Week 7: 6/30/17

7. Support Vector Machines

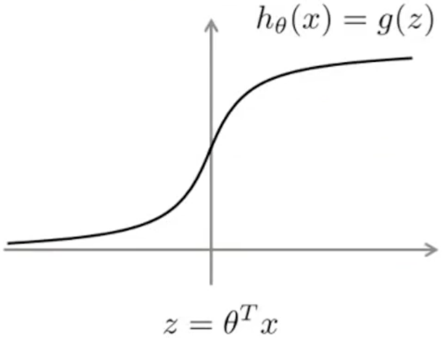
By now, we've seen a range of different learning algorithms. With ***supervised learning***, the performance of many algorithms are expected to be very similar. It doesn't really matter whether we're using algorithm A vs. B, but what does matter is the amount of data used to create the algorithms, as well as your skill in applying them. For instance, the choice of *features* provided to the learning algorithms and how well chosen then colorization parameter is.

There's one more algorithm that is very powerful and is very widely used both within industry and academia, and that's called the ***support vector machine (SVM)***. Compared to both logistic regression and neural networks, the Support Vector Machine, sometimes provides a cleaner, and more powerful way of learning complex non-linear functions.

7.1. Optimization Objective.

In order to describe the SVM, we start with *logistic regression*, and show how we can modify it to obtain an SVM. So in logistic regression, we have our familiar form of the hypothesis there and the associated sigmoid activation function as well.

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



Now, if we're given an *example* data set, in order to do a good job classifying the data, we would like to have the following:

– If $y = 1$, we want $h_\theta(x) \approx 1$, $\theta^\intercal x \gg 0$

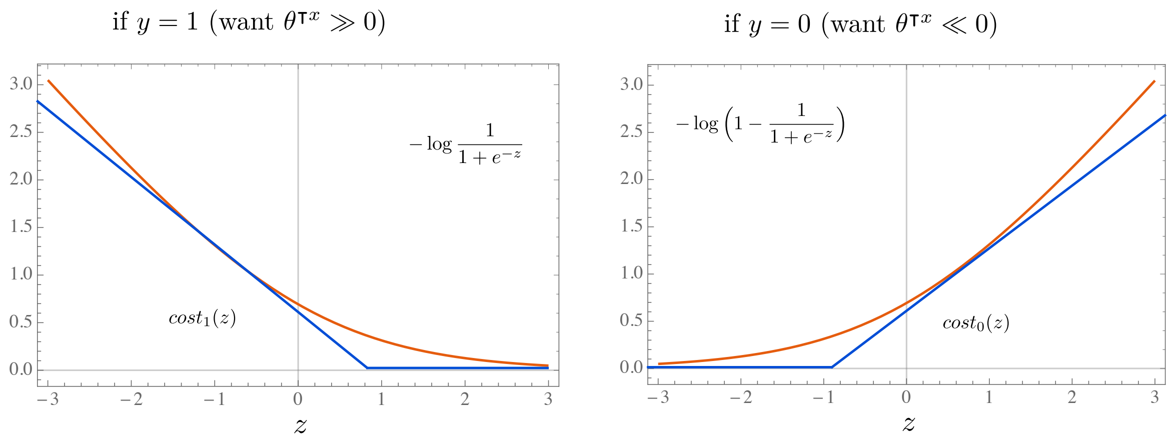
– If $y = 0$, we want $h_\theta(x) \approx 0$, $\theta^\intercal x \ll 0$

Alternative View:

Now, if we take a look at a single example, (x, y), the cost function is:

$$\begin{aligned} Cost &= -\left(y \log h_\theta(x) + (1-y)\log(1 - h_\theta(x))\right) \\ &= -y \log\left(\frac{1}{1+e^{-\theta^\intercal x}}\right) - (1-y)\log\left(1 - \frac{1}{1+e^{-\theta^\intercal x}}\right) \end{aligned}$$

Then, for $y = 1$ & $y = 0$, the cost function value gradually decreases towards zero on the low side. However, if we redefine this cost function to remove the gradual change, we can save computational effort. The new cost functions are now shown below in blue.



if $y = 1$ (want $\theta^{\intercal x} \gg 0$)

if $y = 0$ (want $\theta^{\intercal x} \ll 0$)

Logistic Regression:

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

simply substituting in the functions $cost_1$ & $cost_0$:

$$\min_\theta \frac{1}{m}\sum_{i=1}^{m} y^{(i)}\ cost_1(\theta^\intercal x^{(i)}) + (1-y^{(i)})\ cost_0(\theta^\intercal x^{(i)}) + \frac{\lambda}{2m}\sum_{j=1}^{n}\theta_j^2$$

Then, to conform with standard SVM notation, we make the following changes:

- Remove the $1/m$ term,
- The regularization term, λ is replaced with C and is now imposed on the first term.

$$A + \lambda B \qquad (\textit{Logistic Regression})$$
$$CA + B \qquad (\textit{SVM})$$

Here, $C = \frac{1}{\lambda}$, or is analogous to $1/\lambda$. Then, we have our overall optimization objective function:

SVM:

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

Then, our hypothesis is written as:

Hypothesis:

$$h_\theta(x) = \begin{cases} 1 & \text{if } \theta^\intercal x \geq 0 \\ 0 & \text{otherwise} \end{cases}$$