Constrained optimization: equality and inequality constraints

Lluís Garrido – lluis.garrido@ub.edu

December 2018

Abstract

This laboratory is focused on constrained optimization and, in particular, on equality and inequality constraints that appear in the support vector machines.

1 Support Vector Machines

Assume that we have a set of data points that we have *previously* classified in one of two ways: either they have a certain stated property or they do not. These data points might, for instance, represent the subject titles of email messages, which are classified as either being legitimate mail or spam. Suppose now that we obtain a new data point, i.e. a new email message. Our goal is to determine whether this new point does or does not have the stated property, i.e. if the email is legitimate or not. The set of techniques for doing this is broadly referred as pattern classification.

In its simplest form, pattern classification uses linear functions to provide the characterization. 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 $f(\mathbf{w}) = \mathbf{w}^T\mathbf{x} + b = 0$, $\mathbf{w} \in R^n$ and $b \in R$, which separates the positive points from the negative. For Figure 1 there is a hyperplane that clearly separates both classes. In this case

$$\begin{array}{lll} \mathbf{w}^T\mathbf{x}_i + b \geq +1 & \text{for} & y_i = +1 \\ \mathbf{w}^T\mathbf{x}_i + b \leq -1 & \text{for} & y_i = -1 \end{array}$$

We would like the hyperplane separating the positive points from the negative to be as far apart as possible. From basic geometric principles it can be shown that the distance between the two hyperplanes (that is, the separation margin) is $2/\left\|\mathbf{w}\right\|$. Thus among all separating hyperplanes we should seek the one that maximizes this margin. This is equivalent to minimizing $\mathbf{w}^T\mathbf{w}$. The resulting problem is to determine the coefficients $\mathbf{w}$ and b that solve

$$\begin{array}{lll} \text{minimize} & \frac{1}{2}\mathbf{w}^T\mathbf{w} & \\ \text{subject to} & y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 & i = 1\ldots m \end{array} \tag{1}$$

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 $f(\bar{\mathbf{x}})$.

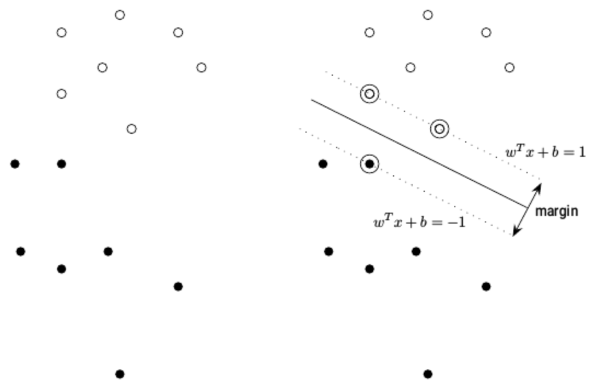

Figure 1: Linear separating hyperplane for the separable case. Image taken from Griva, I.; Nash, S.; Sofer, A., “Linear and nonlinear optimization”, SIAM.

This is the original or *primal* problem. How can this primal problem be solved? Many optimization problems have a companion problem called the dual problem. There are important relations between a primal and its dual, and that these relations sometimes lead to insights for solving the 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 optimal solution to the primal problem can be easily computed. The Lagrangian is used for this purpose

$$\mathcal{L}(\mathbf{w}, b, \alpha) = \frac{1}{2}\mathbf{w}^T\mathbf{w} - \sum_{i=1}^{m} \alpha_i \left[y_i(\mathbf{w}^T\mathbf{x}_i + b) - 1 \right] \tag{2}$$

where $\alpha = (\alpha_1, \ldots, \alpha_m)^T$ is the vector of non-negative Lagrange multipliers, $\alpha_i \geq 0$. It is known that the solution to the optimization problem is determined by the saddle point of this Lagrangian, where the minimum should be taken with respect parameters $\mathbf{w}$ and b, and the maximum should be taken with respect the Lagrange multipliers α. At the point of the minimum (with respect parameters $\mathbf{w}$ and b)

$$\frac{\partial \mathcal{L}}{\partial \mathbf{w}} = \left(\mathbf{w} - \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i \right) \tag{3}$$

$$\frac{\partial \mathcal{L}}{\partial b} = \sum_{i=1}^{m} y_i \alpha_i = 0 \tag{4}$$

The first equation, Eq. (3), leads us to

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i \mathbf{x}_i \tag{5}$$

which expresses that the optimal plane solution can be written as a linear combination of training vectors. Only training vectors $\mathbf{x}_i$ with $\alpha_i > 0$ (strictly positive) have an effective contribution to the previous sum. These vectors $\mathbf{x}_i$ satisfy $y_i(\mathbf{w}^T\mathbf{x}_i + b) = 1$ and are termed *support vectors*. The value of b can be computed easily for any support vector ensuring that $y_i(\mathbf{w}^T\mathbf{x}_i + b) = 1$.

Substituting Eq. (4) and Eq. (5) into Eq. (2) one obtains

$$\begin{aligned}\mathcal{W}(\alpha) &= \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\mathbf{w}^T\mathbf{w}\\ &= \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j \mathbf{x}_i^T\mathbf{x}_j^T\end{aligned}$$

Denote by $\mathbf{X}$ the $n \times m$ matrix whose columns are the training vectors $\mathbf{x}_i$. Let $\mathbf{Y} = \mathrm{diag}(\mathbf{y})$ be the diagonal matrix whose i-th diagonal term is y_i. The previous equation can be written as

$$\mathcal{W}(\alpha) = \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\alpha^T(\mathbf{Y}\mathbf{X}^T\mathbf{X}\mathbf{Y})\alpha$$

The original problem has thus been transformed into

$$\begin{array}{ll}\text{maximize} & \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\alpha^T(\mathbf{Y}\mathbf{X}^T\mathbf{X}\mathbf{Y})\alpha\\ \text{subject to} & \sum_{i=1}^{m} y_i\alpha_i = 0\\ & \alpha_i \geq 0\end{array} \tag{6}$$

We have transformed the original Eq. (1) into a quadratic problem with a linear constraint which is much easier to solve than the original primal problem. We will see this afterwards.

So far we have assumed that the data set was separable, that is, a hyperplane separating the positive points from the negative points exists. For the case where the data set is not separable, we can refine the approach to the separable case, see Figure 2. 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 require

$$\begin{array}{lll}\mathbf{w}^T\mathbf{x}_i + b \geq +1 - \xi_i & \text{for} & y_i = +1\\ \mathbf{w}^T\mathbf{x}_i + b \leq -1 + \xi_i & \text{for} & y_i = -1\end{array}$$

A common way to impose the penalty is to 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

$$\begin{array}{lll}\text{minimize} & \frac{1}{2}\mathbf{w}^T\mathbf{w} + K\sum\xi_i & \\ \text{subject to} & y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i & i = 1\dots m\\ & \xi_i \geq 0 & \end{array} \tag{7}$$

Let us know formulate the dual of the primal Eq. (7). We are not going into the details of how this problem it transformed into the dual space. If you are interested in it take a look at the paper of C. Cortes and V. Vapnik, “Suport Vector Networks”, Machine Learning.

The dual problem can be written just in terms of α

$$\begin{array}{ll}\text{maximize} & f(\alpha) = \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\alpha^T(\mathbf{Y}\mathbf{X}^T\mathbf{X}\mathbf{Y})\alpha\\ \text{subject to} & \sum_{i=1}^{m} y_i\alpha_i = 0\\ & 0 \leq \alpha_i \leq K\end{array}$$

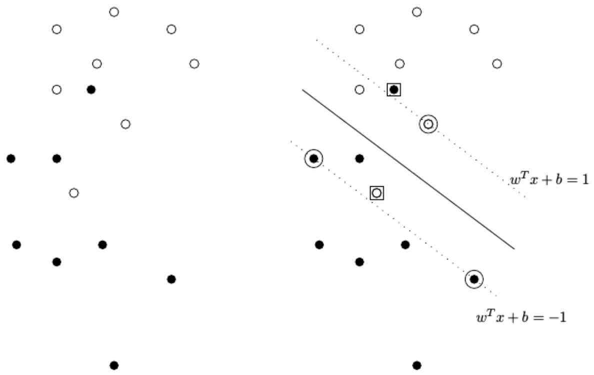

Figure 2: Linear separating hyperplane for the non separable case. Image taken from Griva, I.; Nash, S.; Sofer, A., "Linear and nonlinear optimization", SIAM.

The dual, like the primal, is a quadratic problem. However, it is usually easier to solve because, with the exception of one equality, all constraints are simple upper and lower bounds. Since the function is quadratic it is a convex function and thus it only has one mininum.

Assume that the dual problem has been solved, that is, we know the optimal values of the vector α. The parameters w and b are then computed as is done for the separable case

$$\mathbf{w} = \sum_{i=1}^{m} \alpha_i y_i x_i$$

A point $\mathbf{x}_j \in R^n$ for which $0 < \alpha_j < K$ satisfies $\xi_j = 0$ and thus $y_j(\mathbf{w}^T\mathbf{x}_j + b) = 1$. Any such point j can be used to compute the value of b

$$b = y_j - \mathbf{w}^T\mathbf{x}_j$$

If there are several such points, the average computed value of b is commonly taken to ensure the highest accuracy.

In general, the significance of the dual formulation is its computational ease. But, for the support vector machine, it also has another important advantage: it allows us to expand the power of support vector machines to data that are not linearly separable. In other words, it allows to use other functions (called kernels) to separate data points. This is out of the scope of this laboratory.

2 Implementation of the dual formulation

This is the problem that has to be solved

$$\begin{array}{ll} \text{maximize} & f(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\alpha^T(\mathbf{Y}\mathbf{X}^T\mathbf{X}\mathbf{Y})\alpha \\ \text{subject to} & \sum_{i=1}^{m} y_i\alpha_i = 0 \\ & 0 \le \alpha_i \le K \end{array} \tag{8}$$

Are you able to recognize the problem that has to be solved? In the course of numerical linear algebra, you have learned to solve problems like this ones

$$\begin{array}{ll} \text{minimize} & f(x) = \frac{1}{2}x^T\mathbf{G}x + g^Tx \\ \text{subject to} & \mathbf{A}^Tx = b,\ \mathbf{C}^Tx \geq d \end{array} \tag{9}$$

where $x \in R^n$, $\mathbf{G} \in R^{n\times n}$ is symmetric semidefinite positive, $g \in R^n$, $\mathbf{A} \in R^{n\times p}$, $\mathbf{C} \in R^{n\times m}$, $b \in R^p$ and $d \in R^m$ (the notation used in this equation does not have any relationship with the notation used in the problem defined in the previous section).

Do you recognize now the similarities of the two problems to be solved? The difference is the fact that in our problem, Equation (8), we have the inequality $0 \leq \alpha_i \leq K$. How is this "translated" to the problem you have learned in numerical linear algebra? In the case of the problem you have analyzed in numerical algebra this translates to consider $d_{low} \leq \mathbf{C}^Tx \leq d_{high}$: we have two inequalities, $\mathbf{C}^Tx \geq d_{low}$ and $-\mathbf{C}^Tx \geq -d_{high}$. You just need to consider these two inequalities in the equations of the KKT you have implemented, which can be written as

$$\left(\begin{array}{c} \mathbf{C}^T \\ -\mathbf{C}^T \end{array} \right) x \geq \left(\begin{array}{c} d_{low} \\ -d_{high} \end{array} \right)$$

which corresponds to the inequality you have to consider.

Exercise

This exercise assumes that you have implemented the code associated to (9). You will need to modify the code to adapt it to the problem we are considering. Implement the previous proposed algorithm and use the dataset with the following mean and covariance

```
m1 = [0.,0.]
s1 = [[1,-0.9],[-0.9,1]]
m2 = [3.,6.]  # Separable dataset
#m2 = [1.,2.]  s2 = [[1,0],[0,1]] # Non-separable dataset
```

- You are recommended to begin with the simplest case, i.e. the case in which data is separable, and with a small dataset (2 points for each class, for instance). You are recommended to start with a relatively small value of K, e.g. $K = 1$.

- Once it works, perform several experiments with different values of K and test the stability of the solution you obtain. You may use small values of K, e.g. $K = 1$, as well as large values of K, e.g. $K = 10^6$. Repeat the experiment several times. Does the value of K influence the solution you obtain? Can you explain the results? Recall that you are in the separable case, i.e. Eq (6) has no upper threshold for α.

- Once it works, you may test your algorithm with the non-separable case using different values of K. Repeat the experiment several times. Again, does the value of K influence the solution you obtain? Can you explain the results?

- Check if the solutions you obtain are similar to those obtained with the ones obtained in the lectures of Oriol.

Report

There is no need to deliver a report of this lab, since you will be requested to deliver one report that includes this lab, lab 5, and lab 6, the next one. You may begin doing the report, by including in it he steps you have followed as well as the results and plots you obtain. Do not expect the reader (i.e. me) to interpret the results for you. I would like to see if you are able to understand the results you have obtained. The objective is to deliver a joint report of lab 5 and lab 6 in which you comment the results you have obtained with both methods. Whereas in this lab $\mathbf{w}$ is solved via the dual formulation, in lab 6 the weights $\mathbf{w}$ will be solved using the primal formulation using stochastic gradient descent. This will allow you to compare both methods for solving the same problem.

Please include the Python code within the lab indicating with comments the changes you have done to the original code you had. You may include it as separate files if you wish so, there is not need to include it within the the notebook. You may just deliver the Python notebook if you want.

Note

Most of the text of section 1 has been copied from different chapters of Griva, I.; Nash, S.; Sofer, A., “Linear and nonlinear optimization”, SIAM. The demonstrations of the dual problem of the original primal can be obtained in C. Cortes and V. Vapnik, “Suport Vector Networks”, Machine Learning.