

Support Vector Machines 支持向量机

Support vector machine

Logistic regression:

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

Support vector machine:

$$\min_{\theta} \underbrace{\cancel{\frac{1}{m} \sum_{i=1}^m y^{(i)} cost_1(\theta^T x^{(i)}) + (1-y^{(i)}) cost_0(\theta^T x^{(i)})}}_A + \frac{\lambda}{2} \sum_{j=0}^n \theta_j^2$$

$$\min_u \underbrace{\frac{(u-s)^2 + 1}{10}}_{\text{red}} \rightarrow u=5 \quad \left| \begin{array}{l} A + \frac{\lambda}{2} B \Leftarrow \\ C = \frac{1}{\lambda} \end{array} \right.$$

$$\min_u \underbrace{10(u-s)^2 + 10}_{\text{red}} \rightarrow u=5 \quad \left| \begin{array}{l} C = A + B \Leftarrow \\ C = \frac{1}{\lambda} \end{array} \right.$$

这里首先把常量m去掉，然后用新的变量C代替lambda的作用，我们可以得到一种新的算法：支持向量机

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

支持向量机做出假设方法如下：

SVM hypothesis

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

Hypothesis:

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

Large Margin Classifier 大间距分离法

是一个特殊的栗子，当正则化参数C非常大，并且数据是线性可分的，那么支持向量机将会以最大距离分开两种类型，但是这种算法会受到单个特殊样本的影响从而产生非常不一样的结果，因此这只不过是一种特殊情况用来说明C非常大的一种情形

支持向量机的决策边界

SVM Decision Boundary

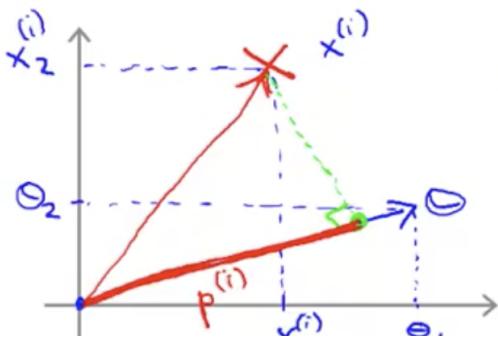
$$\min_{\theta} \frac{1}{2} \sum_{j=1}^n \theta_j^2 = \frac{1}{2} (\Theta_1^2 + \Theta_2^2) = \frac{1}{2} (\underbrace{\Theta_1^2 + \Theta_2^2}_{= \|\theta\|^2})^2 = \frac{1}{2} \|\theta\|^2$$

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

Simplification: $\Theta_0 = 0$, $n=2$

$$\Theta^T x^{(i)} = ?$$

\uparrow
 $U^T V$



$$\Theta^T x^{(i)} = \boxed{p^{(i)} \cdot \|\theta\|} \leftarrow$$

$$= \Theta_1 x_1^{(i)} + \Theta_2 x_2^{(i)} \leftarrow$$

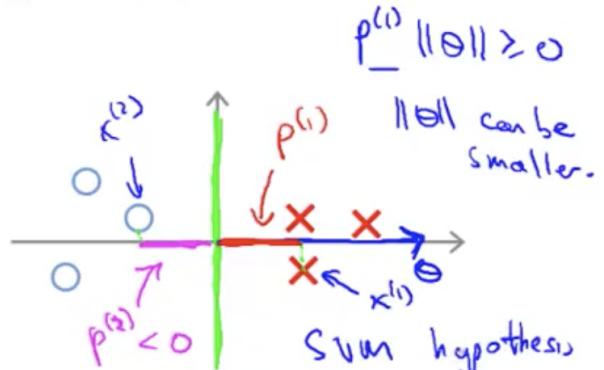
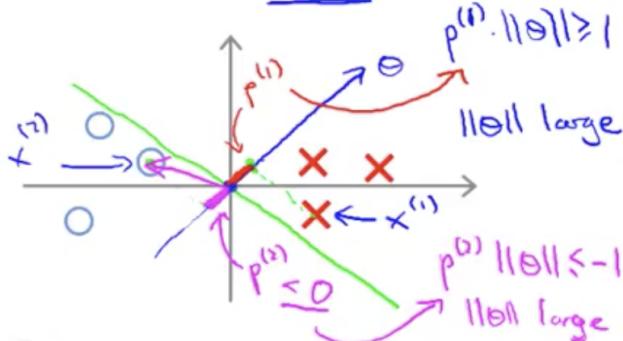
SVM Decision Boundary

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

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

where $p^{(i)}$ is the projection of $x^{(i)}$ onto the vector θ .

Simplification: $\Theta_0 = 0$



上图中绿色的线就是决策边界，而向量theta总是垂直于决策边界。

我们的目标是让theta最小，因此SVM会自然地选择右边的情况，这样样本在theta上的投影p会相对大一些，因此theta就会达到最小值

这里为了简化说明，我们设theta0 = 0，这意味着决策边界会通过原点

$$K(x, z) = \exp\left(-\frac{\|x - z\|^2}{2\sigma^2}\right)$$

这个函数中，如果x和z很相近 ($\|x - z\| \approx 0$)，那么核函数值为1，如果x和z相差很大 ($\|x - z\| \gg 0$)，那么核函数值约等于0。由于这个函数类似于高斯分布，因此称为高斯核函数，也叫做径向基函数(Radial Basis Function 简称RBF)。它能够把原始特征映射到无穷维。

既然高斯核函数能够比较x和z的相似度，并映射到0到1，回想logistic回归，sigmoid函数可以，因此还有sigmoid核函数等等。

Kernels and Similarity

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

If $x \approx l^{(1)}$:

$$f_1 \underset{\uparrow}{\approx} \exp\left(-\frac{\underset{\downarrow}{0}}{2\sigma^2}\right) \approx 1$$

$$\begin{aligned} l^{(1)} &\rightarrow f_1 \\ l^{(2)} &\rightarrow f_2 \\ l^{(3)} &\rightarrow f_3. \end{aligned}$$

If x if far from $l^{(1)}$:

$$f_1 = \exp\left(-\frac{(\text{large number})^2}{2\sigma^2}\right) \approx 0.$$

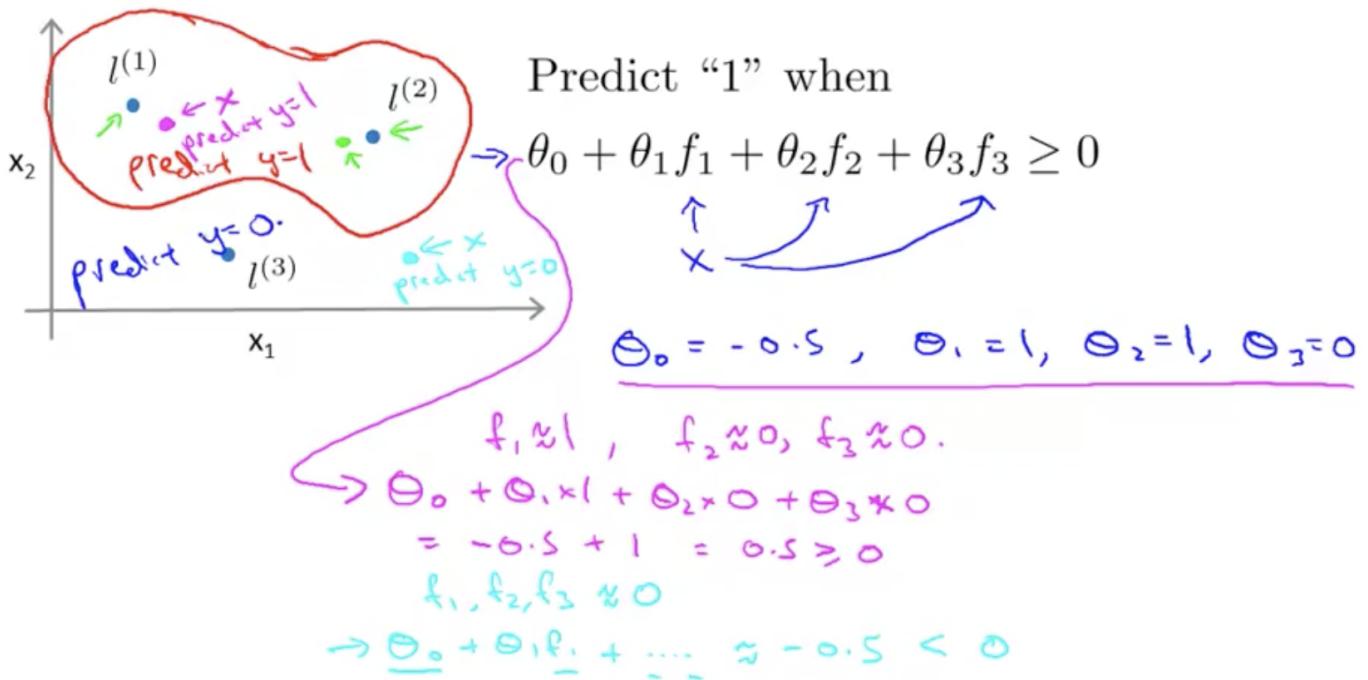
\uparrow

核函数用来计算训练样本x与标记l (landmark) 的相似度，当我们为一个真样本指定若干个标记 l(1), l(2), l(3)...l(n)时，他们以点的形式分布在坐标系呢

(这里的示例图以二维的形式出现只是为了方便理解并且已经省略了x0，实际问题中x的特征不仅限于两个)

下图中，当训练样本x距离l(1)很近，则特征f(1)的相似度接近于1，而x距离l(2)和l(3)很远，则特征f(2)和f(3)的相似度趋近于0 (注意不是真正意义上的0)

在这里，计算特征f的相似度时所用的函数即高斯核函数 (Gaussian kernel)



使用支持向量机时，要考虑以下参数的选取

C过大时会导致过拟合（即lambda过小，相当于没有正规化）

而sigma过大时会导致高偏差低方差，如下图中那个十分平滑的特征曲线，

而sigma过小会导致低偏差高方差，如下图中那个迷之突起，可以理解为符合特征的范围很狭窄，仅有少数特征可以算作符合

值得注意的是这里特征曲线的顶点数值不变，就是1

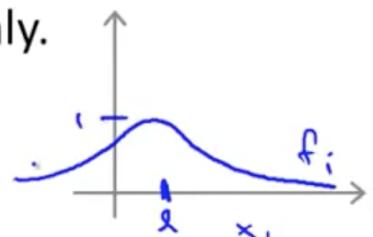
SVM parameters:

$C \left(= \frac{1}{\lambda} \right)$. → Large C: Lower bias, high variance. (small λ)
 → Small C: Higher bias, low variance. (large λ)

σ^2 Large σ^2 : Features f_i vary more smoothly.

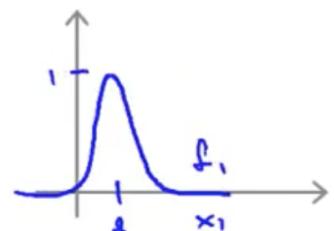
→ Higher bias, lower variance.

$$\exp \left(-\frac{\|x - l^{(i)}\|^2}{2\sigma^2} \right)$$



Small σ^2 : Features f_i vary less smoothly.

Lower bias, higher variance.



当特征数n很多，而训练集数量m较少时，我们倾向于选择线性核函数（linear kernel），即不使用核函数的支持向量机，或者说选择逻辑回归算法，

例如垃圾邮件分类问题，你可能有1万个特征对应1万个单词，而你只有10封垃圾邮件作为训练样本，这是你既不需要也没有能力做一个非常复杂的拟合

相反，当特征数n较少，而训练集数量m相对较多时，我们会用支持向量机并选取圈状物来得出学

习算法

最后，当特征数n较少，而训练集数量m超级多时（例如100万个），支持向量机的计算将会非常慢，因此应当选用线性核函数或逻辑回归

Use SVM software package (e.g. liblinear, libsvm, ...) to solve for parameters θ .

Need to specify:

→ Choice of parameter C.

Choice of kernel (similarity function):

E.g. No kernel ("linear kernel")

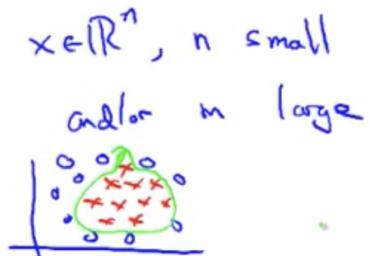
Predict "y = 1" if $\underline{\theta^T x} \geq 0$

$$\Theta_0 + \Theta_1 x_1 + \dots + \Theta_n x_n \geq 0 \quad \rightarrow \underline{n} \text{ large}, \underline{m} \text{ small} \quad \underline{x \in \mathbb{R}^{n+1}}$$

Gaussian kernel:

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

Need to choose $\underline{\sigma^2}$.



下图中总结了何时该选取线性回归，何时该用SVM，

值得注意的是，尽管线性回归和不带核函数的SVM (linear kernel) 非常相似，做的事也差不多，但对于不同问题仍可能有不同表现。

另外，当特征n较少，而训练集m的数量适中的时候，SVM的表现非常耀眼，将会是一个非常强大的算法。

关于神经网络，尽管对于这里的大部分情况都能表现得很好，但在计算速度方面并不如SVM，并且还要面对局部最优解的问题

相比之下，对于特定的nm组合，SVM表现得更好，并且速度更快，也不用担心局部最优解的问题，因为SVM总能找到全局最优解

Logistic regression vs. SVMs

n = number of features ($x \in \mathbb{R}^{n+1}$), m = number of training examples

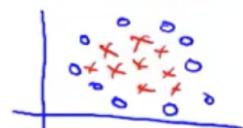
→ If n is large (relative to m): (e.g. $n \geq m$, $n = 10,000$, $m = 10 \dots 1000$)

→ Use logistic regression, or SVM without a kernel ("linear kernel")

→ If n is small, m is intermediate: ($n = 1 \dots 1000$, $m = 10 \dots 10,000$) ←

→ Use SVM with Gaussian kernel

If n is small, m is large: ($n = 1 \dots 1000$, $m = 50,000 \dots$)



→ Create/add more features, then use logistic regression or SVM without a kernel

→ Neural network likely to work well for most of these settings, but may be slower to train.

