Note on Support Vector Machine

1. Why (1.9)-(1.10) is equivalent to (1.8) ?

In (1.8) , we can restrict to those w with $||w|| = 1$; that is in (1.6), the normal vector w has length 1. Then $w \cdot x_i$ is just the projection of x_I along the direction w. Plotting all the values of $w \cdot x_i, i = 1, \cdots, l$ together in a line, those with $y_i = 1$ must be separated from those with $Y_i = -1$.

$$* \ * \ * \ * \ | \ \cdots$$

So the value of b that maximizes (1.8) must be chosen so that the minimum is equal to a half of the difference between the smallest value from group 1 and the largest value from group 2. Therefore (1.8) is equivalent to

Subject to

$$||w|| = 1 \tag{1}$$

, maximize

$$\min_{i:y_i=1} w \cdot x_i - \max_{i:y_i=-1} w \cdot x_i \tag{2}$$

In contrast, (1.10) says that $\min_{i:y_i=1} w \cdot x_i$ is equal to $1 - b$ and $\max_{i:y_i=-1} w \cdot x_i$ is equal to $-1 - b$. So (1.10) is the same as setting the value from (2) to be equal to 2.

It should be clear now why the two problems are the same. Suppose w is a vector subject to (1.10), then the corresponding normalized vector $w/||w||$, should satisfy (1), and it will have $2/||w||$ as the value of (2). Therefore minimizing over (1.9) is the same as maximizing (2).

2. Convex hull. The dual optimization problem (1.16)(1.17) that the support vector machine method tries to solve can be interpreted as fining the shortest distance between the convex hull of points from one group and the convex hull of points from the other group.

A point , say A, in the convex hull of group 1, by definition, can be represented as

$$A = \sum_{i:y_i=1} \alpha_i x_i$$

with

$$\sum_{i:Y_i=1} \alpha_i = 1 \tag{3}$$

. A point B in the convex hull for group 2 can be defined in the same way. It is clear now the second summation term in (1.16) is equal to the squared distance$||A - B||^2$ between a point A from the first convex hull to a point B in the second convex hull :

$$||A - B||^2 = || \sum_{i:Y_i=1} \alpha_i x_i - \sum_{i:Y_i=-1} \alpha_i x_i||^2 = || \sum_{i=1}^{l} Y_i \alpha_i x_i||^2 = \sum_{i,j} \alpha_i \alpha_j Y_i Y_j (x_i \cdot x_j) \tag{4}$$

Furthermore, (3) and the analogous condition for group 2, imply (1.17). In fact, maximizing (1.16) is the same as minimizing (4). To see this, we first observe that for any α satisfying (1.17) and (3), to maximize (1.16) we can drop the first term in (1.16) because this term is now equal to 2 , which is a constant. Because of the minus sign , the maximization problem of (1.16) is the same as the minimizing (4).

Now suppose we already find the solution (A^*, B^*) for minimizing (4). We can scale this solution by a factor c, the corresponding value for (1.16) will become

$$2c - 1/2c^2||A^* - B^*||^2$$

Maximizing over c, the solution is $c = 2/||A^* - B^*||^2$ and (1.16) becomes $2/||A^* - B^*||^2$.

The connection is now clear. The maximization of (1.16) is the same as maximizing

$$2/||A - B||^2$$

which of course is the same as minimizing the squared distance $||A - B||^2$.

3. Does the factor of 1/2 in the second term of (1.16) matter ? The answer is No. Any positive number will lead to the same weight (up to a proportionality constant) solution. The intercept b in (1.18) is easy to find. An easy way to understand why is to think of rescaling the predictors by a constant. All you have to do is to pick up any support point (a point with the corresponding $\alpha_i > 0$) from each group (say x^*, x^{**}), compute their projection along the normal vector to the separation plane , and then take the average to be the value of -b. Thus $b = -(1/2)\sum_i \alpha_i x_i \cdot (x* + x**)$.

4. Another aspect can be drawn is that if the points from two groups are indeed separable, then we can also separate them after any affine transformation. The decision function (1.18) is not invariant under affine transformation though. Thus it is easy to see that any separation hyperplane can be the optimal solution with some appropriate affine transformation on x. In other words, with an appropriate choice of a matrix A and by setting the inner product (kernel) between two points x_i and x_j as $k(x_i, x_j) = x_i' A' A x_j$, we can make any separation hyperplane a solution of (1.33).

5. In general, no matter which space the data points are embedded into, as long as the dimension of the space is greater than the sample size (n) , the data points will occupy a subspace with n dimensions (if no points are collinear). When this happens, separation hyper-planes always exist. This explains why Support vector machine often "works well" when using a kernel of higher order. However, the ability to separate points in the training set does not guarantee its performance when generalized to the test set.

6. Which kernel to use ? This is still an unanswered question.

7. Connection to the concept of e.d.r. directions.

Consider the conditional density functions of x given Y, $f(x|Y = 1), f(x|Y = -1)$. Let $A = \{x : f(x|Y = 1) > 0\}$ and $B = \{x : f(x|Y = -1) > 0\}$. If A and B can be separated by a hyperplane, this would mean that we can find a direction w and a constant c so that $w^x > c$ for any $x \in A$ and $w'x < c$ for any $x \in B$. We can write $Y = g(w^x)$ where $g(u) = 1, for, u > c$ and $g(u) = -1, for\, u < c$. Thus u is an e.d.r. direction. If there are more than one hyperplanes that can separate A from B, then each of them can provide an e.d.r. direction. Thus in general, the e.d.r. space is not well-defined. Cook(1994) attempts to resolve this difficulty by considering the intersection of all e.d.r. space and call it the "central space". While this notion is useful in discussing regression graphics, it does not help much here.

Let A^* and B^* be the convex hull of A and B respectively. The convex hull of the sample points from each group, (denoted by C_1 and C_2), will be contained in A^* and B^* respectively. Thus from the training sample we can only find separation directions between C_1 and C_2.

8. Suppose that $f(x)$ is any function such that $f(x) = 1, for\, x \in A$ and $f(x) = -1, for, x \in B$. Suppose a kernel function $k(\cdot, \cdot)$ is used in applying support vector machine. Then if $f(\cdot)$ can be approximated well by linear combinations of $\sum_i \alpha_i y_i k(x_i, \cdot)$, then the resulting classification rule would be nearly optimal.

9. The ability of Support vector machine to handle large noises in the training sample may be questionable. More regularization conditions need to be imposed.