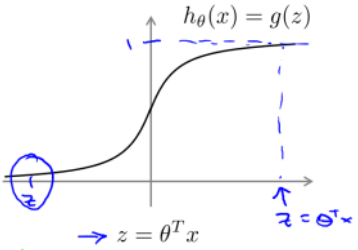
12 支持向量机

2022年11月22日　　18:14

12.1优化目标

依然从逻辑回归出发，

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

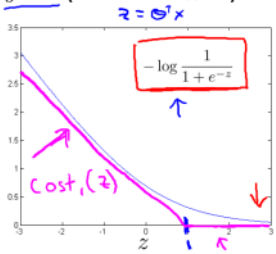


对于一个逻辑回归的模型，要想让假设函数的输出值为1，那么需要z远大于0；若想让假设函数的输出为0，那么z需远小于0。

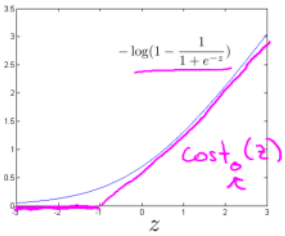
Cost of example: $-(y \log h_\theta(x) + (1-y)\log(1-h_\theta(x)))$

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

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



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



对于一个样本的代价函数，它由两项构成。如果想要y为1，那么只有第一项起作用，从而绘制出此时的代价函数曲线；而如果想要y为0，那么只有第二项起作用，同样地绘制出曲线。分别在z=1和-1处取一点，作如图所示的分段折线，可以看出它是能够逼近起始的曲线的。有了这两个函数就可以构建支持向量机(Support vector machine)。

Logistic regression:

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

Support vector machine:

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

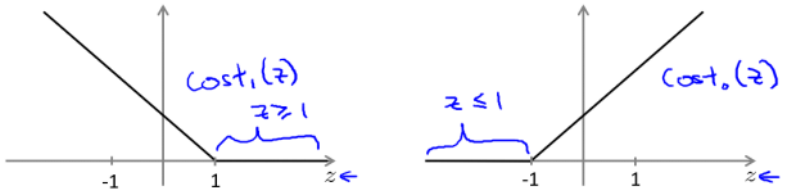
可以看到，和逻辑回归的式子不同的是，将两个代价项替换成了刚刚的所设的函数，同时去掉了相乘的常数1/m，显然这个相乘的常数对求取最小值对应的变量是没有影响的；此外，将正则化因子转移到了前面的代价项上，因为本质上正则化因子的设置就是用来平衡第一项和第二项的权重的，所以把它移到前面依然能够行使这样的功能。

最后，和逻辑回归不同的是，支持向量机输出的并不是一个概率，它作出的直接是一个预测，那就是y到底是1还是0。

12.2直观上对大间隔的理解

有时候人们会把支持向量机叫做大间隔分类器。

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



- If $y=1$, we want $\theta^T x \geq 1$ (not just ≥ 0)
- If $y=0$, we want $\theta^T x \leq -1$ (not just < 0)

C = 100,000

根据上一节所讲的支持向量机的性质，如果我们想让代价值尽可能地小，那么当y=1时，只需z大于等于1，其对应的代价值就为0；而当y=0时，只要z小于等于-1，其对应的代价值也就为0了。

我们已知在逻辑回归中，可以通过判断$\theta^T X$是否大于零来对其进行分类；而支持向量机则对其提出了更高的要求。

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

Whenever $y^{(i)}=1$:

$\theta^T x^{(i)} \geq 1$

Whenever $y^{(i)}=0$:

$\theta^T x^{(i)} \leq -1$

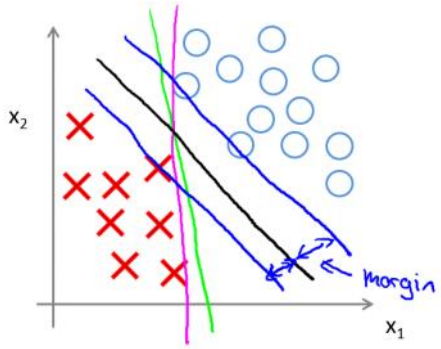
$$\min_\theta \frac{1}{2}\sum_{i=1}^{n}\theta_j^2$$

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

$\theta^T x^{(i)} \leq -1$ if $y^{(i)}=0$

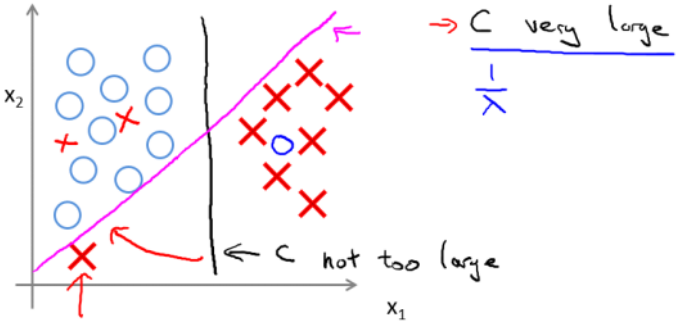
当我们把C设置得特别大的时候，我们希望找到一个合适的 θ 值使代价项为0。

SVM Decision Boundary: Linearly separable case



Large margin classifier

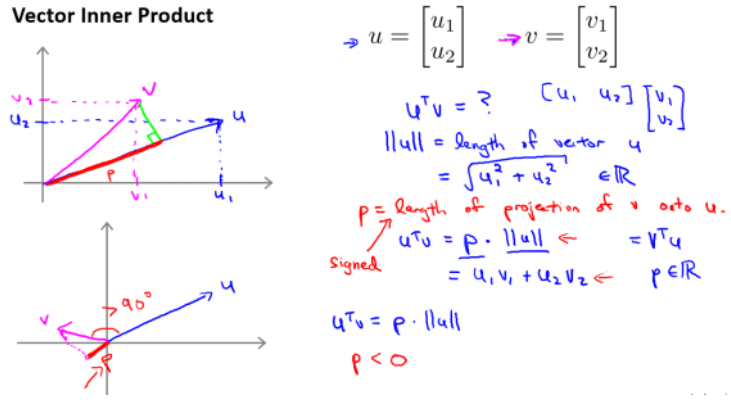
对于决策边界的选择可能有很多种，比如上图粉色和绿色的极端情况，而支持向量机则倾向于选择黑色的那条更稳健的边界。它在数学上的具体表现是它到样本的最小距离要比其他边界更大一些。这个距离叫做支持向量机的间距(margin)，这使得它在分离数据时具有鲁棒性，因为它会用尽量大的间距来分离。这也是为什么支持向量机被称为最大间距分类器。



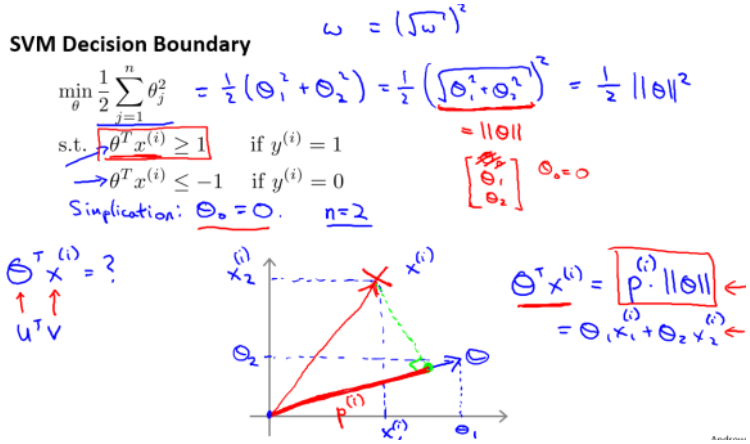
但是最大间距分类器的论断的得到是我们基于C很大的前提，此时模型会对样本中的异常点非常敏感，为了得到最大间距的决策边界，得到的结果可能就像上图中粉色的直线所示，它相对黑色的直线并不是一个很好的边界。而当C不那么大的时候，得到的边界会更接近于黑色的直线。

12.3大间隔分类器的数学原理(选学)

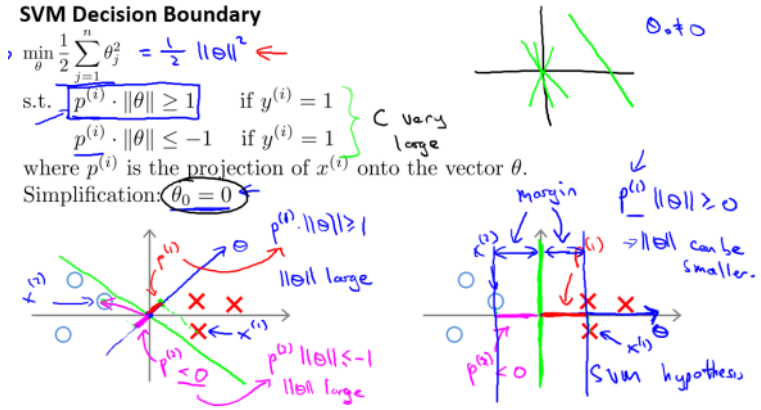
内积的性质：



那么z的求取其实就是内积的应用：



那么就可以作如下的变换：

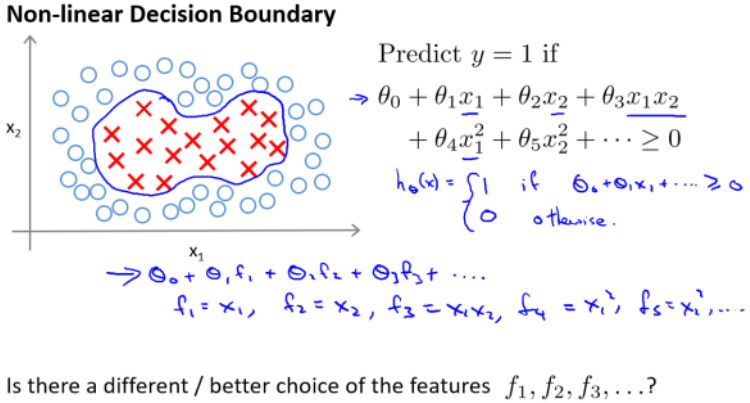


上图演示了上一节中的结论，那就是支持向量机为什么不会选择擦边而过的极端情况的决策边界。是因为参数所构成的向量和上图左边绿色的决策边界是正交的，而由于部分点到向量 θ 上的投影的模会非常小，那么为了让内积的结果大于或小于1，就需要 θ 的范数极大，而又由于支持向量机的最小化函数致力于寻找使函数最小的结果，这个函数又恰等于 θ 的范数的平方的一半，所以它绝不会将这条绿色的线作为最终的决策边界。

而如上图右所示，当边界和样本点的最小距离最大时，内积和代价函数的目的趋于一致，那就是都希望最小化 θ 的范数，它也将被作为支持向量机选择的决策边界。

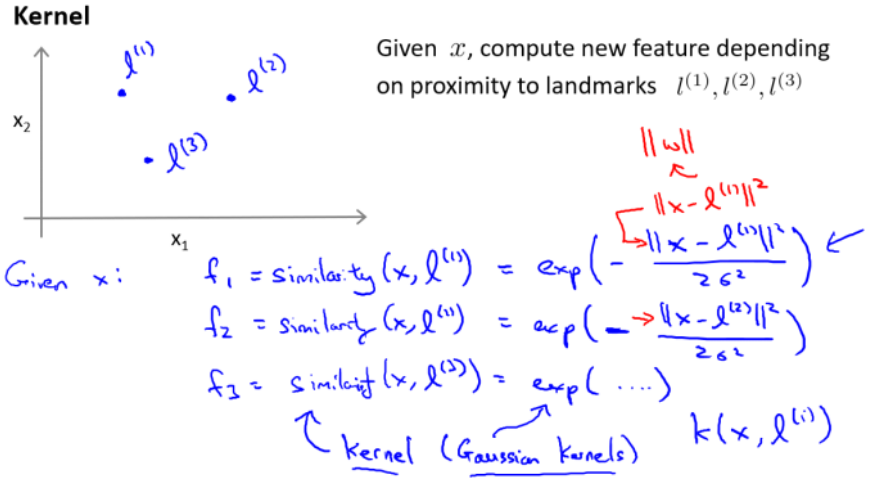
12.4核函数1

对于一个复杂的分类问题，

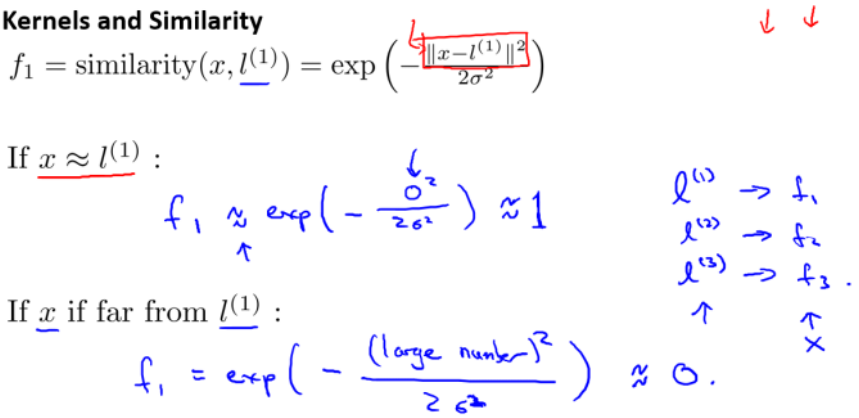


Is there a different / better choice of the features $f_1, f_2, f_3, \ldots$?

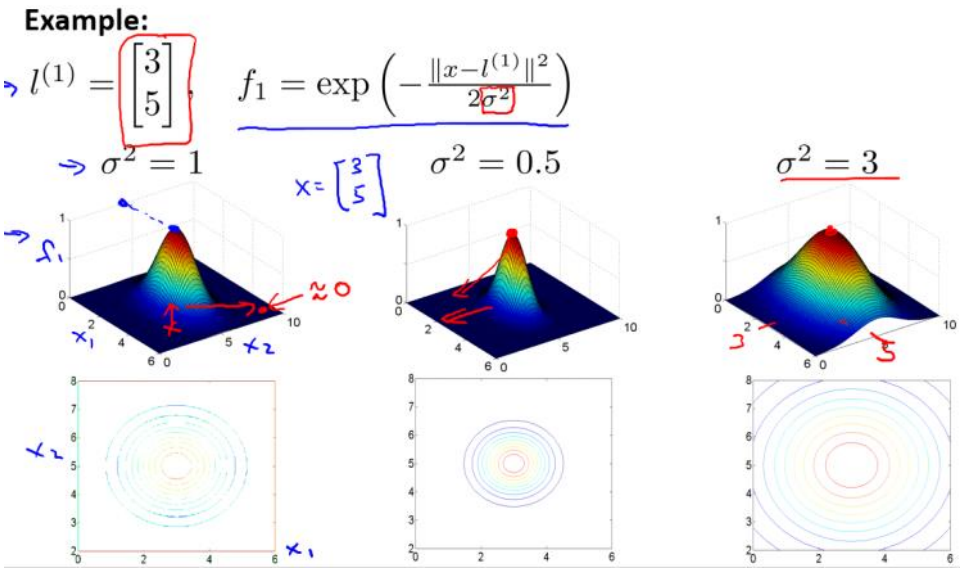
可以通过构造高阶多项式的假设函数来对其进行边界的拟合，此时定义如上蓝色字体所写的函数取代高阶多项式的假设函数。由于我们不知道这些高阶项是否有用以及是否有更好的特征可以嵌入到函数中，所以如何选择是一个问题。



定义f为一种x和标记(landmark)l的相似度量。这个相似度函数就叫做核函数(Kernel)。这里特指高斯核函数。

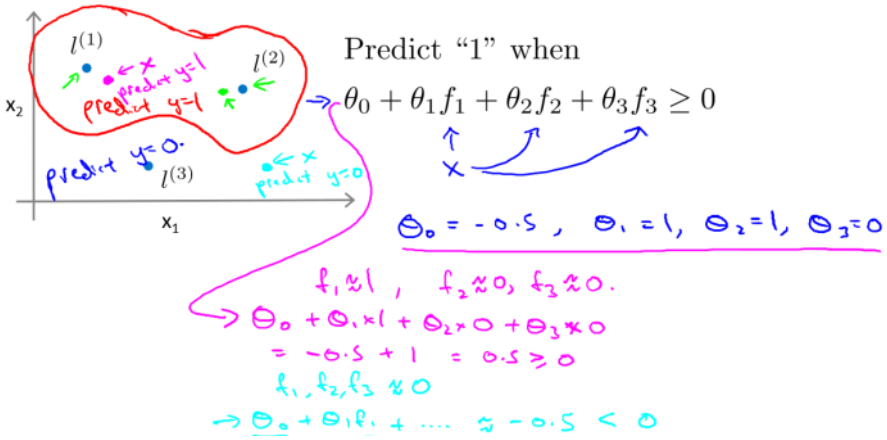


当x和l很接近时，核函数的值趋于1；当x和l很远时，核函数的值趋于0。
核函数的直观表示如下图：



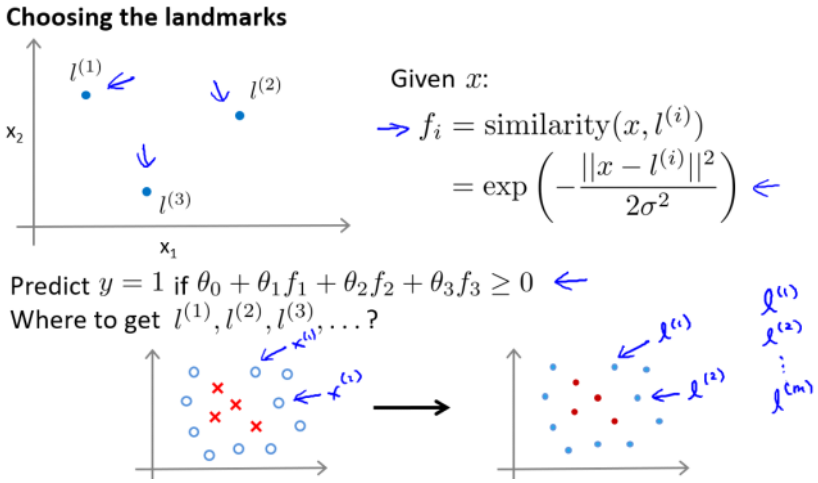
可以看出，当x与1相等时达到最高点；而随着核函数的参数的增加，最高点向周围降低的速度变得越来越缓慢。

举一个实际预测的例子：

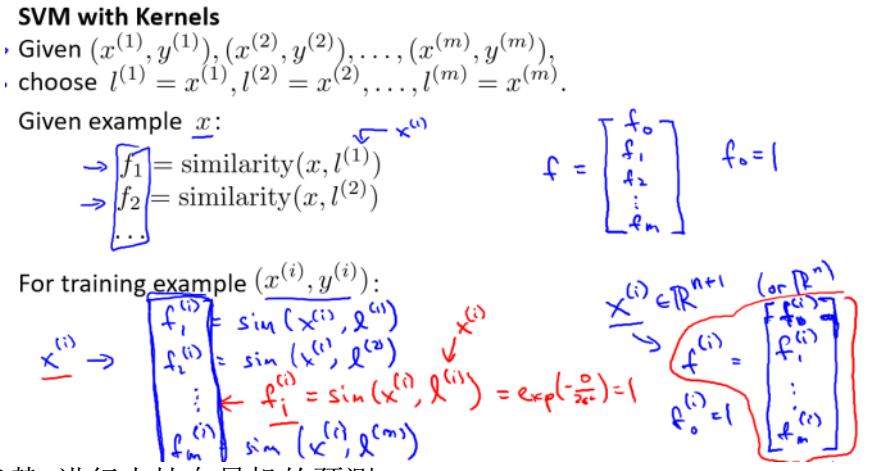


12.5核函数2

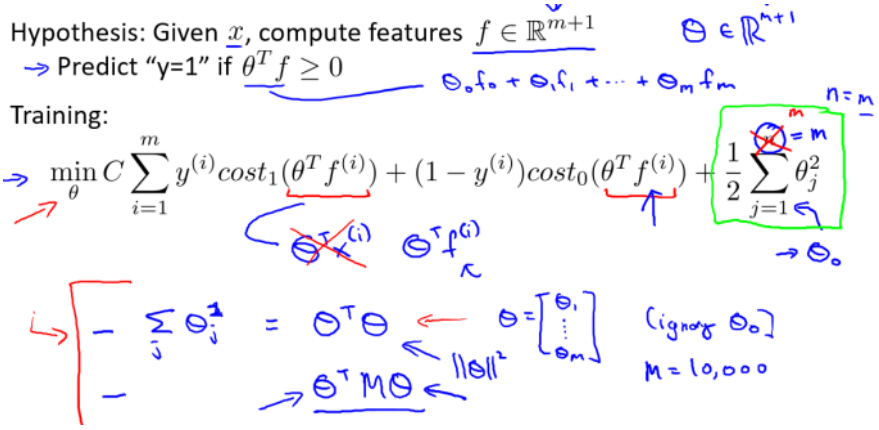
怎么得到标记点？



一般直接将样本作为标记点：

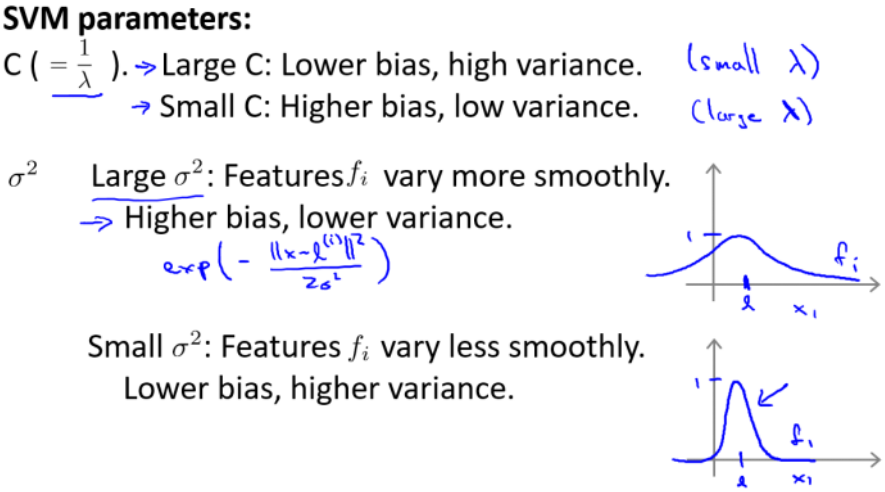


然后，用f代替x进行支持向量机的预测：



有研究表明，支持向量机和核函数可以相得益彰以及基于其二者的更高级的优化算法，而逻辑回归和核函数则显得十分缓慢。
老师不推荐自己写最小化代价函数的代码，而应该使用成熟的软件包。

另一个问题是怎么使用支持向量机中的参数。其依据依然是偏差和方差的折中。



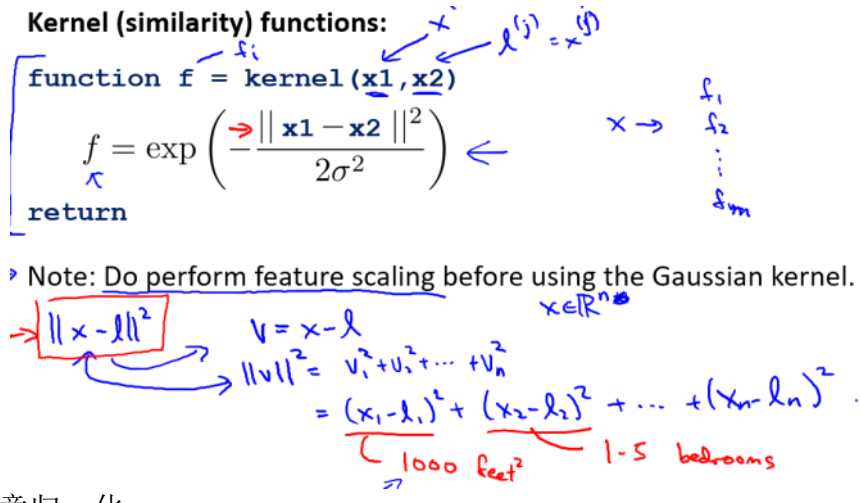
同我们在逻辑回归中分析的一样，在逻辑回归中，当λ太小时会得到低偏差和高方差；当λ太大时会得到高偏差和低方差。由于C和λ的反比趋势，对于C所造成的影响就恰好相反。即大的C值会倾向于过拟合，小的C值会倾向于欠拟合。
而对于高斯核函数中σ²的选取，前面已经分析过，当它更大时核函数会更平滑和缓慢，所得的模型倾向于高偏差和低方差；而当它较小时核函数的变化会剧烈一些、斜率较大，所得模型倾向于低偏差和高方差。

12.6支持向量机的使用

一般建议用已有的软件库实现SVM算法，如liblinear,libsvm。
特别的，我们需要自己选择C值；其次，对于有内参的核函数我们需要选择参数值，而

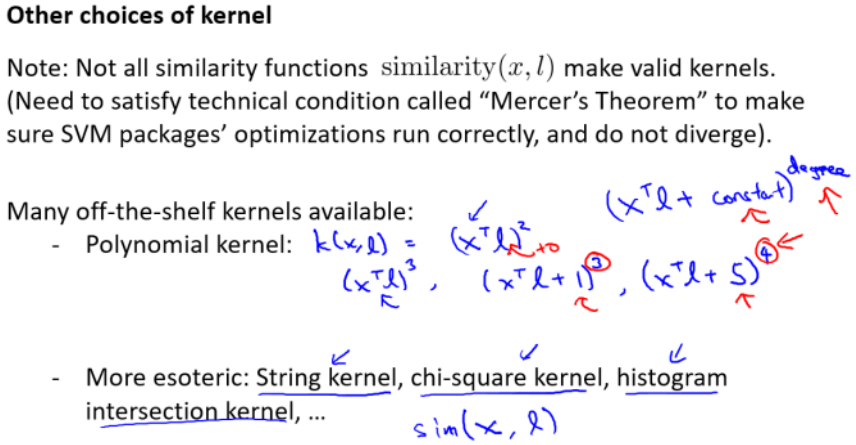
像线性核函数(linear kernel)这种就是无内核参数函数，我们不需要自己选择参数。比如，当特征数量很多而样本数量很少的时候，为了防止过拟合，线性的核函数就是一个好的选择。

核函数的实现：



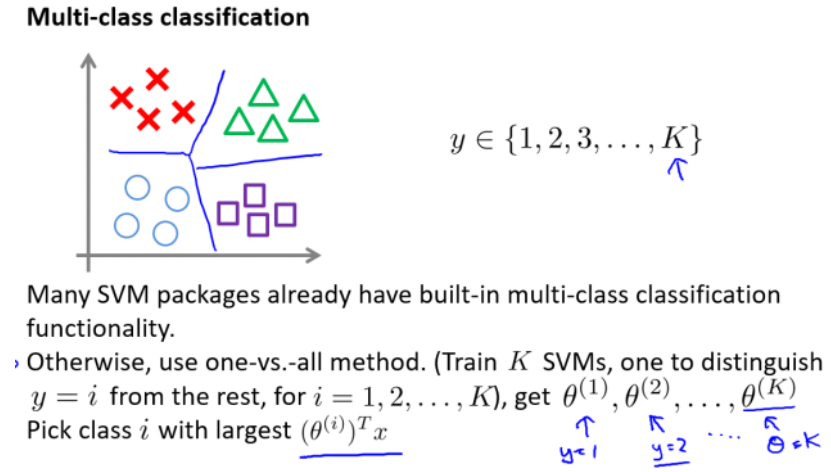
其中应当注意归一化。

另外，我们选择的核函数应当满足默塞尔定理：



这个定理确保所有的SVM算法大类的优化方法并从而迅速得到参数 θ 。多项式核函数等是其它符合默塞尔定理的核函数。

下面再来讨论多分类的判定边界：



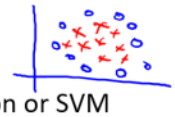
一些SVM算法包已经能够很好地做到多分类；而我们也可以采用之前提到的一对多的分类方法，通过k个SVM来进行分类。

最后来比较一下逻辑回归和支持向量机。

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.



- 相较于样本数量，特征数目很多的情况下，我们通常选择逻辑回归或者是使用了线性核的支持向量机。
- 当特征数目很小，而样本数量适中时，用如高斯核函数的有核函数的支持向量会工作得很好。
- 当特征数目很小，但样本数量很大时，由于数据数量太大，用有核函数的支持向量机可能会工作得比较慢。在这种情况下可以尝试手动创建或添加更多特征变量，然后用逻辑回归或者不带核函数的支持向量机。
- 神经网络可能对大多数情况都很有效，但是训练起来会比较慢。