CS281B/Stat241B: Advanced Topics in Learning & Decision Making

Optimal Margin Classifiers

Lecturer: Michael I. Jordan *Scribes: Weidong Cui*

1 Lagrangian Duality

For the primal problem:

$$\min f(x), \quad s.t.\ g(x) \leq 0 \tag{1}$$

we can express its Lagrangian as

$$\mathcal{L}(x,u) = f(x) + u^T g(x) \tag{2}$$

Moreover, we can easily see

$$\max_{u \geq 0} \mathcal{L}(x,u) = \begin{cases} \infty & \text{if } g(x) > 0 \\ f(x) & \text{otherwise} \end{cases} \tag{3}$$

So we can conclude

$$\min_x \max_{u \geq 0} \mathcal{L}(x,u) \equiv \textit{the primal problem} \tag{4}$$

The following theorem provides us with a way to find the dual problem.

Theorem 1 *If and only if there exists a saddle point of $\mathcal{L}(x,u)$, we have*

$$\max_{u \geq 0} \min_x \mathcal{L}(x,u) = \min_x \max_{u \geq 0} \mathcal{L}(x,u) \tag{5}$$

We define a new function $\Theta(u)$

$$\Theta(u) \stackrel{\triangle}{=} \min_x \mathcal{L}(x,u) \tag{6}$$

Then the dual problem is

$$\max \Theta(u), \ s.t.\ u \geq 0 \tag{7}$$

It's easy to prove that

$$\textit{duality gap} \stackrel{\text{def}}{=} \min_x \max_{u \geq 0} \mathcal{L}(x,u) - \max_{u \geq 0} \min_x \mathcal{L}(x,u) \geq 0 \tag{8}$$

which is called *weak duality*. If there exists a saddle point, then we have *strong duality*, where the inequality reduces to an equality.

If there exists some saddle point of $\mathcal{L}$, then the saddle point satisfies the following KKT conditions:

$$\bigtriangledown f + u^T \bigtriangledown g = 0 \tag{9}$$
$$u^T g = 0 \tag{10}$$
$$u \geq 0 \tag{11}$$

On the other hand, KKT conditions plus convexity and a constraint qualification imply that the duality gap is zero. Please note in particular that, in the primal problem, if $g(x)$ is strictly less than 0, then u must be 0.

2 Optimal Margin Classifiers

In the context of the optimal margin classifier, the primal problem is

$$\min \frac{1}{2} w^T w \tag{12}$$

such that

$$y_i(w^T x_i + b) \geq 1 \quad i = 1, \cdots, n \tag{13}$$

Then the Lagrangian is

$$\mathcal{L}(w, b, \alpha) = \frac{1}{2} w^T w - \sum_i \alpha_i (y_i (w^T x_i + b) - 1) \tag{14}$$

This gives the dual function:

$$\Theta(\alpha) = \min_{w,b} \mathcal{L}(w, b, \alpha) \tag{15}$$

At the saddle point, we have

$$\frac{\partial \mathcal{L}}{\partial w} = 0 \tag{16}$$

$$\frac{\partial \mathcal{L}}{\partial b} = 0 \tag{17}$$

which implies

$$w^* = \sum_i \alpha_i y_i x_i \tag{18}$$

$$\sum_i \alpha_i y_i = 0 \tag{19}$$

From the equation above, we can see that the optimal weight is a sum of all the input vectors x_i, which makes this method efficient in the case that there are few data points in a high dimension space. Now we can rewrite the dual function:

$$\begin{aligned} \Theta(\alpha) &= \mathcal{L}(w^*, b^*) \\ &= \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i^T x_j - \sum_i \alpha_i y_i b^* \\ &\quad - \sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i^T x_j + \sum_i \alpha_i \\ &= \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i^T x_j \end{aligned} \tag{20}$$

So the dual problem

$$\max \Theta(\alpha), \; s.t. \;\; \alpha \geq 0 \tag{21}$$

is a Quadratic Programming problem. Taking into account the complementary conditions:

$$\alpha_i (y_i (w^T x_i + b) - 1) = 0 \tag{22}$$

We can see that, only if (x_i, y_i) is on the border (we call it a support vector, see Figure 1), α_i may be non-zero. Then we can rewrite w^*

$$w^* = \sum_{support\ vectors} \alpha_i y_i x_i \tag{23}$$

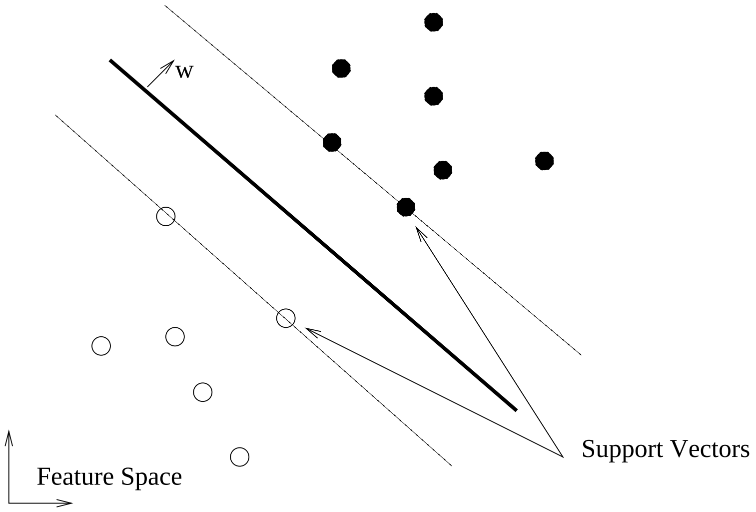

Figure 1: Support Vectors

To classify a new data point x, we juct need to calculate:

$$w^{*T}x + b = \sum_i \alpha_i y_i (x_i^T x) + b \tag{24}$$

and make a judgement based on its sign.

3 Misc

Training SVMs is a Quadratic Programming (QP) problem. Once the size of the data is huge (say $>$ $1,000,000$), it is infeasible to use any standard numerial QP techniques to solve it. Vapnik and Cortes [1] proposed an algorithm, called "chunking", to train SVMs, which is limited because, at the last step, it needs to solve a QP problem on all support vectors (it may be very big!). Platt [2] proposed a SVM training algorithm, called Sequential Minimal Optimization (SMO), which breaks the large QP problem into a series of smallest possible (only 2 point) QP problems that can be solved analytically. To my knowledge, SMO is the state-of-the-art training algorithm for SVMs.

By comparing SVMs with Neural Nets, we can see that support vectors can be regarded as the 'hidden nodes' in a Support Vector Machine (see Figure 2). After solving the QP problem, we actually get these support vectors. This makes us avoid the hard model selection problem in training Neural Nets!

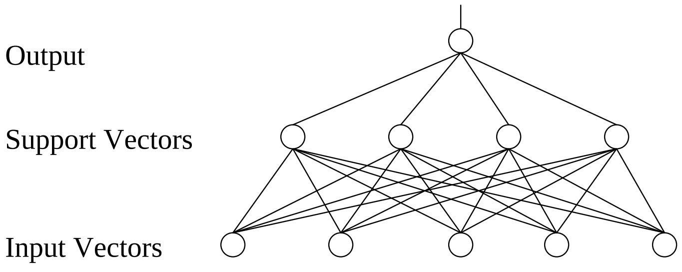

Figure 2: Representing a SVM as a NN

References

[1] C. Cortes and V. Vapnik, "Support Vector Networks", *Machine Learning*, 20:273-297, 1995.

[2] J. Platt, "Fast Training of Support Vector Machines using Sequential Minimal Optimization", *Advances in Kernel Methods - Support Vector Learning*, B. Schlkopf, C. Burges, and A. Smola, eds., MIT Press, 1999.