Large margins optimization

Input: m linearly separable training examples (x_i, y_i), $i = 1, \ldots, m$, where x_i is the ith feature vector and y_i is either -1 or $+1$.

Output: A weights vector w and a scalar b such that:

$$\text{for } i = 1, \ldots, m, \quad \begin{cases} \text{if } y_i = +1 & w'x_i + b > 0 \\ \text{if } y_i = -1 & w'x_i + b < 0 \end{cases}$$

In addition, w, b should "maximize the margins".

The margins

What does it mean that w, b maximize the margins? We may try to define it as follows: Find max $\gamma > 0$ such that:

$$\text{for } i = 1, \ldots, m, \quad \begin{cases} \text{if } y_i = +1 & w'x_i + b \geq \gamma \\ \text{if } y_i = -1 & w'x_i + b \leq -\gamma \end{cases}$$

This is not a good definition since one can always multiply w, b, γ by a large constant to increase γ. Therefore, without loss of generality we can fix the value of γ to 1 and compute the margins in terms of w, b. We have:

$$\text{for } i = 1, \ldots, m, \quad \begin{cases} \text{if } y_i = +1 & w'x + b \geq 1 \\ \text{if } y_i = -1 & w'x + b \leq -1 \end{cases} \tag{1}$$

This means that all positive training examples are "above" the hyperplane $w'x = 1 - b$, and all negative training examples are "below" the hyperplane $w'x = -1 - b$. Our goal is to maximize the distance between these two hyperplanes.

As previously shown the distance between the hyperplane $w'x = s$ and the origin is $\frac{|s|}{|w|}$. Therefore, the distance between the two hyperplanes if they are on the same side of the origin is: $\mid \frac{|1-b|}{|w|} - \frac{|-1-b|}{|w|} \mid$, and the distance between the two hyperplanes if they are on different sides of the origin is: $\frac{|1-b|}{|w|} + \frac{|-1-b|}{|w|}$. In both cases this can be shown to be $\frac{2}{|w|}$.
Therefore, maximizing the margins is achieved by minimizing $|w|$, or $|w|^2$, or $\frac{1}{2}|w|^2$.

The primal problem:

Minimize $\frac{1}{2}|w|^2$ subject to the m linear inequality constraints:

$$\text{for } i = 1, \ldots, m, \qquad y_i(w'x_i + b) \geq 1$$

Derivation of the dual problem

The Lagrangian of the primal problem:

$$L(w, b, \alpha_1, \ldots, \alpha_m) = \frac{1}{2}|w|^2 + \sum_{i=1}^{m} \alpha_i(1 - y_i(w'x_i + b))$$

To compute the dual problem we need to maximize L with respect to w, b so that it is a function of only the alphas.

$$\begin{array}{ll} \text{The derivative of } L \text{ w.r.t. } w \text{ gives:} & w = \sum_{i=1}^m \alpha_i y_i x_i \\ \text{The derivative of } L \text{ w.r.t. } b \text{ gives:} & \sum_{i=1}^m \alpha_i y_i = 0 \end{array} \tag{2}$$

Substituting the above value of w in L:

$$\begin{aligned} L =& \frac{1}{2}(\sum_{i=1}^m \alpha_i y_i x_i')(\sum_{j=1}^m \alpha_j y_j x_j) + \sum_{i=1}^m \alpha_i \left(1 - y_i(\sum_{j=1}^m \alpha_j y_j x_j' x_j + b)\right) \\ =& \frac{1}{2}\sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j x_i' x_j + \sum_{i=1}^m \alpha_i - \left(\sum_{i=1}^m \alpha_i y_i \sum_{j=1}^m \alpha_j y_j x_j' x_j\right) - b \sum_{i=1}^m \alpha_i y_i \\ =& -\frac{1}{2}\sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j x_i' x_j + \sum_{i=1}^m \alpha_i \end{aligned}$$

where in the last step we use the fact that $\sum_{i=1}^m \alpha_i y_i = 0$.

The dual problem:

$$\text{Maximize} \quad L(\alpha_1, \ldots, \alpha_m) = \sum_{i=1}^m \alpha_i - \frac{1}{2}\sum_{i=1}^m \sum_{j=1}^m \alpha_i \alpha_j y_i y_j x_i' x_j$$

$$\text{subject to:} \quad \alpha_i \geq 0, \quad \sum_{i=1}^m \alpha_i y_i = 0$$

This is a quadratic programming problem and we assume that there is a black-box that solves it. The solution gives the values of the α_i, and typically most of these values are 0. The points corresponding to nonzero α_i are called **support vectors**. To simplify notation we assume without loss of generality that only $\alpha_1, \ldots, \alpha_k$ are nonzero, so that $x_1, \ldots, x_k$ are the support vectors.

Recovering w, b

From (2) it follows that w can be recovered from the support vectors with no need to consider any of the other points:

$$w = \sum_{j=1}^k \alpha_j y_j x_j \tag{3}$$

Once w is determined the value of b can be computed from any one of the support vectors. From the Karush-Kuhn-Tucker Complementary Condition each support vectors (x_s, y_s) satisfies: $y_s(w' x_s + b) = 1$

$$\text{so that} \quad b = \frac{1}{y_s} - w' x_s \tag{4.1}$$

When working with inexact arithmetic it is not desirable to rely on a single support vector. A more robust equation for b is:

$$b = -\frac{1}{2}\left(\min_{y_s=1} w' x_s + \max_{y_s=-1} w' x_s\right) \tag{4.2}$$

Example

i	0	1	2
x_i	0	1	2
y_i	-1	-1	1
Lagrangian multiplier	α_0	α_1	α_2

The dual problem:

$$\begin{aligned}
\text{maximize} \quad L(\alpha_0, \alpha_1, \alpha_2) &= \alpha_0 + \alpha_1 + \alpha_2 - \frac{1}{2}(0 + 0 + 0 + 0 + \alpha_1^2 - 2\alpha_1\alpha_2 + 0 - 2\alpha_1\alpha_2 + 4\alpha_2^2) \\
&= \alpha_0 + \alpha_1 + \alpha_2 - \frac{1}{2}(\alpha_1^2 - 4\alpha_1\alpha_2 + 4\alpha_2^2) \\
\text{subject to:} \ \ \alpha_0 \geq 0, \ \ \alpha_1 &\geq 0, \ \ \alpha_2 \geq 0, \quad -\alpha_0 - \alpha_1 + \alpha_2 = 0
\end{aligned}$$

The solution (computed by the black box quadratic optimizer) is: $\alpha_0 = 0$, $\alpha_1 = \alpha_2 = 2$. Therefore, the support vectors are x_1, x_2.
We can now compute w from (3):

$$w = -2 + 4 = 2$$

The value of b can be computed, for example, from the first support vector: using (4.1):

$$b = -1 - 2 = -3$$

It can also be computed from the second support vector: using (4.1):

$$b = 1 - 2 \times 2 = -3$$

It can also be computed from all the vectors using (4.2):

$$b = -\frac{1}{2}\left(\min\{4\} + \max\{0, 2\}\right) = -3$$

Verify that the "entire" training data is correctly classified and that the distance between the two hyperplanes is, indeed, $2/|w| = 1$.

It is not necessary to compute w explicitly

Given a test point z, its classification is determined from the sign of $w'z + b$. Using (3), it is clear that there is no need to compute w explicitly since $w'z = \sum_{j=1}^{k} \alpha_j y_j x_j' z$. Define the function $K(x, z)$ to be $x'z$:

$$K(x, z) = x'z \tag{5}$$

Then we have:

$$w'z = \sum_{j=1}^{k} \alpha_j y_j K(x_j, z)$$

Also observe that the value of b, computed in (4.1), can be written as:

$$b = \frac{1}{y_s} - \sum_{j=1}^{k} \alpha_j y_j K(x_j, x_s)$$

where (x_s, y_s) is any one of the support vectors. Alternatively, using (4.2) b can be computed as:

$$b = -\frac{1}{2}\left(\min_{y_i=1}\sum_{j=1}^{k}\alpha_j y_j K(x_j, x_i) + \max_{y_i=-1}\sum_{j=1}^{k}\alpha_j y_j K(x_j, x_i)\right)$$

Therefore, once b is computed the following condition can be used to compute the classification of z:

$$\text{Classify } z \text{ according to the sign of } \quad \sum_{j=1}^{k}\alpha_j y_j K(x_j, z) + b$$

It is not necessary to know the x_i explicitly

As shown above, when we use the hyperplane to classify test data we don't need to know the x_j or the vector z explicitly. It is enough to know the values of the function $K(,)$, when applied to these vectors. Observe now that the same holds for the definition of the dual problem. It can be stated in terms of K as:

$$\text{Maximize} \quad L(\alpha_1, \ldots, \alpha_m) = \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m}\alpha_i\alpha_j y_i y_j K(x_i, x_j)$$

$$\text{subject to:} \quad \alpha_i \geq 0, \quad \sum_{i=1}^{m}\alpha_i y_i = 0$$

Therefore, it is enough to be able to compute $K(x_i, x_j)$, and this can sometimes be computed without the explicit vectors.

Generalization

Suppose there are k support vectors and m examples. If we perform leave-one-out cross validation, all non support vectors will be correctly classified. A rough estimate to a bound on the error is, therefore, k/m.

A PAC Learning style bound can also be proved. Let k be the number of support vectors and let m be the total number of training examples. Let ϵ be the error of the support vector on randomly chosen examples. Then with probability (confidence) of at least $1 - \delta$

$$\epsilon \leq \frac{1}{m-k}\left(k \log_2 \frac{em}{d} + \log_2 \frac{m}{\delta}\right)$$