CSC 446 Notes: Lecture 8

1 Review Support Vector Machines

Goal: To solve equation:

$$\begin{aligned} \min_{w} \ & \left(\frac{1}{2}\|w\|^2 + C\sum_n \xi_n \right) \\ \text{s.t. } & y^n(w^T x^n + b) + \xi_n \geq 1 \\ & \xi_n \geq 0 \end{aligned}$$

where

$$x^n = [x_1, x_2, ..., x_K]^T, n \in 1, ..., N$$

This is a K-dimensional problem, which means the more features the data has, the more complicated to solve this problem.

In the meantime, this equation is equal to

$$\begin{aligned} \max_{\alpha} \ & \left(-\frac{1}{2}\sum_n \sum_m \alpha_n \alpha_m y^n y^m x^{nT} x^m + \sum_n \alpha_n \right) \\ \text{s.t. } & \alpha_n \geq 0 \\ & \alpha_n \leq C \end{aligned}$$

This is a N-dimensional problem, which means the more data points we include in the training set, the more time it takes to find the optimized classifier.

To train the classifier is actually to solve this problem inside the box of alpha.

According to KKT,

$$\begin{aligned} \lambda_i f_i(x) &= 0 \\ \lambda_i &\geq 0 \\ f_i(x) &\leq 0 \end{aligned}$$

As shown in the figure below,

$$w = \sum_n \alpha_n y^n x_n$$

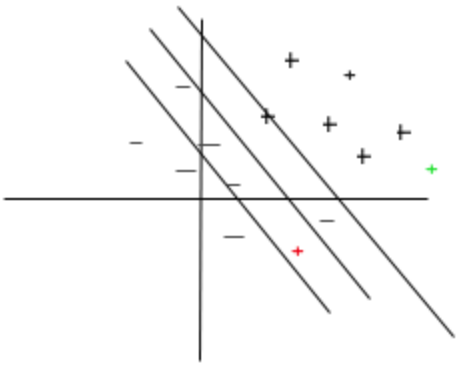
Points on the right side but not on the margin contribute nothing because alpha equals to 0. (The green point)

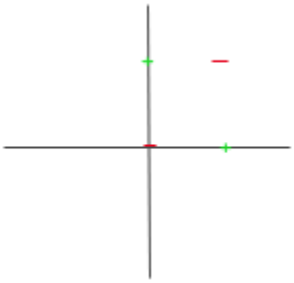
For points on the wrong side (the red point), alpha equals to C, and

$$\xi_n > 0$$

so they along with points on the margin contribute to the vector, but no point is allowed to contribute more than C.

SVM can train classifier better than naive bayes in the most of time, but since its still binary classification it is not able to deal with situation like this one below:





2 Kernel Function

Now when we look back, the classification formula is

$$Sign\left(w^T x\right) = Sign\left(\left(\sum_n \alpha_n y^n x_n\right)^T x\right) = Sign\left(\sum_n \alpha_n y^n \left(x^{nT} x\right)\right)$$

We can introduce Kernel Function K now, the simplest one is:

$$x^{nT} x = K\left(x^n, x\right)$$

Now the problem is transformed into:

$$\max_\alpha \left(-\frac{1}{2}\sum_n \sum_m \alpha_n \alpha_m y^n y^m K\left(x^n, x^m\right) + \sum_n \alpha_n\right)$$

where

$$K\left(x, y\right) = \phi\left(x\right)^T \phi\left(y\right)$$

for some ϕ.

The most commonly seen Kernel Functions are:

$$K\left(x, y\right) = x^T y$$

$$K\left(x, y\right) = \left(x^T y\right)^m$$

$$K\left(x, y\right) = e^{-c\|x-y\|^2}$$

Generally, Kernel function is a measure of how x and y are similar, then they are the same, it has the peak output.

3 Proof that ϕ exists

For a two dimensional

$$x = [x_1, x_2]^T y = [y_1, y_2]^T$$
$$K(x, y) = (x_1 y_1 + x_2 y_2)^m$$

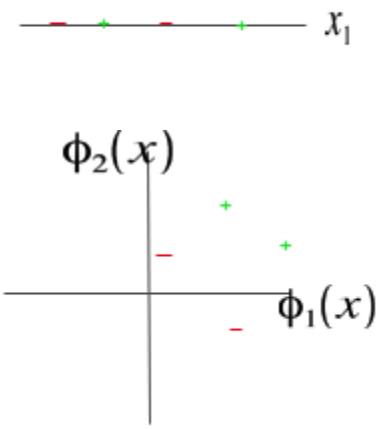
Let m = 2, then

$$K(x, y) = (x_1 y_1)^2 + (x_2 y_2)^2 + 2(x_1 y_1 x_2 y_2) = \phi(x)^T \phi(y)$$

Thus, we can conclude that

$$\phi(x) = \left[\sqrt{2} x_1 x_2, x_1^2, x_2^2\right]^T$$

Basically, ϕ transforms x from a linear space to a multi nominal space like shown below:



so that the points can be classified.

For

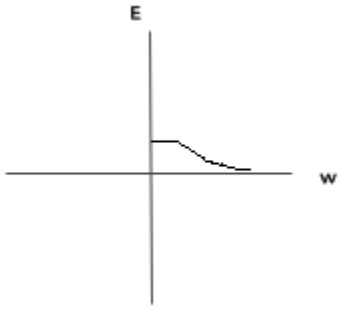
$$K(x, y) = e^{-\|x-y\|^2}$$

because we have

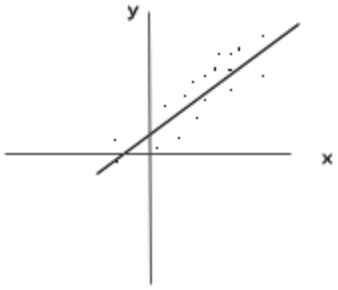
$$e^x = 1 + x + \frac{x^2}{2} + ...$$

it transforms feature into a infinite dimensional space. Generally Kernel Functions lead to more dimension of w which is K-dimensional so solve dual is more practical.

For perceptron, its error term and w have a relationship drawn as below: (not convex) so that we cant



do the same thing to perceptron.



4 Regression

When we are predicting output we actually have a space like this: The line is the prediction line, the points around it are the data set we have. We predict y with formula:

$$\hat{y} = w^T x$$
$$w = \left(X^T X\right)^{-1} X^T \vec{y}$$

its known as linear regression. The goal is to

$$\min_w \sum_n \frac{1}{2} \|\hat{y}^n - y^n\|^2$$

which leads us to Support Vector Regression:

$$\begin{aligned} \min_w \ & \frac{1}{2}\|w\|^2 + C\sum_n \left(\xi_n + \hat{\xi}_n\right) \\ \text{s.t. } & y^n - w^T x - \xi_n \leq \epsilon \\ & -(y^n - w^T x) - \hat{\xi}_n \leq \epsilon \\ & \xi_n \geq 0 \\ & \hat{\xi}_n \geq 0 \end{aligned}$$

Yu Zhong 2/12; DG 2/13