Crash course on constrained optimization problems

Doina Precup

October 15, 2003

Notation: in this document, like in all lecture notes, we denote vectors with bold letters. So

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

would be a vector, and x_i, $i = 1 \dots n$ are the elements of the vector. We denote the transpose of a vector by $\mathbf{x}^T$.

Suppose you have a constrained optimization problem in which you want to find:

$$\min_{\mathbf{w}} f(\mathbf{w})$$

subject to k equality constraints:

$$h_i(\mathbf{w}) = 0, \ i = 1 \dots k,$$

where f and h_i are arbitrary real-valued functions. In order to solve the problem, we will define a "helper", called the **Lagrangian**:

$$L(\mathbf{w}, \beta) = f(\mathbf{w}) + \sum_{i=1}^{k} \beta_i h_i(\mathbf{w})$$

The newly introduced variables, $\beta_i, i = 1 \dots k$ are called Lagrange multipliers. Now, instead of finding $\min_{\mathbf{w}} f(\mathbf{w})$, we will try to find $\min_{\mathbf{w}} \max_{\beta} L(\mathbf{w}, \beta)$.

Why does this work? Consider that we are given some arbitrary $\mathbf{w}$. Suppose that one of the constraints is violated, i.e., $h_i(\mathbf{w}) \neq 0$ for some i. In this case, we can take the Lagrangian to ∞, by setting the corresponding β_i to an arbitrarily large positive (or negative) number (depending on whether $h_i(\mathbf{w}) > 0$ or $h_i(\mathbf{w}) < 0$). If all constraints are satisfied, then $L(\mathbf{w}, \beta) = f(\mathbf{w})$. So we have:

$$\max_{\beta} L(\mathbf{w}, \beta) = \begin{cases} f(\mathbf{w}) & \text{if all constraints are satisfied} \\ +\infty & \text{otherwise} \end{cases}$$

So $\min_{\mathbf{w}} f(\mathbf{w}) = \min_{\mathbf{w}} \max_{\beta} L(\mathbf{w}, \beta)$.

Now we converted our constrained optimization problem into an unconstrained optimization problem. In principle, we can solve this by taking the partial derivatives of L with respect to all the β_i and w_j, and setting them to 0. If we cannot solve the resulting system for all parameters, then we can do gradient descent or use other, fancier methods.

Now let us assume we have a more complicated constraint optimization problem, in which there are equality as well as inequality constraints:

$$\begin{aligned} \min_{\mathbf{w}} f(\mathbf{w}) & \\ \text{such that } g_i(\mathbf{w}) &\leq 0,\ i = 1 \dots k \\ h_j(\mathbf{w}) &= 0, j = 1 \dots l \end{aligned}$$

In order to solve it, we will apply the same "helper" idea as above and define the **generalized Lagrangian**:

$$L(\mathbf{w}, \alpha, \beta) = f(\mathbf{w}) + \sum_{i=1}^{k} \alpha_i g_i(\mathbf{w}) + \sum_{j=1}^{l} \beta_j h_j(\mathbf{w}), \tag{1}$$

where $\alpha_i,\ i = 1 \dots k$ and $\beta_j,\ j = 1 \dots l$ are the Lagrange multipliers.

Now consider the quantity:

$$\mathcal{P}(\mathbf{w}) = \max_{\alpha, \beta : \alpha_i \geq 0} L(\mathbf{w}, \alpha, \beta)$$

Consider again an arbitrary setting of $\mathbf{w}$. If it violates some constraints (either $g_i(\mathbf{w}) > 0$ for some i or $h_j(\mathbf{w}) \neq 0$ for some j), then we can use the same reasoning as above to show that $\mathcal{P}(\mathbf{w}) = \infty$. Now suppose that the constraints are satisfied. Then $h_j(\mathbf{w}) = 0$, $j = 1 \dots l$, so the third term in (1) is 0. For the second term, since $g_i(\mathbf{w}) \leq 0\, i = 1 \dots k$, and since we want to maximize, the best thing to do is to set $\alpha_i = 0$ for all i for which $g_i(\mathbf{w}) < 0$. In this case, the second term is also 0. Putting all these together, we get:

$$\mathcal{P}(\mathbf{w}) = \begin{cases} f(\mathbf{w}) & \text{if all constraints are satisfied} \\ +\infty & \text{otherwise} \end{cases}$$

Hence, instead of computing $\min_{\mathbf{w}} f(\mathbf{w})$ subject to the original constraints, we can compute:

$$\min_{\mathbf{w}} \mathcal{P}(\mathbf{w}) = \min_{\mathbf{w}} \max_{\alpha, \beta : \alpha_i \geq 0} L(\mathbf{w}, \alpha, \beta)$$

Let $p^* = \min_{\mathbf{w}} \mathcal{P}(\mathbf{w})$ denote the solution of the primal problem.

Now let us consider a similar-looking (and related) problem:

$$\max_{\alpha, \beta : \alpha_i \geq 0} \min_{\mathbf{w}} L(\mathbf{w}, \alpha, \beta)$$

This is called the **dual** optimization problem, and it is the same as the primal, except that the min and max are reversed. Let d^* be the value of this dual problem. Then it is easy to show that $d^* \leq p^*$ (you will show that this is true in homework 6). Hence, often the solution of the dual problem is used to bound the solution of the primal problem. However, under certain conditions, we have $d^* = p^*$ (and hence we can solve the dual problem instead of the primal problem). One set of conditions is given the **strong duality theorem** (see, e.g, the book on support vector machines by Cristianini and Shawne-Taylor):

- g_i and h_j have to be **affine**, i.e., they have to be of the form ${\mathbf{a_i}}^T \mathbf{w} + b_i$
- the domain of $\mathbf{w}$ has to be convex

Note that there are other sets of assumptions (less restrictive) which also allow for $p^* = d^*$. However, for the purpose of SVMs, we only need to consider the conditions above.

When $p^* = d^*$, let $\mathbf{w}^*$, α^* and β^* be the parameters settings which achieve the solution of these problems. These parameters settings have to satisfy the following conditions (which should be obvious based on our discussion so far):

$$\frac{\partial}{\partial w_i} L(\mathbf{w}^*, \alpha^*, \beta^*) = 0, \ i = 1 \ldots n \tag{2}$$

$$\frac{\partial}{\partial \beta_j} L(\mathbf{w}^*, \alpha^*, \beta^*) = 0, \ j = 1 \ldots l \tag{3}$$

$$\alpha_i^* g_i(\mathbf{w}^*) = 0, \ i = 1 \ldots k \tag{4}$$

$$g_i(\mathbf{w}^*) \leq 0, \ i = 1 \ldots k \tag{5}$$

$$\alpha_i^* \geq 0, \ i = 1 \ldots k \tag{6}$$

These are called the **Karush-Kuhn-Tucker (KKT) conditions**. Condition (4), which is called the **complementarity condition** will be especially important from our point of view. It implies that, if $\alpha_i^* > 0$, then $g_i(\mathbf{w}^*) = 0$ (i.e., the constraint is **active**, which means that it holds with equality rather than inequality).

Acknowledgments

This description is inspired to a large extent by the course notes of Andrew Ng.