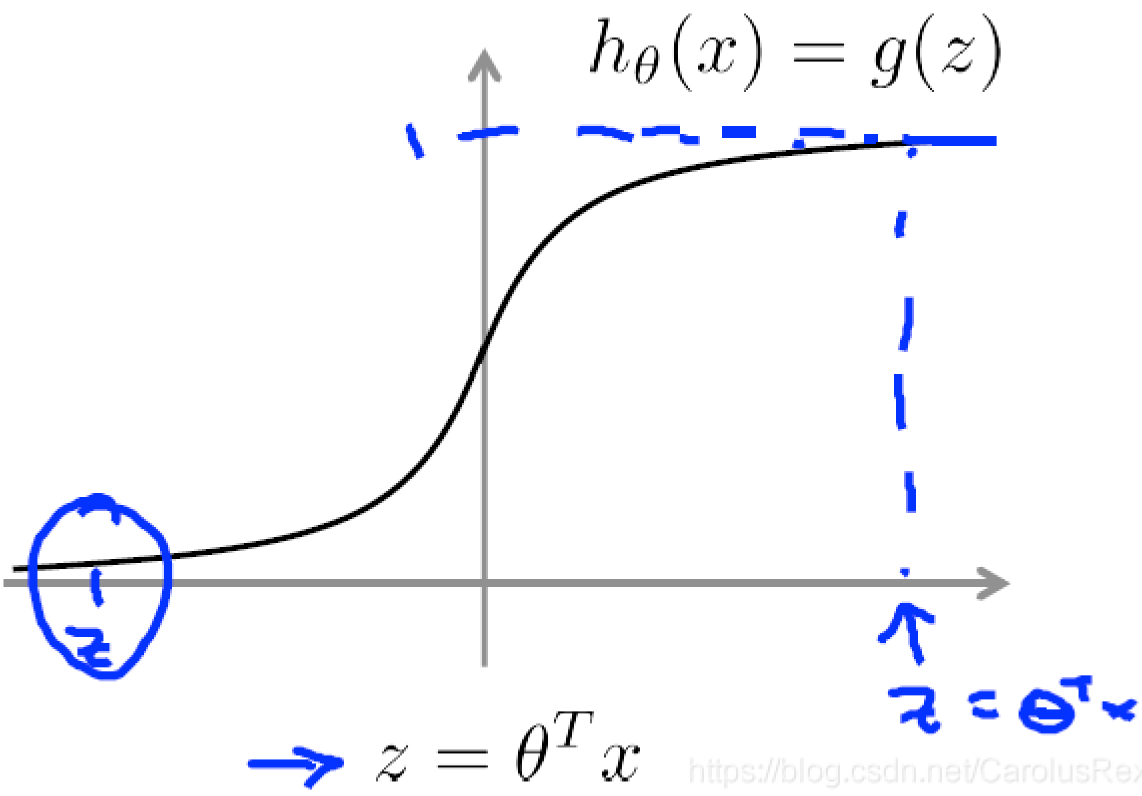
Suppore Vector Machines

- Optimization objective
- Large Margin Intuition
- SVM derivation
- Multi-class classification

Optimization objective

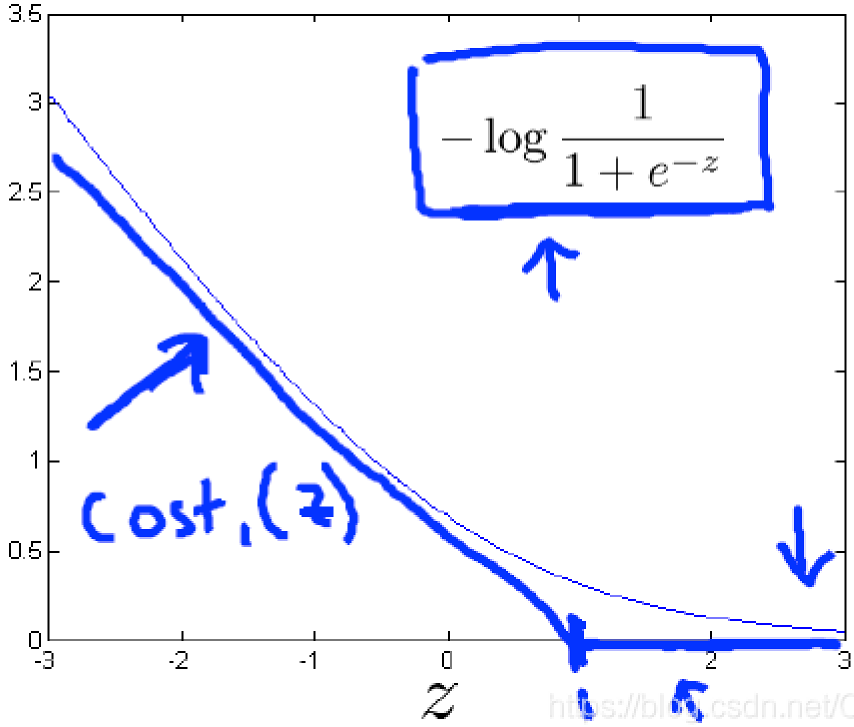
alternative view of logistic regression
$h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$



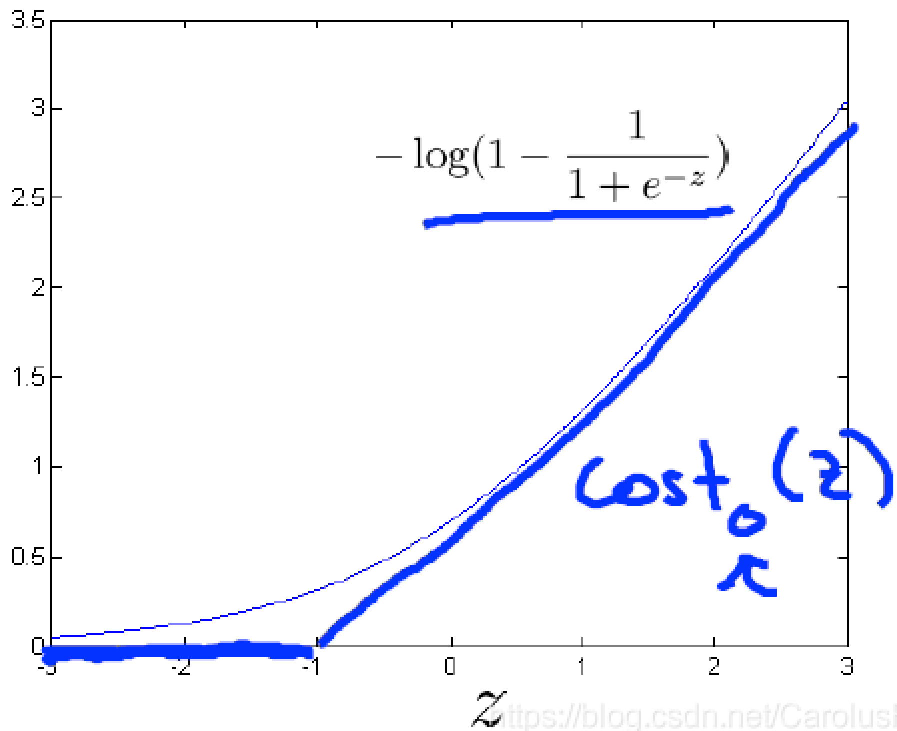
If $y == 1$, we want $h_\theta(x) \approx 1,\ \theta^T x >> 0$;
If $y == 0$, we want $h_\theta(x) \approx 0,\ \theta^T x << 0$.
And we use linear to replace the cost function, like fellow.

If $y = 1$ (want $\theta^T x \gg 0$):

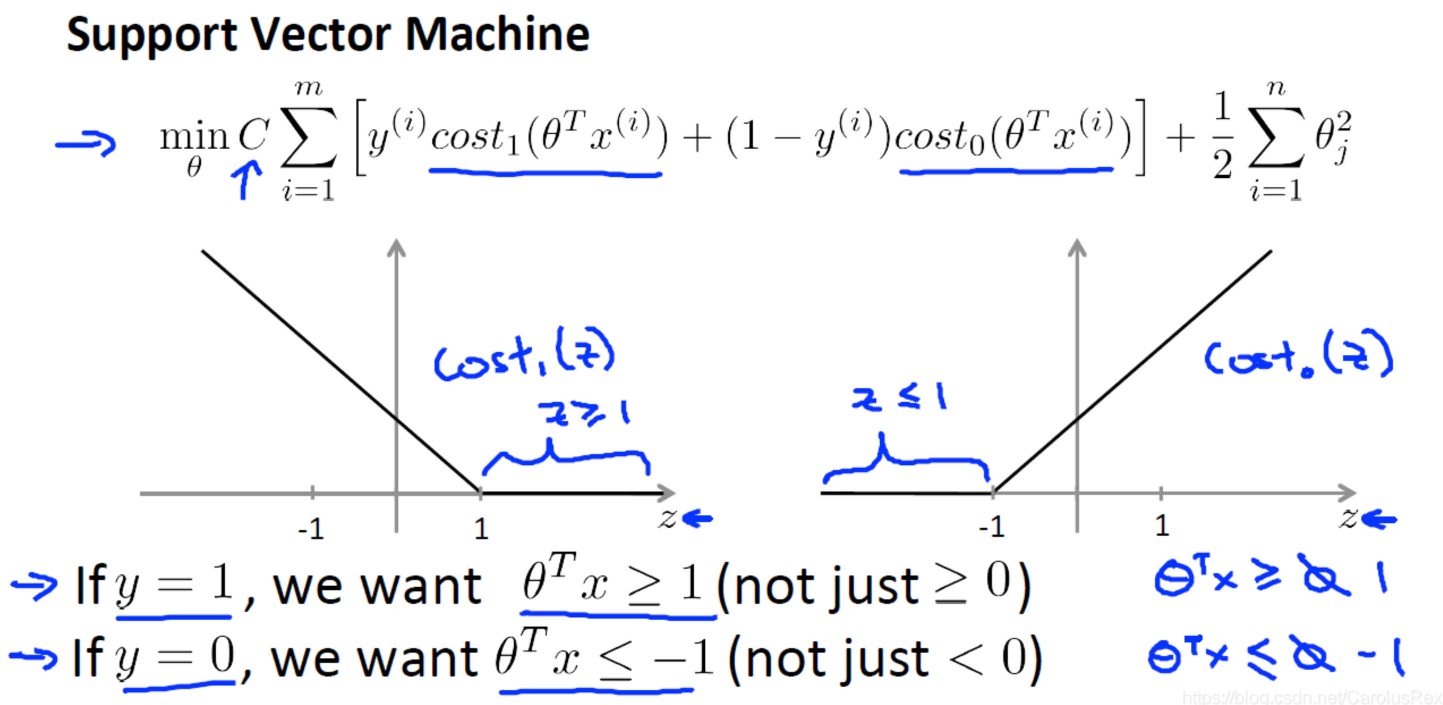
$z = \theta^T x$



If $y = 0$ (want $\theta^T x \ll 0$):



Large Margin Intuition



Here the cost function J has the regularization.
$C = \frac{1}{\lambda}$
When C is large, we want $C \cdot 0$ and $min \frac{1}{2} \sum \theta_j^2$, which is prone to overfitting.
If C is large, then we get higher variance/lower bias
If C is small, then we get lower variance/higher bias

The other parameter we must choose is σ^2 from the Gaussian Kernel function:

With a large σ^2, the features fi vary more smoothly, causing higher bias and lower variance.

With a small σ^2, the features fi vary less smoothly, causing lower bias and higher variance.

SVM derivation

SVM-1——derivation of target and convex optimization(blog) or SVM-1(github)
SVM-2——nonlinear, kernel and SMO derivation(blog) or SVM-2(github)

Mercer's Theorem: 任何半正定矩阵都能作为核函数。

Multi-class classification

one-vs-all method, pick class i with the largest $(\Theta^{(i)})^T x$.

If n is large (relative to m), then use logistic regression, or SVM without a kernel (the "linear kernel")

If n is small and m is intermediate, then use SVM with a Gaussian Kernel

If n is small and m is large, then manually create/add more features, then use logistic regression or SVM without a kernel.

In the first case, we don't have enough examples to need a complicated polynomial hypothesis. In the second example, we have enough examples that we may need a complex non-linear hypothesis. In the last case, we want to increase our features so that logistic regression becomes applicable.

Note: a neural network is likely to work well for any of these situations, but may be slower to train.