smaller number of samples are allowed to violate. It is easy to see for $L0$ norm. But that is not what is used in practice.

Q: Derive the dual problem for softmargin L1 and L2 SVMs.

14.91 Primal to Dual

Before we end this lecture, let us also have a quick look how did we arrive at the dual problem from the primal. Some amount of understanding of primal and dual problems in optimization is needed to appreciate this fully (specially to know how the minimization problem became a maximization problem). 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

$$J(\mathbf{w},b,\alpha)=\frac{1}{2}\mathbf{w}^T\mathbf{w}-\sum_{i=1}^{N}\alpha_i\left[y_i(\mathbf{w^T x_i}+b)-1\right]$$

where $\alpha_i \geq 0$ are the nonnegative Lagrangian multipliers. The optimality conditions are:

$$\frac{\partial J(\mathbf{w},b,\alpha)}{\partial \mathbf{w}}=\mathbf{0} \text{ and } \frac{\partial J(\mathbf{w},b,\alpha)}{\partial \mathbf{b}}=\mathbf{0}$$

The optimality conditions $\frac{\partial J(\mathbf{w},b,\alpha)}{\partial \mathbf{w}}=\mathbf{0}$ and $\frac{\partial J(\mathbf{w},b,\alpha)}{\partial \mathbf{b}}=\mathbf{0}$ leads to

$$\mathbf{w}=\sum_{i=1}^{N}\alpha_i y_i \mathbf{x}_i$$

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

To find the optimal values of α which can give the optimal values of $J(\cdot)$,

$$\begin{aligned} J(\mathbf{w},b,\alpha) &= \frac{1}{2}\mathbf{w}^T\mathbf{w}-\sum_{i=1}^{N}\alpha_i\left[y_i(\mathbf{w^T x_i}+b)-1\right] \\ &= \frac{1}{2}\mathbf{w}^T\mathbf{w}-\sum_{i=1}^{N}\alpha_i y_i \mathbf{w^T x_i} - b\sum_{i=1}^{N}\alpha_i y_i + \sum_{i=1}^{N}\alpha_i \end{aligned}$$

The third term of the above objective function is zero and

$$\mathbf{w^T w}=\sum_{i=1}^{N}\alpha_i y_i \mathbf{w^T x_i}=\sum_{i=1}^{N}\sum_{i=1}^{N}\alpha_i\alpha_j y_i y_j \mathbf{x_i^T x_j},$$

Optimal Hyperplane: Solution(Cont.)

The objective function $J_d(\alpha)$ to be maximised becomes

$$J_d(\alpha)=\sum_{i=1}^{N}\alpha_i-\frac{1}{2}\sum_{i=1}^{N}\sum_{i=1}^{N}\alpha_i\alpha_j y_i y_j \mathbf{x_i^T x_j}$$

Thus find maxima of $J_d(\alpha)$ subject to $\sum_{i=1}^{N}\alpha_i y_i = 0$ and $\alpha_i \geq 0$.

14.91.1 Discussions(*)

Minima of $J(w,b,\alpha)$ is same as Maxima of $J_d(\alpha)$. Why?

A small detour and explanation:**Primal Vs Dual** . You may want to read appropriate material from the optimization literature to appreciate this fully.

Consider a problem of minimizing $f(x)$ such that $\mathbf{g(x)} \geq \mathbf{0}$.

The corresponding lagrangian function is

$$L(\mathbf{x},\lambda)=f(\mathbf{x})-\lambda^{\mathbf{T}}\mathbf{g(x)}$$

Now,

$$\max_{\lambda\geq 0} L(\mathbf{x},\lambda)=\begin{cases}\infty & \text{if } g(x)<0\\ f(x) & \text{otherwise}\end{cases}$$

$$\textbf{Primal Problem: } \min_{x}\max_{\lambda\geq\mathbf{0}} L(\mathbf{x},\lambda)$$

$$\textbf{Dual Problem: } \max_{\lambda\geq\mathbf{0}}\min_{x} L(\mathbf{x},\lambda)$$

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

a detailed intro to primal and dual problems in optimization is beyond the scope of this course. Please read on Internet, if interested in more details.