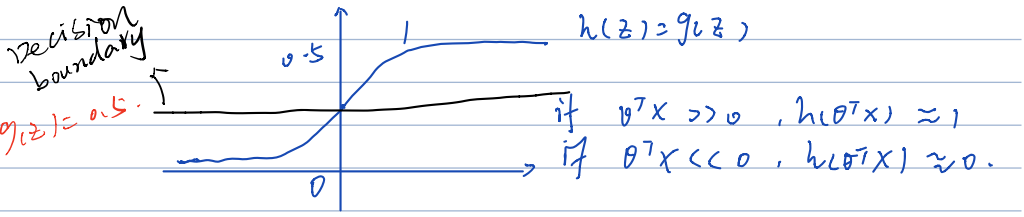
SVM

logistical Regression:

$$h(z) = \frac{1}{1+e^{-z}}$$

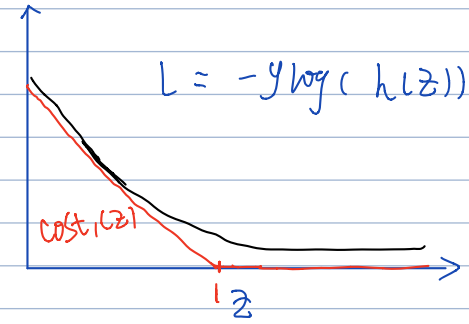
$$z = \theta^T x$$



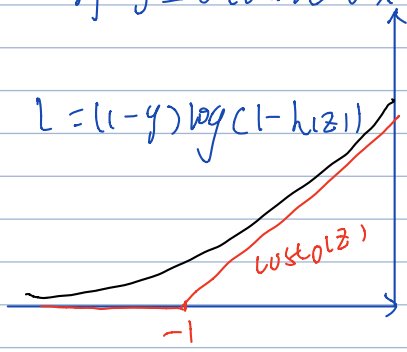
Cost Function:

$$L_i = -y^{(i)} \log(h_\theta(x^{(i)})) - (1-y^{(i)}) \log(1-h_\theta(x^{(i)})) \qquad h_\theta(x) = \frac{1}{1+e^{-\theta^T x}}$$

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



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



$$L = \frac{1}{m} \sum_{i=1}^{m} L_i + \frac{\lambda}{2m} \sum_{j=1}^{n} \theta_j^2$$

Logistic Regression Function:

$$\min \frac{1}{m} \left[\sum_{i=1}^{m} y^{(i)} \underbrace{(-\log h_\theta(x^{(i)}))}_{\text{cost}_1(z)} + (1-y^{(i)}) \underbrace{(-(\log(1-h_\theta(x^{(i)}))))}_{\text{cost}_0(z)} \right] + \frac{\lambda}{2m} \sum_{j=1}^{n} \theta_j^2$$

Support vector machine:

$$\min_\theta \frac{1}{m} \left[\sum y^{(i)} \cdot \text{cost}_1(\theta^T x) + (1-y^{(i)}) \text{cost}_0(\theta^T x) \right] + \frac{\lambda}{2m} \sum_{i=1}^{n} \theta_j^2$$

$$\Downarrow$$

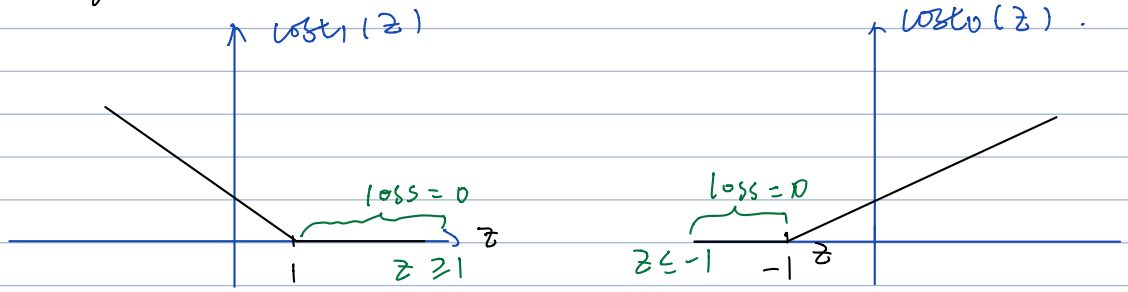
$$\min_\theta C \left[\sum y^{(i)} \text{cost}_1(\theta^T x) + (1-y^{(i)}) \text{cost}_0(\theta^T x) \right] + \frac{1}{2} \sum \theta_j^2$$

$C = \frac{1}{\lambda}$ → the parameter optimized by SVM.

SVM hypothesis:

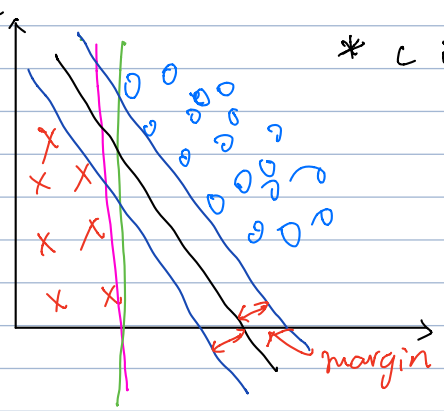
$$h_\theta(x) = \begin{cases} 1, & \theta^T x \geq 0 \\ 0, & \text{otherwise} \end{cases}$$

large margin intuition:



If $y=1$, we want $\theta^T x \geq 1$ (not just $\theta^T x \geq 0$)
If $y=0$, we want $\theta^T x \leq -1$ (not just $\theta^T x \leq 0$)
↳ safty margin. / safety factor

linear Separable



* C is very large.

当C不是很大的时候，SVM可以忽略一些异常点的影响，得到更好的效果。

甚至当data不是线性可分的时候，SVM也可以给出相对合理的决策。

What we minimize in the SVM is

$$\min_\theta \frac{1}{2}\sum_{j=1}^{n}\theta_j^2$$

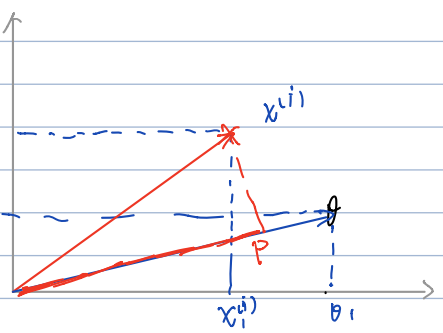
SVM Decision Boundary:

$$\min_\theta \frac{1}{2}\sum_{j=1}^{n}\theta_j^2 = \frac{1}{2}(\theta_1^2+\theta_2^2) = \frac{1}{2}\sqrt{(\theta_1^2+\theta_2^2)}^2 = \frac{1}{2}\|\theta\|^2$$

$\|\theta\|$

$$\theta = \begin{bmatrix}\theta_0\\ \theta_1\\ \theta_2\end{bmatrix} \quad \theta_0 = 0$$

s.t. $\theta^T x \geq 1$, $y^{(i)}=1$
$\theta^T x \leq -1$, $y^{(i)}=0$



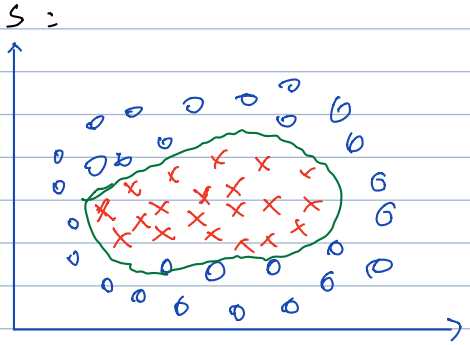
If $n=2$. $\theta_0 = 0$
↓
the boundary has to pass origin (0,0)

$$\theta^T x^{(i)} = p^{(i)} \cdot \|\theta\| = \theta_1 x_1^{(i)} + \theta_2 x_2^{(i)}$$

s.t. $p^{(i)} \cdot \|\theta\| \geq 1$, $y^{(i)}=1$
$p^{(i)} \cdot \|\theta\| \leq -1$, $y^{(i)}=0$

使 $p^{(i)}$ 最大，即样本到 Decision boundary 的距离最大，让样本 vector 投影在 parameters vector 上的值最大，即能找到 θ minimize $\|\theta\|$，使 $p^{(i)}\|\theta\|$ 满足约束。

kernels :

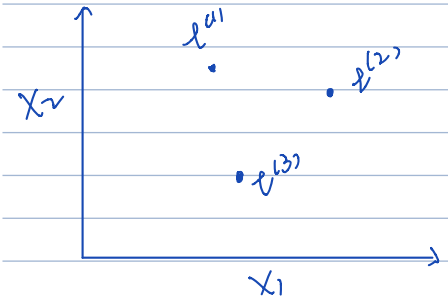


Given the dataset, our model might be
$\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2 + \theta_4 x_1^2 + \cdots$

If we use a set of new features to replace those : $f_1 = x_1$, $f_2 = x_2$, $f_3 = x_1 x_2$. then,

$h_\theta(x) = \theta_1 f_1 + \theta_2 f_2 + \theta_3 f_3 + \cdots$

How to find better features :
use the similarity between x and a bounch of landmarks $l^{(1)}, l^{(2)}, l^{(3)}$.



Given x :

$f_1 = \text{similarity}(x, l^{(1)}) = \exp\left(-\frac{\|x - l^{(1)}\|^2}{2\sigma^2}\right)$

$f_2 = \text{similarity}(x, l^{(2)}) = \exp\left(-\frac{\|x - l^{(2)}\|^2}{2\sigma^2}\right)$

$\vdots$

$\|x - l^{(1)}\|^2 = \sum_{j=1}^{n} (x_j - l_j^{(1)})^2$ → x中所有特征与 $l^{(1)}$ 的距离的和.

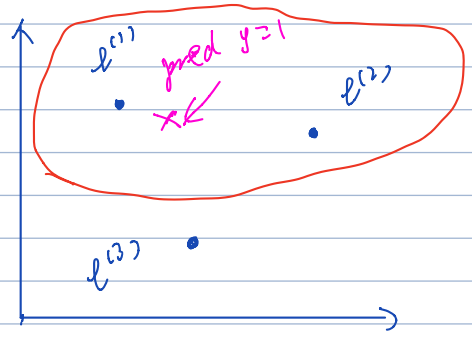
核函数，此处的为高斯核函数 (Gaussian Kernel)

若 x is close to landmarks $l^{(1)}$: $f_1 \approx e^{-0} = 1$

若 x is far from $l^{(1)}$: $f_1 \approx e^{-\infty} = 0$.

注：只有当 x 与 $l^{(i)}$ 重合时 f_i 才具有最大值，且改变的速率受 σ^2 的影响.

Example :



predict "1" when

$h_\theta(x) = \theta_0 + \theta_1 f_1 + \theta_2 f_2 + \theta_3 f_3 \geq 0$.

$\theta_0 = -0.5 \quad \theta_1 = \theta_2 = 1 \quad \theta_3 = 0$.

$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_0 + \theta_3 x_0 = -0.5 \geq 0$.

$\Rightarrow y = 1$

在训练中我们采取的特征为是训练实例本身，并是通过 kernel 计算出来的.

How to choose/set landmarks :

training examples : $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}) \cdots, (x^{(m)}, y^{(m)})$

训练集中有 m 个实例，则我们选择 m 个地标，并令 $l^{(1)} = x^{(1)}$, $l^{(2)} = x^{(2)}$, $\cdots$, $l^{(m)} = x^{(m)}$. 我们得到的新的特征是建立在原有特征与训练集中所有其他特征之间的距离的基础之上的，即：

Given example x.

$f_1 = \text{similiarity}(x, l^{(1)})$

$f_2 = \text{similiarity}(x, l^{(2)})$

$\vdots$

Given $(x^{(i)}, y^{(i)})$

$$f^{(i)} = \begin{bmatrix} f_0^{(i)} = 1 \\ f_1^{(i)} = sim(x^{(i)}, l^{(1)}) \\ f_2^{(i)} = sim(x^{(i)}, l^{(2)}) \\ \vdots \\ f_i^{(i)} = sim(x^{(i)}, l^{(i)}) = 1 \\ \vdots \\ f_m^{(i)} = sim(x^{(i)}, l^{(m)}) \end{bmatrix}$$

Hypothesis: Given x, compute features $f \in R^{m+1}$

predict "$y=1$" if $\theta^T f \geq 0$ $\quad \theta^T f = \theta_0 + \theta_1 f_1 + \theta_2 f_2 + \cdots$

Loss Function: $\quad z = \theta^T f$.

$$\min C\left[\sum y^{(i)} \text{cost}_1(\theta^T f^{(i)}) + (1-y^{(i)}) \text{cost}_0(\theta^T f^{(i)})\right] + \frac{1}{2}\sum_{j=1}^{n} \theta_j^2$$

在实际应用中，我们需要对正则项进行微调：

$$\sum_{j=1}^{n} \theta_j^2 = \theta^T \theta \Rightarrow \theta^T M \theta$$ ：M是根据 kernel 函数定义的一个矩阵.

High bias / variance in SVM:

$C = \frac{1}{\lambda}$

- C is large $\rightarrow \lambda$ is small $\Rightarrow$ lower bias, high variance $\Rightarrow$ overfitting.
- C is small $\Rightarrow \lambda$ is large $\Rightarrow$ high bias, low variance $\Rightarrow$ underfit.

σ^2

- Large σ^2  : f_i vary more smoothly $\Rightarrow$ High bias, low variance $\Rightarrow$ underfit.
- Small σ^2  : f_i vary less smoothly $\Rightarrow$ low bias, high variance $\Rightarrow$ overfit

SVM 的代价函数是凸函数，不存在局部最小值，所以其优化是一种凸优化问题.

A linearly separable dataset can usually be separated by many different lines. Varying the parameter C will cause the SVM's decision boundary to vary among those possibilities. for example, very large C might learn larger values of θ in order to increase the margin on certain examples.