Machine Learning notes

LWang

January 10, 2017

1 Support Vextor Machines(SVMs) and Optimization objective

The two key ideas of support vector machines are

- The maximum margin solution for a linear classifier.
- The "kernel trick"; a method of expanding up from a linear classifier to a non–linear one in an efficient manner.

1.1 Decesion boundary

Alternative view of logistic regression

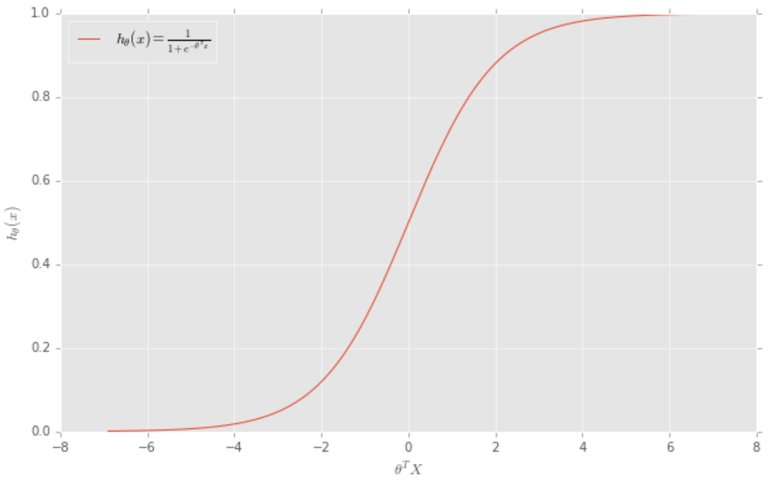

Figure 1: Logistic regression

$$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$$

One example of cost function

$$\begin{aligned} J_\theta &= -(y \log h_\theta(x) + (1-y)\log(1-h_\theta(x))) \\ &= -y \log \frac{1}{1+e^{-\theta^T x}} + (1-y)\log\big(1 - \frac{1}{1+e^{-\theta^T x}}\big) \end{aligned}$$

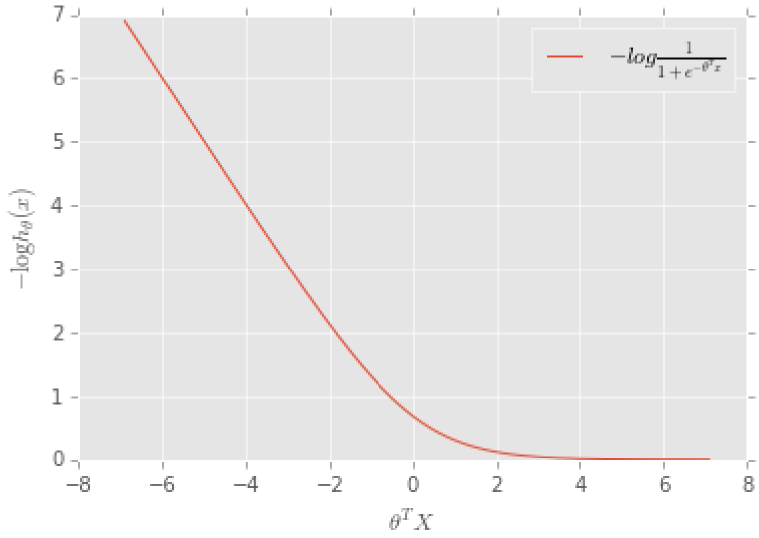

Figure 2: Cost function

If y=1, we want $h_\theta(x) \approx 1$, so $\theta^T x \gg 0$.

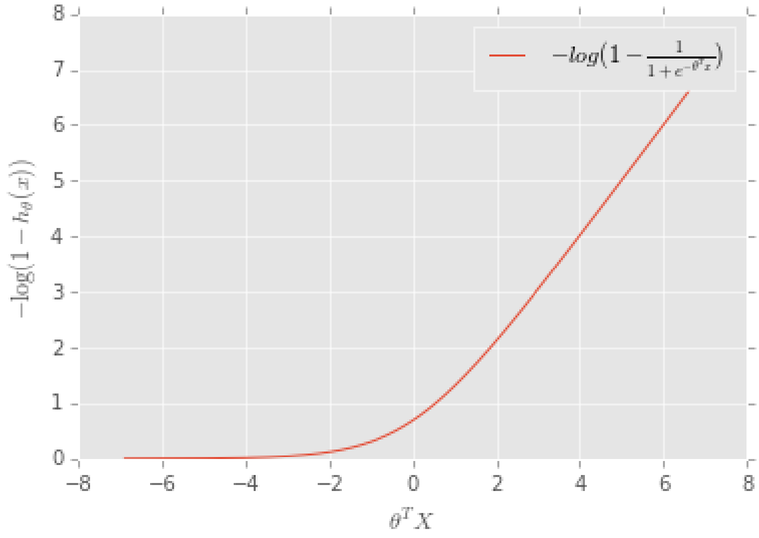

Figure 3: Cost function

If y=0, we want $h_\theta(x) \approx 0$, then $\theta^T x \ll 0$.

Running the perceptron learning algorithm on the training data, with the decision boundary ($\theta^T x = 0$) we will obtain θ that would classify the training examples correctly. However, there is a whole version space of weight vectors that yield to the same classification of the training points. The SVMs algorithm chooses a particular weight vector, that gives rise to the "maximum margin" of separation. The goal of Logistic regression is to minimize the cost function

$$min_\theta \left\{ \frac{1}{m} \sum_{i=1}^{m} [y^{(i)}(-\log(h_\theta(x^{(i)}))) + (1-y^{(i)})\big(-\log(1-h_\theta(x^{(i)}))] + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2 \right\}$$

Rewrite the above equation in SVM (Support vector machine) hypothesis

$$min_\theta \left\{ C\sum_{i=1}^{m} \big[y^{(i)}cost_1(\theta^T x^{(i)}) + (1-y^{(i)})cost_0(\theta^T x^{(i)})] + \frac{\lambda}{2}\sum_{j=1}^{n}\theta_j^2 \right\}$$

1.2 Widest street approach

In binary classification problems, we want to label a dataset as positive(+1) or negative examples(-1), and suppose we have built a logistic regression hypothesis and run the perceptron algorithm to linearly separate the data, one can obtain a bunch of parameters that can do the job(e.g. Fig 9), but there exists a particular solution that seperates the positive and negative examples as wide as possible, we call this is the widest street approach, and the decesion boundary is the median line of the two "streets", where variables lie on the two lines.

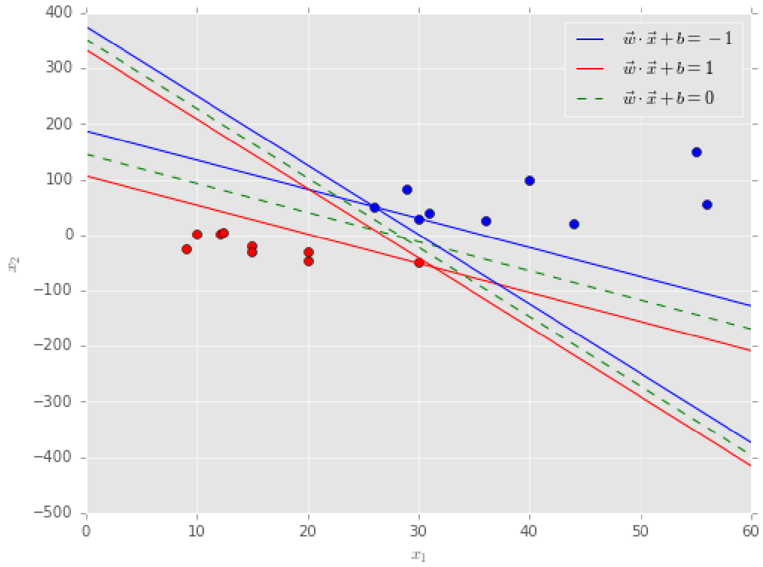

Figure 4: Decesion boundary

Let's denote w and intercept b as our parameters, and the vector that perpendicular to the median line of the "streets" is $\vec{w}$.
The decesion rule works like this: For a unknown vector $\vec{u}$, if

$$\vec{w}\cdot\vec{u}+b\geq 0 \tag{1.2.1}$$

the we say this new example $\vec{u}$ belongs to the positive group. For the existed positive examples, we have

$$\vec{w}\cdot\vec{x_+}+b\geq 1 \tag{1.2.2}$$

while the negative examples have the property

$$\vec{w}\cdot\vec{x_-}+b\leq -1 \tag{1.2.3}$$

To be mathematically convenient, we introduce a new variable y, such that y=1 if positive examples, and $y=-1$ for negative examples.

Therefore, we can combine eq(1.2.2) and eq(1.2.3) as one equation, this is for example vectors.

$$y_i(\vec{w}\cdot\vec{x_i}+b)\geq 1 \tag{1.2.4}$$

1.3 Optimization

The goal is to find the parameters w and b to maximaize the width of the street.

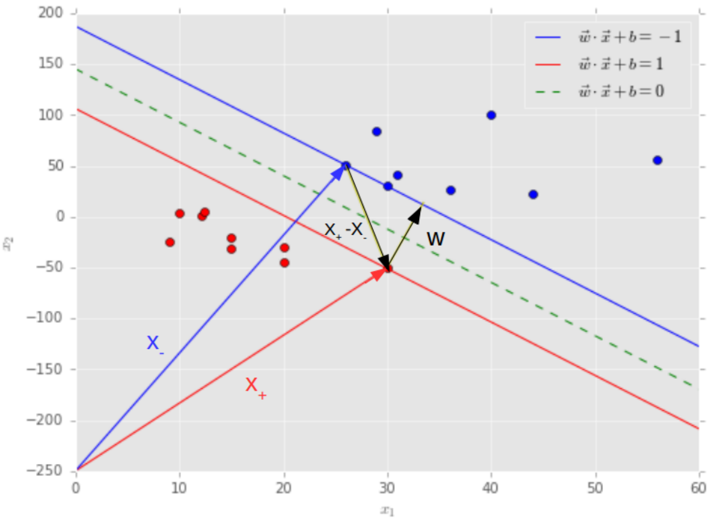

Figure 5: Width of street

The width can be computed as $(x_+ - x_-)\frac{\vec{w}}{\|\vec{w}\|} = \frac{2}{\|\vec{w}\|}$ (using eq(1.2.4)). Now the question transforms to minimize $\|\vec{w}\|$, for the sake of mathematical convenience, it is equivalent to minimize $\frac{1}{2}\|\vec{w}\|^2$.

Since we have constraints eq(1.2.3) and the goal of minimizing $\frac{1}{2}\|\vec{w}\|^2$, we now employ Lagrange multiplier.

$$L = \frac{1}{2}\|\vec{w}\|^2 - \sum_i \alpha_i \left[y_i(\vec{w}\cdot\vec{x_i} + b) - 1 \right] \tag{1.3.1}$$

The two partial derivatives:

$$\frac{\partial L}{\partial \vec{w}} = \vec{w} - \sum_i \alpha_i y_i \vec{x_i} = 0 \tag{1.3.2}$$

$$\frac{\partial L}{\partial b} = \sum_i \alpha_i y_i = 0 \tag{1.3.3}$$

then we have

$$\vec{w} = \sum_i \alpha_i y_i \vec{x_i} \tag{1.3.4}$$

$$\sum_i \alpha_i y_i = 0 \tag{1.3.5}$$

Next we subsitute eq(1.3.4) back into to eq(8.3.1),

$$L = \frac{1}{2}\sum_i \alpha_i y_i \vec{x_i} \sum_j \alpha_j y_j \vec{x_j} - \sum_i \alpha_i \left[y_i (\sum_j \alpha_j y_j \vec{x_j} \vec{x_i}) \right] - \sum_i \alpha_i y_i b + \sum_i \alpha_i$$

then using eq(1.3.5)

$$\begin{aligned} L &= \sum_i \alpha_i - \frac{1}{2}\sum_i \alpha_i y_i \vec{x_i} \sum_j \alpha_j y_j \vec{x_j} \\ &= \sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j \vec{x_i} \vec{x_j} \end{aligned} \tag{1.3.6}$$

It turns out that the optimization depends on the dot product of sample vectors. And of course, we can rewrite the decesion boundary eq(1.2.2) as

$$\sum_i \alpha_i y_i \vec{x_i} \vec{u} + b \geq 0 \tag{1.3.7}$$

For an unknown vector $\vec{u}$, as long as e1(8.4.1) holds, this example can be labeled as positive.

1.4 Kernels

The Kernel trick is used to transform non–linear space into linear space. For linear case, we tend to come up a function like

$$\begin{aligned} \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_n x_n \geq c \rightarrow class1 \\ \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots + \theta_n x_n \leq c \rightarrow class2 \end{aligned}$$

In the case of non–linear separation, one can have hypothesis function of

$$\theta_0 + \theta_1 x_1^2 + \theta_2 x_2^3 \ldots$$

To make the computation more easier, we can introduce new variables like $f_1 = x_1^2$, this can provide linearly separable space. More generally, we use a kernel function totransfer variables into other space which is linear separable,

$$K(x_i, x_j) = \Phi(x_i)\Phi(x_j) \tag{1.4.1}$$

SVM parameters:

- C:$\frac{1}{\lambda}$
  - Large C: lower bias, high variance
  - Small C: Higher bias, low variance
- σ^2
  - Large σ^2,fearures vary more smoothly Higher bias, lower variance
  - small σ^2,fearures vary less smoothly lower bias, higher variance