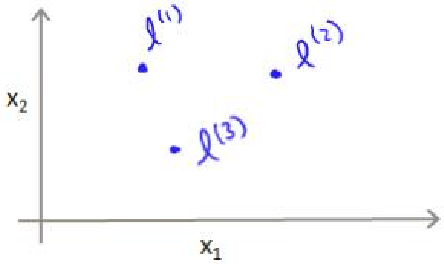
17.5.2（SVM核函数）

Kernels（核函数）

用核函数就是为了把原来的样本特征进行重新组合，构成新的特征，便于计算。把低维度线性不可分的特征变量，转变为高维度超平面可分的。

给定一个训练实例 x，我们利用 x 的各个特征与我们预先选定的**地标**（landmarks）$l^{(1)}, l^{(2)}, l^{(3)}$ 的近似程度来选取新的特征 f_1, f_2, f_3。



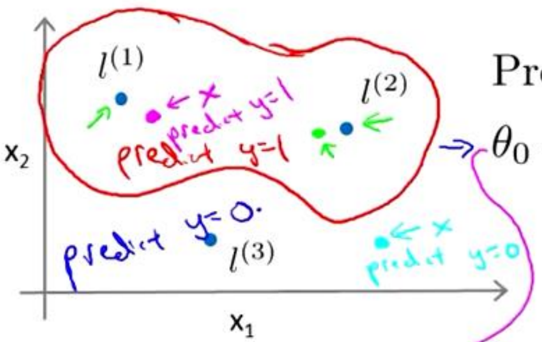
例如：

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

其中：$\|x - l^{(1)}\|^2 = \sum_{j=1}^{n}\left(x_j - l_j^{(1)}\right)^2$，为实例 x 中所有特征与地标 $l^{(1)}$之间的距离的和。上例中的 similarity（$x, l^{(1)}$）就是核函数，具体而言，这里是一个**高斯核函数**（Gaussian

上述的高斯核函数所示。高斯核函数就是求每个x 与我们所选取的landmark 的相似率，也就是说x离landmarks 越近，f 越接近于1，x离landmarks 越远，f 越接近于0，l 就是我们所选取的landmarks ，sigma表示的是每个landmark所覆盖的范围，也就是随着x 的改变f 的变化速率。图像呈正态分布式。

假设我们已经得到了theta参数，用下图中的假设函数来预测1 的情况，可以画出图中的紫红色线就是决策边界，在接近l1或l2的时候，预测结果为1，否则预测结果为0.



Predict "1" when

$$\theta_0 + \theta_1 f_1 + \theta_2 f_2 + \theta_3 f_3 \geq 0$$

x

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

$f_1 \approx 1, \quad f_2 \approx 0, \quad f_3 \approx 0.$

$\rightarrow \theta_0 + \theta_1 \times 1 + \theta_2 \times 0 + \theta_3 \times 0$

$= -0.5 + 1 = 0.5 \geq 0$

$f_1, f_2, f_3 \approx 0$

$\rightarrow \theta_0 + \theta_1 f_1 + \dots \approx -0.5 < 0$

SVM with Kernels

→ Given $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})$,

→ choose $l^{(1)} = x^{(1)}, l^{(2)} = x^{(2)}, \ldots, l^{(m)} = x^{(m)}$.

Given example $\underline{x}$:

→ $f_1 = \text{similarity}(x, l^{(1)})$ ← $x^{(1)}$

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

$\ldots$

$$f = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ \vdots \\ f_m \end{bmatrix} \qquad f_0 = 1$$

For training example $(x^{(i)}, y^{(i)})$:

$x^{(i)}$ →

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

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

$\vdots$

$f_i^{(i)} = \text{sim}(x^{(i)}, l^{(i)}) = \exp(-\frac{0}{2\sigma^2}) = 1$ ← $x^{(i)}$

$f_m^{(i)} = \text{sim}(x^{(i)}, l^{(m)})$

$x^{(i)} \in \mathbb{R}^{n+1}$ (or $\mathbb{R}^n$)

$$f^{(i)} = \begin{bmatrix} f_0^{(i)} \\ f_1^{(i)} \\ \vdots \\ f_m^{(i)} \end{bmatrix} \qquad f_0^{(i)} = 1$$

Andrew Ng

上图表示，选择landmarks 标记点时，可以把每一个样本当成一个landmark，也就有了m个landmarks，对于每一个x ，都可以用高斯核函数得到这个x 与其余所有x样本间的相似度，这些相似度就构成了每个x 新的特征向量f ，也就是训练SVM要用到的特征向量。

SVM parameters:

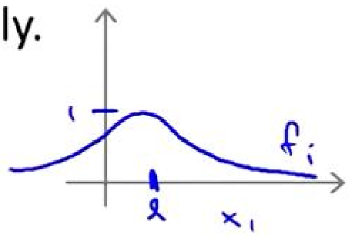
C ($= \frac{1}{\lambda}$). → Large C: Lower bias, high variance. (small λ)

→ Small C: Higher bias, low variance. (large λ)

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

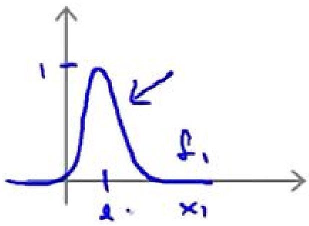
→ Higher bias, lower variance.

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



Small σ^2: Features f_i vary less smoothly.

Lower bias, higher variance.



上图所示SVM的参数变化对模型的影响。

C就相当于1/lambda，C太大时会导致过拟合，低偏差，高方差，相当于逻辑回归中的小lambda；C太小时会导致欠拟合，高偏差，低方差，相当于逻辑回归中的大lambda。

sigma^2太大时，会导致欠拟合，高偏差，低方差；反之，会导致过拟合，低偏差，高方差。

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 $\theta^T x \geq 0$

$\theta_0 + \theta_1 x_1 + \cdots + \theta_n x_n \geq 0$

→ n large, m small $x \in \mathbb{R}^{n+1}$

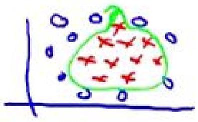
Gaussian kernel:

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

Need to choose σ^2.

$x \in \mathbb{R}^n$, n small

and/or m large



使用SVM算法时，只需要用软件库里面的软件包（比如：liblinear，libsvm，。。。）去求出需要的theta。在用软件包时，需要选择参数C ，和需要用的核函数（求样本和landmark的相似率的方法），当没有选择核函数时就是使用的"线性核函数"，线性核函数中的约束条件还是原来的theta的转置乘x是否大于0.。一般是在特征量特别多，样本量相对特别少的情况下。在相反的情况下用高斯核函数效果更好。用标注点就是样本，高斯核函数就需要选择sigma。注意：使用高斯核函数的情况下要先对特征进行归一化处理。

Other choices of kernel

Note: Not all similarity functions $\text{similarity}(x, l)$ make valid kernels.

→ (Need to satisfy technical condition called "Mercer's Theorem" to make sure SVM packages' optimizations run correctly, and do not diverge).

Many off-the-shelf kernels available:

- Polynomial kernel: $k(x, l) = (x^T l)^2$, $(x^T l)^3$, $(x^T l + 1)^3$, $(x^T l + 5)^4$

  $(x^T l + \text{constant})^{\text{degree}}$

- More esoteric: String kernel, chi-square kernel, histogram intersection kernel, ...

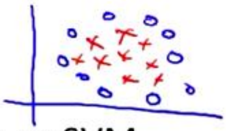
  $\text{sim}(x, l)$

除了线性核函数和高斯核函数之外，还有一些求相似率的函数可以作为核函数，但是它们必须要满足默赛尔定理，满足默赛尔定理可以保证SVM软件包中的大量优化方法都可以使用，可以快速地进行计算theta。还有一些核函数，比如：多项式核函数（一般形式为（x的转置乘l + 一个常数）^次数），字符串核函数，卡方核函数，直方图交集核函数。

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. $n \geq m$, $n = 10{,}000$, $m = 10 \cdots 1000$)
- Use logistic regression, or SVM without a kernel ("linear kernel")
- If n is small, m is intermediate: ($n = 1 - 1000$, $m = 10 - 10{,}000$)
  - Use SVM with Gaussian kernel



- If n is small, m is large: ($n = 1 - 1000$, $m = 50{,}000+$)
  - 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.

n 为样本的特征数量，m 为样本数。当n 远大于m 时，一般用逻辑回归或者SVM的线性核函数，当n 相对较小，m 也只是n 的十倍大小时，SVM的高斯核函数效果更好，当n 相对较小，m 相对较大，可以使用逻辑回归，创建更多的特征，或者用SVM线性核函数，神经网络都能适用，但训练速度可能比较慢。