Notes on Support Vector Machine Optimization

22nd October 2002

The basic support vector machine cost function is

$$\min_{\alpha} E(\alpha) = \min_{\alpha} -\sum_{i=1}^{N} \alpha_i + \frac{1}{2}\sum_{i=1}^{N} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \tag{1}$$

subject to the constraints

$$\alpha_i \geq 0, \; and \sum_{i=1}^{N} \alpha_i y_i = 0.$$

For the moment, we avoid including the constraint $\alpha_i \leq C, \forall i$ since it complicates matters. We now use a Bregman distance (or I-divergence) *barrier function* to express the positivity constraint and a Lagrange parameter μ to express the constraint $\sum_{i=1}^{N} \alpha_i y_i = 0$:

$$\begin{aligned} E_{Bregman}(\alpha, \sigma, \mu) \;\; = \;\; & -\sum_{i=1}^{N} \alpha_i + \frac{1}{2}\sum_{i=1}^{N} \alpha_i \alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) + \frac{1}{\beta}\sum_{i=1}^{N} (\alpha_i \log \frac{\alpha_i}{\sigma_i} - \alpha_i + \sigma_i) \\ & +\mu \sum_{i=1}^{N} \alpha_i y_i \end{aligned}$$

where σ is a new variable which will subsequently be identified with $\alpha^{(t)}$. Differentiating w.r.t. (α, σ, μ) and setting the results to zero, we get

$$\frac{\partial E_{Bregman}}{\partial \alpha_i} = 0 \;\; \Rightarrow \;\; -1 + \sum_{j=1}^{N} y_i y_j k(x_i, x_j) \alpha_j + \frac{1}{\beta} \log \frac{\alpha_i}{\sigma_i} + \mu y_i = 0 \tag{2}$$

$$\frac{\partial E_{Bregman}}{\partial \sigma_i} = 0 \;\; \Rightarrow \;\; \frac{1}{\beta}(-\frac{\alpha_i}{\sigma_i} + 1) = 0 \tag{3}$$

$$\frac{\partial E_{Bregman}}{\partial \mu} = 0 \;\; \Rightarrow \;\; \sum_{i=1}^{N} \alpha_i y_i = 0 \tag{4}$$

We cannot invert (2) to get a closed form solution for α_i. However, we can solve for σ in (3) to get the solution $\sigma_i = \alpha_i$. This will interpreted to mean that $\sigma_i^{(t+1)} = \alpha_i^{(t)}$, $\forall i$. The chosen update equation for α_i is

$$\alpha_i^{(t+1)} = \alpha_i^{(t)}(1 - \gamma^{(t+1)}) + \gamma^{(t+1)} \sigma_i^{(t+1)} \exp\{-\beta(1 + \sum_{j=1}^{N} y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \alpha_j^{(t)}) - \beta \mu^{(t+1)} y_i\} \tag{5}$$

In (5), $\sigma^{(t+1)}$may be set to $\alpha^{(t)}$. Also, we can solve for $\mu^{(t+1)}$by using the constraint equation (4). Assuming that $\alpha_i^{(t)}$ satisfies the constraint $\sum_{i=1}^{N} \alpha_i^{(t)} y_i = 0$ and by using $y_i = +1, \forall i \in C_1$ and $y_j = -1, \forall j \in C_2$, we get (after some algebra)

$$\begin{aligned} \alpha_i^{(t+1)} &= \alpha_i^{(t)}(1 - \gamma^{(t+1)}) + \gamma^{(t+1)} \alpha_i^{(t)} \exp\{-\beta(1 + \sum_{k=1}^{N} y_i y_k k(\mathbf{x}_i, \mathbf{x}_k) \alpha_k^{(t)})\} \\ &\quad \cdot \sqrt{\frac{\sum_{j \in C_2} \alpha_j^{(t)} \exp\{-\beta(1 + \sum_{k=1}^{N} y_j y_k k(\mathbf{x}_j, \mathbf{x}_k) \alpha_k^{(t)})\}}{\sum_{i \in C_1} \alpha_i^{(t)} \exp\{-\beta(1 + \sum_{k=1}^{N} y_i y_k k(\mathbf{x}_i, \mathbf{x}_k) \alpha_k^{(t)})\}}}, \forall i \in C_1, \end{aligned}$$

and

$$\begin{aligned} \alpha_j^{(t+1)} &= \alpha_j^{(t)}(1 - \gamma^{(t+1)}) + \gamma^{(t+1)} \alpha_j^{(t)} \exp\{-\beta(1 + \sum_{k=1}^{N} y_j y_k k(\mathbf{x}_j, \mathbf{x}_k) \alpha_k^{(t)})\} \\ &\quad \cdot \sqrt{\frac{\sum_{i \in C_1} \alpha_i^{(t)} \exp\{-\beta(1 + \sum_{k=1}^{N} y_i y_k k(\mathbf{x}_i, \mathbf{x}_k) \alpha_k^{(t)})\}}{\sum_{j \in C_2} \alpha_j^{(t)} \exp\{-\beta(1 + \sum_{k=1}^{N} y_j y_k k(\mathbf{x}_j, \mathbf{x}_k) \alpha_k^{(t)})\}}}, \forall j \in C_2. \end{aligned}$$

The initial condition must satisfy the constraints $\alpha_i^{(0)} \geq 0$ and $\sum_{i=1}^{N} \alpha_i^{(0)} y_i = 0$. The stepsize parameter $\gamma^{(t+1)}$ should be chosen to decrease the cost function in (1). The parameter β is a free parameter.

Enforcing the constraint $\alpha_i \leq C$ is somewhat tricky. One way to do it is to set $\alpha_i = C$ whenever it increases the limit and set the stepsize parameter to zero for that α_i. Unfortunately, this messes up the solution for μ above. A second approach is to use slack variables and set up a second Bregman distances for the slacks and an extra constraint $\alpha_i + slack_i = C$. This has been implemented in my code and you are free to use this approach. A third approach is to lower the stepsize parameter until all $\alpha_i \leq C$. This results in a very slow algorithm.