Stochastic gradient descent for SVM

Lluís Garrido – lluis.garrido@ub.edu

December 2018

Abstract

This laboratory is focused on stochastic subgradietn methods in the support vector machines.

1 Support Vector Machines

In the previous lab, lab 5, we have formulated pattern classification using linear functions to provide the characterization of a sample. We recall here the main aspects of the formulation just as a reminder. Suppose we have a set of m training data, $x_i \in R^n$, with classification y_i, where either $y_i = 1$ or $y_i = -1$ (the data point has a certain property or not), see Figure 1 on the left. Suppose it is possible to find some hyperplane $\mathbf{w}^T\mathbf{x} + b = 0$, $\mathbf{w} \in R^n$ and $b \in R$, which separates the positive points from the negative. The problem is to determine the coefficients $\mathbf{w}$ and b that solve

$$\begin{array}{ll} \text{mimize} & f(\mathbf{w}, b) = \frac{1}{2}\mathbf{w}^T\boldsymbol{w} \\ \text{subject to} & y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 \qquad i = 1 \dots m \end{array}$$

Once the coefficients $\mathbf{w}$ and b of the separating hyperplane are found from the training data, we can use the value of the function $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$ (our "learning machine") to predict whether a new point $\bar{\mathbf{x}}$ has the property of interest or not, depending on the sign of $f(\bar{\mathbf{x}})$. The previous formulation assumes that the data is separable, that is, there exists a hyperplane separating the positive points from the negative points exists. In the previous lab, we presented a formulation that allowed to transform the problem to the dual space

For the case where the data set is not separable, we can refine the approach to the separable case, see Figure 1. We will now allow the points to violate the equations of the separating hyperplane, but we will impose a penalty for the violation. Letting the nonnegative variable ξ_i denote the amount by which the point x_i violates the constraint at the margin, we now add to the objective a term proportional to the sum of the violations. The added penalty term takes the form $K \sum \xi_i$ and is added to the objective, where the larger the value of the parameter K, the larger the penalty for violating the separation. Our problem is now to find $\mathbf{w}$, b and ξ that solve

$$\begin{array}{lll} \text{mimize} & f(\mathbf{w}, b, \xi) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + K \sum \xi_i & \\ \text{subject to} & y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i & i = 1 \dots m \\ & \xi_i \geq 0 & \end{array} \tag{1}$$

This is the original or *primal* problem. In the previous lab we have seen that many optimization problems have a companion problem called the dual problem. The dual problem may be easier to solve, and if the optimal solution to the dual problem is known, then (in nondegenerate cases) the

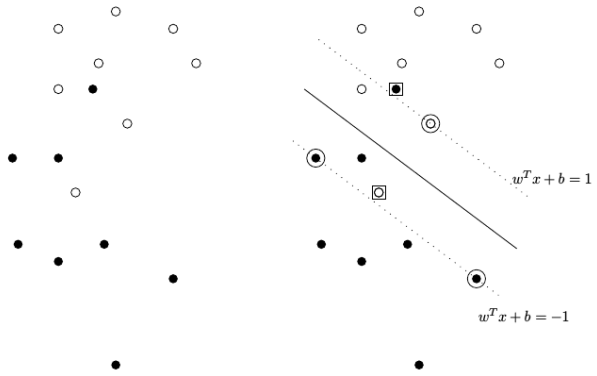

Figure 1: Linear separating hyperplane for the non separable case. Image taken from Griva, I.; Nash, S.; Sofer, A., "Linear and nonlinear optimization", SIAM.

optimal solution to the primal problem can be easily computed. In addition, for the case of the Support Vector Machine, the dual optimization problem can be written in terms of dot products, thereby making it possible to use kernel functions.

These reasons are not a limitation for solving the problem in the primal, mainly by writing the optimization problem as an unconstrained one and by using the Representer theorem to be able to deal with the non-linear case. Intuitively, the primal optimization should be superior than the dual because it directly minimizes the quantity we are interested in, $\mathbf{w}$, whereas in the dual problem we solve the dual variable, α (see Lab 5).

The primal problem of Equation (1) can be rewritten in an unconstrained way as

$$f(\mathbf{w}, b) = \frac{1}{2}\mathbf{w}^T\mathbf{w} + K\sum_{i=0}^{m} \max\left(0,\ 1 - y_i(\mathbf{w}^T\mathbf{x}_i + b)\right) \tag{2}$$

This is the function to be minimized and it can be solved using, for instance, the gradient or the Newton method. In this case, however, we propose to use the Stochastic gradient algorithm. The Stochastic gradient descent is a drastic simplification of the gradient descent. Instead of computing the gradient of Equation (2) using all the terms of the sum, $i = 1 \ldots m$, each iteration estimates the gradient of $f(\mathbf{w}, b)$ on the basis of a single randomly picked example $\mathbf{x}_t$, where t is the stochastic process $t = 1, \ldots$

We just rewrite a bit the previous equation in order to avoid too "large numbers" when doing numerical computations. We just rewrite the previous equation as

$$f(\mathbf{w}, b) = \frac{\lambda}{2}\mathbf{w}^T\mathbf{w} + \sum_{i=0}^{m} \max\left(0,\ 1 - y_i(\mathbf{w}^T\mathbf{x}_i + b)\right)$$

where λ should be a "small value" (since usually K is a large value). The stochastic gradient algorithm is

$$\begin{aligned}
\mathbf{w} &\leftarrow \mathbf{w} - \gamma_t \left(\lambda \mathbf{w} + \begin{cases} 0 & \text{if } y_t(\mathbf{w}^T \mathbf{x}_t + b) > 1 \\ -y_t \mathbf{x}_t & \text{otherwise} \end{cases} \right) \\
b &\leftarrow b - \gamma_t \begin{cases} 0 & \text{if } y_t(\mathbf{w}^T \mathbf{x}_t + b) > 1 \\ -y_t & \text{otherwise} \end{cases}
\end{aligned}$$

The convergence of stochastic gradient descent has been studied extensively in the literature. Convergence results usually require decreasing gains γ_t. The convergence speed of stochastic gradient descent is in fact limited by the noisy approximation of the true gradient. Under sufficient regularity conditions, the best convergence speed is achieved using gains $\gamma_t \sim t^{-1}$.

For the case of the mini-batch a similar procedure is followed: assume you want to perform a mini-batch of a set S of M elements. At each iteration t you select M elements from the m available samples. The stochastic minibatch algorithm is

$$\begin{aligned}
\mathbf{w} &\leftarrow \mathbf{w} - \gamma_t \left(\lambda \mathbf{w} + \sum_{k \in S} \begin{cases} 0 & \text{if } y_k(\mathbf{w}^T \mathbf{x}_k + b) > 1 \\ -y_k \mathbf{x}_k & \text{otherwise} \end{cases} \right) \\
b &\leftarrow b - \gamma_t \left(\sum_{k \in S} \begin{cases} 0 & \text{if } y_k(\mathbf{w}^T \mathbf{x}_k + b) > 1 \\ -y_k & \text{otherwise} \end{cases} \right)
\end{aligned}$$

Report

You are asked to deliver a report (PDF, notebook, or whatever else you prefer) of the work you have performed. You are requested to perform a comparison between the dual method studied in Lab 5, as well as the stochastic and the mini batch using the primal equation (2). Take into account that the problem you are solving is quadratic, convex, and thus it (mathematically) only has only minimum. But the numerical method you use may have difficulties to arrive to such minimum.

It is interesting to perform the next experiments: a) Start with the "pure" stochastic gradient descent, that is, the one in which only one random sample is taken at each iteration, 2) Perform some experiments with the mini-batch. You may try different number of samples for each experiment, e.g. 10 or 50 samples. 3) Perform a plot of the logarithm of $f(\mathbf{w}, b)$ along the iterations of each "epoch" (it is important to plot the logarithm of the function). This allows you to see how fast each of the methods approaches the optimal solution. This will allow you to perform a comparison between the different experiments you have done. **You are recommended to use separable sets to perform such experiments.**

Notes

There is a great paper that treats the problem of training a Support Vector Machine in the primal. It treats the problem of training using Newton optimization. It is "Training a Support Vector Machine in the Primal", by O. Chapelle, 2006.