**Introduction to Machine Learning** **Fall Semester, 2013**

# Recitation 6: November 17

*Lecturer: Mariano Schain* *Scribe: ym*

## 6.1 SVM optimization

In the lecture we saw the following optimization problem, for a maximum margin classifier.

$$\min_{w,b} \frac{1}{2} w^t w$$
$$\text{s.t. } y_n(w^t x_n + b) \geq 1 \qquad \forall n = 1, \ldots, N$$

where $w \in \mathbb{R}^d$ is the weight vector, $b \in \mathbb{R}$ is the bias, and $(x_n, y_n)$ are the examples and $x_n \in \mathbb{R}^d$ and $y_n \in \{+1, -1\}$.

The first step is to write the Lagrangian. In general, for a program

$$\min f(X)$$
$$\text{s.t. } g_i(x) \leq 0 \forall i = 1, \ldots, N$$

the Lagrangian is

$$L(x, \alpha) = f(x) + \sum_{i=1}^{N} \alpha_i g_i(x)$$

where $\alpha$ are called the *Lagrangian multipliers*.

For our SVM program we get

$$L(w, b, \alpha) = \frac{1}{2} w^t w - \sum_{n=1}^{N} \alpha_n (y_n (w^t x_n + b) - 1)$$

We now take the derivative of $L$ and equate it with zero to minimize over $w$ and $b$.

$$\nabla_w L = w - \sum_{n=1}^{N} \alpha_n y_n x_n = 0 \quad \Longrightarrow \quad w = \sum_{n=1}^{N} \alpha_n y_n x_n$$

this give us a way to compute $w$ given $\alpha$. We call this the $w$-constraint. For $b$ we have

$$\frac{d}{db} L = -\sum_{n=1}^{N} \alpha_n y_n = 0 \quad \Longrightarrow \quad \alpha_n y_n = 0$$

We call this the $b$-constraint.

Plugging the constraints back in $L$ we have

$$
\begin{aligned}
L(w,b,\alpha) =& \frac{1}{2}w^t w - w^t \underbrace{(\sum_{n=1}^{N} \alpha_n y_n x_n)}_{w} - b \underbrace{(\sum_{n=1}^{N} \alpha_n y_n)}_{0} + (\sum_{n=1}^{N} \alpha_n) \\
=& -\frac{1}{2}w^t w + (\sum_{n=1}^{N} \alpha_n) \\
=& -\frac{1}{2}(\sum_{i=1}^{N} \alpha_i y_i x_i)^t (\sum_{j=1}^{N} \alpha_j y_j x_j) + (\sum_{n=1}^{N} \alpha_n) \\
=& -\frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_j y_i x_i^t x_j + (\sum_{n=1}^{N} \alpha_n)
\end{aligned}
$$

where we have the constraints $\sum_{n=1}^{N} \alpha_n y_n = 0$ and $\forall n$ we have $\alpha_n \geq 0$.

Formally, the dual problem is

$$
\begin{aligned}
\max_{\alpha} L(w,b,\alpha) = \min_{\alpha} \quad & \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i \alpha_j y_j y_i x_i^t x_j - (\sum_{n=1}^{N} \alpha_n) \\
\text{s.t.} \quad & \sum_{n=1}^{N} \alpha_n y_n = 0 \\
& \forall n \;\; \alpha_n \geq 0
\end{aligned}
$$

## 6.2 Unrealizable case

We add slack variables $\xi_n$ to ensure feasibility. We have,

$$
\begin{aligned}
& \min_{w,b,\xi} \frac{1}{2}w^t w + C\sum_{n=1}^{N} \xi_n \\
& \text{s.t.} \;\; y_n(w^t x_n + b) \geq 1 - \xi_n \qquad \forall n = 1,\ldots,N \qquad \forall n \;\; \xi_n \geq 0
\end{aligned}
$$

We can now write the Lagrangian

$$
L(w,b,\xi,\alpha,r) = \frac{1}{2}w^t w + C\sum_{n=1}^{N} \xi_n - \sum_{n=1}^{N} \alpha_n (y_n(w^t x_n + b) - 1 + \xi_n) - \sum_{n=1}^{N} r_n \xi_n
$$

We now take the derivatives

$$\nabla_w L = w - \sum_{n=1}^{N} \alpha_n y_n x_n = 0 \quad \Longrightarrow \quad w = \sum_{n=1}^{N} \alpha_n y_n x_n$$

identically as before. For $b$ we have

$$\frac{d}{db} L = - \sum_{n=1}^{N} \alpha_n y_n = 0 \quad \Longrightarrow \quad \alpha_n y_n = 0$$

also as before.

For $\xi_n$ we have

$$\frac{d}{d\xi_n} L = C - \alpha_n - r_n = 0 \quad \Longrightarrow \quad \alpha_n = C - r_n$$

Substituting the constraints in $L$ we get

$$\begin{aligned} L(w,b,\alpha) =& \frac{1}{2} w^t w - w^t \underbrace{(\sum_{n=1}^{N} \alpha_n y_n x_n)}_{w} - b \underbrace{(\sum_{n=1}^{N} \alpha_n y_n)}_{0} + (\sum_{n=1}^{N} \alpha_n) + \sum_{n=1}^{N} \xi_n \underbrace{(C - \alpha_n - r_n)}_{0} \\ =& -\frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i \alpha_j y_j y_i x_i^t x_j + (\sum_{n=1}^{N} \alpha_n) \end{aligned}$$

identically as before. The only difference is that now we have two additional constraints, $r_n \geq 0$ and $\alpha_n = C - r_n$. Since $r_n$ does now appear in the optimization, we can drop it, and join then two constraints to $\alpha_n \leq C$. (For any solution of $\alpha_n$ we can set $r_n = C - \alpha_n$.)

Note that when we have an error in classification or in the margin, then $\xi_n > 0$ and therefore $r_n = 0$, which implies that $\alpha_n = C$.

For $C > \alpha_n > 0$ we have $r_n > 0$ and therefore $\xi_n = 0$. Since $\alpha_n > 0$ this implies that it is a support vector.

For $\alpha_n = 0$ we have $r_n = C$ and therefore $\xi_n = 0$ and since $\alpha_n = 0$ this is not an support vector.

## 6.3 Sequential Minimization Optimization (SMO)

For a convex program, we can solve it by doing a gradient ascent, simply choosing a single coordinate and optimizing the value. In our case, since we have a constraint that $\sum_{n=1}^{N} \alpha_n y_n = 0$, relaxing a single variable will be forced back to the same solution. For this we need to relax at least two variables.

Without loss of generality assume we selected $\alpha_1$ and $\alpha_2$. From the constraint we have,

$$\alpha_1 y_1 + \alpha_2 y_2 = -\sum_{i=3}^{N} \alpha_i y_i = F$$

where $F$ is some constant (since we keep $\alpha_i$ for $i > 3$ fixed). Now we can set

$$\alpha_1 = (F - \alpha_2 y_2) y_1$$

This implies that in the maximization we have a single variable $\alpha_2$ we are maximizing over. The weight function is now

$$w((F - \alpha_2 y_2) y_1, \alpha_2, \alpha_3, \ldots, \alpha_N)$$

which is a quadratic function in $\alpha_2$. (Recall that we keep $\alpha_i$ for $i > 3$ fixed).

We can now maximize it as an unconstraint quadratic form and find a maximizer $\bar{\alpha}_2$. We now need to consider the constraints

$$0 \leq \alpha_2 \leq C$$

and

$$0 \leq (F - \alpha_2 y_2) y_1 = \alpha_1 \leq C$$

the two constraints give a feasible range $[L, H]$ of $\alpha_2$. We can now test the unconstraint solution $\bar{\alpha}_2$ to derive the optimal solution $\alpha_2^*$, as follows,

1. If $\bar{\alpha}_2 \in [L, H]$ then $\alpha_2^* = \bar{\alpha}_2$.
2. If $\bar{\alpha}_2 < L < H$ then $\alpha_2^* = L$.
3. If $L < H < \bar{\alpha}_2$ then $\alpha_2^* = H$.