对回归方法的认识

JerryLead
csxulijie@gmail.com
2011 年 2 月 27 日

1 摘要

本报告是在学习斯坦福大学机器学习课程前四节加上配套的讲义后的总结与认识。前四节主要讲述了回归问题，属于有监督学习中的一种方法。该方法的核心思想是从离散的统计数据中得到数学模型，然后将该数学模型用于预测或者分类。该方法处理的数据可以是多维的。

讲义最初介绍了一个基本问题，然后引出了线性回归的解决方法，然后针对误差问题做了概率解释。

2 问题引入

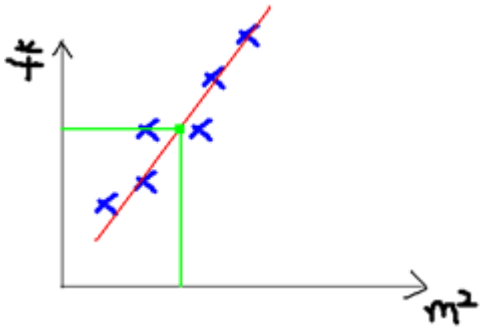
假设有一个房屋销售的数据如下：

面积(m^2)	销售价钱（万元）
123	250
150	320
87	160
102	220
…	…

这个表类似于北京 5 环左右的房屋价钱，我们可以做出一个图，x 轴是房屋的面积。y 轴是房屋的售价，如下：

如果来了一个新的面积，假设在销售价钱的记录中没有的，我们怎么办呢？

我们可以用一条曲线去尽量准的拟合这些数据，然后如果有新的输入过来，我们可以在将曲线上这个点对应的值返回。如果用一条直线去拟合，可能是下面的样子：



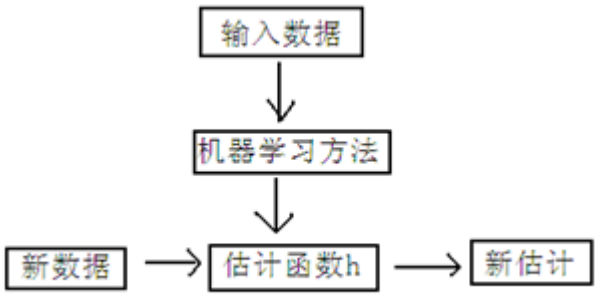
绿色的点就是我们想要预测的点。

首先给出一些概念和常用的符号。
房屋销售记录表：训练集(training set)或者训练数据(training data)，是我们流程中的输入数据，一般称为 x
房屋销售价钱：输出数据，一般称为 y
拟合的函数（或者称为假设或者模型）：一般写做 y = h(x)
训练数据的条目数(#training set),：一条训练数据是由一对输入数据和输出数据组成的输入数据的维度 n (特征的个数，#features)

这个例子的特征是两维的，结果是一维的。然而回归方法能够解决特征多维，结果是一维多离散值或一维连续值的问题。

3 学习过程

下面是一个典型的机器学习的过程，首先给出一个输入数据，我们的算法会通过一系列的过程得到一个估计的函数，这个函数有能力对没有见过的新数据给出一个新的估计，也被称为构建一个模型。就如同上面的线性回归函数。



4 线性回归

线性回归假设特征和结果满足线性关系。其实线性关系的表达能力非常强大，每个特征对结果的影响强弱可以有前面的参数体现，而且每个特征变量可以首先映射到一个函数，然后再参与线性计算。这样就可以表达特征与结果之间的非线性关系。

我们用 X1，X2..Xn 去描述 feature 里面的分量，比如 x1=房间的面积，x2=房间的朝向，等等，我们可以做出一个估计函数：

$$h(x) = h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

θ 在这儿称为参数，在这的意思是调整 feature 中每个分量的影响力，就是到底是房屋的面积更重要还是房屋的地段更重要。为了如果我们令 X0 = 1，就可以用向量的方式来表示了：

$$h_\theta(x) = \theta^T X$$

我们程序也需要一个机制去评估我们 θ 是否比较好，所以说需要对我们做出的 h 函数进行评估，一般这个函数称为损失函数（loss function）或者错误函数(error function)，描述 h 函数不好的程度，在下面，我们称这个函数为 J 函数

在这儿我们可以做出下面的一个错误函数：

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

$$\min_\theta J_\theta$$

这个错误估计函数是去对 x(i)的估计值与真实值 y(i)差的平方和作为错误估计函数，前面乘上的 1/2 是为了在求导的时候，这个系数就不见了。

至于为何选择平方和作为错误估计函数，讲义后面从概率分布的角度讲解了该公式的来源。

如何调整 θ 以使得 J(θ)取得最小值有很多方法，其中有最小二乘法(min square)，是一种完全是数学描述的方法，和梯度下降法。

5 梯度下降法

在选定线性回归模型后，只需要确定参数 θ，就可以将模型用来预测。然而 θ 需要在 J(θ)最小的情况下才能确定。因此问题归结为求极小值问题，使用梯度下降法。梯度下降法最大的问题是求得有可能是全局极小值，这与初始点的选取有关。

梯度下降法是按下面的流程进行的：
1）首先对 θ 赋值，这个值可以是随机的，也可以让 θ 是一个全零的向量。
2）改变 θ 的值，使得 J(θ)按梯度下降的方向进行减少。

梯度方向由 J(θ)对 θ 的偏导数确定，由于求的是极小值，因此梯度方向是偏导数的反方向。结果为

$$\theta_j := \theta_j + \alpha\left(y^{(i)} - h_\theta(x^{(i)})\right)x_j^{(i)}.$$

迭代更新的方式有两种，一种是批梯度下降，也就是对全部的训练数据求得误差后再对 θ 进行更新，另外一种是增量梯度下降，每扫描一步都要对 θ 进行更新。前一种方法能够不断收敛，后一种方法结果可能不断在收敛处徘徊。

一般来说，梯度下降法收敛速度还是比较慢的。

另一种直接计算结果的方法是最小二乘法。

6 最小二乘法

将训练特征表示为 X 矩阵，结果表示成 y 向量，仍然是线性回归模型，误差函数不变。那么 θ 可以直接由下面公式得出

$$\theta = (X^T X)^{-1} X^T \vec{y}.$$

但此方法要求 X 是列满秩的，而且求矩阵的逆比较慢。

7 选用误差函数为平方和的概率解释

假设根据特征的预测结果与实际结果有误差$\epsilon^{(i)}$，那么预测结果$\theta^T x^{(i)}$和真实结果$y^{(i)}$满足下式：

$$y^{(i)} = \theta^T x^{(i)} + \epsilon^{(i)},$$

一般来讲，误差满足平均值为 0 的高斯分布，也就是正态分布。那么 x 和 y 的条件概率也就是

$$p(y^{(i)}|x^{(i)};\theta) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right).$$

这样就估计了一条样本的结果概率，然而我们期待的是模型能够在全部样本上预测最准，也就是概率积最大。这个概率积成为最大似然估计。我们希望在最大似然估计得到最大值时确定 θ。那么需要对最大似然估计公式求导，求导结果既是

$$\frac{1}{2}\sum_{i=1}^{m}(y^{(i)} - \theta^T x^{(i)})^2,$$

这就解释了为何误差函数要使用平方和。
当然推导过程中也做了一些假定，但这个假定符合客观规律。

8 带权重的线性回归

上面提到的线性回归的误差函数里系统都是 1，没有权重。带权重的线性回归加入了权重信

息。
基本假设是

1. Fit θ to minimize $\sum_i w^{(i)}(y^{(i)} - \theta^T x^{(i)})^2$.
2. Output $\theta^T x$.

其中假设$w^{(i)}$符合公式

$$w^{(i)} = \exp\left(-\frac{(x^{(i)} - x)^2}{2\tau^2}\right)$$

其中 x 是要预测的特征，这样假设的道理是离 x 越近的样本权重越大，越远的影响越小。这个公式与高斯分布类似，但不一样，因为$w^{(i)}$不是随机变量。

此方法成为非参数学习算法，因为误差函数随着预测值的不同而不同，这样 θ 无法事先确定，预测一次需要临时计算，感觉类似 KNN。

9 分类和对数回归

一般来说，回归不用在分类问题上，因为回归是连续型模型，而且受噪声影响比较大。如果非要应用进入，可以使用对数回归。

对数回归本质上是线性回归，只是在特征到结果的映射中加入了一层函数映射，即先把特征线性求和，然后使用函数 g(z)将最为假设函数来预测。g(z)可以将连续值映射到 0 和 1 上。

对数回归的假设函数如下，线性回归假设函数只是$\theta^T x$。

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

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

对数回归用来分类 0/1 问题，也就是预测结果属于 0 或者 1 的二值分类问题。这里假设了二值满足伯努利分布，也就是

$$\begin{aligned} P(y = 1 \mid x; \theta) &= h_\theta(x) \\ P(y = 0 \mid x; \theta) &= 1 - h_\theta(x) \end{aligned}$$

当然假设它满足泊松分布、指数分布等等也可以，只是比较复杂，后面会提到线性回归的一般形式。

与第 7 节一样，仍然求的是最大似然估计，然后求导，得到迭代公式结果为

$$\theta_j := \theta_j + \alpha \left(y^{(i)} - h_\theta(x^{(i)}) \right) x_j^{(i)}$$

可以看到与线性回归类似，只是$\theta^T x^{(i)}$换成了$h_\theta(x^{(i)})$，而$h_\theta(x^{(i)})$实际上就是$\theta^T x^{(i)}$经过g(z)映射过来的。

10 牛顿法来解最大似然估计

第 7 和第 9 节使用的解最大似然估计的方法都是求导迭代的方法，这里介绍了牛顿下降法，使结果能够快速的收敛。

当要求解f(θ) = 0时，如果 f 可导，那么可以通过迭代公式

$$\theta := \theta - \frac{f(\theta)}{f'(\theta)}.$$

来迭代求解最小值。

当应用于求解最大似然估计的最大值时，变成求解$\ell'(\theta) = 0$的问题。

那么迭代公式写作

$$\theta := \theta - \frac{\ell'(\theta)}{\ell''(\theta)}.$$

当 θ 是向量时，牛顿法可以使用下面式子表示

$$\theta := \theta - H^{-1} \nabla_\theta \ell(\theta).$$

$$H_{ij} = \frac{\partial^2 \ell(\theta)}{\partial \theta_i \partial \theta_j}.$$

其中 是 n×n 的 Hessian 矩阵。

牛顿法收敛速度虽然很快，但求 Hessian 矩阵的逆的时候比较耗费时间。

当初始点 X0 靠近极小值 X 时，牛顿法的收敛速度是最快的。但是当 X0 远离极小值时，牛顿法可能不收敛，甚至连下降都保证不了。原因是迭代点 Xk+1 不一定是目标函数 f 在牛顿方向上的极小点。

11 一般线性模型

之所以在对数回归时使用

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

的公式是由一套理论作支持的。

这个理论便是一般线性模型。

首先，如果一个概率分布可以表示成

$$p(y; \eta) = b(y) \exp(\eta^T T(y) - a(\eta))$$

时，那么这个概率分布可以称作是指数分布。
伯努利分布，高斯分布，泊松分布，贝塔分布，狄特里特分布都属于指数分布。
在对数回归时采用的是伯努利分布，伯努利分布的概率可以表示成

$$\begin{aligned} p(y;\phi) &= \phi^y(1-\phi)^{1-y} \\ &= \exp(y\log\phi + (1-y)\log(1-\phi)) \\ &= \exp\left(\left(\log\left(\frac{\phi}{1-\phi}\right)\right)y + \log(1-\phi)\right). \end{aligned}$$

其中

$$\eta = \log(\phi/(1-\phi)).$$

得到

$$\Phi = \frac{1}{1+e^{\eta}}$$

这就解释了对数回归时为了要用这个函数。

一般线性模型的要点是
1） y|x; θ 满足一个以η为参数的指数分布，那么可以求得η的表达式。
2） 给定 x，我们的目标是要确定T(y)，大多数情况下T(y) = y，那么我们实际上要确定的是h(x)，而h(x) = E[y|x]。（在对数回归中期望值是Φ，因此 h 是Φ；在线性回归中期望值是μ，而高斯分布中η = μ，因此线性回归中 h=$\theta^T x$）。
3） $\eta = \theta^T x$

12 Softmax 回归

最后举了一个利用一般线性模型的例子。
假设预测值 y 有 k 种可能，即 y∈{1,2,…,k}
比如 k=3 时，可以看作是要将一封未知邮件分为垃圾邮件、个人邮件还是工作邮件这三类。
定义

$$\phi_i = p(y=i;\phi)$$

那么

$$\sum_{i=1}^{k}\phi_i = 1$$

这样

$$p(y=k;\phi) = 1 - \sum_{i=1}^{k-1}\phi_i.$$

即式子左边可以有其他的概率表示，因此可以当做是 k-1 维的问题。
T(y)这时候一组 k-1 维的向量，不再是 y。即 T(y)要给出 y=i（i 从 1 到 k-1）的概率

$$T(1) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, T(2) = \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, T(3) = \begin{bmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ 0 \end{bmatrix}, \cdots, T(k-1) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}, T(k) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix},$$

应用于一般线性模型

$$\begin{aligned} p(y;\phi) &= \phi_1^{1\{y=1\}}\phi_2^{1\{y=2\}}\cdots\phi_k^{1\{y=k\}} \\ &= \phi_1^{1\{y=1\}}\phi_2^{1\{y=2\}}\cdots\phi_k^{1-\sum_{i=1}^{k-1}1\{y=i\}} \\ &= \phi_1^{(T(y))_1}\phi_2^{(T(y))_2}\cdots\phi_k^{1-\sum_{i=1}^{k-1}(T(y))_i} \\ &= \exp((T(y))_1\log(\phi_1) + (T(y))_2\log(\phi_2) + \\ &\qquad \cdots + \left(1-\sum_{i=1}^{k-1}(T(y))_i\right)\log(\phi_k)) \\ &= \exp((T(y))_1\log(\phi_1/\phi_k) + (T(y))_2\log(\phi_2/\phi_k) + \\ &\qquad \cdots + (T(y))_{k-1}\log(\phi_{k-1}/\phi_k) + \log(\phi_k)) \\ &= b(y)\exp(\eta^T T(y) - a(\eta)) \end{aligned}$$

那么

$$\begin{aligned} \eta &= \begin{bmatrix} \log(\phi_1/\phi_k) \\ \log(\phi_2/\phi_k) \\ \vdots \\ \log(\phi_{k-1}/\phi_k) \end{bmatrix}, \\ a(\eta) &= -\log(\phi_k) \\ b(y) &= 1. \end{aligned}$$

最后求得

$$\phi_i = \frac{e^{\eta_i}}{\sum_{j=1}^{k} e^{\eta_j}}$$

而 y=i 时

$$\begin{aligned} p(y=i|x;\theta) &= \phi_i \\ &= \frac{e^{\eta_i}}{\sum_{j=1}^{k} e^{\eta_j}} \\ &= \frac{e^{\theta_i^T x}}{\sum_{j=1}^{k} e^{\theta_j^T x}} \end{aligned}$$

求得期望值

$$
\begin{aligned}
h_\theta(x) &= \mathrm{E}[T(y)|x;\theta] \\
&= \mathrm{E}\left[\begin{array}{c} 1\{y=1\} \\ 1\{y=2\} \\ \vdots \\ 1\{y=k-1\} \end{array} \middle| x;\theta \right] \\
&= \begin{bmatrix} \phi_1 \\ \phi_2 \\ \vdots \\ \phi_{k-1} \end{bmatrix} \\
&= \begin{bmatrix} \frac{\exp(\theta_1^T x)}{\sum_{j=1}^k \exp(\theta_j^T x)} \\ \frac{\exp(\theta_2^T x)}{\sum_{j=1}^k \exp(\theta_j^T x)} \\ \vdots \\ \frac{\exp(\theta_{k-1}^T x)}{\sum_{j=1}^k \exp(\theta_j^T x)} \end{bmatrix}.
\end{aligned}
$$

那么就建立了假设函数，最后就获得了最大似然估计

$$
\begin{aligned}
\ell(\theta) &= \sum_{i=1}^m \log p(y^{(i)}|x^{(i)};\theta) \\
&= \sum_{i=1}^m \log \prod_{l=1}^k \left(\frac{e^{\theta_l^T x^{(i)}}}{\sum_{j=1}^k e^{\theta_j^T x^{(i)}}} \right)^{1\{y^{(i)}=l\}}
\end{aligned}
$$

对该公式可以使用梯度下降或者牛顿法迭代求解。
解决了多值模型建立与预测问题。

学习总结

该讲义组织结构清晰，思路独特，讲原因，也讲推导。可贵的是讲出了问题的基本解决思路和扩展思路，更重要的是讲出了为什么要使用相关方法以及问题根源。在看似具体的解题思路中能引出更为抽象的一般解题思路，理论化水平很高。
该方法可以用在对数据多维分析和多值预测上，更适用于数据背后蕴含某种概率模型的情景。

判别模型、生成模型与朴素贝叶斯方法

JerryLead
csxulijie@gmail.com
2011 年 3 月 5 日星期六

1 判别模型与生成模型

上篇报告中提到的回归模型是判别模型，也就是根据特征值来求结果的概率。形式化表示为$p(y|x;\theta)$，在参数θ确定的情况下，求解条件概率$p(y|x)$。通俗的解释为在给定特征后预测结果出现的概率。

比如说要确定一只羊是山羊还是绵羊，用判别模型的方法是先从历史数据中学习到模型，然后通过提取这只羊的特征来预测出这只羊是山羊的概率，是绵羊的概率。换一种思路，我们可以根据山羊的特征首先学习出一个山羊模型，然后根据绵羊的特征学习出一个绵羊模型。然后从这只羊中提取特征，放到山羊模型中看概率是多少，再放到绵羊模型中看概率是多少，哪个大就是哪个。形式化表示为求$p(x|y)$（也包括$p(y)$），y 是模型结果，x 是特征。

利用贝叶斯公式发现两个模型的统一性：

$$p(y|x) = \frac{p(x|y)p(y)}{p(x)}.$$

由于我们关注的是 y 的离散值结果中哪个概率大（比如山羊概率和绵羊概率哪个大），而并不是关心具体的概率，因此上式改写为：

$$\begin{aligned}\arg\max_y p(y|x) &= \arg\max_y \frac{p(x|y)p(y)}{p(x)} \\ &= \arg\max_y p(x|y)p(y).\end{aligned}$$

其中$p(x|y)$称为后验概率，$p(y)$称为先验概率。

由$p(x|y) * p(y) = p(x,y)$，因此有时称判别模型求的是条件概率，生成模型求的是联合概率。

常见的判别模型有线性回归、对数回归、线性判别分析、支持向量机、boosting、条件随机场、神经网络等。

常见的生产模型有隐马尔科夫模型、朴素贝叶斯模型、高斯混合模型、LDA、Restricted Boltzmann Machine 等。

这篇博客较为详细地介绍了两个模型：

http://blog.sciencenet.cn/home.php?mod=space&uid=248173&do=blog&id=227964

2 高斯判别分析（Gaussian discriminant analysis）

1） 多值正态分布

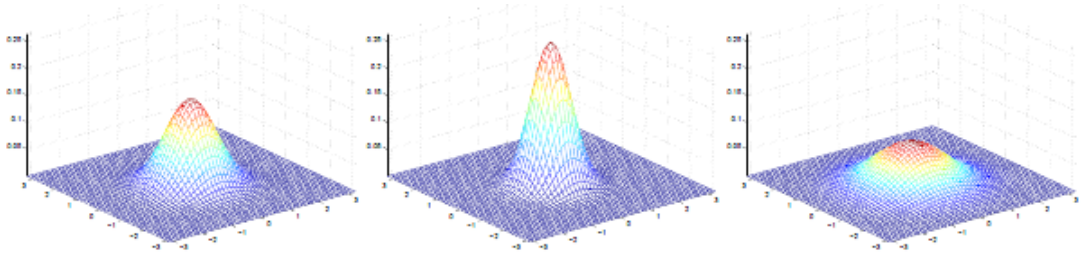
多变量正态分布描述的是 n 维随机变量的分布情况，这里的μ变成了向量，σ也变成了矩阵Σ。写作$N(\mu, \Sigma)$。假设有 n 个随机变量$X_1, X_2, \ldots, X_n$。μ的第 i 个分量是$\mathrm{E}(X_i)$，而$\Sigma_{ii} = \mathrm{Var}(X_i)$，$\Sigma_{ij} = \mathrm{Cov}(X_i, X_j)$。

概率密度函数如下：

$$p(x;\mu,\Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right).$$

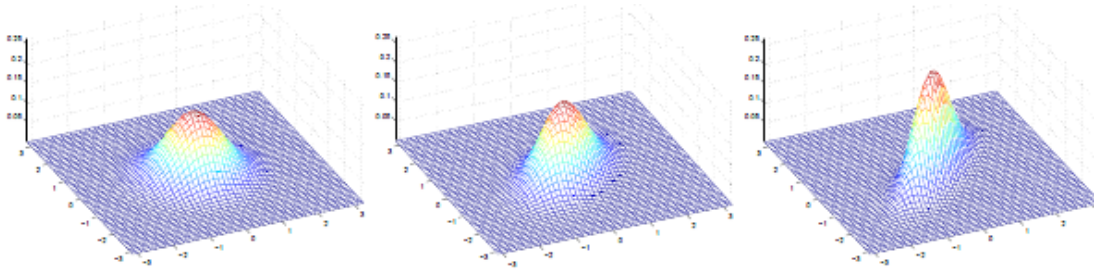
其中|Σ|是Σ的行列式，Σ是协方差矩阵，而且是对称半正定的。

当μ是二维的时候可以如下图表示：



其中μ决定中心位置，Σ决定投影椭圆的朝向和大小。

如下图：



The figures above show Gaussians with mean 0, and with covarianc matrices respectively

$$\Sigma = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}; \quad \Sigma = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}; \quad .\Sigma = \begin{bmatrix} 1 & 0.8 \\ 0.8 & 1 \end{bmatrix}.$$

对应的Σ都不同。

2） 模型分析与应用

如果输入特征x是连续型随机变量，那么可以使用高斯判别分析模型来确定p(x|y)。

模型如下：

$$
\begin{aligned}
y &\sim \text{Bernoulli}(\phi) \\
x|y=0 &\sim \mathcal{N}(\mu_0, \Sigma) \\
x|y=1 &\sim \mathcal{N}(\mu_1, \Sigma)
\end{aligned}
$$

输出结果服从伯努利分布，在给定模型下特征符合多值高斯分布。通俗地讲，在山羊模型下，它的胡须长度，角大小，毛长度等连续型变量符合高斯分布，他们组成的特征向量符合多值高斯分布。
这样，可以给出概率密度函数：

$$
\begin{aligned}
p(y) &= \phi^y(1-\phi)^{1-y} \\
p(x|y=0) &= \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_0)^T\Sigma^{-1}(x-\mu_0)\right) \\
p(x|y=1) &= \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(x-\mu_1)^T\Sigma^{-1}(x-\mu_1)\right)
\end{aligned}
$$

最大似然估计如下：

$$
\begin{aligned}
\ell(\phi, \mu_0, \mu_1, \Sigma) &= \log \prod_{i=1}^{m} p(x^{(i)}, y^{(i)}; \phi, \mu_0, \mu_1, \Sigma) \\
&= \log \prod_{i=1}^{m} p(x^{(i)}|y^{(i)}; \mu_0, \mu_1, \Sigma) p(y^{(i)}; \phi).
\end{aligned}
$$

注意这里的参数有两个μ，表示在不同的结果模型下，特征均值不同，但我们假设协方差相同。反映在图上就是不同模型中心位置不同，但形状相同。这样就可以用直线来进行分隔判别。

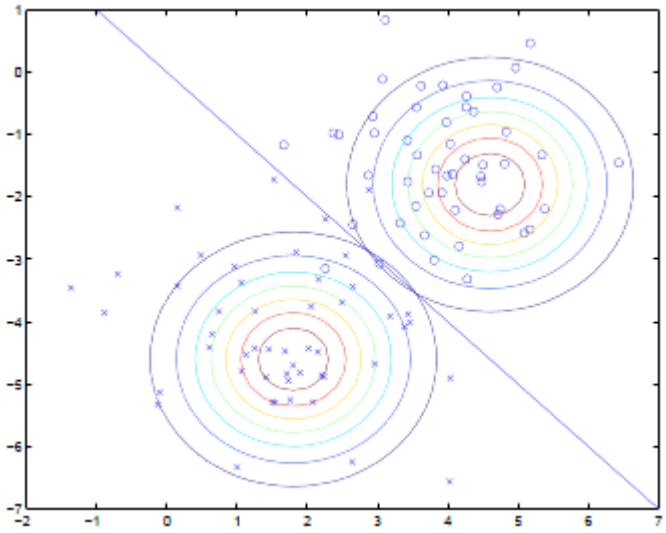
求导后，得到参数估计公式：

$$
\begin{aligned}
\phi &= \frac{1}{m}\sum_{i=1}^{m} 1\{y^{(i)}=1\} \\
\mu_0 &= \frac{\sum_{i=1}^{m} 1\{y^{(i)}=0\}x^{(i)}}{\sum_{i=1}^{m} 1\{y^{(i)}=0\}} \\
\mu_1 &= \frac{\sum_{i=1}^{m} 1\{y^{(i)}=1\}x^{(i)}}{\sum_{i=1}^{m} 1\{y^{(i)}=1\}} \\
\Sigma &= \frac{1}{m}\sum_{i=1}^{m} (x^{(i)}-\mu_{y^{(i)}})(x^{(i)}-\mu_{y^{(i)}})^T.
\end{aligned}
$$

Φ是训练样本中结果 y=1 占有的比例。
μ_0是 y=0 的样本中特征均值。

μ_1是 y=1 的样本中特征均值。
Σ是样本特征方差均值。

如前面所述，在图上表示为：



直线两边的 y 值不同，但协方差矩阵相同，因此形状相同。μ不同，因此位置不同。

3） 高斯判别分析（GDA）与 logistic 回归的关系
将 GDA 用条件概率方式来表述的话，如下：

$$p(y = 1|x; \phi, \mu_0, \mu_1, \Sigma)$$

y 是 x 的函数，其中 $\phi, \mu_0, \mu_1, \Sigma$ 都是参数。
进一步推导出

$$p(y = 1|x; \phi, \Sigma, \mu_0, \mu_1) = \frac{1}{1 + \exp(-\theta^T x)},$$

这里的θ是 $\phi, \Sigma, \mu_0, \mu_1$ 的函数。
这个形式就是 logistic 回归的形式。
也就是说如果 p(x|y)符合多元高斯分布，那么 p(y|x)符合 logistic 回归模型。反之，不成立。为什么反过来不成立呢？因为 GDA 有着更强的假设条件和约束。

如果认定训练数据满足多元高斯分布，那么 GDA 能够在训练集上是最好的模型。然而，我们往往事先不知道训练数据满足什么样的分布，不能做很强的假设。Logistic 回归的条件假设要弱于 GDA，因此更多的时候采用 logistic 回归的方法。

例如，训练数据满足泊松分布，$x|y = 0 \sim \text{Poisson}(\lambda_0)$

$x|y=1 \sim \text{Poisson}(\lambda_1)$，那么 p(y|x)也是 logistic 回归的。这个时候如果采用 GDA，那么效果会比较差，因为训练数据特征的分布不是多元高斯分布，而是泊松分布。

这也是 logistic 回归用的更多的原因。

3 朴素贝叶斯模型

在 GDA 中，我们要求特征向量 x 是连续实数向量。如果 x 是离散值的话，可以考虑采用朴素贝叶斯的分类方法。

假如要分类垃圾邮件和正常邮件。分类邮件是文本分类的一种应用。

假设采用最简单的特征描述方法，首先找一部英语词典，将里面的单词全部列出来。然后将每封邮件表示成一个向量，向量中每一维都是字典中的一个词的 0/1 值，1 表示该词在邮件中出现，0 表示未出现。

比如一封邮件中出现了“a”和“buy”，没有出现“aardvark”、“aardwolf”和“zygmurgy”，那么可以形式化表示为：

$$x=\begin{bmatrix}1\\0\\0\\\vdots\\1\\\vdots\\0\end{bmatrix}\begin{matrix}\text{a}\\\text{aardvark}\\\text{aardwolf}\\\vdots\\\text{buy}\\\vdots\\\text{zygmurgy}\end{matrix}$$

假设字典中总共有 5000 个词，那么 x 是 5000 维的。这时候如果要建立多项式分布模型（二项分布的扩展）。

多项式分布（multinomial distribution）

某随机实验如果有 k 个可能结局 A1，A2，…，Ak，它们的概率分布分别是 p1，p2，…，pk，那么在 N 次采样的总结果中，A1 出现 n1 次，A2 出现 n2 次，…，Ak 出现 nk 次的这种事件的出现概率 P 有下面公式：（Xi 代表出现 ni 次）

$$P(X_1=x_1,\ldots,X_k=x_k)=\begin{cases}\frac{n!}{x_1!\cdots x_k!}p_1^{x_1}\cdots p_k^{x_k} & \text{when } \sum_{i=1}^{k}x_i=n\\0 & \text{otherwise.}\end{cases}$$

对应到上面的问题上来，把每封邮件当做一次随机试验，那么结果的可能性有2^{5000}种。意味着 pi 有2^{5000}个，参数太多，不可能用来建模。

换种思路，我们要求的是 p(y|x)，根据生成模型定义我们可以求 p(x|y)和 p(y)。假设 x 中的特征是条件独立的。这个称作朴素贝叶斯假设。如果一封邮件是垃圾邮件（y=1），且这封邮件出现词“buy”与这封邮件是否出现“price”无关，那么“buy”和“price”之间是条件独立的。

形式化表示为，（如果给定 Z 的情况下，X 和 Y 条件独立）：

$P(X|Z) = P(X|Y,Z)$

也可以表示为：

$P(X,Y|Z) = P(X|Z)P(Y|Z)$

回到问题中

$$\begin{aligned} & p(x_1, \ldots, x_{50000}|y) \\ &= p(x_1|y)p(x_2|y,x_1)p(x_3|y,x_1,x_2)\cdots p(x_{50000}|y,x_1,\ldots,x_{49999}) \\ &= p(x_1|y)p(x_2|y)p(x_3|y)\cdots p(x_{50000}|y) \\ &= \prod_{i=1}^{n} p(x_i|y) \end{aligned}$$

这个与 NLP 中的 n 元语法模型有点类似，这里相当于 unigram。

这里我们发现朴素贝叶斯假设是约束性很强的假设，“buy”从通常上讲与“price”是有关系，我们这里假设的是条件独立。（注意条件独立和独立是不一样的）

建立形式化的模型表示：

$\phi_{i|y=1} = p(x_i = 1|y = 1)$

$\phi_{i|y=0} = p(x_i = 0|y = 1)$

$\phi_{y} = p(y = 1)$

那么我们想要的是模型在训练数据上概率积能够最大，即最大似然估计如下：

$$\mathcal{L}(\phi_y, \phi_{i|y=0}, \phi_{i|y=1}) = \prod_{i=1}^{m} p(x^{(i)}, y^{(i)}).$$

注意这里是联合概率分布积最大，说明朴素贝叶斯是生成模型。

求解得：

$$\begin{aligned} \phi_{j|y=1} &= \frac{\sum_{i=1}^{m} 1\{x_j^{(i)} = 1 \wedge y^{(i)} = 1\}}{\sum_{i=1}^{m} 1\{y^{(i)} = 1\}} \\ \phi_{j|y=0} &= \frac{\sum_{i=1}^{m} 1\{x_j^{(i)} = 1 \wedge y^{(i)} = 0\}}{\sum_{i=1}^{m} 1\{y^{(i)} = 0\}} \\ \phi_y &= \frac{\sum_{i=1}^{m} 1\{y^{(i)} = 1\}}{m} \end{aligned}$$

最后一个式子是表示 y=1 的样本数占全部样本数的比例，前两个表示在 y=1 或 0 的样本中，特征 Xj=1 的比例。

然而我们要求的是

$$
\begin{aligned}
p(y=1|x) &= \frac{p(x|y=1)p(y=1)}{p(x)} \\
&= \frac{\left(\prod_{i=1}^{n} p(x_i|y=1)\right)p(y=1)}{\left(\prod_{i=1}^{n} p(x_i|y=1)\right)p(y=1)+\left(\prod_{i=1}^{n} p(x_i|y=0)\right)p(y=0)},
\end{aligned}
$$

实际是求出分子即可，分母对 y=1 和 y=0 都一样。

当然，朴素贝叶斯方法可以扩展到 x 和 y 都有多个离散值的情况。对于特征是连续值的情况，我们也可以采用分段的方法来将连续值转化为离散值。具体怎么转化能够最优，我们可以采用信息增益的度量方法来确定（参见 Mitchell 的《机器学习》决策树那一章）。比如房子大小可以如下划分成离散值：

Living area (sq. feet)	< 400	400-800	800-1200	1200-1600	>1600
x_i	1	2	3	4	5

4 拉普拉斯平滑

朴素贝叶斯方法有个致命的缺点就是对数据稀疏问题过于敏感。

比如前面提到的邮件分类，现在新来了一封邮件，邮件标题是“NIPS call for papers”。我们使用更大的网络词典（词的数目由 5000 变为 35000）来分类，假设 NIPS 这个词在字典中的位置是 35000。然而 NIPS 这个词没有在训练数据中出现过，这封邮件第一次出现了 NIPS。那我们算概率的时候如下：

$$
\begin{aligned}
\phi_{35000|y=1} &= \frac{\sum_{i=1}^{m} 1\{x_{35000}^{(i)}=1 \wedge y^{(i)}=1\}}{\sum_{i=1}^{m} 1\{y^{(i)}=1\}} = 0 \\
\phi_{35000|y=0} &= \frac{\sum_{i=1}^{m} 1\{x_{35000}^{(i)}=1 \wedge y^{(i)}=0\}}{\sum_{i=1}^{m} 1\{y^{(i)}=0\}} = 0
\end{aligned}
$$

由于 NIPS 在以前的不管是垃圾邮件还是正常邮件都没出现过，那么结果只能是 0 了。显然最终的条件概率也是 0。

$$
\begin{aligned}
p(y=1|x) &= \frac{\prod_{i=1}^{n} p(x_i|y=1)p(y=1)}{\prod_{i=1}^{n} p(x_i|y=1)p(y=1)+\prod_{i=1}^{n} p(x_i|y=0)p(y=0)} \\
&= \frac{0}{0}.
\end{aligned}
$$

原因就是我们的特征概率条件独立，使用的是相乘的方式来得到结果。

为了解决这个问题，我们打算给未出现特征值，赋予一个“小”的值而不是 0。

具体平滑方法如下：

假设离散型随机变量 z 有{1,2,…,k}个值，我们用Φ_i = p(z = i)来表示每个值的概率。假设有 m 个训练样本中，z 的观察值是$\{z^{(1)},\ldots,z^{(m)}\}$，其中每一个观察值对应 k 个值中的一个。那么根据原来的估计方法可以得到

$$\phi_j = \frac{\sum_{i=1}^{m} 1\{z^{(i)} = j\}}{m}.$$

说白了就是 z=j 出现的比例。

拉普拉斯平滑法将每个 k 值出现次数事先都加 1，通俗讲就是假设他们都出现过一次。

那么修改后的表达式为：

$$\phi_j = \frac{\sum_{i=1}^{m} 1\{z^{(i)} = j\} + 1}{m + k}.$$

每个 z=j 的分子都加 1，分母加 k。可见$\sum_{j=1}^{k} \phi_j = 1$ 。

这个有点像 NLP 里面的加一平滑法，当然还有 n 多平滑法了，这里不再详述。

回到邮件分类的问题，修改后的公式为：

$$\phi_{j|y=1} = \frac{\sum_{i=1}^{m} 1\{x_j^{(i)} = 1 \wedge y^{(i)} = 1\} + 1}{\sum_{i=1}^{m} 1\{y^{(i)} = 1\} + 2}$$

$$\phi_{j|y=0} = \frac{\sum_{i=1}^{m} 1\{x_j^{(i)} = 1 \wedge y^{(i)} = 0\} + 1}{\sum_{i=1}^{m} 1\{y^{(i)} = 0\} + 2}$$

5 文本分类的事件模型

回想一下我们刚刚使用的用于文本分类的朴素贝叶斯模型，这个模型称作多值伯努利事件模型（multi-variate Bernoulli event model）。在这个模型中，我们首先随机选定了邮件的类型（垃圾或者普通邮件，也就是 p(y)），然后一个人翻阅词典，从第一个词到最后一个词，随机决定一个词是否要在邮件中出现，出现标示为 1，否则标示为 0。然后将出现的词组成一封邮件。决定一个词是否出现依照概率 p(xi|y)。那么这封邮件的概率可以标示为 $p(y)\prod_{i=1}^{n} p(x_i|y)$。

让我们换一个思路，这次我们不先从词典入手，而是选择从邮件入手。让 i 表示邮件中的第 i 个词，xi 表示这个词在字典中的位置，那么 xi 取值范围为{1,2,…|V|}，|V|是字典中词的数目。这样一封邮件可以表示成$(x_1, x_2, \ldots, x_n)$，n 可以变化，因为每封邮件的词的个数不同。然后我们对于每个 xi 随机从|V|个值中取一个，这样就形成了一封邮件。这相当于重复投掷|V|面的骰子，将观察值记录下来就形成了一封邮件。当然每个面的概率服从 p(xi|y)，而且每次试验条件独立。这样我们得到的邮件概率是 $p(y)\prod_{i=1}^{n} p(x_i|y)$。居然跟上面的一样，那么不同点在哪呢？注意第一个的 n 是字典中的全部的词，下面这个 n 是邮件中的词个数。上面 xi 表示一个词是否出现，只有 0 和 1 两个值，两者概率和为 1。下面的 xi 表示|V|中的一个值，|V|个 p(xi|y)相加和为 1。是多值二项分布模型。上面的 x 向量都是 0/1 值，下面的 x 的向量都是字典中的位置。

形式化表示为：

m 个训练样本表示为：$\{(x^{(i)}, y^{(i)}); i = 1, \ldots, m\}$

$$x^{(i)} = (x_1^{(i)}, x_2^{(i)}, \ldots, x_{n_i}^{(i)})$$

表示第 i 个样本中，共有 ni 个词，每个词在字典中的编号为$x_j^{(i)}$。

那么我们仍然按照朴素贝叶斯的方法求得最大似然估计概率为

$$\begin{aligned}\mathcal{L}(\phi, \phi_{i|y=0}, \phi_{i|y=1}) &= \prod_{i=1}^{m} p(x^{(i)}, y^{(i)}) \\ &= \prod_{i=1}^{m} \left(\prod_{j=1}^{n_i} p(x_j^{(i)}|y; \phi_{i|y=0}, \phi_{i|y=1}) \right) p(y^{(i)}; \phi_y).\end{aligned}$$

解得，

$$\begin{aligned}\phi_{k|y=1} &= \frac{\sum_{i=1}^{m} \sum_{j=1}^{n_i} 1\{x_j^{(i)} = k \wedge y^{(i)} = 1\}}{\sum_{i=1}^{m} 1\{y^{(i)} = 1\} n_i} \\ \phi_{k|y=0} &= \frac{\sum_{i=1}^{m} \sum_{j=1}^{n_i} 1\{x_j^{(i)} = k \wedge y^{(i)} = 0\}}{\sum_{i=1}^{m} 1\{y^{(i)} = 0\} n_i} \\ \phi_y &= \frac{\sum_{i=1}^{m} 1\{y^{(i)} = 1\}}{m}.\end{aligned}$$

与以前的式子相比，分母多了个 ni，分子由 0/1 变成了 k。

举个例子：

X1	X2	X3	Y
1	2	-	1
2	1	-	0
1	3	2	0
3	3	3	1

假如邮件中只有 a，b，c 这三词，他们在词典的位置分别是 1,2,3，前两封邮件都只有 2 个词，后两封有 3 个词。

Y=1 是垃圾邮件。

那么，

$$\Phi_{1|y=1} = \frac{1+0}{2+3} = \frac{1}{5},\ \Phi_{2|y=1} = \frac{1}{5},\ \Phi_{3|y=1} = \frac{3}{5}$$

$$\Phi_{1|y=0} = \frac{2+0}{2+3} = \frac{2}{5},\ \Phi_{2|y=0} = \frac{2}{5},\ \Phi_{3|y=0} = \frac{1}{5}$$

$$\Phi_{y=1} = \frac{1}{2},\ \Phi_{y=0} = \frac{1}{2}$$

假如新来一封邮件为 b，c 那么特征表示为{2,3}。

那么

$$P(y=1|x)=\frac{p(x,y=1)}{p(x)}=\frac{p(x=\{2,3\}|y=1)p(y=1)}{p(x=\{2,3\})}$$
$$=\frac{\Phi_{2|y=1}\Phi_{3|y=1}\Phi_{y=1}}{\Phi_{2|y=1}\Phi_{3|y=1}\Phi_{y=1}+\Phi_{2|y=0}\Phi_{3|y=0}\Phi_{y=0}}$$
$$=\frac{0.2*0.6*0.5}{0.2*0.6*0.5+0.4*0.2*0.5}=0.6$$

那么该邮件是垃圾邮件概率是 0.6。

注意这个公式与朴素贝叶斯的不同在于这里针对整体样本求的$\Phi_{k|y=1}$，而朴素贝叶斯里面针对每个特征求的$\Phi_{xj=1|y=1}$，而且这里的特征值维度是参差不齐的。

这里如果假如拉普拉斯平滑，得到公式为：

$$\phi_{k|y=1} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n_i}1\{x_j^{(i)}=k\wedge y^{(i)}=1\}+1}{\sum_{i=1}^{m}1\{y^{(i)}=1\}n_i+|V|}$$
$$\phi_{k|y=0} = \frac{\sum_{i=1}^{m}\sum_{j=1}^{n_i}1\{x_j^{(i)}=k\wedge y^{(i)}=0\}+1}{\sum_{i=1}^{m}1\{y^{(i)}=0\}n_i+|V|}.$$

表示每个 k 值至少发生过一次。

另外朴素贝叶斯虽然有时候不是最好的分类方法，但它简单有效，而且速度快。

支持向量机（上）

JerryLead
csxulijie@gmail.com
2011 年 3 月 12 日星期六

1 简介

支持向量机基本上是最好的有监督学习算法了。最开始接触 SVM 是去年暑假的时候，老师要求交《统计学习理论》的报告，那时去网上下了一份入门教程，里面讲的很通俗，当时只是大致了解了一些相关概念。这次斯坦福提供的学习材料，让我重新学习了一些 SVM 知识。我看很多正统的讲法都是从 VC 维理论和结构风险最小原理出发，然后引出 SVM 什么的，还有些资料上来就讲分类超平面什么的。这份材料从前几节讲的 logistic 回归出发，引出了 SVM，既揭示了模型间的联系，也让人觉得过渡更自然。

2 重新审视 logistic 回归

Logistic 回归目的是从特征学习出一个 0/1 分类模型，而这个模型是将特性的线性组合作为自变量，由于自变量的取值范围是负无穷到正无穷。因此，使用 logistic 函数（或称作 sigmoid 函数）将自变量映射到(0,1)上，映射后的值被认为是属于 y=1 的概率。

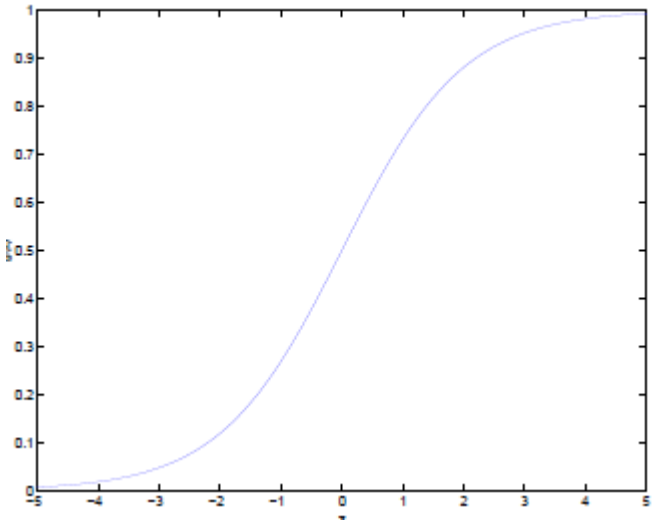
形式化表示就是

假设函数

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

其中 x 是 n 维特征向量，函数 g 就是 logistic 函数。

$g(z) = \frac{1}{1 + e^{-z}}$ 的图像是



可以看到，将无穷映射到了(0,1)。

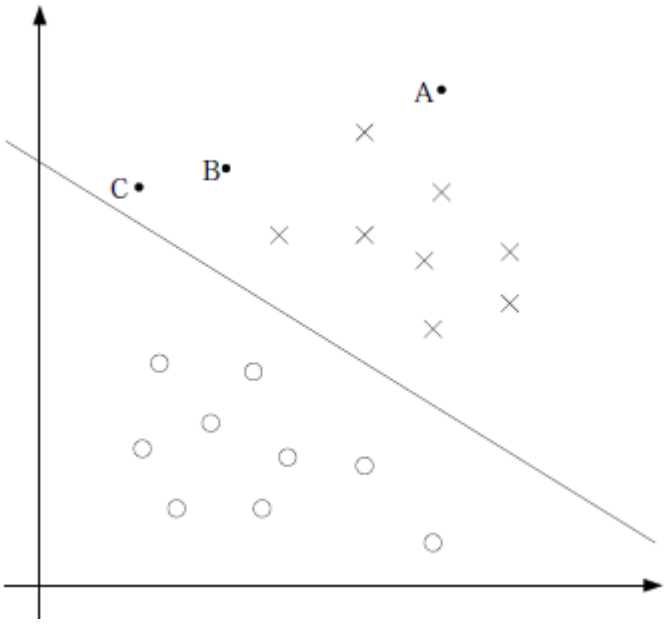
而假设函数就是特征属于 y=1 的概率。

$$P(y=1 \mid x;\theta) = h_\theta(x)$$
$$P(y=0 \mid x;\theta) = 1 - h_\theta(x)$$

当我们要判别一个新来的特征属于哪个类时，只需求$h_\theta(x)$，若大于 0.5 就是 y=1 的类，反之属于 y=0 类。

再审视一下$h_\theta(x)$，发现$h_\theta(x)$只和$\theta^T x$有关，$\theta^T x>0$，那么$h_\theta(x) > 0.5$，g(z)只不过是用来映射，真实的类别决定权还在$\theta^T x$。还有当$\theta^T x \gg 0$时，$h_\theta(x)=1$，反之$h_\theta(x)=0$。如果我们只从$\theta^T x$出发，希望模型达到的目标无非就是让训练数据中 y=1 的特征$\theta^T x \gg 0$，而是 y=0 的特征$\theta^T x \ll 0$。Logistic 回归就是要学习得到θ，使得正例的特征远大于 0，负例的特征远小于 0，强调在全部训练实例上达到这个目标。

图形化表示如下：



中间那条线是$\theta^T x = 0$，logistic 回顾强调所有点尽可能地远离中间那条线。学习出的结

果也就中间那条线。考虑上面 3 个点 A、B 和 C。从图中我们可以确定 A 是×类别的，然而 C 我们是不太确定的，B 还算能够确定。这样我们可以得出结论，我们更应该关心靠近中间分割线的点，让他们尽可能地远离中间线，而不是在所有点上达到最优。因为那样的话，要使得一部分点靠近中间线来换取另外一部分点更加远离中间线。我想这就是支持向量机的思路和 logistic 回归的不同点，一个考虑局部（不关心已经确定远离的点），一个考虑全局（已经远离的点可能通过调整中间线使其能够更加远离）。这是我的个人直观理解。

3 形式化表示

我们这次使用的结果标签是 y=-1,y=1，替换在 logistic 回归中使用的 y=0 和 y=1。同时将 θ替换成 w 和 b。以前的$\theta^T x = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$，其中认为$x_0 = 1$。现在我们替换$\theta_0$为 b，后面替换$\theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$为$w_1 x_1 + w_2 x_2 + \cdots + w_n x_n$（即$w^T x$）。这样，我们让$\theta^T x = w^T x + b$，进一步$h_\theta(x) = g(\theta^T x) = g(w^T x + b)$。也就是说除了 y 由 y=0 变为 y=-1，只是标记不同外，与 logistic 回归的形式化表示没区别。再明确下假设函数

$$h_{w,b}(x) = g(w^T x + b)$$

上一节提到过我们只需考虑$\theta^T x$的正负问题，而不用关心 g(z)，因此我们这里将 g(z)做一个简化，将其简单映射到 y=-1 和 y=1 上。映射关系如下：

$$g(z) = \begin{cases} 1, & z \geq 0 \\ -1, & z < 0 \end{cases}$$

4 函数间隔（functional margin）和几何间隔（geometric margin）

给定一个训练样本$(x^{(i)}, y^{(i)})$，x 是特征，y 是结果标签。i 表示第 i 个样本。我们定义函数间隔如下：

$$\hat{\gamma}^{(i)} = y^{(i)}(w^T x^{(i)} + b)$$

可想而知，当$y^{(i)} = 1$时，在我们的 g(z)定义中，$w^T x^{(i)} + b \geq 0$，$\hat{\gamma}^{(i)}$的值实际上就是$|w^T x^{(i)} + b|$。反之亦然。为了使函数间隔最大（更大的信心确定该例是正例还是反例），当$y^{(i)} = 1$时，$w^T x^{(i)} + b$应该是个大正数，反之是个大负数。因此函数间隔代表了我们认为特征是正例还是反例的确信度。

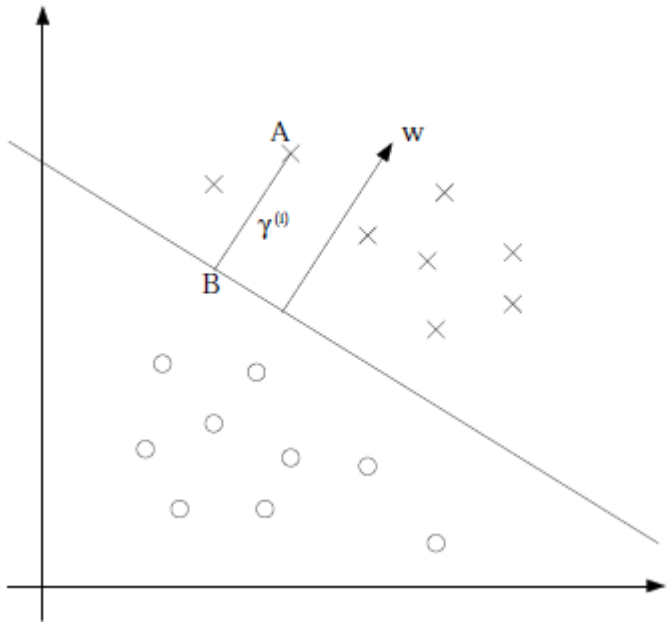
继续考虑 w 和 b，如果同时加大 w 和 b，比如在$(w^T x^{(i)} + b)$前面乘个系数比如 2，那么所有点的函数间隔都会增大二倍，这个对求解问题来说不应该有影响，因为我们要求解的是$w^T x + b = 0$，同时扩大 w 和 b 对结果是无影响的。这样，我们为了限制 w 和 b，可能需要加入归一化条件，毕竟求解的目标是确定唯一一个 w 和 b，而不是多组线性相关的向量。这个归一化一会再考虑。

刚刚我们定义的函数间隔是针对某一个样本的，现在我们定义全局样本上的函数间隔

$$\hat{\gamma} = \min_{i=1,\ldots,m} \hat{\gamma}^{(i)}.$$

说白了就是在训练样本上分类正例和负例确信度最小那个函数间隔。

接下来定义几何间隔，先看图



假设我们有了 B 点所在的$w^T x + b = 0$分割面。任何其他一点，比如 A 到该面的距离以$\gamma^{(i)}$表示，假设 B 就是 A 在分割面上的投影。我们知道向量 BA 的方向是w（分割面的梯度），单位向量是$\frac{w}{||w||}$。A 点是$(x^{(i)}, y^{(i)})$，所以 B 点是 x=$x^{(i)} - \gamma^{(i)} \frac{w}{||w||}$（利用初中的几何知识），带入$w^T x + b = 0$得，

$$w^T(x^{(i)} - \gamma^{(i)} \frac{w}{||w||}) + b = 0$$

进一步得到

$$\gamma^{(i)} = \frac{w^T x^{(i)} + b}{||w||} = \left(\frac{w}{||w||}\right)^T x^{(i)} + \frac{b}{||w||}.$$

$\gamma^{(i)}$实际上就是点到平面距离。

再换种更加优雅的写法：

$$\gamma^{(i)} = y^{(i)} \left(\left(\frac{w}{||w||}\right)^T x^{(i)} + \frac{b}{||w||} \right).$$

当$||w|| = 1$时，不就是函数间隔吗？是的，前面提到的函数间隔归一化结果就是几何间隔。他们为什么会一样呢？因为函数间隔是我们定义的，在定义的时候就有几何间隔的色彩。

同样，同时扩大 w 和 b，w 扩大几倍，||w||就扩大几倍，结果无影响。同样定义全局的几何间隔 $\gamma = \min_{i=1,\dots,m} \gamma^{(i)}.$

5 最优间隔分类器（optimal margin classifier）

回想前面我们提到我们的目标是寻找一个超平面，使得离超平面比较近的点能有更大的间距。也就是我们不考虑所有的点都必须远离超平面，我们关心求得的超平面能够让所有点中离它最近的点具有最大间距。形象的说，我们将上面的图看作是一张纸，我们要找一条折线，按照这条折线折叠后，离折线最近的点的间距比其他折线都要大。形式化表示为：

$$\begin{aligned} \max_{\gamma,w,b} \quad & \gamma \\ \text{s.t.} \quad & y^{(i)}(w^T x^{(i)} + b) \geq \gamma, \quad i = 1, \dots, m \\ & ||w|| = 1. \end{aligned}$$

这里用||w||=1 规约 w，使得$w^T x + b$是几何间隔。

到此，我们已经将模型定义出来了。如果求得了 w 和 b，那么来一个特征 x，我们就能够分类了，称为最优间隔分类器。接下的问题就是如何求解 w 和 b 的问题了。

由于||w|| = 1不是凸函数，我们想先处理转化一下，考虑几何间隔和函数间隔的关系，$\gamma = \frac{\hat{\gamma}}{||w||}$，我们改写一下上面的式子：

$$\begin{aligned} \max_{\gamma,w,b} \quad & \frac{\hat{\gamma}}{||w||} \\ \text{s.t.} \quad & y^{(i)}(w^T x^{(i)} + b) \geq \hat{\gamma}, \quad i = 1, \dots, m \end{aligned}$$

这时候其实我们求的最大值仍然是几何间隔，只不过此时的 w 不受||w|| = 1的约束了。然而这个时候目标函数仍然不是凸函数，没法直接代入优化软件里计算。我们还要改写。前面说到同时扩大 w 和 b 对结果没有影响，但我们最后要求的仍然是 w 和 b 的确定值，不是他们的一组倍数值，因此，我们需要对$\hat{\gamma}$做一些限制，以保证我们解是唯一的。这里为了简便我们取$\hat{\gamma} = 1$。这样的意义是将全局的函数间隔定义为 1，也即是将离超平面最近的点的距离定义为$\frac{1}{||w||}$。由于求$\frac{1}{||w||}$的最大值相当于求$\frac{1}{2}||w||^2$的最小值，因此改写后结果为：

$$\begin{aligned} \min_{\gamma,w,b} \quad & \frac{1}{2}||w||^2 \\ \text{s.t.} \quad & y^{(i)}(w^T x^{(i)} + b) \geq 1, \quad i = 1, \dots, m \end{aligned}$$

这下好了，只有线性约束了，而且是个典型的二次规划问题（目标函数是自变量的二次函数）。代入优化软件可解。

到这里发现，这个讲义虽然没有像其他讲义一样先画好图，画好分类超平面，在图上标示出间隔那么直观，但每一步推导有理有据，依靠思路的流畅性来推导出目标函数和约束。

接下来介绍的是手工求解的方法了，一种更优的求解方法。

6 拉格朗日对偶（Lagrange duality）

先抛开上面的二次规划问题，先来看看存在等式约束的极值问题求法，比如下面的最优化问题：

$$\begin{aligned} \min_w \quad & f(w) \\ \text{s.t.} \quad & h_i(w) = 0, \quad i = 1, \ldots, l. \end{aligned}$$

目标函数是 f(w)，下面是等式约束。通常解法是引入拉格朗日算子，这里使用β来表示算子，得到拉格朗日公式为

$$\mathcal{L}(w, \beta) = f(w) + \sum_{i=1}^{l} \beta_i h_i(w)$$

L 是等式约束的个数。

然后分别对 w 和β求偏导，使得偏导数等于 0，然后解出 w 和β_i。至于为什么引入拉格朗日算子可以求出极值，原因是 f(w)的 dw 变化方向受其他不等式的约束，dw 的变化方向与 f(w)的梯度垂直时才能获得极值，而且在极值处，f(w)的梯度与其他等式梯度的线性组合平行，因此他们之间存在线性关系。（参考《最优化与 KKT 条件》）

然后我们探讨有不等式约束的极值问题求法，问题如下：

$$\begin{aligned} \min_w \quad & f(w) \\ \text{s.t.} \quad & g_i(w) \le 0, \quad i = 1, \ldots, k \\ & h_i(w) = 0, \quad i = 1, \ldots, l. \end{aligned}$$

我们定义一般化的拉格朗日公式

$$\mathcal{L}(w, \alpha, \beta) = f(w) + \sum_{i=1}^{k} \alpha_i g_i(w) + \sum_{i=1}^{l} \beta_i h_i(w).$$

这里的α_i和β_i都是拉格朗日算子。如果按这个公式求解，会出现问题，因为我们求解的是最小值，而这里的$g_i(w) \le 0$，我们可以将α_i调整成很大的正值，来使最后的函数结果是负无穷。因此我们需要排除这种情况，我们定义下面的函数：

$$\theta_{\mathcal{P}}(w) = \max_{\alpha,\beta:\alpha_i\geq 0} \mathcal{L}(w,\alpha,\beta).$$

这里的 P 代表 primal。假设$g_i(w) > 0$或者$h_i(w) \neq 0$，那么我们总是可以调整α_i和β_i来使得$\theta_{\mathcal{P}}$(w)有最大值为正无穷。而只有 g 和 h 满足约束时，$\theta_{\mathcal{P}}$(w)为 f(w)。这个函数的精妙之处在于$\alpha_i \geq 0$，而且求极大值。

因此我们可以写作

$$\theta_{\mathcal{P}}(w) = \begin{cases} f(w) & \text{if } w \text{ satisfies primal constraints} \\ \infty & \text{otherwise.} \end{cases}$$

这样我们原来要求的 min f(w)可以转换成求$\min_w \theta_{\mathcal{P}}$(w)了。

$$\min_w \theta_{\mathcal{P}}(w) = \min_w \max_{\alpha,\beta:\alpha_i\geq 0} \mathcal{L}(w,\alpha,\beta),$$

我们使用p^*来表示$\min_w \theta_{\mathcal{P}}$(w)。如果直接求解，首先面对的是两个参数，而$\alpha_i$也是不等式约束，然后再在 w 上求最小值。这个过程不容易做，那么怎么办呢？

我们先考虑另外一个问题 $\theta_{\mathcal{D}}(\alpha,\beta) = \min_w \mathcal{L}(w,\alpha,\beta).$

D 的意思是对偶，$\theta_{\mathcal{D}}(\alpha,\beta)$将问题转化为先求拉格朗日关于 w 的最小值，将α和β看作是固定值。之后在 $\theta_{\mathcal{D}}(\alpha,\beta)$求最大值的话：

$$\max_{\alpha,\beta:\alpha_i\geq 0} \theta_{\mathcal{D}}(\alpha,\beta) = \max_{\alpha,\beta:\alpha_i\geq 0} \min_w \mathcal{L}(w,\alpha,\beta).$$

这个问题是原问题的对偶问题，相对于原问题只是更换了 min 和 max 的顺序，而一般更换顺序的结果是 Max Min(X) <= Min Max(X)。然而在这里两者相等。用d^*来表示对偶问题如下：

$$d^* = \max_{\alpha,\beta:\alpha_i\geq 0} \min_w \mathcal{L}(w,\alpha,\beta) \leq \min_w \max_{\alpha,\beta:\alpha_i\geq 0} \mathcal{L}(w,\alpha,\beta) = p^*.$$

下面解释在什么条件下两者会等价。假设 f 和 g 都是凸函数，h 是仿射的（affine，there exists a_i, b_i, so that $h_i(w) = a_i^T w + b_i$）。并且存在 w 使得对于所有的 i，$g_i(w) < 0$。在这种假设下，一定存在w^*, α^*, β^*使得w^*是原问题的解，α^*, β^*是对偶问题的解。还有

$p^* = d^* = \mathcal{L}(w^*, \alpha^*, \beta^*)$.另外，w*, α^*, β^*满足库恩-塔克条件（Karush-Kuhn-Tucker, KKT condition），该条件如下：

$$\frac{\partial}{\partial w_i}\mathcal{L}(w^*, \alpha^*, \beta^*) = 0, \quad i = 1, \ldots, n \tag{3}$$

$$\frac{\partial}{\partial \beta_i}\mathcal{L}(w^*, \alpha^*, \beta^*) = 0, \quad i = 1, \ldots, l \tag{4}$$

$$\alpha_i^* g_i(w^*) = 0, \quad i = 1, \ldots, k \tag{5}$$

$$g_i(w^*) \leq 0, \quad i = 1, \ldots, k \tag{6}$$

$$\alpha^* \geq 0, \quad i = 1, \ldots, k \tag{7}$$

所以如果w*, α^*, β^*满足了库恩-塔克条件，那么他们就是原问题和对偶问题的解。让我们再次审视公式（5），这个条件称作是 KKT dual complementarity 条件。这个条件隐含了如果$\alpha^* > 0$，那么$g_i(w^*) = 0$。也就是说，$g_i(w^*) = 0$时，w 处于可行域的边界上，这时才是起作用的约束。而其他位于可行域内部（$g_i(w^*) < 0$的）点都是不起作用的约束，其$\alpha^* = 0$。这个 KKT 双重补足条件会用来解释支持向量和 SMO 的收敛测试。

这部分内容思路比较凌乱，还需要先研究下《非线性规划》中的约束极值问题，再回头看看。KKT 的总体思想是认为极值会在可行域边界上取得，也就是不等式为 0 或等式约束里取得，而最优下降方向一般是这些等式的线性组合，其中每个元素要么是不等式为 0 的约束，要么是等式约束。对于在可行域边界内的点，对最优解不起作用，因此前面的系数为 0。

7 最优间隔分类器（optimal margin classifier）

重新回到 SVM 的优化问题：

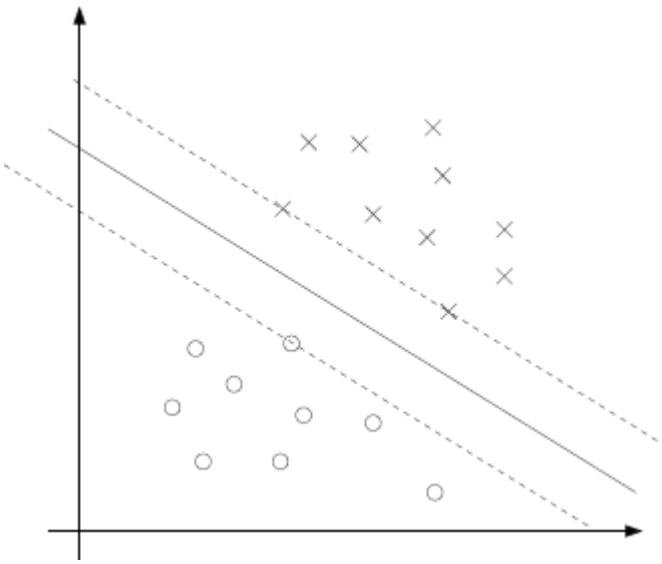
$$\begin{aligned} \min_{\gamma, w, b} \quad & \frac{1}{2}||w||^2 \\ \text{s.t.} \quad & y^{(i)}(w^T x^{(i)} + b) \geq 1, \quad i = 1, \ldots, m \end{aligned}$$

我们将约束条件改写为：

$$g_i(w) = -y^{(i)}(w^T x^{(i)} + b) + 1 \leq 0.$$

从 KKT 条件得知只有函数间隔是 1（离超平面最近的点）的线性约束式前面的系数$\alpha_i > 0$，也就是说这些约束式$g_i(\mathrm{w}) = 0$，对于其他的不在线上的点($g_i(\mathrm{w}) < 0$)，极值不会在他们所在的范围内取得，因此前面的系数$\alpha_i = 0$.注意每一个约束式实际就是一个训练样本。

看下面的图：



实线是最大间隔超平面，假设×号的是正例，圆圈的是负例。在虚线上的点就是函数间隔是 1 的点，那么他们前面的系数$\alpha_i > 0$，其他点都是$\alpha_i = 0$。这三个点称作支持向量。构造拉格朗日函数如下：

$$\mathcal{L}(w,b,\alpha) = \frac{1}{2}||w||^2 - \sum_{i=1}^{m} \alpha_i \left[y^{(i)}(w^T x^{(i)} + b) - 1\right].$$

注意到这里只有α_i没有β_i是因为原问题中没有等式约束，只有不等式约束。

下面我们按照对偶问题的求解步骤来一步步进行，

$$d^* = \max_{\alpha,\beta:\alpha_i \geq 0} \min_{w} \mathcal{L}(w,\alpha,\beta)$$

首先求解 $\mathcal{L}(w,b,\alpha)$ 的最小值，对于固定的α_i， $\mathcal{L}(w,b,\alpha)$ 的最小值只与 w 和 b 有关。对 w 和 b 分别求偏导数。

$$\nabla_w \mathcal{L}(w,b,\alpha) = w - \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)} = 0$$

$$\frac{\partial}{\partial b}\mathcal{L}(w,b,\alpha) = \sum_{i=1}^{m} \alpha_i y^{(i)} = 0.$$

并得到

$$w = \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)}.$$

将上式带回到拉格朗日函数中得到，此时得到的是该函数的最小值（目标函数是凸函数）

化简过程如下：

$$\mathcal{L}(\mathrm{w},\mathrm{b},\alpha)=\frac{1}{2}\|w\|^2-\sum_{i=1}^{m}\alpha_i\left[y^{(i)}\left(w^Tx^{(i)}+b\right)-1\right]$$

$$=\frac{1}{2}\mathrm{w}^T\mathrm{w}-\sum_{i=1}^{m}\alpha_iy^{(i)}w^Tx^{(i)}-\sum_{i=1}^{m}\alpha_iy^{(i)}b+\sum_{i=1}^{m}\alpha_i$$

$$=\frac{1}{2}\mathrm{w}^T\sum_{i=1}^{m}\alpha_iy^{(i)}x^{(i)}-\sum_{i=1}^{m}\alpha_iy^{(i)}w^Tx^{(i)}-\sum_{i=1}^{m}\alpha_iy^{(i)}b+\sum_{i=1}^{m}\alpha_i$$

$$=\frac{1}{2}\mathrm{w}^T\sum_{i=1}^{m}\alpha_iy^{(i)}x^{(i)}-w^T\sum_{i=1}^{m}\alpha_iy^{(i)}x^{(i)}-\sum_{i=1}^{m}\alpha_iy^{(i)}b+\sum_{i=1}^{m}\alpha_i$$

$$=-\frac{1}{2}\mathrm{w}^T\sum_{i=1}^{m}\alpha_iy^{(i)}x^{(i)}-\sum_{i=1}^{m}\alpha_iy^{(i)}b+\sum_{i=1}^{m}\alpha_i$$

$$=-\frac{1}{2}\mathrm{w}^T\sum_{i=1}^{m}\alpha_iy^{(i)}x^{(i)}-b\sum_{i=1}^{m}\alpha_iy^{(i)}+\sum_{i=1}^{m}\alpha_i$$

$$=-\frac{1}{2}\left(\sum_{i=1}^{m}\alpha_iy^{(i)}x^{(i)}\right)^T\sum_{i=1}^{m}\alpha_iy^{(i)}x^{(i)}-b\sum_{i=1}^{m}\alpha_iy^{(i)}+\sum_{i=1}^{m}\alpha_i$$

$$=-\frac{1}{2}\sum_{i=1}^{m}\alpha_iy^{(i)}\left(x^{(i)}\right)^T\sum_{i=1}^{m}\alpha_iy^{(i)}x^{(i)}-b\sum_{i=1}^{m}\alpha_iy^{(i)}+\sum_{i=1}^{m}\alpha_i$$

$$=-\frac{1}{2}\sum_{i=1,j=1}^{m}\alpha_iy^{(i)}\left(x^{(i)}\right)^T\alpha_jy^{(j)}x^{(j)}-b\sum_{i=1}^{m}\alpha_iy^{(i)}+\sum_{i=1}^{m}\alpha_i$$

$$=\sum_{i=1}^{m}\alpha_i-\frac{1}{2}\sum_{i=1,j=1}^{m}y^{(i)}y^{(i)}\alpha_i\alpha_j(x^{(i)})^T\,x^{(j)}-b\sum_{i=1}^{m}\alpha_iy^{(i)}$$

“倒数第 4 步”推导到“倒数第 3 步”使用了线性代数的转置运算，由于α_i和$y^{(i)}$都是实数，因此转置后与自身一样。“倒数第 3 步”推导到“倒数第 2 步”使用了(a+b+c+…)(a+b+c+…)=aa+ab+ac+ba+bb+bc+…的乘法运算法则。最后一步是上一步的顺序调整。

最后得到：

$$\mathcal{L}(w,b,\alpha)=\sum_{i=1}^{m}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{m}y^{(i)}y^{(j)}\alpha_i\alpha_j(x^{(i)})^Tx^{(j)}-b\sum_{i=1}^{m}\alpha_i y^{(i)}.$$

由于最后一项是 0，因此简化为

$$\mathcal{L}(w,b,\alpha)=\sum_{i=1}^{m}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{m}y^{(i)}y^{(j)}\alpha_i\alpha_j(x^{(i)})^Tx^{(j)}.$$

这里我们将向量内积$(x^{(i)})^Tx^{(j)}$表示为$\langle x^{(i)},x^{(j)}\rangle$.

此时的拉格朗日函数只包含了变量α_i。然而我们求出了α_i才能得到 w 和 b。

接着是极大化的过程 $d^*=\max_{\alpha,\beta:\alpha_i\geq 0}\min_w\mathcal{L}(w,\alpha,\beta)$，

$$\begin{aligned}\max_\alpha\quad & W(\alpha)=\sum_{i=1}^{m}\alpha_i-\frac{1}{2}\sum_{i,j=1}^{m}y^{(i)}y^{(j)}\alpha_i\alpha_j\langle x^{(i)},x^{(j)}\rangle.\\ \text{s.t.}\quad & \alpha_i\geq 0,\quad i=1,\dots,m\\ & \sum_{i=1}^{m}\alpha_i y^{(i)}=0,\end{aligned}$$

前面提到过对偶问题和原问题满足的几个条件，首先由于目标函数和线性约束都是凸函数，而且这里不存在等式约束 h。存在 w 使得对于所有的 i，$g_i(w)<0$。因此，一定存在w^*,α^*使得w^*是原问题的解，α^*是对偶问题的解。在这里，求α_i就是求α^*了。

如果求出了α_i，根据 $w=\sum_{i=1}^{m}\alpha_i y^{(i)}x^{(i)}$. 即可求出 w（也是$w^*$，原问题的解）。然后

$$b^*=-\frac{\max_{i:y^{(i)}=-1}w^{*T}x^{(i)}+\min_{i:y^{(i)}=1}w^{*T}x^{(i)}}{2}.$$

即可求出 b。即离超平面最近的正的函数间隔要等于离超平面最近的负的函数间隔。

关于上面的对偶问题如何求解，将留给下一篇中的 SMO 算法来阐明。

这里考虑另外一个问题，由于前面求解中得到

$$\mathrm{w} = \sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)}$$

我们通篇考虑问题的出发点是$w^T x + b$，根据求解得到的α_i，我们代入前式得到

$$\begin{aligned} w^T x + b &= \left(\sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)} \right)^T x + b \\ &= \sum_{i=1}^{m} \alpha_i y^{(i)} \langle x^{(i)}, x \rangle + b. \end{aligned}$$

也就是说，以前新来的要分类的样本首先根据 w 和 b 做一次线性运算，然后看求的结果是大于 0 还是小于 0,来判断正例还是负例。现在有了α_i，我们不需要求出 w，只需将新来的样本和训练数据中的所有样本做内积和即可。那有人会说，与前面所有的样本都做运算是不是太耗时了？其实不然，我们从 KKT 条件中得到，只有支持向量的$\alpha_i > 0$，其他情况$\alpha_i = 0$。因此，我们只需求新来的样本和支持向量的内积，然后运算即可。这种写法为下面要提到的核函数（kernel）做了很好的铺垫。这是上篇，先写这么多了。

支持向量机 SVM（下）

JerryLead
csxulijie@gmail.com
2011 年 3 月 17 日星期四

7 核函数（Kernels）

考虑我们最初在“线性回归”中提出的问题，特征是房子的面积 x，这里的 x 是实数，结果 y 是房子的价格。假设我们从样本点的分布中看到 x 和 y 符合 3 次曲线，那么我们希望使用 x 的三次多项式来逼近这些样本点。那么首先需要将特征 x 扩展到三维(x, x^2, x^3)，然后寻找特征和结果之间的模型。我们将这种特征变换称作特征映射（feature mapping）。映射函数称作ϕ，在这个例子中

$$\phi(x) = \begin{bmatrix} x \\ x^2 \\ x^3 \end{bmatrix}$$

我们希望将得到的特征映射后的特征应用于 SVM 分类，而不是最初的特征。这样，我们需要将前面$w^T x + b$公式中的内积从$< x^{(i)}, x >$，映射到$< \phi(x^{(i)}), \phi(x) >$。

至于为什么需要映射后的特征而不是最初的特征来参与计算，上面提到的（为了更好地拟合）是其中一个原因，另外的一个重要原因是样例可能存在线性不可分的情况，而将特征映射到高维空间后，往往就可分了。（在《数据挖掘导论》Pang-Ning Tan 等人著的《支持向量机》那一章有个很好的例子说明）

将核函数形式化定义，如果原始特征内积是$< x, z >$，映射后为$< \phi(x), \phi(z) >$，那么定义核函数（Kernel）为

$K(x, z) = \phi(x)^T \phi(z)$

到这里，我们可以得出结论，如果要实现该节开头的效果，只需先计算$\phi(x)$，然后计算$\phi(x)^T \phi(z)$即可，然而这种计算方式是非常低效的。比如最初的特征是 n 维的，我们将其映射到n^2维，然后再计算，这样需要$O(n^2)$的时间。那么我们能不能想办法减少计算时间呢？

先看一个例子，假设 x 和 z 都是 n 维的，

$$K(x, z) = (x^T z)^2$$

展开后，得

$$K(x, z) = (x^T z)^2 = \left(\sum_{i=1}^{n} x_i z_i \right) \left(\sum_{j=1}^{n} x_j z_j \right) = \sum_{i=1}^{n} \sum_{j=1}^{n} x_i x_j z_i z_j$$

$$= \sum_{i=1}^{n} \sum_{j=1}^{n} (x_i x_j)(z_i z_j) = \phi(x)^T \phi(z)$$

这个时候发现我们可以只计算原始特征 x 和 z 内积的平方（时间复杂度是 O(n)），就等价与计算映射后特征的内积。也就是说我们不需要花$O(n^2)$时间了。

现在看一下映射函数（n=3 时），根据上面的公式，得到

$$\phi(x) = \begin{bmatrix} x_1x_1 \\ x_1x_2 \\ x_1x_3 \\ x_2x_1 \\ x_2x_2 \\ x_2x_3 \\ x_3x_1 \\ x_3x_2 \\ x_3x_3 \end{bmatrix}.$$

也就是说核函数$K(x,z) = (x^Tz)^2$只能在选择这样的ϕ作为映射函数时才能够等价于映射后特征的内积。

再看一个核函数

$$\begin{aligned} K(x,z) &= (x^Tz + c)^2 \\ &= \sum_{i,j=1}^{n} (x_ix_j)(z_iz_j) + \sum_{i=1}^{n} (\sqrt{2c}x_i)(\sqrt{2c}z_i) + c^2. \end{aligned}$$

对应的映射函数（n=3 时）是

$$\phi(x) = \begin{bmatrix} x_1x_1 \\ x_1x_2 \\ x_1x_3 \\ x_2x_1 \\ x_2x_2 \\ x_2x_3 \\ x_3x_1 \\ x_3x_2 \\ x_3x_3 \\ \sqrt{2c}x_1 \\ \sqrt{2c}x_2 \\ \sqrt{2c}x_3 \\ c \end{bmatrix},$$

更一般地，核函数$K(x,z) = (x^Tz + c)^d$对应的映射后特征维度为$\binom{\mathrm{n}+\mathrm{d}}{d}$。（这个我一直没有理解）。

由于计算的是内积，我们可以想到 IR 中的余弦相似度，如果 x 和 z 向量夹角越小，那么核函数值越大，反之，越小。因此，核函数值是$\phi(x)$和$\phi(z)$的相似度。

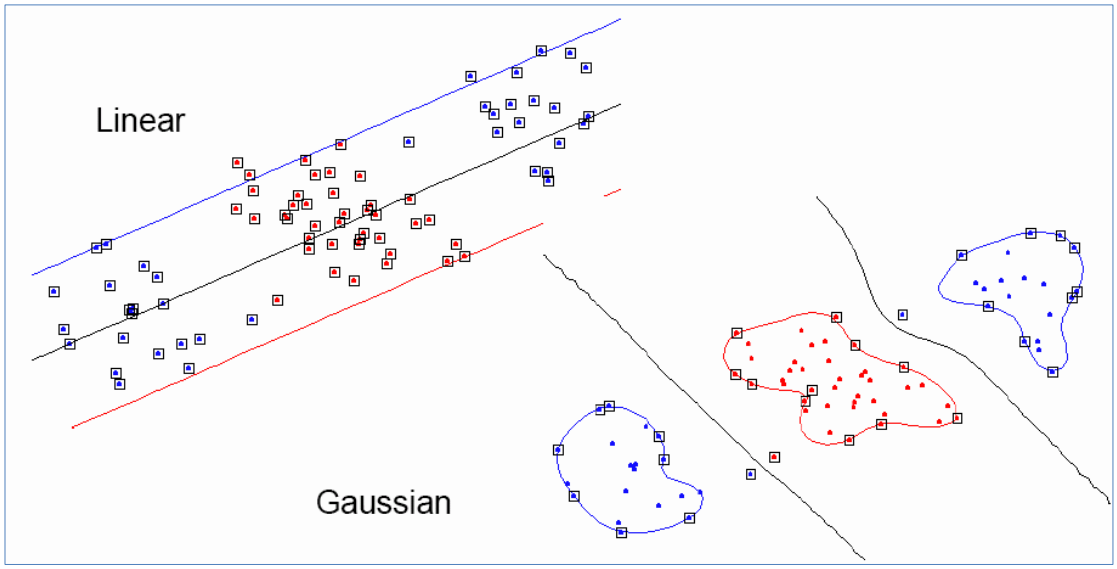
再看另外一个核函数

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

这时，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。

下面有张图说明在低维线性不可分时，映射到高维后就可分了，使用高斯核函数。



来自 Eric Xing 的 slides

注意，使用核函数后，怎么分类新来的样本呢？线性的时候我们使用 SVM 学习出 w 和 b，新来样本 x 的话，我们使用$w^T x + b$来判断，如果值大于等于 1，那么是正类，小于等于是负类。在两者之间，认为无法确定。如果使用了核函数后，$w^T x + b$就变成了$w^T \phi(x) + b$，是否先要找到$\phi(x)$，然后再预测？答案肯定不是了，找$\phi(x)$很麻烦，回想我们之前说过的

$$\begin{aligned} w^T x + b &= \left(\sum_{i=1}^{m} \alpha_i y^{(i)} x^{(i)} \right)^T x + b \\ &= \sum_{i=1}^{m} \alpha_i y^{(i)} \langle x^{(i)}, x \rangle + b. \end{aligned}$$

只需将$< x^{(i)}, x >$替换成$K(x^{(i)}, x)$，然后值的判断同上。

8 核函数有效性判定

问题：给定一个函数 K，我们能否使用 K 来替代计算 $\phi(x)^T \phi(z)$，也就说，是否能够找出一个ϕ，使得对于所有的 x 和 z，都有$K(x,z) = \phi(x)^T \phi(z)$?

比如给出了$K(x,z) = (x^T z)^2$，是否能够认为 K 是一个有效的核函数。

下面来解决这个问题，给定 m 个训练样本$\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$，每一个$x^{(i)}$对应一个特征向量。那么，我们可以将任意两个$x^{(i)}$和$x^{(j)}$带入 K 中，计算得到$K_{ij} = K(x^{(i)}, x^{(j)})$。I 可以从 1 到 m，j 可以从 1 到 m，这样可以计算出 m*m 的核函数矩阵（Kernel Matrix）。为了方便，我们将核函数矩阵和$K(x,z)$都使用 K 来表示。

如果假设 K 是有效的核函数，那么根据核函数定义

$$K_{ij} = K\left(x^{(i)}, x^{(j)}\right) = \phi\left(x^{(i)}\right)^T \phi\left(x^{(j)}\right) = \phi\left(x^{(j)}\right)^T \phi\left(x^{(i)}\right) = K\left(x^{(j)}, x^{(i)}\right) = K_{ji}$$

可见，矩阵 K 应该是个对称阵。让我们得出一个更强的结论，首先使用符号$\phi_k(x)$来表示映射函数$\phi(x)$的第 k 维属性值。那么对于任意向量 z，得

$$\begin{aligned} z^T K z &= \sum_i \sum_j z_i K_{ij} z_j \\ &= \sum_i \sum_j z_i \phi(x^{(i)})^T \phi(x^{(j)}) z_j \\ &= \sum_i \sum_j z_i \sum_k \phi_k(x^{(i)}) \phi_k(x^{(j)}) z_j \\ &= \sum_k \sum_i \sum_j z_i \phi_k(x^{(i)}) \phi_k(x^{(j)}) z_j \\ &= \sum_k \left(\sum_i z_i \phi_k(x^{(i)}) \right)^2 \\ &\geq 0. \end{aligned}$$

最后一步和前面计算$K(x,z) = (x^T z)^2$时类似。从这个公式我们可以看出，如果 K 是个有效的核函数（即$K(x,z)$和$\phi(x)^T\phi(z)$等价），那么，在训练集上得到的核函数矩阵 K 应该是半正定的（$K \geq 0$）

这样我们得到一个核函数的必要条件：

K 是有效的核函数 ==> 核函数矩阵 K 是对称半正定的。

可幸的是，这个条件也是充分的，由 Mercer 定理来表达。

Mercer 定理：

如果函数 K 是$\mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$上的映射（也就是从两个 n 维向量映射到实数域）。那么如果 K 是一个有效核函数（也称为 Mercer 核函数），那么当且仅当对于训练样例$\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$，其相应的核函数矩阵是对称半正定的。

Mercer 定理表明为了证明 K 是有效的核函数，那么我们不用去寻找ϕ，而只需要在训练集上求出各个K_{ij}，然后判断矩阵 K 是否是半正定（使用左上角主子式大于等于零等方法）即可。

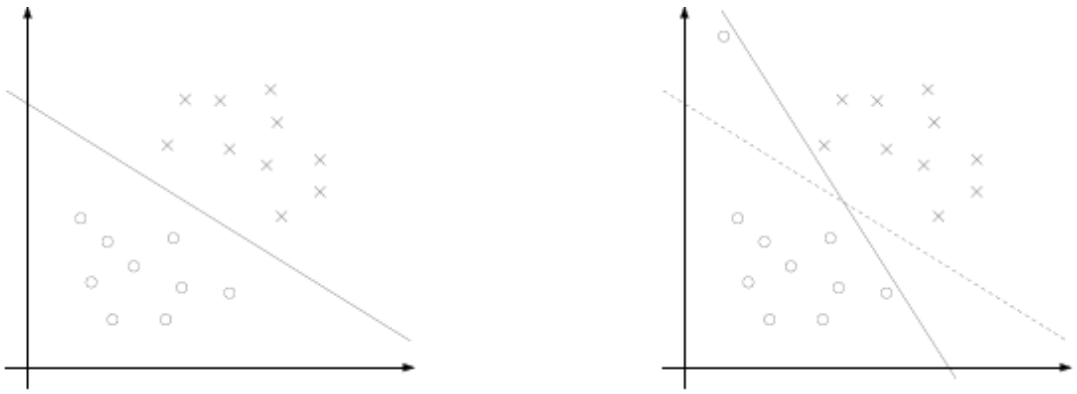
许多其他的教科书在 Mercer 定理证明过程中使用了L^2范数和再生希尔伯特空间等概念，但在特征是 n 维的情况下，这里给出的证明是等价的。

核函数不仅仅用在 SVM 上，但凡在一个模型后算法中出现了$< x, z >$，我们都可以常使用$K(x,z)$去替换，这可能能够很好地改善我们的算法。

9 规则化和不可分情况处理（Regularization and the non-separable case）

我们之前讨论的情况都是建立在样例线性可分的假设上，当样例线性不可分时，我们可以尝试使用核函数来将特征映射到高维，这样很可能就可分了。然而，映射后我们也不能100%保证可分。那怎么办呢，我们需要将模型进行调整，以保证在不可分的情况下，也能够尽可能地找出分隔超平面。

看下面两张图：



可以看到一个离群点（可能是噪声）可以造成超平面的移动，间隔缩小，可见以前的模型对噪声非常敏感。再有甚者，如果离群点在另外一个类中，那么这时候就是线性不可分了。

这时候我们应该允许一些点游离并在在模型中违背限制条件（函数间隔大于 1）。我们设计得到新的模型如下（也称软间隔）：

$$\begin{aligned} \min_{\gamma,w,b} \quad & \frac{1}{2}||w||^2 + C\sum_{i=1}^{m}\xi_i \\ \text{s.t.} \quad & y^{(i)}(w^T x^{(i)} + b) \geq 1 - \xi_i, \quad i = 1, \ldots, m \\ & \xi_i \geq 0, \quad i = 1, \ldots, m. \end{aligned}$$

引入非负参数ξ_i后（称为松弛变量），就允许某些样本点的函数间隔小于 1，即在最大间隔区间里面，或者函数间隔是负数，即样本点在对方的区域中。而放松限制条件后，我们需要重新调整目标函数，以对离群点进行处罚，目标函数后面加上的$C\sum_{i=1}^{m}\xi_i$就表示离群点越多，目标函数值越大，而我们要求的是尽可能小的目标函数值。这里的 C 是离群点的权重，C 越大表明离群点对目标函数影响越大，也就是越不希望看到离群点。我们看到，目标函数控制了离群点的数目和程度，使大部分样本点仍然遵守限制条件。

模型修改后，拉格朗日公式也要修改如下：

$$\mathcal{L}(w, b, \xi, \alpha, r) = \frac{1}{2}w^T w + C\sum_{i=1}^{m}\xi_i - \sum_{i=1}^{m}\alpha_i\left[y^{(i)}(x^T w + b) - 1 + \xi_i\right] - \sum_{i=1}^{m} r_i\xi_i.$$

这里的α_i和γ_i都是拉格朗日乘子，回想我们在拉格朗日对偶中提到的求法，先写出拉格朗日公式（如上），然后将其看作是变量 w 和 b 的函数，分别对其求偏导，得到 w 和 b 的表达式。然后代入公式中，求带入后公式的极大值。整个推导过程类似以前的模型，这里只写出最后结果如下：

$$\begin{aligned} \max_{\alpha} \quad & W(\alpha) = \sum_{i=1}^{m}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{m} y^{(i)}y^{(j)}\alpha_i\alpha_j\langle x^{(i)}, x^{(j)}\rangle \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, m \\ & \sum_{i=1}^{m}\alpha_i y^{(i)} = 0, \end{aligned}$$

此时，我们发现没有了参数ξ_i，与之前模型唯一不同在于α_i又多了$\alpha_i \leq C$的限制条件。需要提醒的是，b 的求值公式也发生了改变，改变结果在 SMO 算法里面介绍。先看看 KKT

条件的变化：

$$\alpha_i = 0 \Rightarrow y^{(i)}(w^T x^{(i)} + b) \geq 1 \quad (14)$$
$$\alpha_i = C \Rightarrow y^{(i)}(w^T x^{(i)} + b) \leq 1 \quad (15)$$
$$0 < \alpha_i < C \Rightarrow y^{(i)}(w^T x^{(i)} + b) = 1. \quad (16)$$

第一个式子表明在两条间隔线外的样本点前面的系数为 0，离群样本点前面的系数为 C，而支持向量（也就是在超平面两边的最大间隔线上）的样本点前面系数在(0,C)上。通过 KKT 条件可知，某些在最大间隔线上的样本点也不是支持向量，相反也可能是离群点。

10 坐标上升法（Coordinate ascent）

在最后讨论W(α)的求解之前，我们先看看坐标上升法的基本原理。假设要求解下面的优化问题：

$$\max_{\alpha} W(\alpha_1, \alpha_2, \ldots, \alpha_m).$$

这里 W 是α向量的函数。之前我们在回归中提到过两种求最优解的方法，一种是梯度下降法，另外一种是牛顿法。现在我们再讲一种方法称为坐标上升法（求解最小值问题时，称作坐标下降法，原理一样）。

方法过程：

Loop until convergence: {

  For $i = 1, \ldots, m$, {

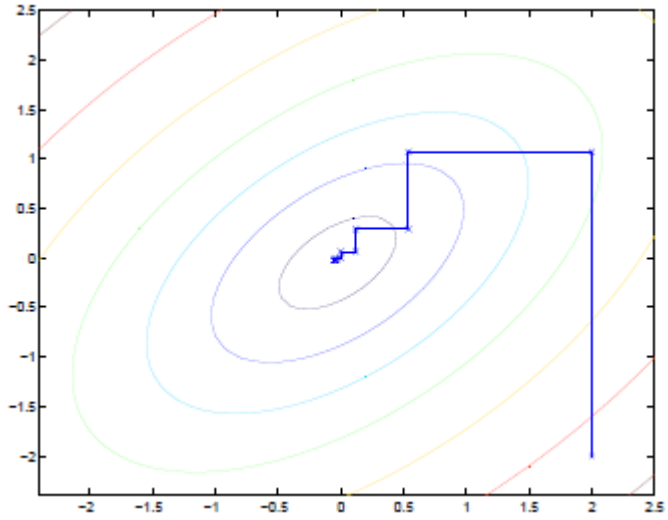
    $\alpha_i := \arg\max_{\hat{\alpha}_i} W(\alpha_1, \ldots, \alpha_{i-1}, \hat{\alpha}_i, \alpha_{i+1}, \ldots, \alpha_m)$.

  }

}

最里面语句的意思是固定除α_i之外的所有$\alpha_j (j \neq i)$，这时 W 可看作只是关于α_i的函数，那么直接对α_i求导优化即可。这里我们进行最大化求导的顺序 i 是从 1 到 m，可以通过更改优化顺序来使 W 能够更快地增加并收敛。如果 W 在内循环中能够很快地达到最优，那么坐标上升法会是一个很高效的求极值方法。

下面通过一张图来展示：



椭圆代表了二次函数的各个等高线，变量数为 2，起始坐标是(2,-2)。图中的直线式迭代优化的路径，可以看到每一步都会向最优值前进一步，而且前进路线是平行于坐标轴的，因为每一步只优化一个变量。

11 SMO 优化算法（Sequential minimal optimization）

SMO 算法由 Microsoft Research 的 John C. Platt 在 1998 年提出，并成为最快的二次规划优化算法，特别针对线性 SVM 和数据稀疏时性能更优。关于 SMO 最好的资料就是他本人写的《Sequential Minimal Optimization A Fast Algorithm for Training Support Vector Machines》了。

我拜读了一下，下面先说讲义上对此方法的总结。

首先回到我们前面一直悬而未解的问题，对偶函数最后的优化问题：

$$\begin{aligned}\max_\alpha \quad & W(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2}\sum_{i,j=1}^{m} y^{(i)} y^{(j)} \alpha_i \alpha_j \langle x^{(i)}, x^{(j)} \rangle \\ \text{s.t.} \quad & 0 \le \alpha_i \le C, \quad i = 1, \ldots, m \\ & \sum_{i=1}^{m} \alpha_i y^{(i)} = 0,\end{aligned}$$

要解决的是在参数$\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$上求最大值 W 的问题，至于$x^{(i)}$和$y^{(i)}$都是已知数。C 由我们预先设定，也是已知数。

按照坐标上升的思路，我们首先固定除α_1以外的所有参数，然后在α_1上求极值。等一下，这个思路有问题，因为如果固定α_1以外的所有参数，那么α_1将不再是变量（可以由其他值推出），因为问题中规定了

$$\alpha_1 y^{(1)} = -\sum_{i=2}^{m} \alpha_i y^{(i)}.$$

因此，我们需要一次选取两个参数做优化，比如α_1和α_2，此时α_2可以由α_1和其他参数表示出来。这样回带到 W 中，W 就只是关于α_1的函数了，可解。

这样，SMO 的主要步骤如下：

Repeat till convergence {

1. Select some pair α_i and α_j to update next (using a heuristic that tries to pick the two that will allow us to make the biggest progress towards the global maximum).
2. Reoptimize $W(\alpha)$ with respect to α_i and α_j, while holding all the other α_k's $(k \neq i, j)$ fixed.

}

意思是，第一步选取一对α_i和α_j，选取方法使用启发式方法（后面讲）。第二步，固定除α_i和α_j之外的其他参数，确定 W 极值条件下的α_i，α_j由α_i表示。

SMO 之所以高效就是因为在固定其他参数后，对一个参数优化过程很高效。

下面讨论具体方法：

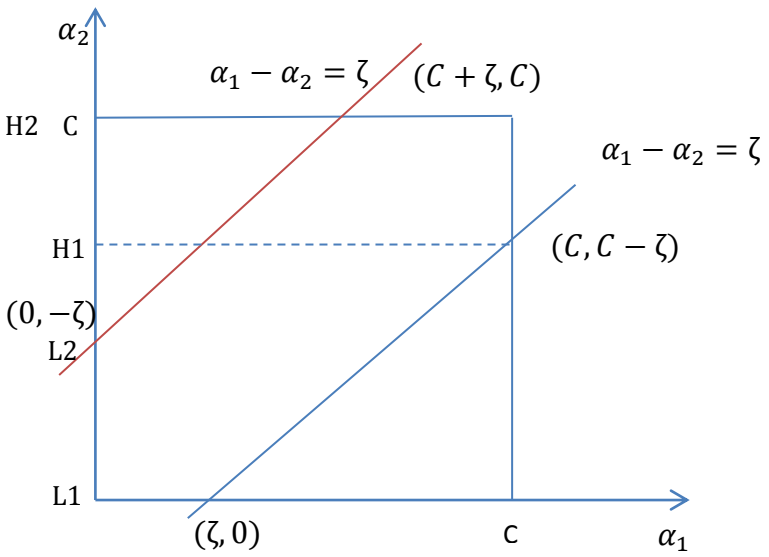
假设我们选取了初始值$\{\alpha_1, \alpha_2, \ldots, \alpha_n\}$满足了问题中的约束条件。接下来，我们固定$\{\alpha_3, \alpha_4, \ldots, \alpha_n\}$，这样 W 就是$\alpha_1$和$\alpha_2$的函数。并且$\alpha_1$和$\alpha_2$满足条件：

$$\alpha_1 y^{(1)} + \alpha_2 y^{(2)} = -\sum_{i=3}^{m} \alpha_i y^{(i)}.$$

由于$\{\alpha_3, \alpha_4, \ldots, \alpha_n\}$都是已知固定值，因此为了方面，可将等式右边标记成实数值ζ。

$$\alpha_1 y^{(1)} + \alpha_2 y^{(2)} = \zeta.$$

当$y^{(1)}$和$y^{(2)}$异号时，也就是一个为 1，一个为-1 时，他们可以表示成一条直线，斜率为 1。如下图：



横轴是α_1，纵轴是α_2，α_1和α_2既要在矩形方框内，也要在直线上，因此

$$L = max(0, \alpha_2 - \alpha_1),\ H = min(C, C + \alpha_2 - \alpha_1)$$

同理，当$y^{(1)}$和$y^{(2)}$同号时，

$$L = max(0, \alpha_2 + \alpha_1 - C),\ H = min(C, \alpha_2 + \alpha_1)$$

然后我们打算将α_1用α_2表示：

$$\alpha_1 = (\zeta - \alpha_2 y^{(2)}) y^{(1)}.$$

然后反代入 W 中，得

$$W(\alpha_1, \alpha_2, \ldots, \alpha_m) = W((\zeta - \alpha_2 y^{(2)}) y^{(1)}, \alpha_2, \ldots, \alpha_m).$$

展开后 W 可以表示成a${\alpha_2}^2 + b\alpha_2 + c$。其中 a,b,c 是固定值。这样，通过对 W 进行求导可以得到α_2，然而要保证α_2满足$L \le \alpha_2 \le H$，我们使用${\alpha_2}^{new,unclipped}$表示求导求出来的$\alpha_2$，然而最后的$\alpha_2$，要根据下面情况得到：

$$\alpha_2^{new} = \begin{cases} H & \text{if } \alpha_2^{new,unclipped} > H \\ \alpha_2^{new,unclipped} & \text{if } L \le \alpha_2^{new,unclipped} \le H \\ L & \text{if } \alpha_2^{new,unclipped} < L \end{cases}$$

这样得到${\alpha_2}^{new}$后，我们可以得到α_1的新值${\alpha_1}^{new}$。

下面进入 Platt 的文章，来找到启发式搜索的方法和求 b 值的公式。

这篇文章使用的符号表示有点不太一样，不过实质是一样的，先来熟悉一下文章中符号的表示。

文章中定义特征到结果的输出函数为

$$u = \vec{w} \cdot \vec{x} - b, \qquad (1)$$

与我们之前的$w^T x^{(i)} + b$实质是一致的。

原始的优化问题为：

$$\min_{w,b} \tfrac{1}{2} \|\vec{w}\|^2 \text{ subject to } y_i(\vec{w} \cdot \vec{x}_i - b) \ge 1, \forall i, \qquad (3)$$

求导得到：

$$\vec{w} = \sum_{i=1}^{N} y_i \alpha_i \vec{x}_i, \quad b = \vec{w} \cdot \vec{x}_k - y_k \text{ for some } \alpha_k > 0. \qquad (7)$$

经过对偶后为：

$$\min_{\vec{\alpha}} \Psi(\vec{\alpha}) = \min_{\vec{\alpha}} \tfrac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} y_i y_j (\vec{x}_i \cdot \vec{x}_j) \alpha_i \alpha_j - \sum_{i=1}^{N} \alpha_i,$$

s.t. $\alpha_i \ge 0, \forall i,$

$$\sum_{i=1}^{N} y_i \alpha_i = 0.$$

这里与 W 函数是一样的，只是符号求反后，变成求最小值了。y_i和$y^{(i)}$是一样的，都表示第 i 个样本的输出结果（1 或-1）。

经过加入松弛变量ξ_i后，模型修改为：

$$\min_{\vec{w},b,\vec{\xi}} \frac{1}{2}\|\vec{w}\|^2 + C\sum_{i=1}^{N}\xi_i \quad \text{subject to } y_i(\vec{w}\cdot\vec{x}_i - b) \geq 1-\xi_i, \forall i, \tag{8}$$

$$0 \leq \alpha_i \leq C, \forall i. \tag{9}$$

由公式（7）代入（1）中可知，

$$u = \sum_{j=1}^{N} y_j \alpha_j K(\vec{x}_j, \vec{x}) - b, \tag{10}$$

这个过程和之前对偶过程一样。

重新整理我们要求的问题为：

$$\begin{aligned} \min_{\vec{\alpha}} \Psi(\vec{\alpha}) = \min_{\vec{\alpha}} \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} y_i y_j K(\vec{x}_i, \vec{x}_j)\alpha_i\alpha_j - \sum_{i=1}^{N}\alpha_i, \\ 0 \leq \alpha_i \leq C, \forall i, \\ \sum_{i=1}^{N} y_i \alpha_i = 0. \end{aligned} \tag{11}$$

与之对应的 KKT 条件为：

$$\begin{aligned} \alpha_i = 0 &\Leftrightarrow y_i u_i \geq 1, \\ 0 < \alpha_i < C &\Leftrightarrow y_i u_i = 1, \\ \alpha_i = C &\Leftrightarrow y_i u_i \leq 1. \end{aligned} \tag{12}$$

这个 KKT 条件说明，在两条间隔线外面的点，对应前面的系数α_i为 0，在两条间隔线里面的对应α_i为 C，在两条间隔线上的对应的系数α_i在 0 和 C 之间。

将我们之前得到 L 和 H 重新拿过来：

$$L = \max(0, \alpha_2 - \alpha_1), \qquad H = \min(C, C + \alpha_2 - \alpha_1). \tag{13}$$

$$L = \max(0, \alpha_2 + \alpha_1 - C), \qquad H = \min(C, \alpha_2 + \alpha_1). \tag{14}$$

之前我们将问题进行到这里，然后说将α_1用α_2表示后代入 W 中，这里将代入Ψ中，得

$$\Psi = \frac{1}{2}K_{11}\alpha_1^2 + \frac{1}{2}K_{22}\alpha_2^2 + sK_{12}\alpha_1\alpha_2 + y_1\alpha_1 v_1 + y_2\alpha_2 v_2 - \alpha_1 - \alpha_2 + \Psi_{\text{constant}}, \tag{24}$$

其中

$$\begin{aligned} K_{ij} &= K(\vec{x}_i, \vec{x}_j), \\ v_i &= \sum_{j=3}^{N} y_j \alpha_j^* K_{ij} = u_i + b^* - y_1\alpha_1^* K_{1i} - y_2\alpha_2^* K_{2i}, \end{aligned} \tag{25}$$

这里的α_1^*和α_2^*代表某次迭代前的原始值，因此是常数，而α_1和α_2是变量，待求。公式（24）中的最后一项是常数。

由于α_1和α_2满足以下公式

$$y_1\alpha_1{}^* + y_2\alpha_2{}^* = -\sum_{i=3}^{n} y_i\alpha_i{}^* = y_1\alpha_1 + y_2\alpha_2$$

因为$\alpha_i{}^*(i > 2)$的值是固定值，在迭代前后不会变。

那么用 s 表示y_1y_2，上式两边乘以y_1时，变为：

$$\alpha_1 + s\alpha_2 = \alpha_1^* + s\alpha_2^* = w. \tag{26}$$

其中

$$w = -y_1\sum_{i=3}^{n} y_i\alpha_i{}^*$$

代入（24）中，得

$$\begin{aligned}\Psi = {} & \tfrac{1}{2}K_{11}(w - s\alpha_2)^2 + \tfrac{1}{2}K_{22}\alpha_2^2 + sK_{12}(w - s\alpha_2)\alpha_2 \\ & + y_1(w - s\alpha_2)v_1 - w + s\alpha_2 + y_2\alpha_2 v_2 - \alpha_2 + \Psi_{\text{constant}}.\end{aligned} \tag{27}$$

这时候只有α_2是变量了，求导

$$\frac{d\Psi}{d\alpha_2} = -sK_{11}(w - s\alpha_2) + K_{22}\alpha_2 - K_{12}\alpha_2 + sK_{12}(w - s\alpha_2) - y_2v_1 + s + y_2v_2 - 1 = 0. \tag{28}$$

如果Ψ的二阶导数大于 0（凹函数），那么一阶导数为 0 时，就是极小值了。

假设其二阶导数为 0（一般成立），那么上式化简为：

$$\alpha_2(K_{11} + K_{22} - 2K_{12}) = s(K_{11} - K_{12})w + y_2(v_1 - v_2) + 1 - s. \tag{29}$$

将 w 和 v 代入后，继续化简推导，得（推导了六七行推出来了）

$$\alpha_2(K_{11} + K_{22} - 2K_{12}) = \alpha_2^*(K_{11} + K_{22} - 2K_{12}) + y_2(u_1 - u_2 + y_2 - y_1). \tag{30}$$

我们使用η来表示：

$$\eta = K(\vec{x}_1, \vec{x}_1) + K(\vec{x}_2, \vec{x}_2) - 2K(\vec{x}_1, \vec{x}_2). \tag{15}$$

通常情况下目标函数是正定的，也就是说，能够在直线约束方向上求得最小值，并且$\eta > 0$。

那么我们在（30）两边都除以η可以得到

$$\alpha_2^{\text{new}} = \alpha_2 + \frac{y_2(E_1 - E_2)}{\eta}, \tag{16}$$

这里我们使用$\alpha_2{}^{new}$表示优化后的值，α_2是迭代前的值，$E_i = u_i - y_i$。

与之前提到的一样$\alpha_2{}^{new}$不是最终迭代后的值，需要进行约束：

$$\alpha_2^{\text{new,clipped}} = \begin{cases} H & \text{if} \quad \alpha_2^{\text{new}} \geq H; \\ \alpha_2^{\text{new}} & \text{if} \quad L < \alpha_2^{\text{new}} < H; \\ \text{L} & \text{if} \quad \alpha_2^{\text{new}} \leq L. \end{cases} \tag{17}$$

那么

$$\alpha_1^{\text{new}} = \alpha_1 + s(\alpha_2 - \alpha_2^{\text{new,clipped}}). \tag{18}$$

在特殊情况下，η可能不为正，如果核函数 K 不满足 Mercer 定理，那么目标函数可能变得非正定，η可能出现负值。即使 K 是有效的核函数，如果训练样本中出现相同的特征 x，那么η仍有可能为 0。SMO 算法在η不为正值的情况下仍有效。为保证有效性，我们可以推导出η就是Ψ的二阶导数，$\eta < 0$，Ψ没有极小值，最小值在边缘处取到（类比$y = -x^2$），$\eta = 0$时更是单调函数了，最小值也在边缘处取得，而α_2的边缘就是 L 和 H。这样将$\alpha_2 = L$和$\alpha_2 = H$分别代入Ψ中即可求得Ψ的最小值，相应的$\alpha_2 =$ L还是$\alpha_2 = H$也可以知道了。具体计算公式如下：

$$\begin{aligned}
f_1 &= y_1(E_1 + b) - \alpha_1 K(\vec{x}_1, \vec{x}_1) - s\alpha_2 K(\vec{x}_1, \vec{x}_2), \\
f_2 &= y_2(E_2 + b) - s\alpha_1 K(\vec{x}_1, \vec{x}_2) - \alpha_2 K(\vec{x}_2, \vec{x}_2), \\
L_1 &= \alpha_1 + s(\alpha_2 - L), \\
H_1 &= \alpha_1 + s(\alpha_2 - H), \\
\Psi_L &= L_1 f_1 + L f_2 + \tfrac{1}{2} L_1^2 K(\vec{x}_1, \vec{x}_1) + \tfrac{1}{2} L^2 K(\vec{x}_2, \vec{x}_2) + sLL_1 K(\vec{x}_1, \vec{x}_2), \\
\Psi_H &= H_1 f_1 + H f_2 + \tfrac{1}{2} H_1^2 K(\vec{x}_1, \vec{x}_1) + \tfrac{1}{2} H^2 K(\vec{x}_2, \vec{x}_2) + sHH_1 K(\vec{x}_1, \vec{x}_2).
\end{aligned} \tag{19}$$

至此，迭代关系式除了 b 的推导式以外，都已经推出。

b 每一步都要更新，因为前面的 KKT 条件指出了α_i和$y_i u_i$的关系，而u_i和 b 有关，在每一步计算出α_i后，根据 KKT 条件来调整 b。

b 的更新有几种情况：

b的更新：选择b使得关于乘子α_1或α_1的KKT条件成立

$$b_1 = E_1 + y_1(\alpha_1^{new} - \alpha_1)k(x_1, x_1) + y_2(\alpha_2^{new,clipped} - \alpha_2)k(x_1, x_2) + b \quad (7)$$

$$b_2 = E_2 + y_1(\alpha_1^{new} - \alpha_1)k(x_1, x_2) + y_2(\alpha_2^{new,clipped} - \alpha_2)k(x_2, x_2) + b \quad (8)$$

如果α_1^{new}在界内，则$b^{new} = b_1$；如果$\alpha_2^{new,clipped}$在界内，$b^{new} = b_2$；

如果α_1^{new}和$\alpha_2^{new,clipped}$都在界内，那么$b_1 = b_2$，则$b^{new} = b_1 = b_2$；

如果α_1^{new}和$\alpha_2^{new,clipped}$都在界上，那么b_1和b_2之间的任何数都满足KKT条件，都可作为b的更新值，一般取$b^{new} = (b_1 + b_2)/2$.

来自罗林开的 ppt

这里的界内指$0 < \alpha_i < C$，界上就是等于 0 或者 C 了。

前面两个的公式推导可以根据

$$y_1\alpha_1^* + y_2\alpha_2^* = -\sum_{i=3}^{n} y_i\alpha_i^* = y_1\alpha_1 + y_2\alpha_2$$

和对于$0 < \alpha_i < C$有$y_i u_i = 1$的 KKT 条件推出。

这样全部参数的更新公式都已经介绍完毕，附加一点，如果使用的是线性核函数，我们就可以继续使用 w 了，这样不用扫描整个样本库来作内积了。

w 值的更新方法为：

$$\vec{w}^{\text{new}} = \vec{w} + y_1(\alpha_1^{\text{new}} - \alpha_1)\vec{x}_1 + y_2(\alpha_2^{\text{new,clipped}} - \alpha_2)\vec{x}_2. \quad (22)$$

根据前面的

$$\vec{w} = \sum_{i=1}^{N} y_i \alpha_i \vec{x}_i, \quad b = \vec{w} \cdot \vec{x}_k - y_k \text{ for some } \alpha_k > 0. \quad (7)$$

公式推导出。

12 SMO 中拉格朗日乘子的启发式选择方法

终于到了最后一个问题了，所谓的启发式选择方法主要思想是每次选择拉格朗日乘子的时候，优先选择样本前面系数$0 < \alpha_i < C$的α_i作优化（论文中称为无界样例），因为在界上（α_i为 0 或 C）的样例对应的系数α_i一般不会更改。

这条启发式搜索方法是选择第一个拉格朗日乘子用的，比如前面的α_2。那么这样选择的话，是否最后会收敛？可幸的是 Osuna 定理告诉我们只要选择出来的两个α_i中有一个违背了 KKT 条件，那么目标函数在一步迭代后值会减小。违背 KKT 条件不代表$0 < \alpha_i < C$，在界上也有可能会违背。是的，因此在给定初始值α_i=0 后，先对所有样例进行循环，循环中碰到违背 KKT 条件的（不管界上还是界内）都进行迭代更新。等这轮过后，如果没有收敛，第二轮就只针对$0 < \alpha_i < C$的样例进行迭代更新。

在第一个乘子选择后，第二个乘子也使用启发式方法选择，第二个乘子的迭代步长大致正比于$|\mathrm{E}_1 - \mathrm{E}_2|$，选择第二个乘子能够最大化$|\mathrm{E}_1 - \mathrm{E}_2|$。即当$\mathrm{E}_1$为正时选择负的绝对值最大的$\mathrm{E}_2$，反之，选择正值最大的$\mathrm{E}_2$。

最后的收敛条件是在界内（$0 < \alpha_i < C$）的样例都能够遵循 KKT 条件，且其对应的α_i只在极小的范围内变动。

至于如何写具体的程序，请参考 John C. Platt 在论文中给出的伪代码。

13 总结

这份 SVM 的讲义重点概括了 SVM 的基本概念和基本推导，中规中矩却又让人醍醐灌顶。起初让我最头疼的是拉格朗日对偶和 SMO，后来逐渐明白拉格朗日对偶的重要作用是将 w 的计算提前并消除 w，使得优化函数变为拉格朗日乘子的单一参数优化问题。而 SMO 里面迭代公式的推导也着实让我花费了不少时间。

对比这么复杂的推导过程，SVM 的思想确实那么简单。它不再像 logistic 回归一样企图去拟合样本点（中间加了一层 sigmoid 函数变换），而是就在样本中去找分隔线，为了评判哪条分界线更好，引入了几何间隔最大化的目标。

之后所有的推导都是去解决目标函数的最优化上了。在解决最优化的过程中，发现了 w 可以由特征向量内积来表示，进而发现了核函数，仅需要调整核函数就可以将特征进行低维到高维的变换，在低维上进行计算，实质结果表现在高维上。由于并不是所有的样本都可分，为了保证 SVM 的通用性，进行了软间隔的处理，导致的结果就是将优化问题变得更加复杂，然而惊奇的是松弛变量没有出现在最后的目标函数中。最后的优化求解问题，也被拉格朗日对偶和 SMO 算法化解，使 SVM 趋向于完美。

另外，其他很多议题如 SVM 背后的学习理论、参数选择问题、二值分类到多值分类等等还没有涉及到，以后有时间再学吧。其实朴素贝叶斯在分类二值分类问题时，如果使用对数比，那么也算作线性分类器。

规则化和模型选择（Regularization and model selection）

JerryLead
csxulijie@gmail.com
2011 年 3 月 24 日星期四

1 问题

模型选择问题：对于一个学习问题，可以有多种模型选择。比如要拟合一组样本点，可以使用线性回归($y = \theta^T x$)，也可以用多项式回归($y = \theta^T x^{1\sim m}$)。那么使用哪种模型好呢（能够在偏差和方差之间达到平衡最优）？

还有一类参数选择问题：如果我们想使用带权值的回归模型，那么怎么选择权重 w 公式里的参数τ？

形式化定义：假设可选的模型集合是$M = \{M_1, M_2, \dots, M_d\}$，比如我们想分类，那么 SVM、logistic 回归、神经网络等模型都包含在 M 中。

1 交叉验证（Cross validation）

我们的第一个任务就是要从 M 中选择最好的模型。
假设训练集使用 S 来表示
如果我们想使用经验风险最小化来度量模型的好坏，那么我们可以这样来选择模型：

1、 使用 S 来训练每一个M_i，训练出参数后，也就可以得到假设函数h_i。（比如，线性模型中得到θ_i后，也就得到了假设函数$h_\theta(x) = \theta^T x$）
2、 选择错误率最小的假设函数。

遗憾的是这个算法不可行，比如我们需要拟合一些样本点，使用高阶的多项式回归肯定比线性回归错误率要小，偏差小，但是方差却很大，会过度拟合。因此，我们改进算法如下：

1、 从全部的训练数据 S 中随机选择 70%的样例作为训练集S_{train}，剩余的 30%作为测试集S_{CV}。
2、 在S_{train}上训练每一个M_i，得到假设函数h_i。
3、 在S_{CV}上测试每一个h_i，得到相应的经验错误$\hat{\varepsilon}_{S_{\text{CV}}}(h_i)$。
4、 选择具有最小经验错误$\hat{\varepsilon}_{S_{\text{CV}}}(h_i)$的$h_i$作为最佳模型。

这种方法称为 hold-out cross validation 或者称为简单交叉验证。

由于测试集是和训练集中是两个世界的，因此我们可以认为这里的经验错误$\hat{\varepsilon}_{S_{\text{CV}}}(h_i)$接近于泛化错误（generalization error）。这里测试集的比例一般占全部数据的 1/4-1/3。30%是典型值。

还可以对模型作改进，当选出最佳的模型M_i后，再在全部数据 S 上做一次训练，显然训练数据越多，模型参数越准确。

简单交叉验证方法的弱点在于得到的最佳模型是在 70%的训练数据上选出来的，不代表在全部训练数据上是最佳的。还有当训练数据本来就很少时，再分出测试集后，训练数据就太少了。

我们对简单交叉验证方法再做一次改进，如下：

1、 将全部训练集 S 分成 k 个不相交的子集，假设 S 中的训练样例个数为 m，那么每一个子集有 m/k 个训练样例，相应的子集称作$\{S_1, S_2, \dots, S_k\}$。
2、 每次从模型集合 M 中拿出来一个M_i，然后在训练子集中选择出 k-1 个$\{S_1, S_2, S_{j-1}, S_{j+1} \dots, S_k\}$（也就是每次只留下一个$S_j$），使用这 k-1 个子集训练$M_i$后，得到假设函数$h_{ij}$。最后使用剩下的一份$S_j$作测试，得到经验错误$\hat{\varepsilon}_{S_j}(h_{ij})$。
3、 由于我们每次留下一个S_j（j 从 1 到 k），因此会得到 k 个经验错误，那么对于一个M_i，它的经验错误是这 k 个经验错误的平均。
4、 选出平均经验错误率最小的M_i，然后使用全部的 S 再做一次训练，得到最后的h_i。

这个方法称为 k-fold cross validation（k-折叠交叉验证）。说白了，这个方法就是将简单交叉验证的测试集改为 1/k，每个模型训练 k 次，测试 k 次，错误率为 k 次的平均。一般讲 k 取值为 10。这样数据稀疏时基本上也能进行。显然，缺点就是训练和测试次数过多。

极端情况下，k 可以取值为 m，意味着每次留一个样例做测试，这个称为 leave-one-out cross validation。

如果我们发明了一种新的学习模型或者算法，那么可以使用交叉验证来对模型进行评价。比如在 NLP 中，我们将训练集中分出一部分训练，一部分做测试。

2 特征选择（Feature selection）

特征选择严格来说也是模型选择中的一种。这里不去辨析他们的关系，重点说明问题。假设我们想对维度为 n 的样本点进行回归，然而，n 可能大多以至于远远大于训练样例数 m。但是我们感觉很多特征对于结果是无用的，想剔除 n 中的无用特征。n 个特征就有2^n种去除情况（每个特征去或者保留），如果我们枚举这些情况，然后利用交叉验证逐一考察在该情况下模型的错误率，太不现实。因此需要一些启发式搜索方法。

第一种，前向搜索：

1、 初始化特征集 F 为空。
2、 扫描 i 从 1 到 n，
如果第 i 个特征不在 F 中，那么将特征 i 和 F 放在一起作为F_i（即$F_i = F \cup \{i\}$）
在只使用F_i中特征的情况下，利用交叉验证来得到F_i的错误率。
3、 从上步中得到的 n 个F_i中选出错误率最小的F_i，更新 F 为F_i。
如果 F 中的特征数达到了 n 或者预设定的阈值（如果有的话），那么输出整个搜索过程中最好的 F，没达到转到 2

前向搜索属于 wrapper model feature selection。Wrapper 这里指不断地使用不同的特征集来测试学习算法。前向搜索说白了就是每次增量地从剩余未选中的特征选出一个加入特征集中，待达到阈值或者 n 时，从所有的 F 中选出错误率最小的。

既然有增量加，那么也会有增量减，后者称为后向搜索。先将 F 设置为{1,2,..,n}，然后每次删除一个特征，并评价，直到达到阈值或者为空，然后选择最佳的 F。

这两种算法都可以工作，但是计算复杂度比较大。时间复杂度为$O(n+(n-1)+(n-2)+\cdots+1)=O(n^2)$。

第二种，过滤特征选择（Filter feature selection）：

过滤特征选择方法的想法是针对每一个特征x_i，i 从 1 到 n，计算x_i相对于类别标签y的信息量S(i)，得到 n 个结果，然后将 n 个S(i)按照从大到小排名，输出前 k 个特征。显然，这样复杂度大大降低，为 O(n)。

那么关键问题就是使用什么样的方法来度量S(i)，我们的目标是选取与 y 关联最密切的一些x_i。而 y 和x_i都是有概率分布的。因此我们想到使用互信息来度量S(i)，对于x_i是离散值的情况更适用，不是离散值，将其转变为离散值，方法在第一篇《回归认识》中已经提到。

互信息（Mutual information）公式：

$$\mathrm{MI}(x_i,y)=\sum_{x_i\in\{0,1\}}\sum_{y\in\{0,1\}}p(x_i,y)\log\frac{p(x_i,y)}{p(x_i)p(y)}.$$

当x_i是 0/1 离散值的时候，这个公式如上。很容易推广到x_i是多个离散值的情况。

这里的$p(x_i,y)$，$p(x_i)$和$p(y)$都是从训练集上得到的。

若问这个 MI 公式如何得来，请看它的 KL 距离（Kullback-Leibler）表述：

$$\mathrm{MI}(x_i,y)=\mathrm{KL}\left(p(x_i,y)||p(x_i)p(y)\right)$$

也就是说，MI 衡量的是x_i和 y 的独立性。如果它俩独立（$p(x_i,y)=p(x_i)p(y)$），那么 KL 距离值为 0，也就是说x_i和 y 不相关了，可以去除x_i。相反，如果两者密切相关，那么 MI 值会很大。在对 MI 进行排名后，最后剩余的问题就是如何选择 k 值（前 k 个x_i）。我们继续使用交叉验证的方法，将 k 从 1 扫描到 n，取最大的 F。不过这次复杂度是线性的了。比如，在使用朴素贝叶斯分类文本的时候，词表长度 n 很大。使用 filter 特征选择方法，能够增加分类器的精度。

3 贝叶斯统计和规则化（Bayesian statistics and regularization）

题目有点绕，说白了就是要找更好的估计方法来减少过度拟合情况的发生。

回顾一下，线性回归中使用的估计方法是最小二乘法，logistic 回归是条件概率的最大似然估计，朴素贝叶斯是联合概率的最大似然估计，SVM 是二次规划。

以前我们使用的估计方法是最大似然估计（比如在 logistic 回归中使用的）：

$$\theta_{\mathrm{ML}}=\arg\max_{\theta}\prod_{i=1}^{m}p(y^{(i)}|x^{(i)};\theta).$$

注意这里的最大似然估计与维基百科中的表述

http://zh.wikipedia.org/wiki/%E6%9C%80%E5%A4%A7%E5%90%8E%E9%AA%8C%E6%A6%82%E7%8E%87

有些出入，是因为维基百科只是将样本（观察数据）记为 X，然后求 P(X)的最大概率。然而，对于我们这里的样本而言，分为特征 x 和类标签 y。我们需要具体计算 P(X)。在判别模型（如 logistic 回归）中，我们看待 P(X)=P(x,y)=P(y|x)P(x)，而 P(x)与θ独立无关，

因此最后的 argmax P(X)由 argmaxP(y|x)决定，也就是上式θ_{ML}。严格来讲θ_{ML}并不等于样本 X 的概率，只是 P(X)决定于θ_{ML}，θ_{ML}最大化时 P(X)也最大化。在生成模型，如朴素贝叶斯中，我们看待 P(X)=P(y)P(x|y)，也就是在某个类标签 y 下出现特征 x 的概率与先验概率之积。而 P(x|y)在 x 各个分量是条件独立情况下可以以概率相乘方式计算出，这里根本没有参数θ。因此最大似然估计直接估计 P(x,y)即可，变成了联合分布概率。

在该上式中，我们视参数θ为未知的常数向量。我们的任务就是估计出未知的θ。

从大范围上说，最大似然估计看待θ的视角称为频率学派（frequentist statistics），认为θ不是随机变量，只是一个未知的常量，因此我们没有把$p(y^{(i)}|x^{(i)};\theta)$写成$p(y^{(i)}|x^{(i)},\theta)$。

另一种视角称为贝叶斯学派（Bayesian），他们看待θ为随机变量，值未知。既然θ为随机变量，那么θ不同的值就有了不同的概率p(θ)（称为先验概率），代表我们对特定的θ的相信度。我们将训练集表示成$S=\{(x^{(i)},y^{(i)})\}$，i 从 1 到 m。我们首先需要求出θ的后验概率：

$$\begin{aligned} p(\theta|S) &= \frac{p(S|\theta)p(\theta)}{p(S)} \\ &= \frac{\left(\prod_{i=1}^{m} p(y^{(i)}|x^{(i)},\theta)\right)p(\theta)}{\int_\theta \left(\prod_{i=1}^{m} p(y^{(i)}|x^{(i)},\theta)p(\theta)\right)d\theta} \end{aligned} \tag{1}$$

这个公式的推导其实比较蹊跷。第一步无可厚非，第二步中先看分子，分子中p(S|θ)最完整的表达方式是$(\prod_{i=1}^{m} p(y^{(i)}|x^{(i)},\theta))p(x^{(i)})$。由于在分母中也会出现$p(x^{(i)})$，所以$p(x^{(i)})$会被约掉。当然作者压根就没有考虑$p(x^{(i)})$，因为他看待 P(S)的观点就是 x->y，而不是(x,y)。再来看分母，分母写成这种形式后，意思是对所有的θ可能值做积分。括号里面的意思是$\prod_{i=1}^{m} p(y^{(i)}|x^{(i)})$，然后将其展开成分母的模样，从宏观上理解，就是在求每个样例的概率时，先以一定的概率确定θ，然后在$x^{(i)}$和θ的作用下再确定$y^{(i)}$的概率。而如果让我推导这个公式，我可能会这样写分母$p(S)=\int_\theta (p(S|\theta)p(\theta))d\theta$，这样推导出的结果是$p(S)=\int_\theta (\prod_{i=1}^{m} p(y^{(i)}|x^{(i)},\theta))p(\theta)d\theta$。我不知道自己的想法对不对，分歧在于如何看待$\theta$，作者是为每个样例都重新选定$\theta$，而我是对总体样本选择一个$\theta$。

后记：我看了 Andrew NG 的教学视频，发现视频上的结果和讲义上的不一致，应该讲义上由于笔误写错了，正确的分母是$p(S)=\int_\theta (\prod_{i=1}^{m} p(y^{(i)}|x^{(i)},\theta))p(\theta)d\theta$

$p(y^{(i)}|x^{(i)},\theta)$在不同的模型下计算方式不同。比如在贝叶斯 logistic 回归中，

$$p(y^{(i)}|x^{(i)},\theta)=h_\theta(x^{(i)})^{y^{(i)}}(1-h_\theta(x^{(i)}))^{(1-y^{(i)})},$$

其中$h_\theta(x^{(i)})=1/(1+\exp(-\theta^T x^{(i)}))$，p 的表现形式也就是伯努利分布了。

在θ是随机变量的情况下，如果新来一个样例特征为 x，那么为了预测 y。我们可以使用下面的公式：

$$p(y|x,S)=\int_\theta p(y|x,\theta)p(\theta|S)d\theta$$

p(θ|S)由前面的公式得到。假若我们要求期望值的话，那么套用求期望的公式即可：

$$\mathrm{E}[y|x,S]=\int_y yp(y|x,S)dy$$

大多数时候我们只需求得p(y|x, S)中最大的 y 即可（在 y 是离散值的情况下）。

这次求解p(y|x, S)与之前的方式不同，以前是先求θ，然后直接预测，这次是对所有可能的θ作积分。

再总结一下两者的区别，最大似然估计没有将θ视作 y 的估计参数，认为θ是一个常数，只是未知其值而已，比如我们经常使用常数 c 作为 y=2x+c 的后缀一样。但是$p(y^{(i)}|x^{(i)};\theta)$的计算公式中含有未知数θ。所以再对极大似然估计求导后，可以求出θ。

而贝叶斯估计将θ视为随机变量，θ的值满足一定的分布，不是固定值，我们无法通过计算获得其值，只能在预测时计算积分。

然而在上述贝叶斯估计方法中，虽然公式合理优美，但后验概率p(θ|S)很难计算，看其公式知道计算分母时需要在所有的θ上作积分，然而对于一个高维的θ来说，枚举其所有的可能性太难了。

为了解决这个问题，我们需要改变思路。看p(θ|S)公式中的分母，分母其实就是 P(S)，而我们就是要让 P(S)在各种参数的影响下能够最大（这里只有参数θ）。因此我们只需求出随机变量θ中最可能的取值，这样求出θ后，可将θ视为固定值，那么预测时就不用积分了，而是直接像最大似然估计中求出θ后一样进行预测，这样就变成了点估计。这种方法称为最大后验概率估计（Maximum a posteriori）方法

Θ估计公式为

$$\theta_{\mathrm{MAP}} = \arg\max_{\theta} \prod_{i=1}^{m} p(y^{(i)}|x^{(i)},\theta)p(\theta).$$

θ_{MAP}与θ_{ML}一样表示的是P(S)，意义是在从随机变量分布中以一定概率p(θ)选定好θ后，在给定样本特征$x^{(i)}$上$y^{(i)}$出现的概率积。

但是如果让我推导这个公式的时候，我会这么做，考虑后验概率p(θ|S)，我们的目标是求出最有可能的θ。而对于θ的所有值来说，分母是一样的，只有分子是不同的。因此argmax $p(\theta|S) = arg\max_{\theta}(\prod_{i=1}^{m} p(y^{(i)}|x^{(i)},\theta))p(x^{(i)})$。也就是$\theta_{\mathrm{MAP}}$的推导式。但这个公式与上面的有些不同，同样还是看待每个样本一个θ，还是总体样本一个θ的问题。

与最大似然估计对比发现，MAP 只是将θ移进了条件概率中，并且多了一项p(θ)。一般情况下我们认为$\theta \sim N(0, \tau^2 I)$，实际上，贝叶斯最大后验概率估计相对于最大似然估计来说更容易克服过度拟合问题。我想原因是这样的，过度拟合一般是极大化$p(y^{(i)}|x^{(i)};\theta)$造成的。而在此公式中多了一个参数θ，整个公式由两项组成，极大化$p(y^{(i)}|x^{(i)},\theta)$时，不代表此时p(θ)也能最大化。相反，θ是多值高斯分布，极大化$p(y^{(i)}|x^{(i)},\theta)$时，p(θ)概率反而可能比较小。因此，要达到最大化θ_{MAP}需要在两者之间达到平衡，也就靠近了偏差和方差线的交叉点。这个跟机器翻译里的噪声信道模型比较类似，由两个概率决定比有一个概率决定更靠谱。作者声称利用贝叶斯 logistic 回归（使用θ_{MAP}的 logistic 回归）应用于文本分类时，即使特征个数 n 远远大于样例个数 m，也很有效。

K-means 聚类算法

JerryLead
csxulijie@gmail.com

K-means 也是聚类算法中最简单的一种了，但是里面包含的思想却是不一般。最早我使用并实现这个算法是在学习韩爷爷那本数据挖掘的书中，那本书比较注重应用。看了 Andrew Ng 的这个讲义后才有些明白 K-means 后面包含的 EM 思想。

聚类属于无监督学习，以往的回归、朴素贝叶斯、SVM 等都是有类别标签 y 的，也就是说样例中已经给出了样例的分类。而聚类的样本中却没有给定 y，只有特征 x，比如假设宇宙中的星星可以表示成三维空间中的点集(x, y, z)。聚类的目的是找到每个样本 x 潜在的类别 y，并将同类别 y 的样本 x 放在一起。比如上面的星星，聚类后结果是一个个星团，星团里面的点相互距离比较近，星团间的星星距离就比较远了。

在聚类问题中，给我们的训练样本是$\{x^{(1)}, \ldots, x^{(m)}\}$，每个$x^{(i)} \in \mathbb{R}^n$，没有了 y。

K-means 算法是将样本聚类成 k 个簇（cluster），具体算法描述如下：

1、 随机选取 k 个聚类质心点（cluster centroids）为$\mu_1, \mu_2, \ldots, \mu_k \in \mathbb{R}^n$。

2、 重复下面过程直到收敛 {

对于每一个样例 i，计算其应该属于的类

$$c^{(i)} := \arg\min_j ||x^{(i)} - \mu_j||^2.$$

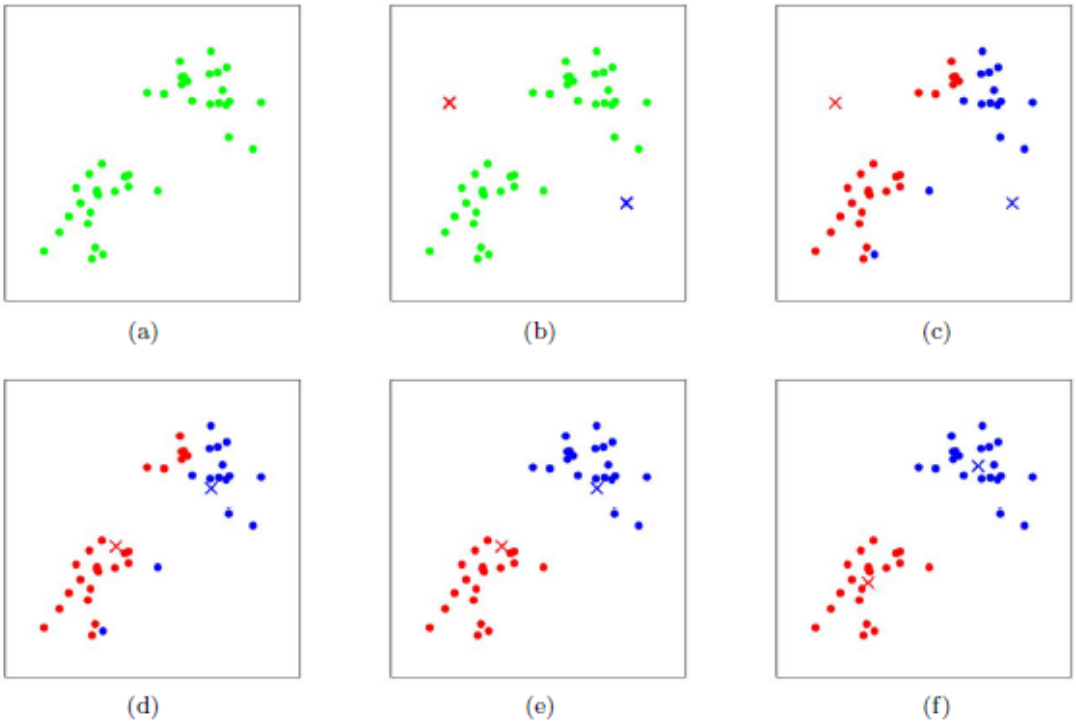
对于每一个类 j，重新计算该类的质心

$$\mu_j := \frac{\sum_{i=1}^{m} 1\{c^{(i)} = j\} x^{(i)}}{\sum_{i=1}^{m} 1\{c^{(i)} = j\}}.$$

}

K 是我们事先给定的聚类数，$c^{(i)}$代表样例 i 与 k 个类中距离最近的那个类，$c^{(i)}$的值是 1 到 k 中的一个。质心μ_j代表我们对属于同一个类的样本中心点的猜测，拿星团模型来解释就是要将所有的星星聚成 k 个星团，首先随机选取 k 个宇宙中的点（或者 k 个星星）作为 k 个星团的质心，然后第一步对于每一个星星计算其到 k 个质心中每一个的距离，然后选取距离最近的那个星团作为$c^{(i)}$，这样经过第一步每一个星星都有了所属的星团；第二步对于每一个星团，重新计算它的质心μ_j（对里面所有的星星坐标求平均）。重复迭代第一步和第二步直到质心不变或者变化很小。

下图展示了对 n 个样本点进行 K-means 聚类的效果，这里 k 取 2。



K-means 面对的第一个问题是如何保证收敛，前面的算法中强调结束条件就是收敛，可以证明的是 K-means 完全可以保证收敛性。下面我们定性的描述一下收敛性，我们定义畸变函数（distortion function）如下：

$$J(c,\mu)=\sum_{i=1}^{m}||x^{(i)}-\mu_{c^{(i)}}||^2$$

J 函数表示每个样本点到其质心的距离平方和。K-means 是要将 J 调整到最小。假设当前 J 没有达到最小值，那么首先可以固定每个类的质心μ_j，调整每个样例的所属的类别$c^{(i)}$来让 J 函数减少，同样，固定$c^{(i)}$，调整每个类的质心μ_j也可以使 J 减小。这两个过程就是内循环中使 J 单调递减的过程。当 J 递减到最小时，μ和 c 也同时收敛。（在理论上，可以有多组不同的μ和 c 值能够使得 J 取得最小值，但这种现象实际上很少见）。

由于畸变函数 J 是非凸函数，意味着我们不能保证取得的最小值是全局最小值，也就是说 k-means 对质心初始位置的选取比较感冒，但一般情况下 k-means 达到的局部最优已经满足需求。但如果你怕陷入局部最优，那么可以选取不同的初始值跑多遍 k-means，然后取其中最小的 J 对应的μ和 c 输出。

下面累述一下 K-means 与 EM 的关系，首先回到初始问题，我们目的是将样本分成 k 个类，其实说白了就是求每个样例 x 的隐含类别 y，然后利用隐含类别将 x 归类。由于我们事先不知道类别 y，那么我们首先可以对每个样例假定一个 y 吧，但是怎么知道假定的对不对呢？怎么评价假定的好不好呢？我们使用样本的极大似然估计来度量，这里是就是 x 和 y 的联合分布 P(x,y)了。如果找到的 y 能够使 P(x,y)最大，那么我们找到的 y 就是样例 x 的最佳类别了，x 顺手就聚类了。但是我们第一次指定的 y 不一定会让 P(x,y)最大，而且 P(x,y)还依赖于其他未知参数，当然在给定 y 的情况下，我们可以调整其他参数让 P(x,y)最大。但是调整完参数后，我们发现有更好的 y 可以指定，那么我们重新指定 y，然后再计算 P(x,y)最大时的参数，反复迭代直至没有更好的 y 可以指定。

这个过程有几个难点，第一怎么假定 y？是每个样例硬指派一个 y 还是不同的 y 有不同的概率，概率如何度量。第二如何估计 P(x,y)，P(x,y)还可能依赖很多其他参数，如何调整里面的参数让 P(x,y)最大。这些问题在以后的篇章里回答。

这里只是指出 EM 的思想，E 步就是估计隐含类别 y 的期望值，M 步调整其他参数使得在给定类别 y 的情况下，极大似然估计 P(x,y)能够达到极大值。然后在其他参数确定的情况下，重新估计 y，周而复始，直至收敛。

上面的阐述有点费解，对应于 K-means 来说就是我们一开始不知道每个样例$x^{(i)}$对应隐含变量也就是最佳类别$c^{(i)}$。最开始可以随便指定一个$c^{(i)}$给它，然后为了让 P(x,y)最大（这里是要让 J 最小），我们求出在给定 c 情况下，J 最小时的μ_j（前面提到的其他未知参数），然而此时发现，可以有更好的$c^{(i)}$（质心与样例$x^{(i)}$距离最小的类别）指定给样例$x^{(i)}$，那么$c^{(i)}$得到重新调整，上述过程就开始重复了，直到没有更好的$c^{(i)}$指定。这样从 K-means 里我们可以看出它其实就是 EM 的体现，E 步是确定隐含类别变量c，M 步更新其他参数μ来使 J 最小化。这里的隐含类别变量指定方法比较特殊，属于硬指定，从 k 个类别中硬选出一个给样例，而不是对每个类别赋予不同的概率。总体思想还是一个迭代优化过程，有目标函数，也有参数变量，只是多了个隐含变量，确定其他参数估计隐含变量，再确定隐含变量估计其他参数，直至目标函数最优。

混合高斯模型（Mixtures of Gaussians）和 EM 算法

JerryLead
csxulijie@gmail.com

这篇讨论使用期望最大化算法（Expectation-Maximization）来进行密度估计（density estimation）。

与 k-means 一样，给定的训练样本是$\{x^{(1)},\ldots,x^{(m)}\}$，我们将隐含类别标签用$z^{(i)}$表示。与 k-means 的硬指定不同，我们首先认为$z^{(i)}$是满足一定的概率分布的，这里我们认为满足多项式分布，$z^{(i)}$~Multinomial(∅)，其中$p(z^{(i)}=j)=\emptyset_j$，$\emptyset_j\geq 0$，$\sum_{j=1}^{k}\emptyset_j=1$，$z^{(i)}$有 k 个值{1,...,k}可以选取。而且我们认为在给定$z^{(i)}$后，$x^{(i)}$满足多值高斯分布，即$(x^{(i)}|z^{(i)}=j)\sim N(\mu_j,\Sigma_j)$。由此可以得到联合分布$p(x^{(i)},z^{(i)})=p(x^{(i)}|z^{(i)})p(z^{(i)})$。

整个模型简单描述为对于每个样例$x^{(i)}$，我们先从 k 个类别中按多项式分布抽取一个$z^{(i)}$，然后根据$z^{(i)}$所对应的 k 个多值高斯分布中的一个生成样例$x^{(i)}$，。整个过程称作混合高斯模型。注意的是这里的$z^{(i)}$仍然是隐含随机变量。模型中还有三个变量∅，μ和Σ。最大似然估计为$p(x,z)$。对数化后如下：

$$\begin{aligned}\ell(\phi,\mu,\Sigma) &= \sum_{i=1}^{m}\log p(x^{(i)};\phi,\mu,\Sigma)\\ &= \sum_{i=1}^{m}\log\sum_{z^{(i)}=1}^{k}p(x^{(i)}|z^{(i)};\mu,\Sigma)p(z^{(i)};\phi).\end{aligned}$$

这个式子的最大值是不能通过前面使用的求导数为 0 的方法解决的，因为求的结果不是 close form。但是假设我们知道了每个样例的$z^{(i)}$，那么上式可以简化为：

$$\ell(\phi,\mu,\Sigma)=\sum_{i=1}^{m}\log p(x^{(i)}|z^{(i)};\mu,\Sigma)+\log p(z^{(i)};\phi).$$

这时候我们再来对∅，μ和Σ进行求导得到：

$$\begin{aligned}\phi_j &= \frac{1}{m}\sum_{i=1}^{m}1\{z^{(i)}=j\},\\ \mu_j &= \frac{\sum_{i=1}^{m}1\{z^{(i)}=j\}x^{(i)}}{\sum_{i=1}^{m}1\{z^{(i)}=j\}},\\ \Sigma_j &= \frac{\sum_{i=1}^{m}1\{z^{(i)}=j\}(x^{(i)}-\mu_j)(x^{(i)}-\mu_j)^T}{\sum_{i=1}^{m}1\{z^{(i)}=j\}}.\end{aligned}$$

$\emptyset_j$就是样本类别中$z^{(i)}=j$的比率。μ_j是类别为 j 的样本特征均值，Σ_j是类别为 j 的样例的特征的协方差矩阵。

实际上，当知道$z^{(i)}$后，最大似然估计就近似于高斯判别分析模型（Gaussian discriminant analysis model）了。所不同的是 GDA 中类别 y 是伯努利分布，而这里的 z 是多项式分布，还有这里的每个样例都有不同的协方差矩阵，而 GDA 中认为只有一个。

之前我们是假设给定了$z^{(i)}$，实际上$z^{(i)}$是不知道的。那么怎么办呢？考虑之前提到的 EM 的思想，第一步是猜测隐含类别变量 z，第二步是更新其他参数，以获得最大的最大似然估计。用到这里就是：

循环下面步骤，直到收敛： {

（E 步）对于每一个 i 和 j，计算

$$w_j^{(i)} := p(z^{(i)} = j | x^{(i)}; \Phi, \mu, \Sigma)$$

（M 步），更新参数：

$$\begin{aligned}
\phi_j &:= \frac{1}{m}\sum_{i=1}^{m} w_j^{(i)}, \\
\mu_j &:= \frac{\sum_{i=1}^{m} w_j^{(i)} x^{(i)}}{\sum_{i=1}^{m} w_j^{(i)}}, \\
\Sigma_j &:= \frac{\sum_{i=1}^{m} w_j^{(i)} (x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^T}{\sum_{i=1}^{m} w_j^{(i)}}
\end{aligned}$$

}

在 E 步中，我们将其他参数Φ, μ, Σ看作常量，计算$z^{(i)}$的后验概率，也就是估计隐含类别变量。估计好后，利用上面的公式重新计算其他参数，计算好后发现最大化最大似然估计时，$w_j^{(i)}$值又不对了，需要重新计算，周而复始，直至收敛。

$w_j^{(i)}$的具体计算公式如下：

$$p(z^{(i)} = j | x^{(i)}; \phi, \mu, \Sigma) = \frac{p(x^{(i)} | z^{(i)} = j; \mu, \Sigma) p(z^{(i)} = j; \phi)}{\sum_{l=1}^{k} p(x^{(i)} | z^{(i)} = l; \mu, \Sigma) p(z^{(i)} = l; \phi)}$$

这个式子利用了贝叶斯公式。

这里我们使用$w_j^{(i)}$代替了前面的$1\{z^{(i)} = j\}$，由简单的 0/1 值变成了概率值。

对比 K-means 可以发现，这里使用了“软”指定，为每个样例分配的类别$z^{(i)}$是有一定的概率的，同时计算量也变大了，每个样例 i 都要计算属于每一个类别 j 的概率。与 K-means 相同的是，结果仍然是局部最优解。对其他参数取不同的初始值进行多次计算不失为一种好方法。

虽然之前再 K-means 中定性描述了 EM 的收敛性，仍然没有定量地给出，还有一般化 EM 的推导过程仍然没有给出。下一篇着重介绍这些内容。

The EM Algorithm

JerryLead
csxulijie@gmail.com

EM 是我一直想深入学习的算法之一，第一次听说是在 NLP 课中的 HMM 那一节，为了解决 HMM 的参数估计问题，使用了 EM 算法。在之后的 MT 中的词对齐中也用到了。在 Mitchell 的书中也提到 EM 可以用于贝叶斯网络中。

下面主要介绍 EM 的整个推导过程。

1. Jensen 不等式

回顾优化理论中的一些概念。设 f 是定义域为实数的函数，如果对于所有的实数 x，$f''(x) \geq 0$，那么 f 是凸函数。当 x 是向量时，如果其 hessian 矩阵 H 是半正定的（$H \geq 0$），那么 f 是凸函数。如果$f''(x) > 0$或者$H > 0$，那么称 f 是严格凸函数。

Jensen 不等式表述如下：

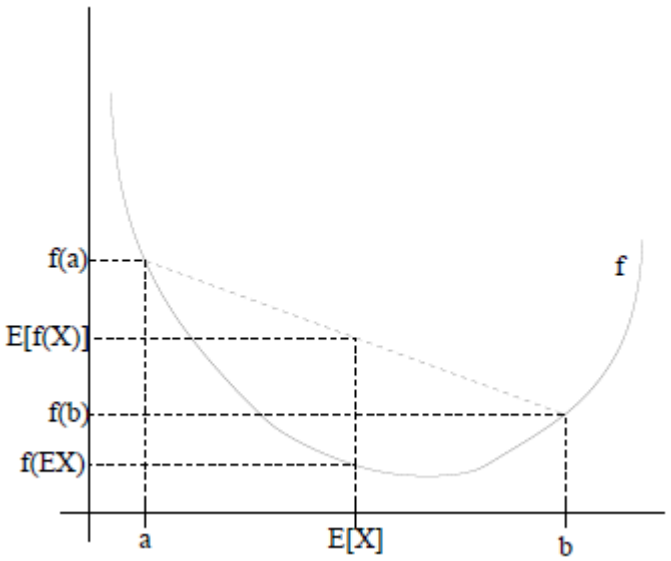
如果 f 是凸函数，X 是随机变量，那么

$E[f(X)] \geq f(EX)$

特别地，如果 f 是严格凸函数，那么$E[f(X)] = f(EX)$当且仅当$p(x = E[X]) = 1$，也就是说 X 是常量。

这里我们将$f(E[X])$简写为$f(EX)$。

如果用图表示会很清晰：



图中，实线 f 是凸函数，X 是随机变量，有 0.5 的概率是 a，有 0.5 的概率是 b。（就像掷硬币一样）。X 的期望值就是 a 和 b 的中值了，图中可以看到$E[f(X)] \geq f(EX)$成立。

当 f 是（严格）凹函数当且仅当-f 是（严格）凸函数。

Jensen 不等式应用于凹函数时，不等号方向反向，也就是$E[f(X)] \leq f(EX)$。

2. EM 算法

给定的训练样本是$\{x^{(1)},\dots,x^{(m)}\}$，样例间独立，我们想找到每个样例隐含的类别 z，能使得 p(x,z)最大。p(x,z)的最大似然估计如下：

$$\begin{aligned}\ell(\theta) &= \sum_{i=1}^{m}\log p(x;\theta)\\ &= \sum_{i=1}^{m}\log\sum_{z}p(x,z;\theta).\end{aligned}$$

第一步是对极大似然取对数，第二步是对每个样例的每个可能类别 z 求联合分布概率和。但是直接求θ一般比较困难，因为有隐藏变量 z 存在，但是一般确定了 z 后，求解就容易了。

EM 是一种解决存在隐含变量优化问题的有效方法。竟然不能直接最大化$\ell(\theta)$，我们可以不断地建立ℓ的下界（E 步），然后优化下界（M 步）。这句话比较抽象，看下面的。

对于每一个样例 i，让Q_i表示该样例隐含变量 z 的某种分布，Q_i满足的条件是$\sum_z Q_i(z)=1, Q_i(z)\geq 0$。（如果 z 是连续性的，那么$Q_i$是概率密度函数，需要将求和符号换做积分符号）。比如要将班上学生聚类，假设隐藏变量 z 是身高，那么就是连续的高斯分布。如果按照隐藏变量是男女，那么就是伯努利分布了。

可以由前面阐述的内容得到下面的公式：

$$\begin{aligned}\sum_i\log p(x^{(i)};\theta) &= \sum_i\log\sum_{z^{(i)}}p(x^{(i)},z^{(i)};\theta) &(1)\\ &= \sum_i\log\sum_{z^{(i)}}Q_i(z^{(i)})\frac{p(x^{(i)},z^{(i)};\theta)}{Q_i(z^{(i)})} &(2)\\ &\geq \sum_i\sum_{z^{(i)}}Q_i(z^{(i)})\log\frac{p(x^{(i)},z^{(i)};\theta)}{Q_i(z^{(i)})} &(3)\end{aligned}$$

（1）到（2）比较直接，就是分子分母同乘以一个相等的函数。（2）到（3）利用了 Jensen 不等式，考虑到$\log(x)$是凹函数（二阶导数小于 0），而且

$$\sum_{z^{(i)}}Q_i(z^{(i)})\left[\frac{p(x^{(i)},z^{(i)};\theta)}{Q_i(z^{(i)})}\right]$$

就是$\left[p(x^{(i)},z^{(i)};\theta)/Q_i(z^{(i)})\right]$的期望（回想期望公式中的 Lazy Statistician 规则）

设 Y 是随机变量 X 的函数，Y = g(X)（g 是连续函数），那么

（1） X 是离散型随机变量，它的分布律为$P(X=x_k)=p_k$，k=1,2,…。若$\sum_{k=1}^{\infty}g(x_k)p_k$绝对收敛，则有

$$E(Y)=E[g(X)]=\sum_{k=1}^{\infty}g(x_k)p_k$$

（2） X 是连续型随机变量，它的概率密度为f(x)，若$\int_{-\infty}^{\infty} g(x)f(x)dx$绝对收敛，则有

$$E(Y) = E[g(X)] = \int_{-\infty}^{\infty} g(x)f(x)dx$$

对应于上述问题，Y 是 $\left[p(x^{(i)}, z^{(i)};\theta)/Q_i(z^{(i)})\right]$，X 是$z^{(i)}$，$Q_i(z^{(i)})$是$p_k$，g 是$z^{(i)}$到 $\left[p(x^{(i)}, z^{(i)};\theta)/Q_i(z^{(i)})\right]$ 的映射。这样解释了式子（2）中的期望，再根据凹函数时的 Jensen 不等式：

$$f\left(\mathrm{E}_{z^{(i)}\sim Q_i}\left[\frac{p(x^{(i)}, z^{(i)};\theta)}{Q_i(z^{(i)})}\right]\right) \geq \mathrm{E}_{z^{(i)}\sim Q_i}\left[f\left(\frac{p(x^{(i)}, z^{(i)};\theta)}{Q_i(z^{(i)})}\right)\right],$$

可以得到（3）。

这个过程可以看作是对$\ell(\theta)$求了下界。对于Q_i的选择，有多种可能，那种更好的？假设θ已经给定，那么$\ell(\theta)$的值就决定于$Q_i(z^{(i)})$和$p(x^{(i)}, z^{(i)})$了。我们可以通过调整这两个概率使下界不断上升，以逼近$\ell(\theta)$的真实值，那么什么时候算是调整好了呢？当不等式变成等式时，说明我们调整后的概率能够等价于$\ell(\theta)$了。按照这个思路，我们要找到等式成立的条件。根据 Jensen 不等式，要想让等式成立，需要让随机变量变成常数值，这里得到：

$$\frac{p(x^{(i)}, z^{(i)};\theta)}{Q_i(z^{(i)})} = c$$

c 为常数，不依赖于$z^{(i)}$。对此式子做进一步推导，我们知道$\sum_z Q_i(z^{(i)}) = 1$，那么也就有$\sum_z p(x^{(i)}, z^{(i)};\theta) = c$，（多个等式分子分母相加不变，这个认为每个样例的两个概率比值都是 c），那么有下式：

$$\begin{aligned} Q_i(z^{(i)}) &= \frac{p(x^{(i)}, z^{(i)};\theta)}{\sum_z p(x^{(i)}, z;\theta)} \\ &= \frac{p(x^{(i)}, z^{(i)};\theta)}{p(x^{(i)};\theta)} \\ &= p(z^{(i)}|x^{(i)};\theta) \end{aligned}$$

至此，我们推出了在固定其他参数θ后，$Q_i(z^{(i)})$的计算公式就是后验概率，解决了$Q_i(z^{(i)})$如何选择的问题。这一步就是 E 步，建立$\ell(\theta)$的下界。接下来的 M 步，就是在给定$Q_i(z^{(i)})$后，调整θ，去极大化$\ell(\theta)$的下界（在固定$Q_i(z^{(i)})$后，下界还可以调整的更大）。那么一般的 EM 算法的步骤如下：

循环重复直到收敛 {

（E 步）对于每一个 i，计算

$$Q_i(z^{(i)}) := p(z^{(i)}|x^{(i)};\theta)$$

（M 步）计算

$$\theta := \arg\max_\theta \sum_i \sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)};\theta)}{Q_i(z^{(i)})}.$$

那么究竟怎么确保 EM 收敛？假定$\theta^{(t)}$和$\theta^{(t+1)}$是 EM 第 t 次和 t+1 次迭代后的结果。如

果我们证明了$\ell(\theta^{(t)}) \le \ell(\theta^{(t+1)})$，也就是说极大似然估计单调增加，那么最终我们会到达最大似然估计的最大值。下面来证明，选定$\theta^{(t)}$后，我们得到 E 步

$$Q_i^{(t)}(z^{(i)}) := p(z^{(i)}|x^{(i)};\theta^{(t)})$$

这一步保证了在给定$\theta^{(t)}$时，Jensen 不等式中的等式成立，也就是

$$\ell(\theta^{(t)}) = \sum_i \sum_{z^{(i)}} Q_i^{(t)}(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta^{(t)})}{Q_i^{(t)}(z^{(i)})}.$$

然后进行 M 步，固定$Q_i^{(t)}(z^{(i)})$，并将$\theta^{(t)}$视作变量，对上面的$\ell(\theta^{(t)})$求导后，得到$\theta^{(t+1)}$，这样经过一些推导会有以下式子成立：

$$\begin{aligned}
\ell(\theta^{(t+1)}) &\ge \sum_i \sum_{z^{(i)}} Q_i^{(t)}(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta^{(t+1)})}{Q_i^{(t)}(z^{(i)})} && (4)\\
&\ge \sum_i \sum_{z^{(i)}} Q_i^{(t)}(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta^{(t)})}{Q_i^{(t)}(z^{(i)})} && (5)\\
&= \ell(\theta^{(t)}) && (6)
\end{aligned}$$

解释第（4）步，得到$\theta^{(t+1)}$时，只是最大化$\ell(\theta^{(t)})$，也就是$\ell(\theta^{(t+1)})$的下界，而没有使等式成立，等式成立只有是在固定θ，并按 E 步得到Q_i时才能成立。

况且根据我们前面得到的下式，对于所有的Q_i和θ都成立

$$\ell(\theta) \ge \sum_i \sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})}$$

第（5）步利用了 M 步的定义，M 步就是将$\theta^{(t)}$调整到$\theta^{(t+1)}$，使得下界最大化。因此（5）成立，（6）是之前的等式结果。

这样就证明了$\ell(\theta)$会单调增加。一种收敛方法是$\ell(\theta)$不再变化，还有一种就是变化幅度很小。

再次解释一下（4）、（5）、（6）。首先（4）对所有的参数都满足，而其等式成立条件只是在固定θ，并调整好 Q 时成立，而第（4）步只是固定 Q，调整θ，不能保证等式一定成立。（4）到（5）就是 M 步的定义，（5）到（6）是前面 E 步所保证等式成立条件。也就是说 E 步会将下界拉到与$\ell(\theta)$一个特定值（这里$\theta^{(t)}$）一样的高度，而此时发现下界仍然可以上升，因此经过 M 步后，下界又被拉升，但达不到与$\ell(\theta)$另外一个特定值一样的高度，之后 E 步又将下界拉到与这个特定值一样的高度，重复下去，直到最大值。

如果我们定义

$$J(Q, \theta) = \sum_i \sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})},$$

从前面的推导中我们知道$\ell(\theta) \ge J(Q, \theta)$，EM 可以看作是 J 的坐标上升法，E 步固定θ，优化Q，M 步固定Q优化θ。

3. 重新审视混合高斯模型

我们已经知道了 EM 的精髓和推导过程，再次审视一下混合高斯模型。之前提到的混合高斯模型的参数∅，μ和Σ计算公式都是根据很多假定得出的，有些没有说明来由。为了简单，这里在 M 步只给出∅和μ的推导方法。

E 步很简单，按照一般 EM 公式得到：

$$w_j^{(i)} = Q_i(z^{(i)} = j) = P(z^{(i)} = j|x^{(i)}; \phi, \mu, \Sigma).$$

简单解释就是每个样例 i 的隐含类别$z^{(i)}$为 j 的概率可以通过后验概率计算得到。

在 M 步中，我们需要在固定$Q_i(z^{(i)})$后最大化最大似然估计，也就是

$$\begin{aligned}
&\sum_{i=1}^{m}\sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \phi, \mu, \Sigma)}{Q_i(z^{(i)})} \\
&= \sum_{i=1}^{m}\sum_{j=1}^{k} Q_i(z^{(i)} = j) \log \frac{p(x^{(i)}|z^{(i)} = j; \mu, \Sigma)p(z^{(i)} = j; \phi)}{Q_i(z^{(i)} = j)} \\
&= \sum_{i=1}^{m}\sum_{j=1}^{k} w_j^{(i)} \log \frac{\frac{1}{(2\pi)^{n/2}|\Sigma_j|^{1/2}} \exp\left(-\frac{1}{2}(x^{(i)} - \mu_j)^T\Sigma_j^{-1}(x^{(i)} - \mu_j)\right) \cdot \phi_j}{w_j^{(i)}}
\end{aligned}$$

这是将$z^{(i)}$的 k 种情况展开后的样子，未知参数∅j，μj和Σj。

固定$\emptyset_j$和Σ_j，对μ_j求导得

$$\begin{aligned}
&\nabla_{\mu_l} \sum_{i=1}^{m}\sum_{j=1}^{k} w_j^{(i)} \log \frac{\frac{1}{(2\pi)^{n/2}|\Sigma_j|^{1/2}} \exp\left(-\frac{1}{2}(x^{(i)} - \mu_j)^T\Sigma_j^{-1}(x^{(i)} - \mu_j)\right) \cdot \phi_j}{w_j^{(i)}} \\
&= -\nabla_{\mu_l} \sum_{i=1}^{m}\sum_{j=1}^{k} w_j^{(i)} \frac{1}{2}(x^{(i)} - \mu_j)^T\Sigma_j^{-1}(x^{(i)} - \mu_j) \\
&= \frac{1}{2}\sum_{i=1}^{m} w_l^{(i)} \nabla_{\mu_l} 2\mu_l^T\Sigma_l^{-1}x^{(i)} - \mu_l^T\Sigma_l^{-1}\mu_l \\
&= \sum_{i=1}^{m} w_l^{(i)} \left(\Sigma_l^{-1}x^{(i)} - \Sigma_l^{-1}\mu_l\right)
\end{aligned}$$

等于 0 时，得到

$$\mu_l := \frac{\sum_{i=1}^{m} w_l^{(i)} x^{(i)}}{\sum_{i=1}^{m} w_l^{(i)}},$$

这就是我们之前模型中的μ的更新公式。

然后推导$\emptyset_j$的更新公式。看之前得到的

$$\sum_{i=1}^{m}\sum_{j=1}^{k} w_j^{(i)} \log \frac{\frac{1}{(2\pi)^{n/2}|\Sigma_j|^{1/2}} \exp\left(-\frac{1}{2}(x^{(i)}-\mu_j)^T\Sigma_j^{-1}(x^{(i)}-\mu_j)\right)\cdot \phi_j}{w_j^{(i)}}$$

在∅和μ确定后，分子上面的一串都是常数了，实际上需要优化的公式是：

$$\sum_{i=1}^{m}\sum_{j=1}^{k} w_j^{(i)} \log \phi_j.$$

需要知道的是，$\emptyset_j$还需要满足一定的约束条件就是$\sum_{j=1}^{k}\emptyset_j = 1$。
这个优化问题我们很熟悉了，直接构造拉格朗日乘子。

$$\mathcal{L}(\phi) = \sum_{i=1}^{m}\sum_{j=1}^{k} w_j^{(i)} \log \phi_j + \beta(\sum_{j=1}^{k}\phi_j - 1),$$

还有一点就是$\emptyset_j \geq 0$，但这一点会在得到的公式里自动满足。
求导得，

$$\frac{\partial}{\partial \emptyset_j}\mathcal{L}(\emptyset) = \sum_{i=1}^{m}\frac{w_j^{(i)}}{\emptyset_j} + \beta$$

等于 0，得到

$$\phi_j = \frac{\sum_{i=1}^{m} w_j^{(i)}}{-\beta}$$

也就是说$\phi_j \propto \sum_{i=1}^{m} w_j^{(i)}$. 再次使用$\sum_{j=1}^{k}\emptyset_j = 1$，得到

$$-\beta = \sum_{i=1}^{m}\sum_{j=1}^{k} w_j^{(i)} = \sum_{i=1}^{m} 1 = m.$$

这样就神奇地得到了β。
那么就顺势得到 M 步中$\emptyset_j$的更新公式：

$$\phi_j := \frac{1}{m}\sum_{i=1}^{m} w_j^{(i)}.$$

Σ的推导也类似，不过稍微复杂一些，毕竟是矩阵。结果在之前的混合高斯模型中已经给出。

4. 总结

如果将样本看作观察值，潜在类别看作是隐藏变量，那么聚类问题也就是参数估计问题，只不过聚类问题中参数分为隐含类别变量和其他参数，这犹如在 x-y 坐标系中找一个曲线的极值，然而曲线函数不能直接求导，因此什么梯度下降方法就不适用了。但固定一个变量后，另外一个可以通过求导得到，因此可以使用坐标上升法，一次固定一个变量，对另外的求极值，最后逐步逼近极值。对应到 EM 上，E 步估计隐含变量，M 步估计其他参数，交替将极

值推向最大。EM 中还有“硬”指定和“软”指定的概念，“软”指定看似更为合理，但计算量要大，“硬”指定在某些场合如 K-means 中更为实用（要是保持一个样本点到其他所有中心的概率，就会很麻烦）。

另外，EM 的收敛性证明方法确实很牛，能够利用 log 的凹函数性质，还能够想到利用创造下界，拉平函数下界，优化下界的方法来逐步逼近极大值。而且每一步迭代都能保证是单调的。最重要的是证明的数学公式非常精妙，硬是分子分母都乘以 z 的概率变成期望来套上 Jensen 不等式，前人都是怎么想到的。

在 Mitchell 的 Machine Learning 书中也举了一个 EM 应用的例子，明白地说就是将班上学生的身高都放在一起，要求聚成两个类。这些身高可以看作是男生身高的高斯分布和女生身高的高斯分布组成。因此变成了如何估计每个样例是男生还是女生，然后在确定男女生情况下，如何估计均值和方差，里面也给出了公式，有兴趣可以参考。

在线学习（Online Learning）

JerryLead
csxulijie@gmail.com

原题目叫做 The perception and large margin classifiers，其实探讨的是在线学习。这里将题目换了换。以前讨论的都是批量学习（batch learning），就是给了一堆样例后，在样例上学习出假设函数 h。而在线学习就是要根据新来的样例，边学习，边给出结果。

假设样例按照到来的先后顺序依次定义为$(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})$。X 为样本特征，y 为类别标签。我们的任务是到来一个样例 x，给出其类别结果 y 的预测值，之后我们会看到 y 的真实值，然后根据真实值来重新调整模型参数，整个过程是重复迭代的过程，直到所有的样例完成。这么看来，我们也可以将原来用于批量学习的样例拿来作为在线学习的样例。在在线学习中我们主要关注在整个预测过程中预测错误的样例数。

拿二值分类来讲，我们用 y=1 表示正例，y=-1 表示负例。回想在讨论支持向量机中提到的感知算法（perception algorithm）。我们的假设函数为

$$h_\theta(x) = g(\theta^T x)$$

其中 x 是 n 维特征向量，θ是 n+1 维参数权重。函数 g 用来将$\theta^T x$计算结果映射到-1 和 1 上。具体公式如下：

$$g(z) = \begin{cases} 1 & \text{if } z \geq 0 \\ -1 & \text{if } z < 0. \end{cases}$$

这个也是 logistic 回归中 g 的简化形式。

现在我们提出一个在线学习算法如下：

新来一个样例(x, y)，我们先用从之前样例学习到的$h_\theta(x)$来得到样例的预测值 y，如果$h_\theta(x) = y$（即预测正确），那么不改变θ，反之

$$\theta := \theta + yx$$

也就是说，如果对于预测错误的样例，θ进行调整时只需加上（实际上为正例）或者减去（实际负例）样本特征 x 值即可。θ初始值为向量 0。这里我们关心的是$\theta^T x$的符号，而不是它的具体值。调整方法非常简单。然而这个简单的调整方法还是很有效的，它的错误率不仅是有上界的，而且这个上界不依赖于样例数和特征维度。

下面定理阐述了错误率上界：

定理（Block and Novikoff）：

给定按照顺序到来的$(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})$样例。假设对于所有的样例$\|x^{(i)}\| \leq D$，也就是说特征向量长度有界为 D。更进一步，假设存在一个单位长度向量 u ($\|u\| = 1$)且$y^{(i)} \cdot (u^T x^{(i)}) \geq \gamma$。也就是说对于 y=1 的正例，$(u^T x^{(i)}) \geq \gamma$，反例$(u^T x^{(i)}) \leq -\gamma$，u 能够有$\gamma$的间隔将正例和反例分开。那么感知算法的预测的错误样例数不超过$\left(\frac{D}{\gamma}\right)^2$。

根据前面对 SVM 的理解，这个定理就可以阐述为：如果训练样本线性可分，并且几何

间距至少是γ，样例样本特征向量最长为 D，那么感知算法错误数不会超过$\left(\frac{D}{\gamma}\right)^2$。这个定理是 62 年提出的，63 年 Vapnik 提出 SVM，可见提出也不是偶然的，感知算法也许是当时的热门。

下面主要讨论这个定理的证明：

感知算法只在样例预测错误时进行更新，定义$\theta^{(k)}$是第 k 次预测错误时使用的样本特征权重，$\theta^{(1)}=\vec{0}$ 初始化为 0 向量。假设第 k 次预测错误发生在样例$(x^{(i)},y^{(i)})$上，利用$\theta^{(k)}$计算$y^{(i)}$值时得到的结果不正确（也就是说$g((x^{(i)})^T\theta^k)\neq y^{(i)}$，调换 x 和θ顺序主要是为了书写方便）。也就是说下面的公式成立：

$$(x^{(i)})^T\theta^{(k)}y^{(i)}\leq 0.$$

根据感知算法的更新方法，我们有$\theta^{(k+1)}=\theta^{(k)}+y^{(i)}x^{(i)}$。这时候，两边都乘以 u 得到

$$\begin{aligned}(\theta^{(k+1)})^Tu &= (\theta^{(k)})^Tu+y^{(i)}(x^{(i)})^Tu\\ &\geq (\theta^{(k)})^Tu+\gamma\end{aligned}$$

两个向量做内积的时候，放在左边还是右边无所谓，转置符号标注正确即可。

这个式子是个递推公式，就像等差数列一样 f(n+1)=f(n)+d。由此我们可得

$$(\theta^{(k+1)})^Tu\geq k\gamma.$$

因为初始θ为 0。

下面我们利用前面推导出的$(x^{(i)})^T\theta^k y^{(i)})\leq 0$和$\|x^{(i)}\|\leq D$得到

$$\begin{aligned}||\theta^{(k+1)}||^2 &= ||\theta^{(k)}+y^{(i)}x^{(i)}||^2\\ &= ||\theta^{(k)}||^2+||x^{(i)}||^2+2y^{(i)}(x^{(i)})^T\theta^{(i)}\\ &\leq ||\theta^{(k)}||^2+||x^{(i)}||^2\\ &\leq ||\theta^{(k)}||^2+D^2\end{aligned}$$

也就是说$\theta^{(k+1)}$的长度平方不会超过$\theta^{(k)}$与 D 的平方和。

又是一个等差不等式，得到：

$$||\theta^{(k+1)}||^2\leq kD^2.$$

两边开根号得：

$$\begin{aligned}\sqrt{k}D &\geq ||\theta^{(k+1)}||\\ &\geq (\theta^{(k+1)})^Tu\\ &\geq k\gamma.\end{aligned}$$

其中第二步可能有点迷惑，我们细想 u 是单位向量的话，

$$z^Tu=||z||\cdot||u||\cos\phi\leq||z||\cdot||u||$$

因此上面的不等式成立，最后得到：

$$k \leq (D/\gamma)^2.$$

也就是预测错误的数目不会超过样本特征向量 x 的最长长度与几何间隔的平方。实际上整个调整过程中θ就是 x 的线性组合。

整个感知算法应该是在线学习中最简单的一种了。

主成分分析（Principal components analysis）

JerryLead
csxulijie@gmail.com

在这一篇之前的内容是《Factor Analysis》，由于非常理论，打算学完整个课程后再写。在写这篇之前，我阅读了 PCA、SVD 和 LDA。这几个模型相近，却都有自己的特点。本篇打算先介绍 PCA，至于他们之间的关系，只能是边学边体会了。PCA 以前也叫做 Principal factor analysis。

1. 问题

真实的训练数据总是存在各种各样的问题：

1、 比如拿到一个汽车的样本，里面既有以“千米/每小时”度量的最大速度特征，也有“英里/小时”的最大速度特征，显然这两个特征有一个多余。

2、 拿到一个数学系的本科生期末考试成绩单，里面有三列，一列是对数学的兴趣程度，一列是复习时间，还有一列是考试成绩。我们知道要学好数学，需要有浓厚的兴趣，所以第二项与第一项强相关，第三项和第二项也是强相关。那是不是可以合并第一项和第二项呢？

3、 拿到一个样本，特征非常多，而样例特别少，这样用回归去直接拟合非常困难，容易过度拟合。比如北京的房价：假设房子的特征是（大小、位置、朝向、是否学区房、建造年代、是否二手、层数、所在层数），搞了这么多特征，结果只有不到十个房子的样例。要拟合房子特征->房价的这么多特征，就会造成过度拟合。

4、 这个与第二个有点类似，假设在 IR 中我们建立的文档-词项矩阵中，有两个词项为“learn”和“study”，在传统的向量空间模型中，认为两者独立。然而从语义的角度来讲，两者是相似的，而且两者出现频率也类似，是不是可以合成为一个特征呢？

5、 在信号传输过程中，由于信道不是理想的，信道另一端收到的信号会有噪音扰动，那么怎么滤去这些噪音呢？

回顾我们之前介绍的《模型选择和规则化》，里面谈到的特征选择的问题。但在那篇中要剔除的特征主要是和类标签无关的特征。比如“学生的名字”就和他的“成绩”无关，使用的是互信息的方法。

而这里的特征很多是和类标签有关的，但里面存在噪声或者冗余。在这种情况下，需要一种特征降维的方法来减少特征数，减少噪音和冗余，减少过度拟合的可能性。

下面探讨一种称作主成分分析（PCA）的方法来解决部分上述问题。PCA 的思想是将 n 维特征映射到 k 维上（k<n），这 k 维是全新的正交特征。这 k 维特征称为主元，是重新构造出来的 k 维特征，而不是简单地从 n 维特征中去除其余 n-k 维特征。

2. PCA 计算过程

首先介绍 PCA 的计算过程：

假设我们得到的 2 维数据如下：

	x	y
	2.5	2.4
	0.5	0.7
	2.2	2.9
	1.9	2.2
Data =	3.1	3.0
	2.3	2.7
	2	1.6
	1	1.1
	1.5	1.6
	1.1	0.9

行代表了样例，列代表特征，这里有 10 个样例，每个样例两个特征。可以这样认为，有 10 篇文档，x 是 10 篇文档中“learn”出现的 TF-IDF，y 是 10 篇文档中“study”出现的 TF-IDF。也可以认为有 10 辆汽车，x 是千米/小时的速度，y 是英里/小时的速度，等等。

第一步分别求 x 和 y 的平均值，然后对于所有的样例，都减去对应的均值。这里 x 的均值是 1.81，y 的均值是 1.91，那么一个样例减去均值后即为（0.69,0.49），得到

	x	y
	.69	.49
	-1.31	-1.21
	.39	.99
	.09	.29
DataAdjust =	1.29	1.09
	.49	.79
	.19	-.31
	-.81	-.81
	-.31	-.31
	-.71	-1.01

第二步，求特征协方差矩阵，如果数据是 3 维，那么协方差矩阵是

$$C = \begin{pmatrix} cov(x,x) & cov(x,y) & cov(x,z) \\ cov(y,x) & cov(y,y) & cov(y,z) \\ cov(z,x) & cov(z,y) & cov(z,z) \end{pmatrix}$$

这里只有 x 和 y，求解得

$$cov = \begin{pmatrix} .616555556 & .615444444 \\ .615444444 & .716555556 \end{pmatrix}$$

对角线上分别是 x 和 y 的方差，非对角线上是协方差。协方差大于 0 表示 x 和 y 若有一

个增，另一个也增；小于 0 表示一个增，一个减；协方差为 0 时，两者独立。协方差绝对值越大，两者对彼此的影响越大，反之越小。

第三步，求协方差的特征值和特征向量，得到

$$eigenvalues = \begin{pmatrix} .0490833989 \\ 1.28402771 \end{pmatrix}$$

$$eigenvectors = \begin{pmatrix} -.735178656 & -.677873399 \\ .677873399 & -.735178656 \end{pmatrix}$$

上面是两个特征值，下面是对应的特征向量，特征值 0.0490833989 对应特征向量为 $(-0.735178656, 0.677873399)^T$，这里的特征向量都归一化为单位向量。

第四步，将特征值按照从大到小的顺序排序，选择其中最大的 k 个，然后将其对应的 k 个特征向量分别作为列向量组成特征向量矩阵。

这里特征值只有两个，我们选择其中最大的那个，这里是 1.28402771，对应的特征向量是$(-0.677873399, -0.735178656)^T$。

第五步，将样本点投影到选取的特征向量上。假设样例数为 m，特征数为 n，减去均值后的样本矩阵为 DataAdjust(m*n)，协方差矩阵是 n*n，选取的 k 个特征向量组成的矩阵为 EigenVectors(n*k)。那么投影后的数据 FinalData 为

$$\text{FinalData}(m * k) = \text{DataAdjust}(m * n) \times \text{EigenVectors}(n * k)$$

这里是

FinalData(10*1) = DataAdjust(10*2 矩阵)×特征向量$(-0.677873399, -0.735178656)^T$

得到结果是

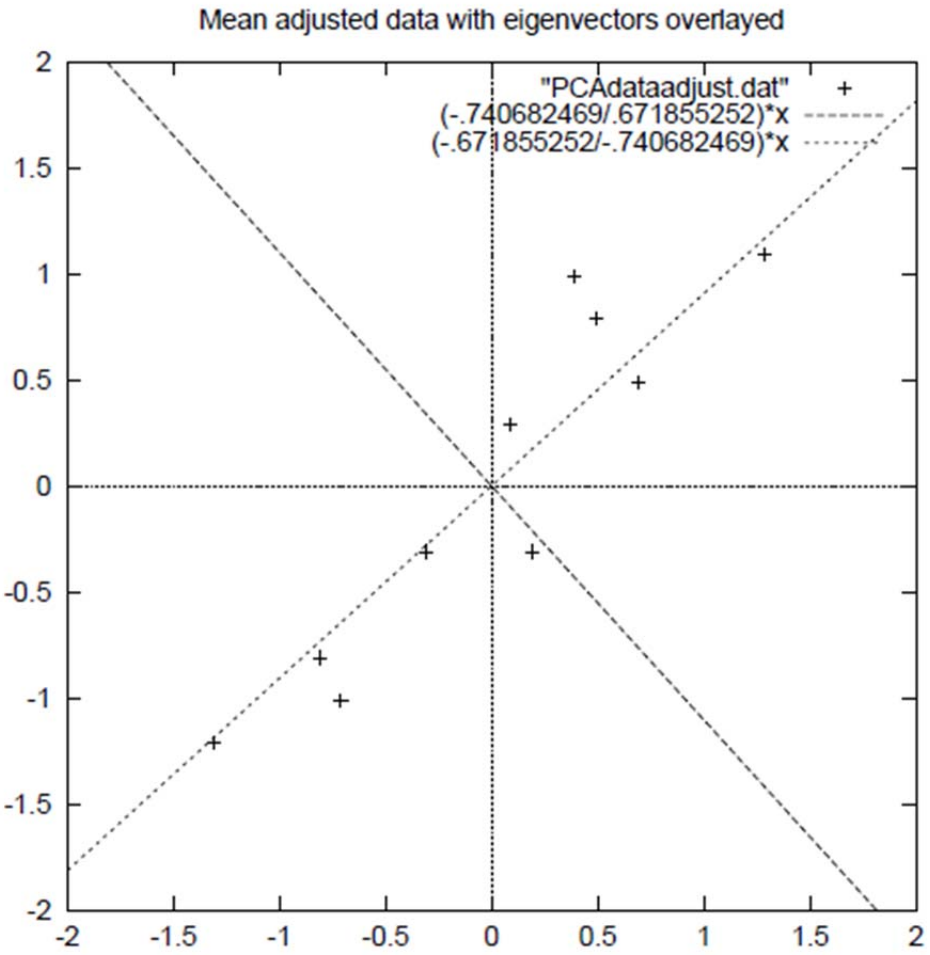
Transformed Data (Single eigenvector)

x
-.827970186
1.77758033
-.992197494
-.274210416
-1.67580142
-.912949103
.0991094375
1.14457216
.438046137
1.22382056

这样，就将原始样例的 n 维特征变成了 k 维，这 k 维就是原始特征在 k 维上的投影。

上面的数据可以认为是 learn 和 study 特征融合为一个新的特征叫做 LS 特征，该特征基本上代表了这两个特征。

上述过程有个图描述：

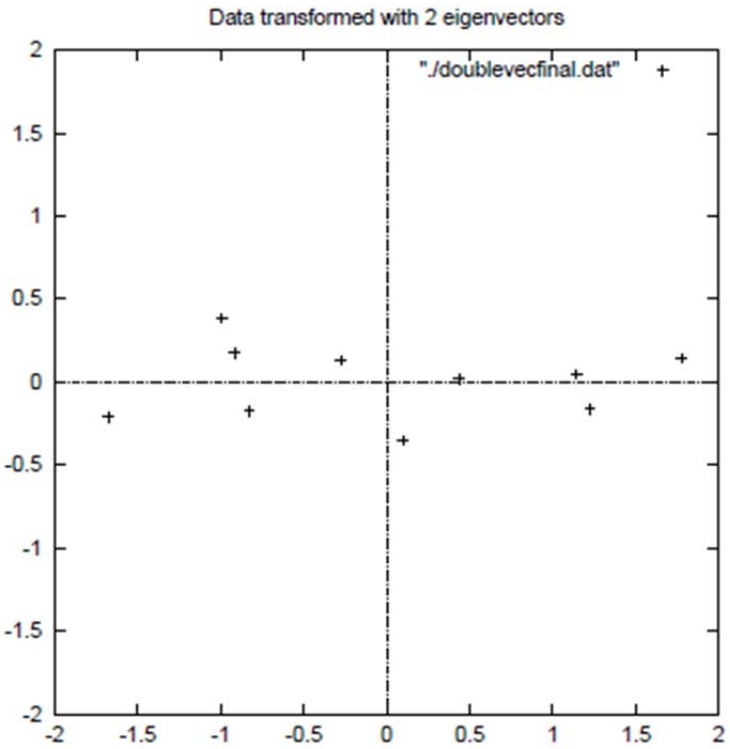


正号表示预处理后的样本点，斜着的两条线就分别是正交的特征向量（由于协方差矩阵是对称的，因此其特征向量正交），最后一步的矩阵乘法就是将原始样本点分别往特征向量对应的轴上做投影。

如果取的 k=2，那么结果是

Transformed Data=

x	y
-.827970186	-.175115307
1.77758033	.142857227
-.992197494	.384374989
-.274210416	.130417207
-1.67580142	-.209498461
-.912949103	.175282444
.0991094375	-.349824698
1.14457216	.0464172582
.438046137	.0177646297
1.22382056	-.162675287



这就是经过 PCA 处理后的样本数据，水平轴（上面举例为 LS 特征）基本上可以代表全部样本点。整个过程看起来就像将坐标系做了旋转，当然二维可以图形化表示，高维就不行了。上面的如果 k=1，那么只会留下这里的水平轴，轴上是所有点在该轴的投影。

这样 PCA 的过程基本结束。在第一步减均值之后，其实应该还有一步对特征做方差归一化。比如一个特征是汽车速度（0 到 100），一个是汽车的座位数（2 到 6），显然第二个的方差比第一个小。因此，如果样本特征中存在这种情况，那么在第一步之后，求每个特征的标准差σ，然后对每个样例在该特征下的数据除以σ。

归纳一下，使用我们之前熟悉的表示方法，在求协方差之前的步骤是：

1. Let $\mu = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}$.
2. Replace each $x^{(i)}$ with $x^{(i)} - \mu$.
3. Let $\sigma_j^2 = \frac{1}{m}\sum_i (x_j^{(i)})^2$
4. Replace each $x_j^{(i)}$ with $x_j^{(i)}/\sigma_j$.

其中$x^{(i)}$是样例，共 m 个，每个样例 n 个特征，也就是说$x^{(i)}$是 n 维向量。$x_j^{(i)}$是第 i 个样例的第 j 个特征。μ是样例均值。σ_j是第 j 个特征的标准差。

整个 PCA 过程貌似及其简单，就是求协方差的特征值和特征向量，然后做数据转换。但是有没有觉得很神奇，为什么求协方差的特征向量就是最理想的 k 维向量？其背后隐藏的意义是什么？整个 PCA 的意义是什么？

3. PCA 理论基础

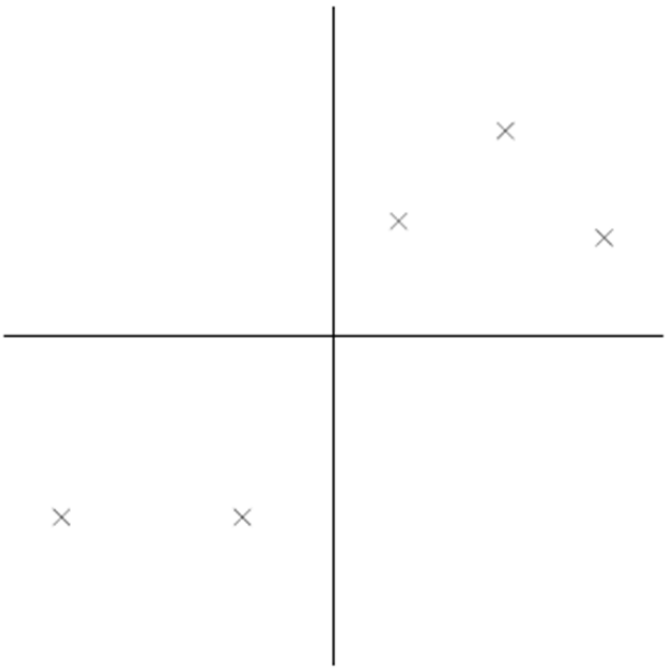
要解释为什么协方差矩阵的特征向量就是 k 维理想特征，我看到的有三个理论：分别是最大方差理论、最小错误理论和坐标轴相关度理论。这里简单探讨前两种，最后一种在讨论 PCA 意义时简单概述。

3.1 最大方差理论

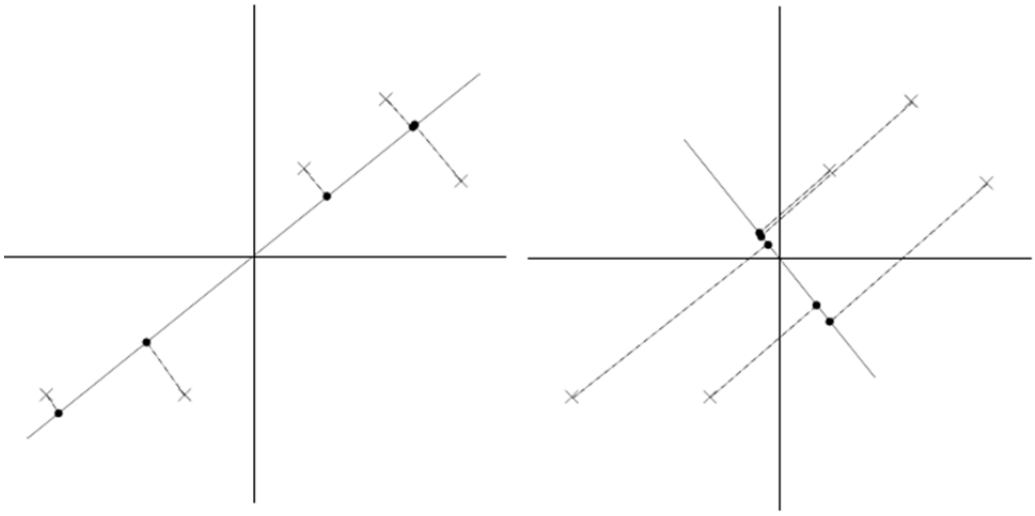
在信号处理中认为信号具有较大的方差，噪声有较小的方差，信噪比就是信号与噪声的方差比，越大越好。如前面的图，样本在横轴上的投影方差较大，在纵轴上的投影方差较小，那么认为纵轴上的投影是由噪声引起的。

因此我们认为，最好的 k 维特征是将 n 维样本点转换为 k 维后，每一维上的样本方差都很大。

比如下图有 5 个样本点：（已经做过预处理，均值为 0，特征方差归一）

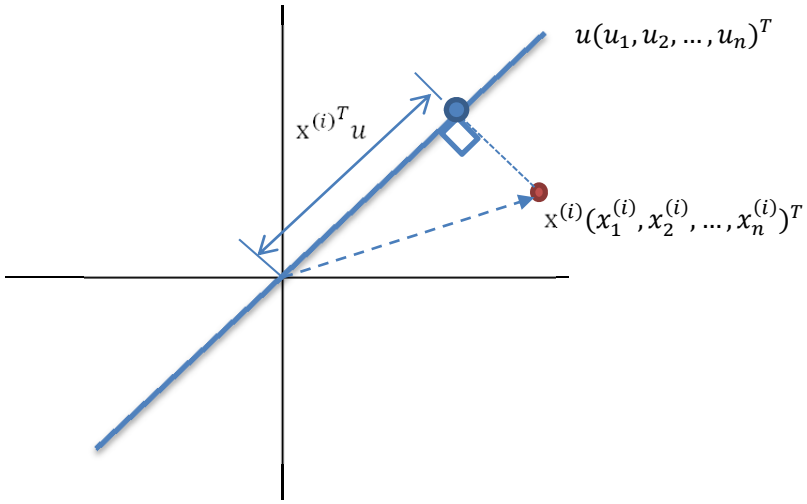


下面将样本投影到某一维上，这里用一条过原点的直线表示（前处理的过程实质是将原点移到样本点的中心点）。



假设我们选择两条不同的直线做投影，那么左右两条中哪个好呢？根据我们之前的方差最大化理论，左边的好，因为投影后的样本点之间方差最大。

这里先解释一下投影的概念：



红色点表示样例$x^{(i)}$，蓝色点表示$x^{(i)}$在 u 上的投影，u 是直线的斜率也是直线的方向向量，而且是单位向量。蓝色点是$x^{(i)}$在 u 上的投影点，离原点的距离是$< x^{(i)}, u >$（即$x^{(i)T}u$或者$u^T x^{(i)}$）由于这些样本点（样例）的每一维特征均值都为 0，因此投影到 u 上的样本点（只有一个到原点的距离值）的均值仍然是 0。

回到上面左右图中的左图，我们要求的是最佳的 u，使得投影后的样本点方差最大。

由于投影后均值为 0，因此方差为：

$$\begin{aligned}\frac{1}{m}\sum_{i=1}^{m}(x^{(i)T}u)^2 &= \frac{1}{m}\sum_{i=1}^{m}u^T x^{(i)}x^{(i)T}u \\ &= u^T\left(\frac{1}{m}\sum_{i=1}^{m}x^{(i)}x^{(i)T}\right)u.\end{aligned}$$

中间那部分很熟悉啊，不就是样本特征的协方差矩阵么（$x^{(i)}$的均值为 0，一般协方差矩阵都除以 m-1，这里用 m）。

用λ来表示$\frac{1}{m}\sum_{i=1}^{m}(x^{(i)T}u)^2$，Σ表示$\frac{1}{m}\sum_{i=1}^{m}x^{(i)}x^{(i)T}$，那么上式写作

$\lambda = u^T\Sigma u$

由于 u 是单位向量，即$u^Tu = 1$，上式两边都左乘 u 得，

$u\lambda = \lambda u = uu^T\Sigma u = \Sigma u$

即$\Sigma u = \lambda u$

We got it！λ就是Σ的特征值，u 是特征向量。最佳的投影直线是特征值λ最大时对应的特征向量，其次是λ第二大对应的特征向量，依次类推。

因此，我们只需要对协方差矩阵进行特征值分解，得到的前 k 大特征值对应的特征向量就是最佳的 k 维新特征，而且这 k 维新特征是正交的。得到前 k 个 u 以后，样例$x^{(i)}$通过以下变换可以得到新的样本。

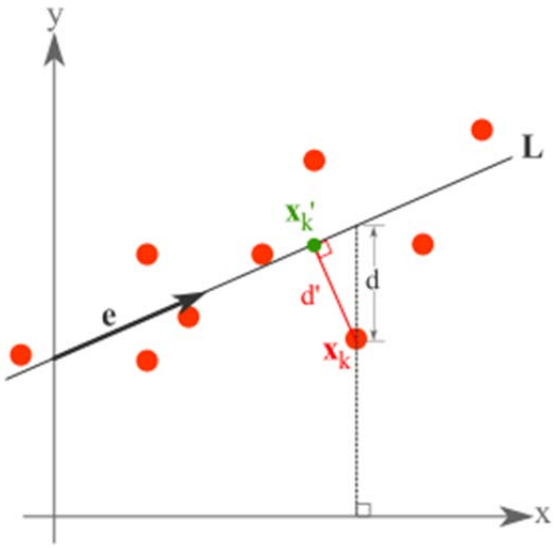
$$y^{(i)} = \begin{bmatrix} u_1^T x^{(i)} \\ u_2^T x^{(i)} \\ \vdots \\ u_k^T x^{(i)} \end{bmatrix} \in \mathbb{R}^k.$$

其中的第 j 维就是$x^{(i)}$在u_j上的投影。

通过选取最大的 k 个 u，使得方差较小的特征（如噪声）被丢弃。

这是其中一种对 PCA 的解释，第二种是错误最小化。

3.2 最小平方误差理论



假设有这样的二维样本点（红色点），回顾我们前面探讨的是求一条直线，使得样本点投影到直线上的点的方差最大。本质是求直线，那么度量直线求的好不好，不仅仅只有方差最大化的方法。再回想我们最开始学习的线性回归等，目的也是求一个线性函数使得直线能够最佳拟合样本点，那么我们能不能认为最佳的直线就是回归后的直线呢？回归时我们的最小二乘法度量的是样本点到直线的坐标轴距离。比如这个问题中，特征是 x，类标签是 y。回归时最小二乘法度量的是距离 d。如果使用回归方法来度量最佳直线，那么就是直接在原始样本上做回归了，跟特征选择就没什么关系了。

因此，我们打算选用另外一种评价直线好坏的方法，使用点到直线的距离 d'来度量。

现在有 n 个样本点$(x_1, x_2, \ldots, x_n)$，每个样本点为 m 维（这节内容中使用的符号与上面的不太一致，需要重新理解符号的意义）。将样本点x_k在直线上的投影记为x_k'，那么我们就是要最小化

$$\sum_{k=1}^{n} ||(x_k' - x_k)||^2$$

这个公式称作最小平方误差（Least Squared Error）。

而确定一条直线，一般只需要确定一个点，并且确定方向即可。

第一步确定点：

假设要在空间中找一点x_0来代表这 n 个样本点，“代表”这个词不是量化的，因此要量化的话，我们就是要找一个 m 维的点x_0，使得

$$J_0(\boldsymbol{x}_0) = \sum_{k=1}^{n} \|\boldsymbol{x}_0 - \boldsymbol{x}_k\|^2, \qquad (1)$$

最小。其中$J_0(x_0)$是平方错误评价函数（squared-error criterion function），假设 **m** 为 n 个样本点的均值：

$$\boldsymbol{m} = \frac{1}{n}\sum_{k=1}^{n} \boldsymbol{x}_k, \qquad (2)$$

那么平方错误可以写作：

$$\begin{aligned} J_0(\boldsymbol{x}_0) &= \sum_{k=1}^{n} \|(\boldsymbol{x}_0 - \boldsymbol{m}) - (\boldsymbol{x}_k - \boldsymbol{m})\|^2 \\ &= \sum_{k=1}^{n} \|\boldsymbol{x}_0 - \boldsymbol{m}\|^2 - 2\sum_{k=1}^{n}(\boldsymbol{x}_0 - \boldsymbol{m})^t(\boldsymbol{x}_k - \boldsymbol{m}) + \sum_{k=1}^{n} \|\boldsymbol{x}_k - \boldsymbol{m}\|^2 \\ &= \sum_{k=1}^{n} \|\boldsymbol{x}_0 - \boldsymbol{m}\|^2 - 2(\boldsymbol{x}_0 - \boldsymbol{m})^t\sum_{k=1}^{n}(\boldsymbol{x}_k - \boldsymbol{m}) + \sum_{k=1}^{n} \|\boldsymbol{x}_k - \boldsymbol{m}\|^2 \\ &= \sum_{k=1}^{n} \|\boldsymbol{x}_0 - \boldsymbol{m}\|^2 + \underbrace{\sum_{k=1}^{n} \|\boldsymbol{x}_k - \boldsymbol{m}\|^2}_{independent of \boldsymbol{x}_0}. \end{aligned} \qquad (3)$$

后项与x_0无关，看做常量，而$J_0(x_0) \geq 0$，因此最小化$J_0(x_0)$时，

$$x_0 = \mathbf{m}$$

x_0是样本点均值。

第一步确定方向：

我们从x_0拉出要求的直线（这条直线要过点 **m**），假设直线的方向是单位向量 **e**。那么直线上任意一点，比如x'_k就可以用点 **m** 和 **e** 来表示

$$x'_k = \mathbf{m} + a_k\boldsymbol{e}$$

其中a_k是x'_k到点 **m** 的距离。

我们重新定义最小平方误差：

$$J_1(a_1, \ldots, a_n, \boldsymbol{e}) = \sum_{k=1}^{n} \|(\boldsymbol{x}'_k - \boldsymbol{x}_k)\|^2 = \sum_{k=1}^{n} \|((\boldsymbol{m} + a_k\boldsymbol{e}) - \boldsymbol{x}_k)\|^2, \qquad (5)$$

这里的 k 只是相当于 *i*。J_1就是最小平方误差函数，其中的未知参数是$a_1, a_2, \ldots, a_n$和 **e**。

实际上是求J_1的最小值。首先将上式展开：

$$J_1(a_1,\ldots,a_n,e)=\sum_{k=1}^{n}\|((\boldsymbol{m}+a_k\boldsymbol{e})-\boldsymbol{x}_k)\|^2=\sum_{k=1}^{n}\|(a_k\boldsymbol{e}-(\boldsymbol{x}_k-\boldsymbol{m}))\|^2$$
$$=\sum_{k=1}^{n}a_k^2\|\boldsymbol{e}\|^2-2\sum_{k=1}^{n}a_k\boldsymbol{e}^t(\boldsymbol{x}_k-\boldsymbol{m})+\sum_{k=1}^{n}\|\boldsymbol{x}_k-\boldsymbol{m}\|^2. \quad (6)$$

我们首先固定 **e**，将其看做是常量，$||e||^2=1$，然后对a_k进行求导，得

$$a_k=\boldsymbol{e}^t(\boldsymbol{x}_k-\boldsymbol{m}). \quad (8)$$

这个结果意思是说，如果知道了 **e**，那么将$x_k'-\mathbf{m}$与 **e** 做内积，就可以知道了x_k在 **e** 上的投影离 **m** 的长度距离，不过这个结果不用求都知道。

然后是固定a_k，对 **e** 求偏导数，我们先将公式（8）代入J_1，得

$$J_1(e)=\sum_{k=1}^{n}a_k^2\|\boldsymbol{e}\|^2-2\sum_{k=1}^{n}a_k^2+\sum_{k=1}^{n}\|\boldsymbol{x}_k-\boldsymbol{m}\|^2$$
$$=-\sum_{k=1}^{n}[\boldsymbol{e}^t(\boldsymbol{x}_k-m)]^2+\sum_{k=1}^{n}\|\boldsymbol{x}_k-\boldsymbol{m}\|^2$$
$$=-\sum_{k=1}^{n}\boldsymbol{e}^t(\boldsymbol{x}_k-\boldsymbol{m})(\boldsymbol{x}_k-\boldsymbol{m})^t\boldsymbol{e}+\sum_{k=1}^{n}\|\boldsymbol{x}_k-\boldsymbol{m}\|^2$$
$$=-\boldsymbol{e}^t\boldsymbol{S}\boldsymbol{e}+\sum_{k=1}^{n}\|\boldsymbol{x}_k-\boldsymbol{m}\|^2. \quad (9)$$

其中$\boldsymbol{S}=\sum_{k=1}^{n}\boldsymbol{e}^t(\boldsymbol{x}_k-\boldsymbol{m})(\boldsymbol{x}_k-\boldsymbol{m})^t\boldsymbol{e}$，与协方差矩阵类似，只是缺少个分母 n-1，我们称之为**散列矩阵**（scatter matrix）。

然后可以对 **e** 求偏导数，但是 **e** 需要首先满足$||e||^2=1$，引入拉格朗日乘子λ，来使$e^T Se$最大（J_1最小），令

$$u=\boldsymbol{e}^t\boldsymbol{S}\boldsymbol{e}-\lambda(\boldsymbol{e}^t\boldsymbol{e}-1) \quad (10)$$

求偏导

$$\frac{\partial u}{\partial \boldsymbol{e}}=2\boldsymbol{S}\boldsymbol{e}-2\lambda\boldsymbol{e}, \quad (11)$$

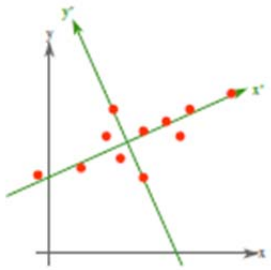
这里存在对向量求导数的技巧，方法这里不多做介绍。可以去看一些关于矩阵微积分的资料，这里求导时可以将$e^T Se$看作是Se^2，将$e^T e$看做是e^2。

导数等于 0 时，得

$$\boldsymbol{S}\boldsymbol{e}=\lambda\boldsymbol{e}. \quad (12)$$

两边除以 n-1 就变成了，对协方差矩阵求特征值向量了。

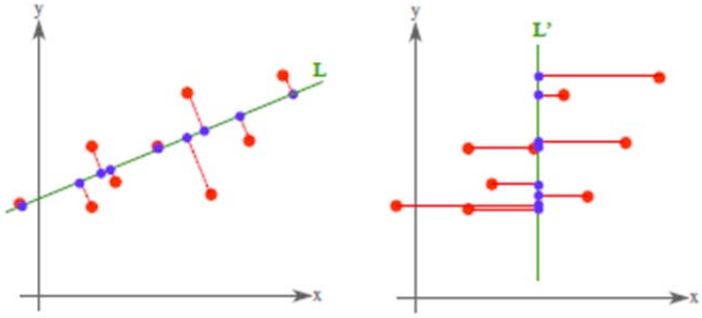
从不同的思路出发，最后得到同一个结果，对协方差矩阵求特征向量，求得后特征向量上就成为了新的坐标，如下图：



这时候点都聚集在新的坐标轴周围，因为我们使用的最小平方误差的意义就在此。

4. PCA 理论意义

PCA 将 n 个特征降维到 k 个，可以用来进行数据压缩，如果 100 维的向量最后可以用 10 维来表示，那么压缩率为 90%。同样图像处理领域的 KL 变换使用 PCA 做图像压缩。但 PCA 要保证降维后，还要保证数据的特性损失最小。再看回顾一下 PCA 的效果。经过 PCA 处理后，二维数据投影到一维上可以有以下几种情况：



我们认为左图好，一方面是投影后方差最大，一方面是点到直线的距离平方和最小，而且直线过样本点的中心点。为什么右边的投影效果比较差？直觉是因为坐标轴之间相关，以至于去掉一个坐标轴，就会使得坐标点无法被单独一个坐标轴确定。

PCA 得到的 k 个坐标轴实际上是 k 个特征向量，由于协方差矩阵对称，因此 k 个特征向量正交。看下面的计算过程。

假设我们还是用$\mathrm{x}^{(i)}(x_1^{(i)}, x_2^{(i)}, \ldots, x_n^{(i)})^T$来表示样例，m 个样例，n 个特征。特征向量为 **e**，$e_1^{(i)}$表示第 i 个特征向量的第 1 维。那么原始样本特征方程可以用下面式子来表示：

前面两个矩阵乘积就是协方差矩阵Σ（除以 m 后），原始的样本矩阵 A 是第二个矩阵 m*n。

$$\begin{bmatrix} | & | & | & | \\ \mathrm{x}^{(1)} & \mathrm{x}^{(2)} & \ldots & \mathrm{x}^{(m)} \\ | & | & | & | \end{bmatrix} \begin{bmatrix} - & \mathrm{x}^{(1)T} & - \\ - & \mathrm{x}^{(2)T} & - \\ - & \vdots & - \\ - & \mathrm{x}^{(m)T} & - \end{bmatrix} \begin{bmatrix} e_1^{(i)} \\ e_2^{(i)} \\ \vdots \\ e_n^{(i)} \end{bmatrix} = \lambda_i \begin{bmatrix} e_1^{(i)} \\ e_2^{(i)} \\ \vdots \\ e_n^{(i)} \end{bmatrix}$$

上式可以简写为$\mathrm{A}^T Ae = \lambda e$

我们最后得到的投影结果是AE，E 是 k 个特征向量组成的矩阵，展开如下：

$$\begin{bmatrix} - & \mathrm{x}^{(1)T} & - \\ - & \mathrm{x}^{(2)T} & - \\ - & \vdots & - \\ - & \mathrm{x}^{(m)T} & - \end{bmatrix}\begin{bmatrix} e_1^{(1)} & e_1^{(2)} & \dots & e_1^{(k)} \\ e_2^{(1)} & e_2^{(2)} & \dots & e_2^{(k)} \\ \vdots & \vdots & \ddots & \vdots \\ e_n^{(1)} & e_n^{(2)} & \dots & e_n^{(k)} \end{bmatrix}$$

得到的新的样例矩阵就是 m 个样例到 k 个特征向量的投影，也是这 k 个特征向量的线性组合。e 之间是正交的。从矩阵乘法中可以看出，PCA 所做的变换是将原始样本点（n 维），投影到 k 个正交的坐标系中去，丢弃其他维度的信息。举个例子，假设宇宙是 n 维的（霍金说是 13 维的），我们得到银河系中每个星星的坐标（相对于银河系中心的 n 维向量），然而我们想用二维坐标去逼近这些样本点，假设算出来的协方差矩阵的特征向量分别是图中的水平和竖直方向，那么我们建议以银河系中心为原点的 x 和 y 坐标轴，所有的星星都投影到 x 和 y 上，得到下面的图片。然而我们丢弃了每个星星离我们的远近距离等信息。

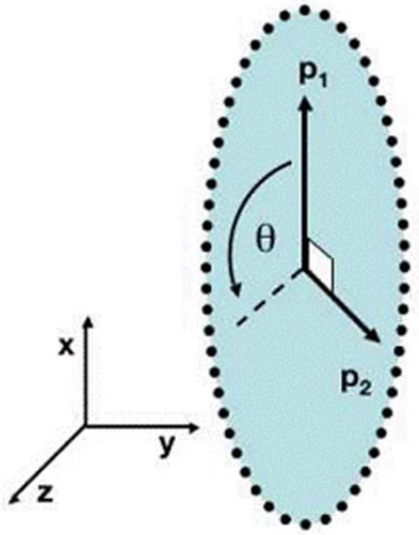
5. 总结与讨论

这一部分来自 http://www.cad.zju.edu.cn/home/chenlu/pca.htm

- PCA 技术的一大好处是对数据进行降维的处理。我们可以对新求出的“主元”向量的重要性进行排序，根据需要取前面最重要的部分，将后面的维数省去，可以达到降维从而简化模型或是对数据进行压缩的效果。同时最大程度的保持了原有数据的信息。

- PCA 技术的一个很大的优点是，它是完全无参数限制的。在 PCA 的计算过程中完全不需要人为的设定参数或是根据任何经验模型对计算进行干预，最后的结果只与数据相关，与用户是独立的。

  但是，这一点同时也可以看作是缺点。如果用户对观测对象有一定的先验知识，掌握了数据的一些特征，却无法通过参数化等方法对处理过程进行干预，可能会得不到预期的效果，效率也不高。



图表 4：黑色点表示采样数据，排列成转盘的形状。

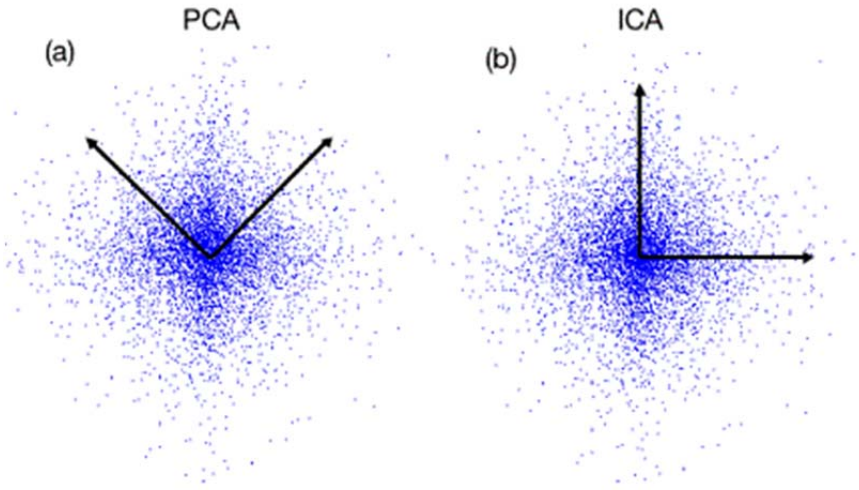
容易想象，该数据的主元是(P_1, P_2)或是旋转角θ。

如图表 4 中的例子，PCA 找出的主元将是(P_1, P_2)。但是这显然不是最优和最简化的主元。(P_1, P_2)之间存在着非线性的关系。根据先验的知识可知旋转角θ是最优的主元（类比极坐标）。则在这种情况下，PCA 就会失效。但是，如果加入先验的知识，对数据进行某种划归，就可以将数据转化为以θ为线性的空间中。这类根据先验知识对数据预先进行非线性转换的方法就成为 *kernel*-PCA，它扩展了 PCA 能够处理的问题的范围，又可以结合一些先验约束，是比较流行的方法。

- 有时数据的分布并不是满足高斯分布。如图表 5 所示，在非高斯分布的情况下，PCA 方法得出的主元可能并不是最优的。在寻找主元时不能将方差作为衡量重要性的标准。要根据数据的分布情况选择合适的描述完全分布的变量，然后根据概率分布式

$$P(y_1, y_2) = P(y_1)P(y_2)$$

来计算两个向量上数据分布的相关性。等价的，保持主元间的正交假设，寻找的主元同样要使$P(y_1, y_2) = 0$。这一类方法被称为独立主元分解(ICA)。



图表 5：数据的分布并不满足高斯分布，呈明显的十字星状。
这种情况下，方差最大的方向并不是最优主元方向。

另外 PCA 还可以用于预测矩阵中缺失的元素。

6. 其他参考文献

A *tutorial* on Principal Components Analysis LI Smith – 2002
A *Tutorial* on Principal Component Analysis J Shlens
http://www.cmlab.csie.ntu.edu.tw/~cyy/learning/tutorials/PCAMissingData.pdf
http://www.cad.zju.edu.cn/home/chenlu/pca.htm

独立成分分析（Independent Component Analysis）

JerryLead
csxulijie@gmail.com

1. 问题：

1、上节提到的 PCA 是一种数据降维的方法，但是只对符合高斯分布的样本点比较有效，那么对于其他分布的样本，有没有主元分解的方法呢？

2、经典的鸡尾酒宴会问题（cocktail party problem）。假设在 party 中有 n 个人，他们可以同时说话，我们也在房间中一些角落里共放置了 n 个声音接收器（Microphone）用来记录声音。宴会过后，我们从 n 个麦克风中得到了一组数据$\{\mathrm{x}^{(i)}(x_1^{(i)}, x_2^{(i)}, \dots, x_n^{(i)}); i = 1, \dots, m\}$，i 表示采样的时间顺序，也就是说共得到了 m 组采样，每一组采样都是 n 维的。我们的目标是单单从这 m 组采样数据中分辨出每个人说话的信号。

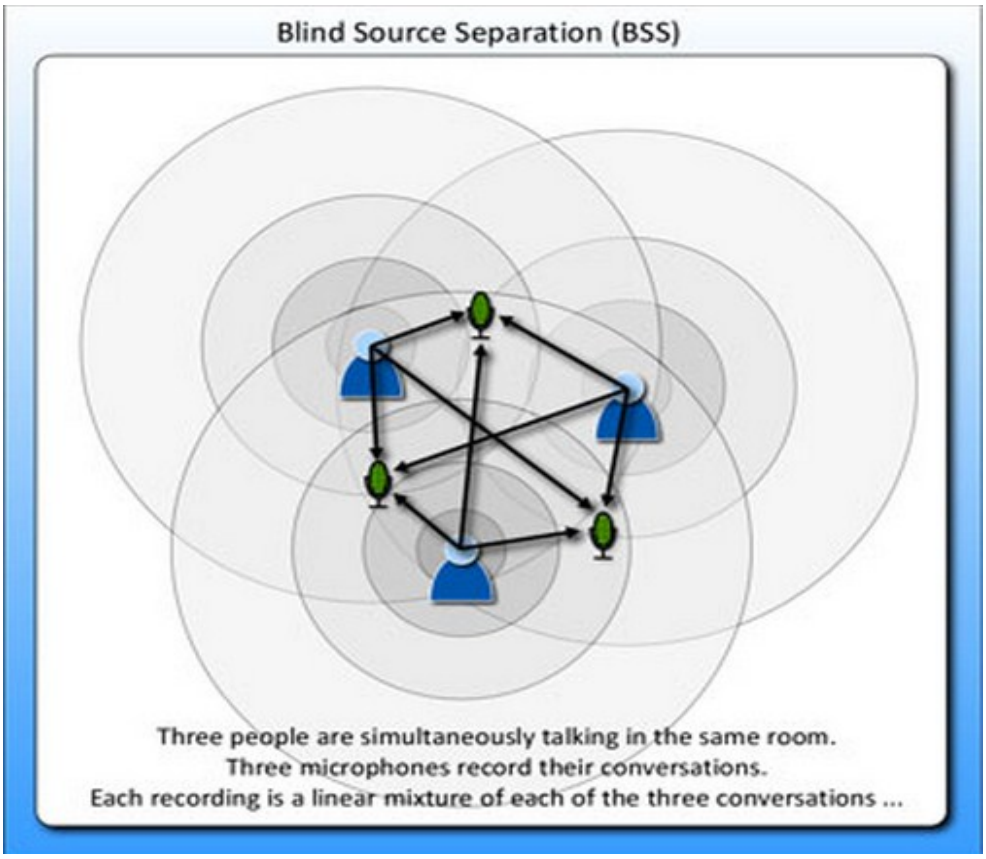
将第二个问题细化一下，有 n 个信号源s $(s_1, s_2, \dots, s_n)^T$，$\mathrm{s} \in \mathbb{R}^n$，每一维都是一个人的声音信号，每个人发出的声音信号独立。A 是一个未知的混合矩阵（mixing matrix），用来组合叠加信号 s，那么

$$\mathrm{x} = \mathrm{A}s$$

x 的意义在上文解释过，这里的 x 不是一个向量，是一个矩阵。其中每个列向量是$\mathrm{x}^{(i)}$，

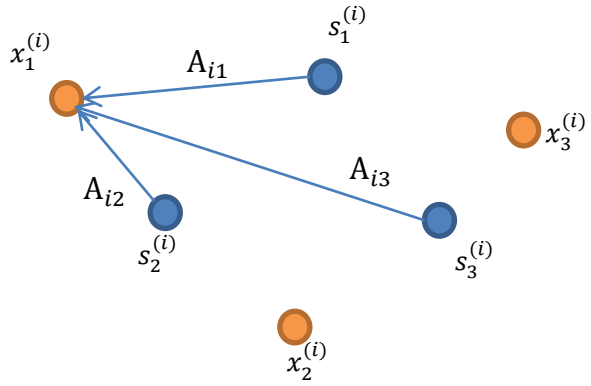
$$x^{(i)} = As^{(i)}$$

表示成图就是



这张图来自

http://amouraux.webnode.com/research-interests/research-interests-erp-analysis/blind-source-separation-bss-of-erps-using-independent-component-analysis-ica/



$x^{(i)}$的每个分量都由$s^{(i)}$的分量线性表示。A 和 s 都是未知的，x 是已知的，我们要想办法根据 x 来推出 s。这个过程也称作为盲信号分离。

令$W = A^{-1}$，那么

$$s^{(i)} = A^{-1}x^{(i)} = Wx^{(i)}$$

将 W 表示成

$$W = \begin{bmatrix} — & w_1^T & — \\ & \vdots & \\ — & w_n^T & — \end{bmatrix}.$$

其中$w_i \in \mathbb{R}^n$，其实就是将w_i写成行向量形式。那么得到：

$$s_j^{(i)} = w_j^T x^{(i)}$$

2. ICA 的不确定性（ICA ambiguities）

由于 w 和 s 都不确定，那么在没有先验知识的情况下，无法同时确定这两个相关参数。比如上面的公式 s=wx。当 w 扩大两倍时，s 只需要同时扩大两倍即可，等式仍然满足，因此无法得到唯一的 s。同时如果将人的编号打乱，变成另外一个顺序，如上图的蓝色节点的编号变为 3,2,1，那么只需要调换 A 的列向量顺序即可，因此也无法单独确定 s。这两种情况称为原信号不确定。

还有一种 ICA 不适用的情况，那就是信号不能是高斯分布的。假设只有两个人发出的声音信号符合多值正态分布，$s \sim N(0, I)$，I 是 2*2 的单位矩阵，s 的概率密度函数就不用说了吧，以均值 0 为中心，投影面是椭圆的山峰状（参见多值高斯分布）。因为$x = As$，因此，x 也是高斯分布的，均值为 0，协方差为$E[xx^T] = E[Ass^TA^T] = AA^T$。

令 R 是正交阵$(RR^T = R^TR = I)$，$A' = AR$。如果将 A 替换成 A'。那么$x' = A's$。s 分布没变，因此 x'仍然是均值为 0，协方差$E[x'(x')^T] = E[A'ss^T(A')^T] = E[ARss^T(AR)^T] = ARR^TA^T = AA^T$。

因此，不管混合矩阵是 A 还是 A'，x 的分布情况是一样的，那么就无法确定混合矩阵，也就无法确定原信号。

3. 密度函数和线性变换

在讨论 ICA 具体算法之前，我们先来回顾一下概率和线性代数里的知识。

假设我们的随机变量 s 有概率密度函数$p_s(s)$（连续值是概率密度函数，离散值是概率）。为了简单，我们再假设 s 是实数，还有一个随机变量 x=As，A 和 x 都是实数。令p_x是 x 的概率密度，那么怎么求p_x？

令$W = A^{-1}$，首先将式子变换成$s = Wx$，然后得到$p_x(x) = p_s(Ws)$，求解完毕。可惜这种方法是错误的。比如 s 符合均匀分布的话（$s \sim \text{Uniform}[0,1]$），那么 s 的概率密度是$p_s(s) = 1\{0 \le s \le 1\}$，现在令 A=2，即 x=2s，也就是说 x 在[0,2]上均匀分布，可知$p_x(x) = 0.5$。然而，前面的推导会得到$p_x(x) = p_s(0.5s) = 1$。正确的公式应该是

$$p_x(x) = p_s(Wx)|W|$$

推导方法

$$F_X(x) = P(X \le x) = P(AS \le x) = P(S \le Wx) = F_S(Wx)$$

$$p_x(x) = F_X'(x) = F_S'(Wx) = p_s(Wx)|W|$$

更一般地，如果 s 是向量，A 可逆的方阵，那么上式子仍然成立。

4. ICA 算法

ICA 算法归功于 Bell 和 Sejnowski，这里使用最大似然估计来解释算法，原始的论文中使用的是一个复杂的方法 Infomax principal。

我们假定每个s_i有概率密度p_s，那么给定时刻原信号的联合分布就是

$$\mathrm{p(s)} = \prod_{i=1}^{n} p_s(s_i)$$

这个公式代表一个假设前提：每个人发出的声音信号各自独立。有了 p(s)，我们可以求得 p(x)

$$\mathrm{p(x)} = \mathrm{p}_s(Wx)|\mathrm{W}| = |\mathrm{W}| \prod_{i=1}^{n} p_s({w_i}^T x)$$

左边是每个采样信号 x（n 维向量）的概率，右边是每个原信号概率的乘积的|W|倍。

前面提到过，如果没有先验知识，我们无法求得 W 和 s。因此我们需要知道$p_s(s_i)$，我们打算选取一个概率密度函数赋给 s，但是我们不能选取高斯分布的密度函数。在概率论里我们知道密度函数 p(x)由累计分布函数（cdf）F(x)求导得到。F(x)要满足两个性质是：单调递增和在[0,1]。我们发现 sigmoid 函数很适合，定义域负无穷到正无穷，值域 0 到 1，缓慢递增。我们假定 s 的累积分布函数符合 sigmoid 函数

$$\mathrm{g(s)} = \frac{1}{1+e^{-s}}$$

求导后

$$p_s(s) = g'(s) = \frac{e^s}{(1+e^s)^2}$$

这就是 s 的密度函数。这里 s 是实数。

如果我们预先知道 s 的分布函数，那就不用假设了，但是在缺失的情况下，sigmoid 函数能够在大多数问题上取得不错的效果。由于上式中$p_s(s)$是个对称函数，因此 E[s]=0（s 的均值为 0），那么 E[x]=E[As]=0，x 的均值也是 0。

知道了$p_s(s)$，就剩下 W 了。给定采样后的训练样本$\{\mathrm{x}^{(i)}(x_1^{(i)}, x_2^{(i)}, \dots, x_n^{(i)}); i = 1, \dots, m\}$，样本对数似然估计如下：

使用前面得到的 x 的概率密度函数，得

$$\ell(W) = \sum_{i=1}^{m} \left(\sum_{j=1}^{n} \log g'(w_j^T x^{(i)}) + \log |W| \right).$$

大括号里面是$\mathrm{p}(\mathrm{x}^{(i)})$。

接下来就是对 W 求导了，这里牵涉一个问题是对行列式|W|进行求导的方法，属于矩阵微积分。这里先给出结果，在文章最后再给出推导公式。

$$\nabla_W |W| = |W|(W^{-1})^T$$

最终得到的求导后公式如下，$\log g'(s)$的导数为$1 - 2g(s)$（可以自己验证）：

$$W := W + \alpha \left(\begin{bmatrix} 1 - 2g(w_1^T x^{(i)}) \\ 1 - 2g(w_2^T x^{(i)}) \\ \vdots \\ 1 - 2g(w_n^T x^{(i)}) \end{bmatrix} x^{(i)^T} + (W^T)^{-1} \right),$$

其中α是梯度上升速率，人为指定。

当迭代求出 W 后，便可得到$s^{(i)} = Wx^{(i)}$来还原出原始信号。

注意：我们计算最大似然估计时，假设了$x^{(i)}$与$x^{(j)}$之间是独立的，然而对于语音信号或者其他具有时间连续依赖特性（比如温度）上，这个假设不能成立。但是在数据足够多时，假设独立对效果影响不大，同时如果事先打乱样例，并运行随机梯度上升算法，那么能够加快收敛速度。

回顾一下鸡尾酒宴会问题，s 是人发出的信号，是连续值，不同时间点的 s 不同，每个人发出的信号之间独立（s_i和s_j之间独立）。s 的累计概率分布函数是 sigmoid 函数，但是所有人发出声音信号都符合这个分布。A（W 的逆阵）代表了 s 相对于 x 的位置变化，x 是 s 和 A 变化后的结果。

5. 实例

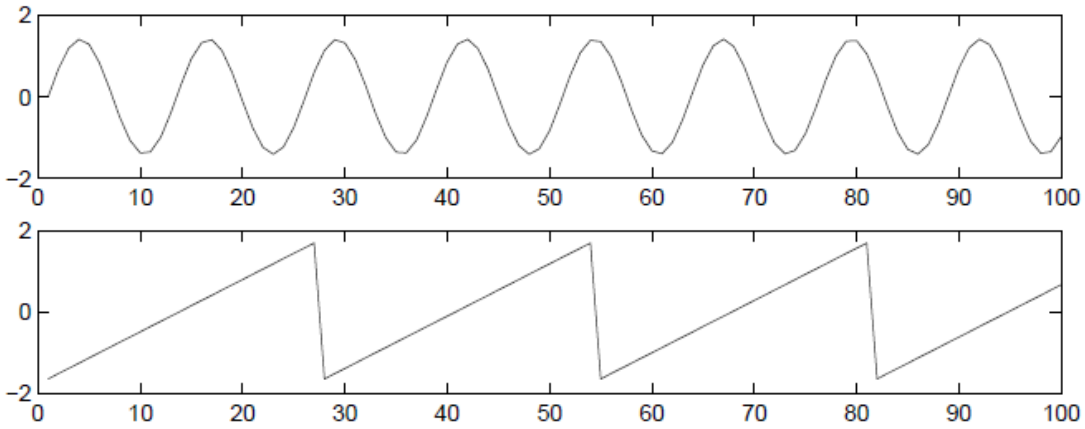


Figure 1: The original signals.

s=2 时的原始信号

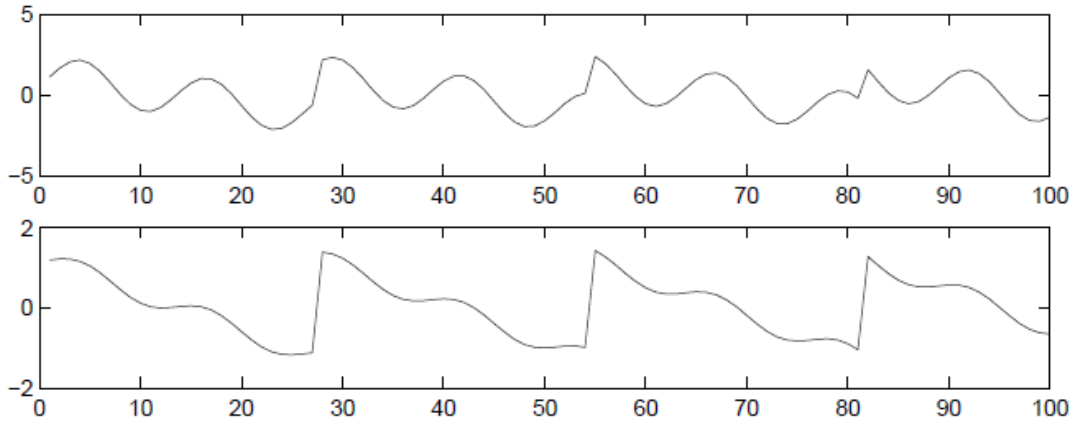
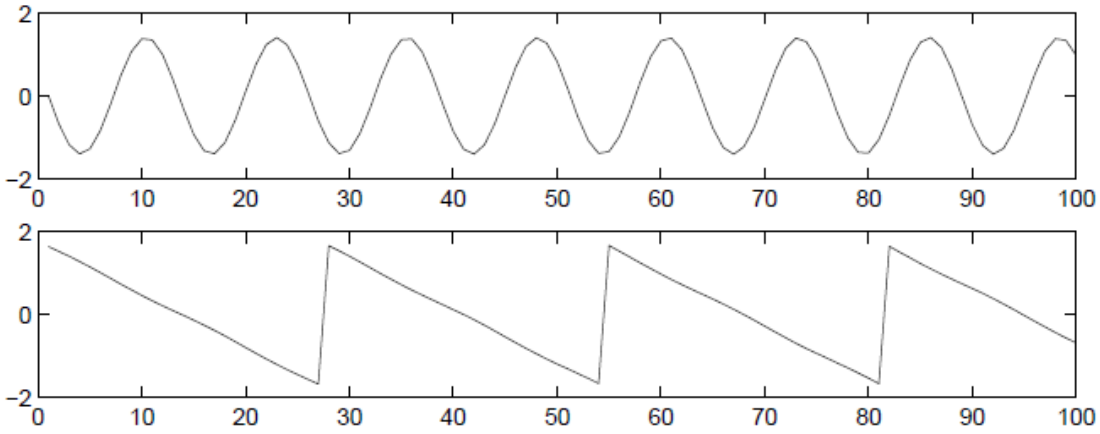


Figure 2: The observed mixtures of the source signals in Fig. 1.

观察到的 x 信号



使用 ICA 还原后的 s 信号

6. 行列式的梯度

对行列式求导，设矩阵 A 是 n×n 的，我们知道行列式与代数余子式有关，

$$|A| = \sum_{i=1}^{n} (-1)^{i+j} A_{ij} |A_{\setminus i,\setminus j}| \quad \text{(for any } j \in 1, \ldots, n)$$

$A_{\setminus i,\setminus j}$是去掉第 i 行第 j 列后的余子式，那么对$A_{k,l}$求导得

$$\frac{\partial}{\partial A_{k\ell}} |A| = \frac{\partial}{\partial A_{k\ell}} \sum_{i=1}^{n} (-1)^{i+j} A_{ij} |A_{\setminus i,\setminus j}| = (-1)^{k+\ell} |A_{\setminus k,\setminus \ell}| = (\operatorname{adj}(A))_{\ell k}.$$

adj(A)跟我们线性代数中学的A^*是一个意思，因此

$$\nabla_A |A| = (\operatorname{adj}(A))^T = |A| A^{-T}.$$

7. ICA 算法扩展描述

上面介绍的内容基本上是讲义上的，与我看的另一篇《Independent Component Analysis: Algorithms and Applications》（Aapo Hyvärinen and Erkki Oja）有点出入。下面总结一下这篇文章里提到的一些内容（有些我也没看明白）。

首先里面提到了一个与“独立”相似的概念“不相关（uncorrelated）”。Uncorrelated 属于部分独立，而不是完全独立，怎么刻画呢？

如果随机变量y_1和y_2是独立的，当且仅当$p(y_1, y_2) = p(y_1)p(y_2)$。

如果随机变量y_1和y_2是不相关的，当且仅当$E(y_1, y_2) = E(y_1)E(y_2)$

第二个不相关的条件要比第一个独立的条件“松”一些。因为独立能推出不相关，不相关推不出独立。

证明如下：

$$p_1(y_1) = \int p(y_1, y_2)dy_2,$$

$$p(y_1, y_2) = p_1(y_1)p_2(y_2).$$

$$E\{h_1(y_1)h_2(y_2)\} = \int\int h_1(y_1)h_2(y_2)p(y_1, y_2)dy_1dy_2$$
$$= \int\int h_1(y_1)p_1(y_1)h_2(y_2)p_2(y_2)dy_1dy_2 = \int h_1(y_1)p_1(y_1)dy_1 \int h_2(y_2)p_2(y_2)dy_2$$
$$= E\{h_1(y_1)\}E\{h_2(y_2)\}.$$

反过来不能推出。

比如，y_1和y_2的联合分布如下(0,1)，(0,-1)，(1,0)，(-1,0)。

$$E(y_1, y_2) = E(y_1)E(y_2) = 0$$

因此y_1和y_2不相关，但是

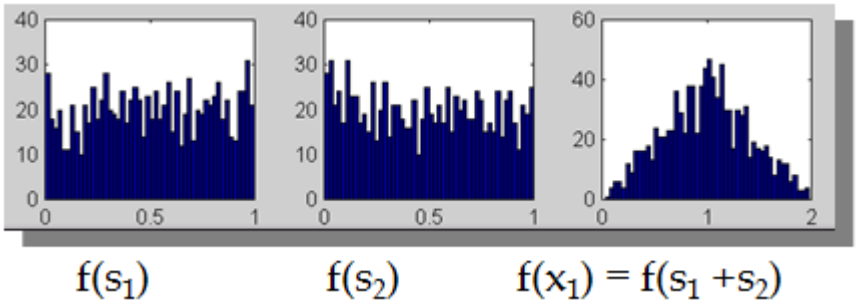
$$\mathrm{E}(y_1^2 y_1^2) = 0 \neq \frac{1}{4} = E(y_1^2)E(y_2^2)$$

因此y_1和y_2不满足上面的积分公式，y_1和y_2不是独立的。

上面提到过，如果$s^{(i)}$是高斯分布的，A 是正交的，那么$x^{(i)}$也是高斯分布的，且$x^{(i)}$与$x^{(j)}$之间是独立的。那么无法确定 A，因为任何正交变换都可以让$x^{(i)}$达到同分布的效果。但是如果$s^{(i)}$中只有一个分量是高斯分布的，仍然可以使用 ICA。

那么 ICA 要解决的问题变为：如何从 x 中推出 s，使得 s 最不可能满足高斯分布？

中心极限定理告诉我们：大量独立同分布随机变量之和满足高斯分布。



我们一直假设的是$x^{(i)}$是由独立同分布的主元$s^{(i)}$经过混合矩阵 A 生成。那么为了求$s^{(i)}$，我们需要计算$s^{(i)}$的每个分量$y_j^{(i)} = w_j^T x^{(i)}$。定义$z_j = A^T w_j$，那么$y_j^{(i)} = w_j^T x^{(i)} = w_j^T A s^{(i)} = z_j^T s^{(i)}$，之所以这么麻烦再定义 z 是想说明一个关系，我们想通过整出一个w_j来对$x^{(i)}$进行线性组合，得出 y。而我们不知道得出的 y 是否是真正的 s 的分量，但我们知道 y 是 s 的真正分量的线性组合。由于我们不能使 s 的分量成为高斯分布，因此我们的目标求是让 y（也就是$w_j^T x^{(i)}$）最不可能是高斯分布时的 w。

那么问题递归到如何度量 y 是否是高斯分布的了。

一种度量方法是 kurtosis 方法，公式如下：

$$\mathrm{kurt}(y) = E\{y^4\} - 3(E\{y^2\})^2$$

如果 y 是高斯分布，那么该函数值为 0，否则绝大多数情况下值不为 0。

但这种度量方法不怎么好，有很多问题。看下一种方法：

负熵（Negentropy）度量方法。

我们在信息论里面知道对于离散的随机变量 Y，其熵是

$$H(Y) = -\sum_i P(Y = a_i)\log P(Y = a_i)$$

连续值时是

$$H(\mathbf{y}) = -\int f(\mathbf{y})\log f(\mathbf{y})\mathrm{d}\mathbf{y}.$$

在信息论里有一个强有力的结论是：高斯分布的随机变量是同方差分布中熵最大的。也就是说对于一个随机变量来说，满足高斯分布时，最随机。

定义负熵的计算公式如下：

$$J(\mathbf{y}) = H(\mathbf{y}_{gauss}) - H(\mathbf{y})$$

也就是随机变量 y 相对于高斯分布时的熵差，这个公式的问题就是直接计算时较为复杂，一般采用逼近策略。

$$J(y) \approx \frac{1}{12}E\{y^3\}^2 + \frac{1}{48}\mathrm{kurt}(y)^2$$

这种逼近策略不够好，作者提出了基于最大熵的更优的公式：

$$J(y) \approx \sum_{i=1}^{p} k_i[E\{G_i(y)\} - E\{G_i(\nu)\}]^2,$$

之后的 FastICA 就基于这个公式。

另外一种度量方法是最小互信息方法：

$$I(y_1, y_2, \ldots, y_m) = \sum_{i=1}^{m} H(y_i) - H(\mathbf{y}).$$

这个公式可以这样解释，前一个 H 是y_i的编码长度（以信息编码的方式理解），第二个 H 是 y 成为随机变量时的平均编码长度。之后的内容包括 FastICA 就不再介绍了，我也没看懂。

8. ICA 的投影追踪解释（Projection Pursuit）

投影追踪在统计学中的意思是去寻找多维数据的“interesting”投影。这些投影可用在数据可视化、密度估计和回归中。比如在一维的投影追踪中，我们寻找一条直线，使得所有的数据点投影到直线上后，能够反映出数据的分布。然而我们最不想要的是高斯分布，最不像高斯分布的数据点最 interesting。这个与我们的 ICA 思想是一直的，寻找独立的最不可能是高斯分布的 s。

在下图中，主元是纵轴，拥有最大的方差，但最 interesting 的是横轴，因为它可以将两个类分开（信号分离）。

9. ICA 算法的前处理步骤

1、**中心化：**也就是求 x 均值，然后让所有 x 减去均值，这一步与 PCA 一致。

2、**漂白：**目的是将 x 乘以一个矩阵变成$\tilde{x}$，使得$\tilde{x}$的协方差矩阵是I。解释一下吧，原始的向量是 x。转换后的是$\tilde{x}$。

$\tilde{x}$的协方差矩阵是I，即

$$E\{\tilde{\mathbf{x}}\tilde{\mathbf{x}}^T\} = \mathbf{I}$$

我们只需用下面的变换，就可以从 x 得到想要的$\tilde{x}$。

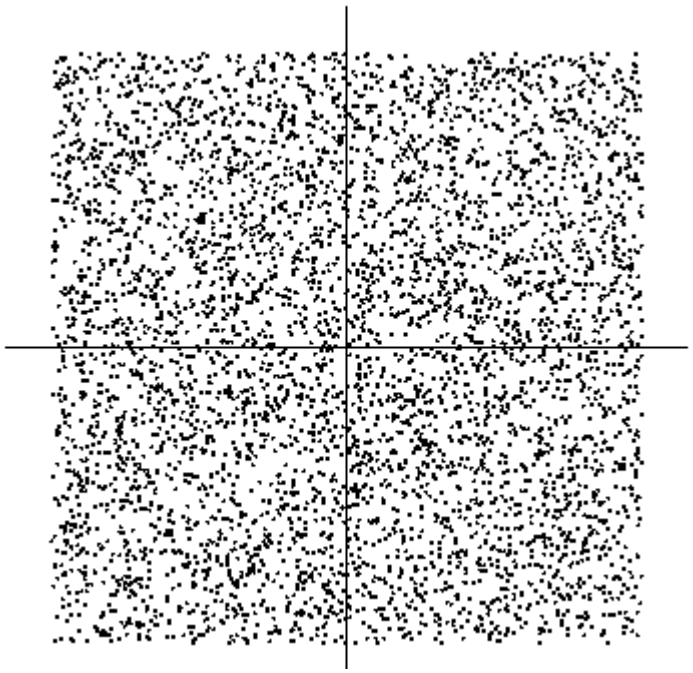
$$\tilde{\mathbf{x}} = \mathbf{E}\mathbf{D}^{-1/2}\mathbf{E}^T\mathbf{x}$$

其中使用特征值分解来得到 E（特征向量矩阵）和 D（特征值对角矩阵），计算公式为

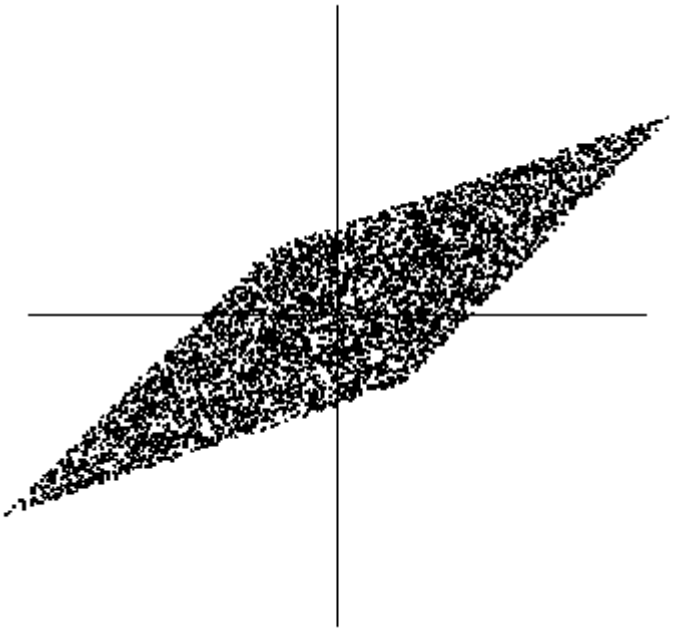
$$E\{\mathbf{x}\mathbf{x}^T\} = \mathbf{E}\mathbf{D}\mathbf{E}^T$$

下面用个图来直观描述一下：

假设信号源 s1 和 s2 是独立的，比如下图横轴是 s1，纵轴是 s2，根据 s1 得不到 s2。

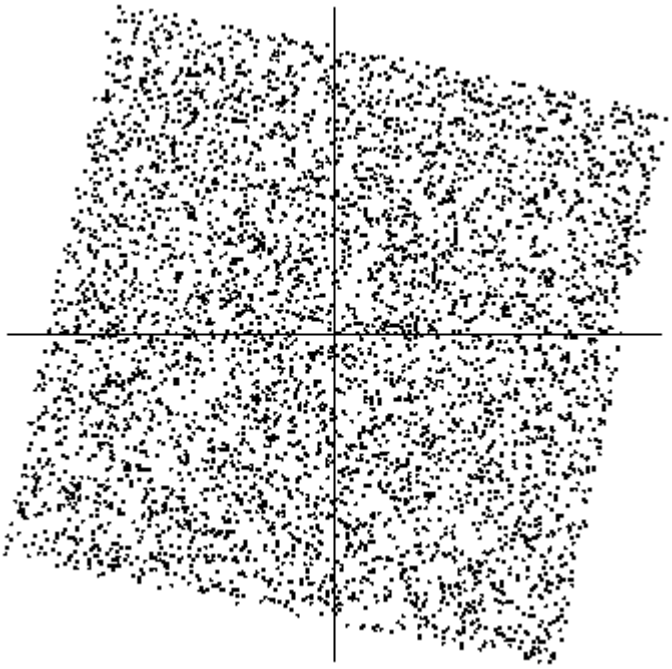


我们只知道他们合成后的信号 x，如下



此时 x1 和 x2 不是独立的（比如看最上面的尖角，知道了 x1 就知道了 x2）。那么直接代入我们之前的极大似然概率估计会有问题，因为我们假定 x 是独立的。

因此，漂白这一步为了让 x 独立。漂白结果如下：



可以看到数据变成了方阵，在$\tilde{x}$的维度上已经达到了独立。

然而这时 x 分布很好的情况下能够这样转换，当有噪音时怎么办呢？可以先使用前面提到的 PCA 方法来对数据进行降维，滤去噪声信号，得到 k 维的正交向量，然后再使用 ICA。

10. 小结

ICA 的盲信号分析领域的一个强有力方法，也是求非高斯分布数据隐含因子的方法。从之前我们熟悉的样本-特征角度看，我们使用 ICA 的前提条件是，认为样本数据由独立非高斯分布的隐含因子产生，隐含因子个数等于特征数。而 PCA 认为特征是由 k 个正交的特征（也可看作是隐含因子）生成的。同是因子分析，一个用来更适合用来还原信号（因为信号比较有规律，经常不是高斯分布的），一个更适合用来降维（用那么多特征干嘛，k 个正交的即可）。有时候也需要组合两者一起使用。这段是我的个人理解，仅供参考。

线性判别分析（Linear Discriminant Analysis）

JerryLead
csxulijie@gmail.com

1. 问题

之前我们讨论的 PCA、ICA 也好，对样本数据来言，可以是没有类别标签 y 的。回想我们做回归时，如果特征太多，那么会产生不相关特征引入、过度拟合等问题。我们可以使用 PCA 来降维，但 PCA 没有将类别标签考虑进去，属于无监督的。

比如回到上次提出的文档中含有“learn”和“study”的问题，使用 PCA 后，也许可以将这两个特征合并为一个，降了维度。但假设我们的类别标签 y 是判断这篇文章的 topic 是不是有关学习方面的。那么这两个特征对 y 几乎没什么影响，完全可以去除。

再举一个例子，假设我们对一张 100*100 像素的图片做人脸识别，每个像素是一个特征，那么会有 10000 个特征，而对应的类别标签 y 仅仅是 0/1 值，1 代表是人脸。这么多特征不仅训练复杂，而且不必要特征对结果会带来不可预知的影响，但我们想得到降维后的一些最佳特征（与 y 关系最密切的），怎么办呢？

2. 线性判别分析（二类情况）

回顾我们之前的 logistic 回归方法，给定 m 个 n 维特征的训练样例$x^{(i)}\{x_1^{(i)}, x_2^{(i)}, \dots, x_n^{(i)}\}$（i 从 1 到 m），每个$x^{(i)}$对应一个类标签$y^{(i)}$。我们就是要学习出参数θ，使得$y^{(i)} = g(\theta)^T x^{(i)}$（g 是 sigmoid 函数）。

现在只考虑二值分类情况，也就是 y=1 或者 y=0。

为了方便表示，我们先换符号重新定义问题，给定特征为 d 维的 N 个样例，$x^{(i)}\{x_1^{(i)}, x_2^{(i)}, \dots, x_d^{(i)}\}$，其中有$N_1$个样例属于类别$\omega_1$，另外$N_2$个样例属于类别$\omega_2$。

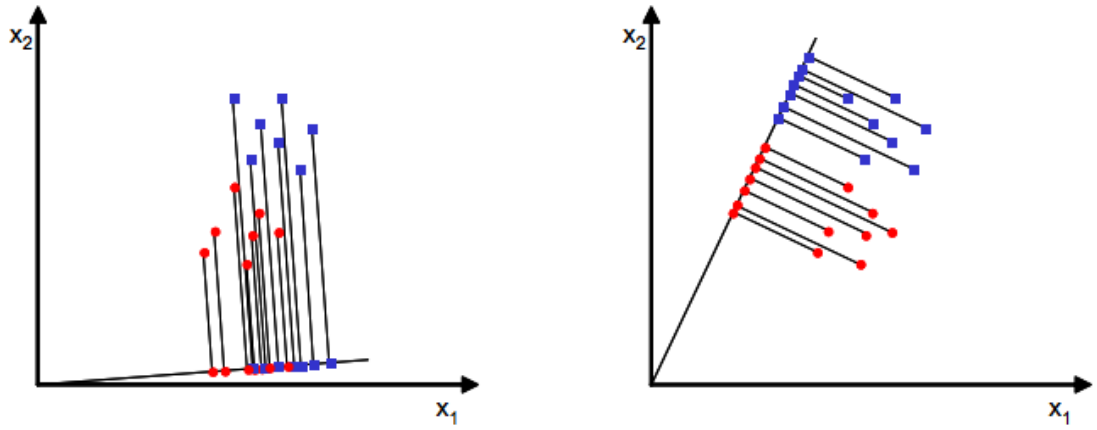
现在我们觉得原始特征数太多，想将 d 维特征降到**只有一维**，而又要保证类别能够“清晰”地反映在低维数据上，也就是这一维就能决定每个样例的类别。

我们将这个最佳的向量称为 w（d 维），那么样例 x（d 维）到 w 上的投影可以用下式来计算

$$y = w^T x$$

这里得到的 y 值不是 0/1 值，而是 x 投影到直线上的点到原点的距离。

当 x 是二维的，我们就是要找一条直线（方向为 w）来做投影，然后寻找最能使样本点分离的直线。如下图：



从直观上来看，右图比较好，可以很好地将不同类别的样本点分离。

接下来我们从定量的角度来找到这个最佳的 w。
首先我们寻找每类样例的均值（中心点），这里 i 只有两个

$$\mu_i = \frac{1}{N_i}\sum_{x\in\omega_i} x$$

由于 x 到 w 投影后的样本点均值为

$$\widetilde{\mu_i} = \frac{1}{N_i}\sum_{y\in\omega_i} y = \frac{1}{N_i}\sum_{x\in\omega_i} w^T x = w^T\mu_i$$

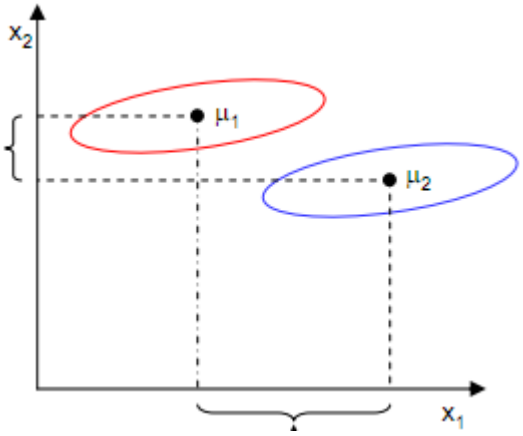
由此可知，投影后的的均值也就是样本中心点的投影。

什么是最佳的直线（w）呢？我们首先发现，能够使投影后的两类样本中心点尽量分离的直线是好的直线，定量表示就是：

$$J(w) = |\widetilde{\mu_1} - \widetilde{\mu_2}| = |w^T(\mu_1 - \mu_2)|$$

J(w)越大越好。

但是只考虑 J(w)行不行呢？不行，看下图



样本点均匀分布在椭圆里，投影到横轴 x1 上时能够获得更大的中心点间距 J(w)，但是由于有重叠，x1 不能分离样本点。投影到纵轴 x2 上，虽然 J(w)较小，但是能够分离样本点。因此我们还需要考虑样本点之间的方差，方差越大，样本点越难以分离。

我们使用另外一个度量值，称作散列值（scatter），对投影后的类求散列值，如下

$$\tilde{s}_i^2 = \sum_{y \in \omega_i} (y - \tilde{\mu}_i)^2$$

从公式中可以看出，只是少除以样本数量的方差值，散列值的几何意义是样本点的密集程度，值越大，越分散，反之，越集中。

而我们想要的投影后的样本点的样子是：不同类别的样本点越分开越好，同类的越聚集越好，也就是均值差越大越好，散列值越小越好。正好，我们可以使用 J(w)和 S 来度量，最终的度量公式是

$$\mathrm{J(w)} = \frac{|\widetilde{\mu_1} - \widetilde{\mu_2}|^2}{\widetilde{s_1}^2 + \widetilde{s_2}^2}$$

接下来的事就比较明显了，我们只需寻找使 J(w)最大的 w 即可。

先把散列值公式展开

$$\tilde{s}_i^2 = \sum_{y \in \omega_i} (y - \tilde{\mu}_i)^2 = \sum_{x \in \omega_i} (w^T x - w^T \mu_i)^2 = \sum_{x \in \omega_i} w^T (x - \mu_i)(x - \mu_i)^T w$$

我们定义上式中中间那部分

$$\mathrm{S}_i = \sum_{x \in \omega_i} (x - \mu_i)(x - \mu_i)^T$$

这个公式的样子不就是少除以样例数的协方差矩阵么，称为散列矩阵（scatter matrices）

我们继续定义

$$\mathrm{S}_w = \mathrm{S}_1 + \mathrm{S}_2$$

S_w称为 **Within**-class scatter matrix。

那么回到上面$\tilde{s}_i^2$的公式，使用S_i替换中间部分，得

$$\tilde{s}_i^2 = w^{\mathrm{T}} \mathrm{S}_i w$$

$$\tilde{s}_1^2 + \tilde{s}_2^2 = w^{\mathrm{T}} \mathrm{S}_w w$$

然后，我们展开分子

$$(\tilde{\mu}_1 - \tilde{\mu}_2)^2 = (\mathrm{w}^{\mathrm{T}}\mu_1 - \mathrm{w}^{\mathrm{T}}\mu_2)^2 = \mathrm{w}^{\mathrm{T}} \underbrace{(\mu_1 - \mu_2)(\mu_1 - \mu_2)^{\mathrm{T}}}_{\mathrm{S_B}} \mathrm{w} = \mathrm{w}^{\mathrm{T}} \mathrm{S_B} \mathrm{w}$$

S_B称为 **Between**-class scatter，是两个向量的外积，虽然是个矩阵，但秩为 1。

那么 J(w)最终可以表示为

$$J(w) = \frac{w^T S_B w}{w^T S_w w}$$

在我们求导之前，需要对分母进行归一化，因为不做归一的话，w 扩大任何倍，都成立，我们就无法确定 w。因此我们打算令$||w^T S_w w|| = 1$，那么加入拉格朗日乘子后，求导

$$c(w) = w^t S_B w - \lambda(w^t S_W w - 1)$$

$$\Rightarrow \frac{dc}{dw} = 2S_B w - 2\lambda S_W w = 0$$

$$\Rightarrow S_B w = \lambda S_W w$$

其中用到了矩阵微积分，求导时可以简单地把$w^T S_w w$当做$S_w w^2$看待。
如果S_w可逆，那么将求导后的结果两边都乘以S_w^{-1}，得

$$S_W{}^{-1} S_B w = \lambda w$$

这个可喜的结果就是 w 就是矩阵$S_w^{-1} S_B$的特征向量了。
这个公式称为 Fisher linear discrimination。

等等，让我们再观察一下，发现前面S_B的公式

$$S_B = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$$

那么

$$S_B w = (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T w = (\mu_1 - \mu_2) * \lambda_w$$

代入最后的特征值公式得

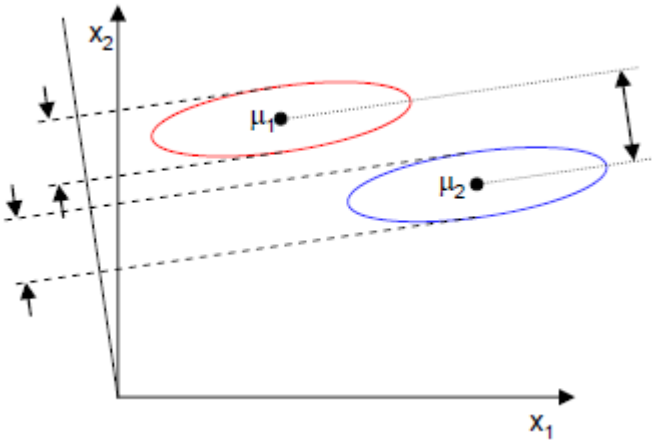
$$S_w^{-1} S_B w = S_w^{-1}(\mu_1 - \mu_2) * \lambda_w = \lambda w$$

由于对 w 扩大缩小任何倍不影响结果，因此可以约去两边的未知常数λ和λ_w，得到

$$w = S_w^{-1}(\mu_1 - \mu_2)$$

至此，我们只需要求出原始样本的均值和方差就可以求出最佳的方向 w，这就是 Fisher 于 1936 年提出的线性判别分析。

看上面二维样本的投影结果图：



3. 线性判别分析（多类情况）

前面是针对只有两个类的情况，假设类别变成多个了，那么要怎么改变，才能保证投影后类别能够分离呢？

我们之前讨论的是如何将 d 维降到一维，现在类别多了，一维可能已经不能满足要求。假设我们有 C 个类别，需要 K 维向量（或者叫做基向量）来做投影。

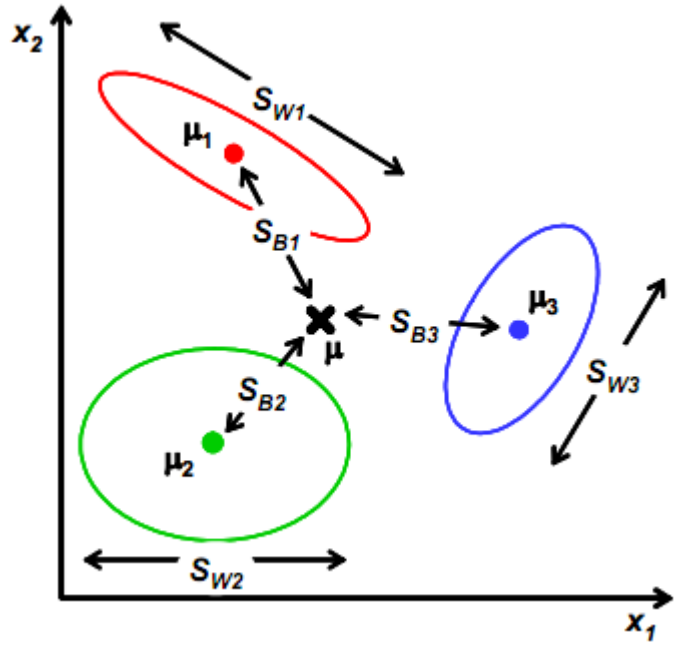
将这 K 维向量表示为$W = [w_1|w_2|\dots|w_K]$。

我们将样本点在这 K 维向量投影后结果表示为$[y_1, y_2, \dots, y_K]$，有以下公式成立

$$y_i = w_i^T x$$
$$y = W^T x$$

为了像上节一样度量 J(w)，我们打算仍然从类间散列度和类内散列度来考虑。

当样本是二维时，我们从几何意义上考虑：



其中μ_i和S_w与上节的意义一样，S_{w1}是类别 1 里的样本点相对于该类中心点μ_1的散列程度。S_{B1}变成类别 1 中心点相对于样本中心点μ的协方差矩阵，即类 1 相对于μ的散列程度。

S_w为

$$S_w = \sum_{i=1}^{C} S_{wi}$$

S_{wi}的计算公式不变，仍然类似于类内部样本点的协方差矩阵

$$S_{wi} = \sum_{x \in \omega_i} (x - \mu_i)(x - \mu_i)^T$$

S_B需要变，原来度量的是两个均值点的散列情况，现在度量的是每类均值点相对于样本中心的散列情况。类似于将μ_i看作样本点，μ是均值的协方差矩阵，如果某类里面的样本点较多，那么其权重稍大，权重用 Ni/N 表示，但由于 J(w)对倍数不敏感，因此使用 Ni。

$$S_B = \sum_{i=1}^{C} N_i \, (\mu_i - \mu)(\mu_i - \mu)^T$$

其中

$$\mu = \frac{1}{N} \sum_{\forall x} x = \frac{1}{N} \sum_{x \in \omega_i} N_i \mu_i$$

μ是所有样本的均值。

上面讨论的都是在投影前的公式变化，但真正的 J(w)的分子分母都是在投影后计算的。下面我们看样本点投影后的公式改变：

这两个是第 i 类样本点在某基向量上投影后的均值计算公式。

$$\tilde{\mu}_i = \frac{1}{N_i}\sum_{y\in\omega_i} y$$

$$\tilde{\mu} = \frac{1}{N}\sum_{\forall y} y$$

下面两个是在某基向量上投影后的S_w和S_B

$$\widetilde{S_w} = \sum_{i=1}^{C}\sum_{y\in\omega_i}(y-\tilde{\mu}_i)(y-\tilde{\mu}_i)^T$$

$$\widetilde{S_B} = \sum_{i=1}^{C} N_i(\tilde{\mu}_i-\tilde{\mu})(\tilde{\mu}_i-\tilde{\mu})^T$$

其实就是将μ换成了$\tilde{\mu}$。

综合各个投影向量（w）上的$\widetilde{S_w}$和$\widetilde{S_B}$，更新这两个参数，得到

$$\widetilde{S_w} = W^T S_w W$$
$$\widetilde{S_B} = W^T S_B W$$

W 是基向量矩阵，$\widetilde{S_w}$是投影后的各个类内部的散列矩阵之和，$\widetilde{S_B}$是投影后各个类中心相对于全样本中心投影的散列矩阵之和。

回想我们上节的公式 J(w)，分子是两类中心距，分母是每个类自己的散列度。现在投影方向是多维了（好几条直线），分子需要做一些改变，我们不是求两两样本中心距之和（这个对描述类别间的分散程度没有用），而是求每类中心相对于全样本中心的散列度之和。

然而，最后的 J(w)的形式是

$$J(w) = \frac{|\widetilde{S_B}|}{|\widetilde{S_w}|} = \frac{|W^T S_B W|}{|W^T S_w W|}$$

由于我们得到的分子分母都是散列矩阵，要将矩阵变成实数，需要取行列式。又因为行列式的值实际上是矩阵特征值的积，一个特征值可以表示在该特征向量上的发散程度。因此我们使用行列式来计算（此处我感觉有点牵强，道理不是那么有说服力）。

整个问题又回归为求 J(w)的最大值了，我们固定分母为 1，然后求导，得出最后结果（我翻查了很多讲义和文章，没有找到求导的过程）

$$S_B w_i = \lambda S_w w_i$$

与上节得出的结论一样

$$S_w^{-1} S_B w_i = \lambda w_i$$

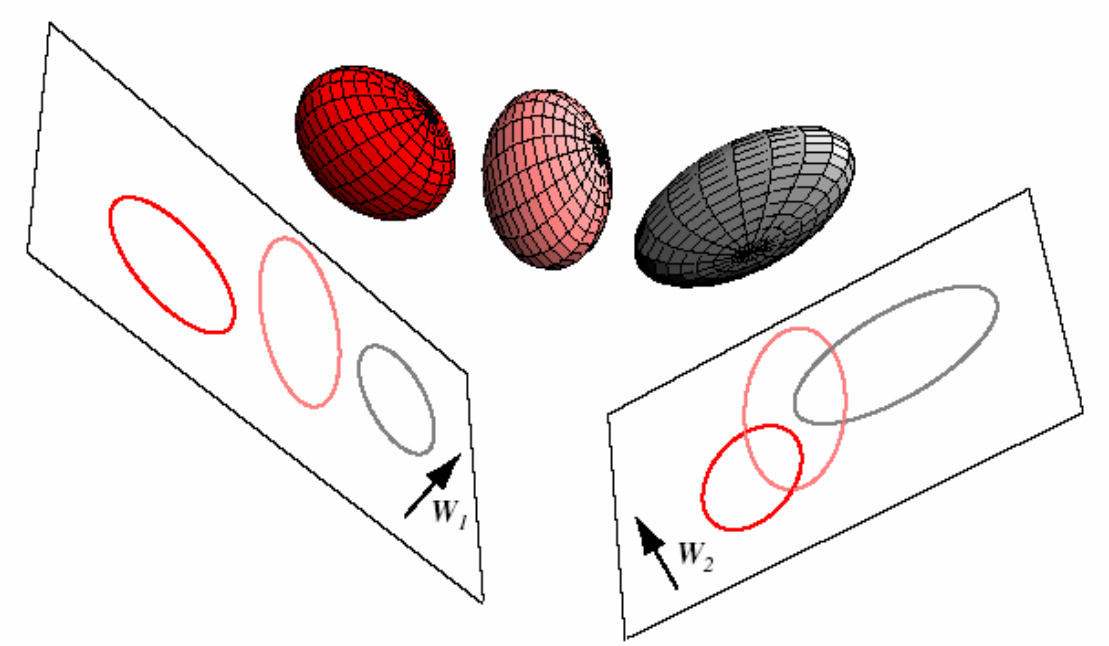
最后还归结到了求矩阵的特征值上来了。首先求出$S_w^{-1}S_B$的特征值，然后取前 K 个特征向量组成 W 矩阵即可。

注意：由于S_B中的$(\mu_i - \mu)$ 秩为 1，因此S_B的秩至多为 C（矩阵的秩小于等于各个相加矩阵的秩的和）。由于知道了前 C-1 个μ_i后，最后一个μ_C可以有前面的μ_i来线性表示，因此S_B的秩至多为 C-1。那么 K 最大为 C-1，即特征向量最多有 C-1 个。特征值大的对应的特征向量分割性能最好。

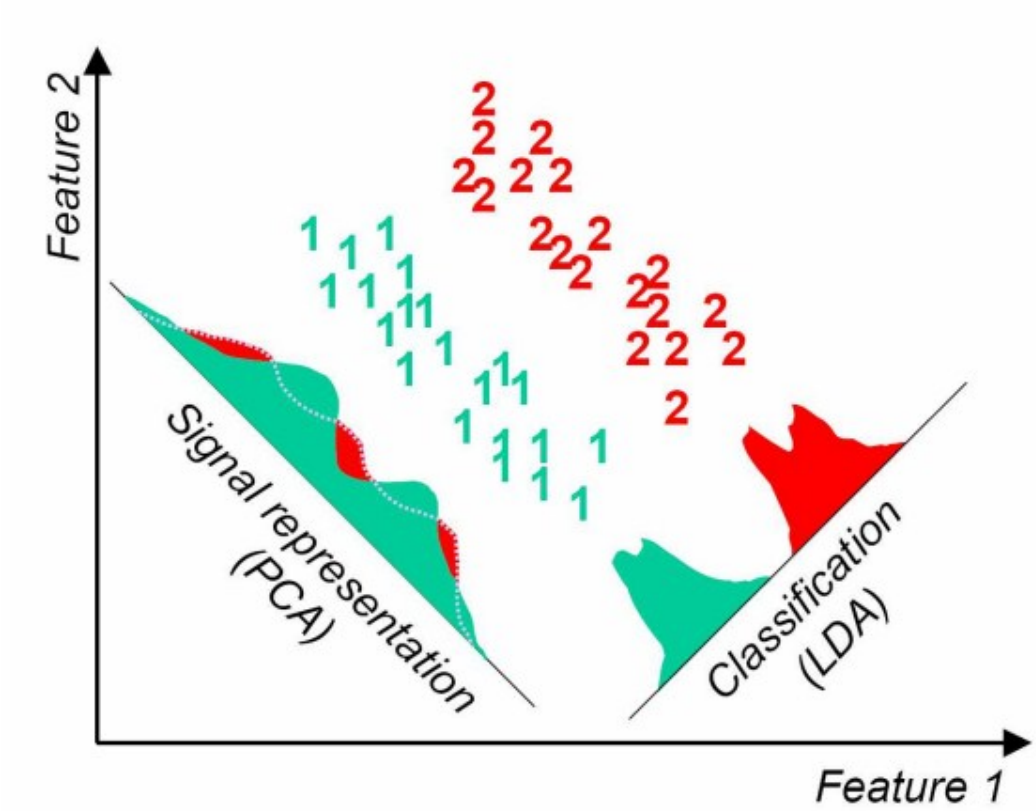
由于$S_w^{-1}S_B$不一定是对称阵，因此得到的 K 个特征向量不一定正交，这也是与 PCA 不同的地方。

4. 实例

将 3 维空间上的球体样本点投影到二维上，W1 相比 W2 能够获得更好的分离效果。



PCA 与 LDA 的降维对比：



PCA 选择样本点投影具有最大方差的方向，LDA 选择分类性能最好的方向。

LDA 既然叫做线性判别分析，应该具有一定的预测功能，比如新来一个样例 x，如何确定其类别？

拿二值分来来说，我们可以将其投影到直线上，得到 y，然后看看 y 是否在超过某个阈值 y0，超过是某一类，否则是另一类。而怎么寻找这个 y0 呢？

看

$$y = w^T x$$

根据中心极限定理，独立同分布的随机变量和符合高斯分布，然后利用极大似然估计求

$$P(y|C_i)$$

然后用决策理论里的公式来寻找最佳的 y0，详情请参阅 PRML。

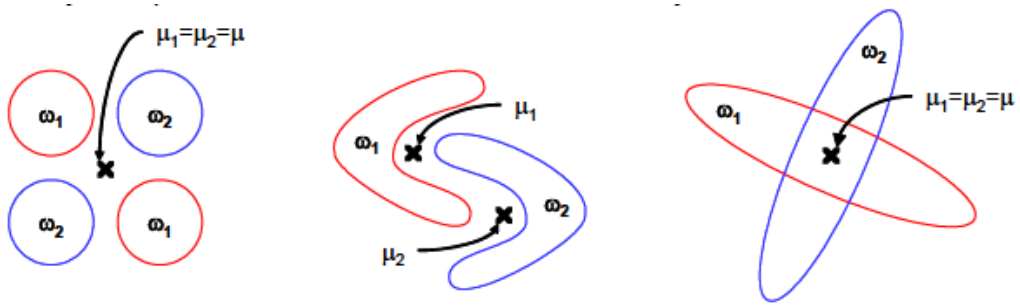
这是一种可行但比较繁琐的选取方法，可以看第 7 节（一些问题）来得到简单的答案。

5. 使用 LDA 的一些限制

1、 LDA 至多可生成 C-1 维子空间

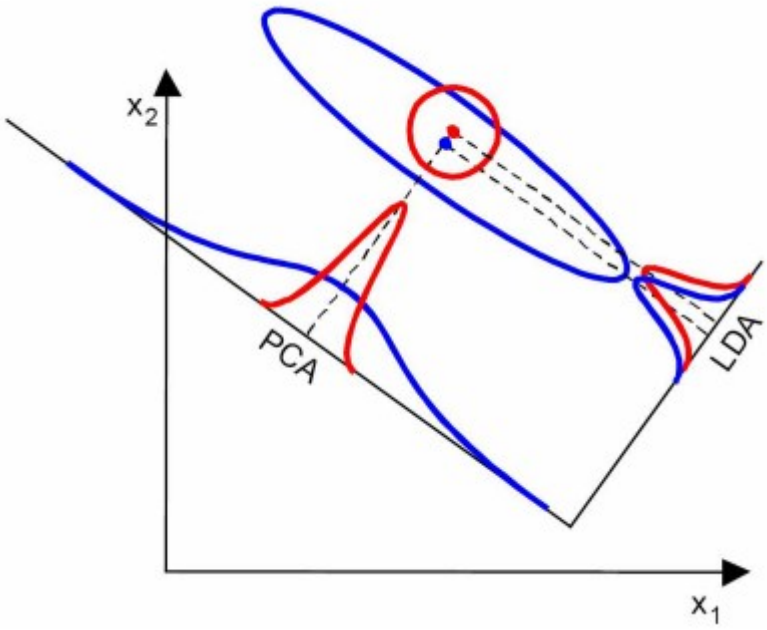
LDA 降维后的维度区间在[1,C-1]，与原始特征数 n 无关，对于二值分类，最多投影到 1 维。

2、 LDA 不适合对非高斯分布样本进行降维。



上图中红色区域表示一类样本，蓝色区域表示另一类，由于是 2 类，所以最多投影到 1 维上。不管在直线上怎么投影，都难使红色点和蓝色点内部凝聚，类间分离。

3、 LDA 在样本分类信息依赖方差而不是均值时，效果不好。



上图中，样本点依靠方差信息进行分类，而不是均值信息。LDA 不能够进行有效分类，因为 LDA 过度依靠均值信息。

4、 LDA 可能过度拟合数据。

6. LDA 的一些变种

1、 非参数 LDA

非参数 LDA 使用本地信息和 K 临近样本点来计算S_B,使得S_B是全秩的，这样我们可以抽取多余 C-1 个特征向量。而且投影后分离效果更好。

2、 正交 LDA

先找到最佳的特征向量，然后找与这个特征向量正交且最大化 fisher 条件的向量。这种方法也能摆脱 C-1 的限制。

3、 一般化 LDA

引入了贝叶斯风险等理论

4、 核函数 LDA

将特征x → Φ(x)，使用核函数来计算。

7. 一些问题

上面在多值分类中使用的

$$S_B = \sum_{i=1}^{C} N_i (\mu_i - \mu)(\mu_i - \mu)^T$$

是带权重的各类样本中心到全样本中心的散列矩阵。如果 C=2（也就是二值分类时）套用这个公式，不能够得出在二值分类中使用的S_B。

$$S_B = \sum_{i=1}^{C} (\mu_1 - \mu_2)(\mu_1 - \mu_2)^T$$

因此二值分类和多值分类时求得的S_B会不同，而S_W意义是一致的。

对于二值分类问题，令人惊奇的是最小二乘法和 Fisher 线性判别分析是一致的。

下面我们证明这个结论，并且给出第 4 节提出的 y0 值得选取问题。

回顾之前的线性回归，给定 N 个 d 维特征的训练样例$x^{(i)}\{x_1^{(i)}, x_2^{(i)}, \dots, x_d^{(i)}\}$（i 从 1 到 N），每个$x^{(i)}$对应一个类标签$y^{(i)}$。我们之前令 y=0 表示一类，y=1 表示另一类，现在我们为了证明最小二乘法和 LDA 的关系，我们需要做一些改变

$$\begin{cases} y = \dfrac{N}{N_1}，样例属于有N_1个元素的类C_1 \\ y = -\dfrac{N}{N_2}，样例属于有N_2个元素的类C_2 \end{cases}$$

就是将 0/1 做了值替换。

我们列出最小二乘法公式

$$E = \frac{1}{2}\sum_{i=1}^{N} (w^T x^{(i)} + w_0 - y^{(i)})^2$$

w 和w_0是拟合权重参数。

分别对w_0和 w 求导得

$$\sum_{i=1}^{N} (w^T x^{(i)} + w_0 - y^{(i)}) = 0$$

$$\sum_{i=1}^{N} (w^T x^{(i)} + w_0 - y^{(i)}) x^{(i)} = 0$$

从第一个式子展开可以得到

$$w^T N\mu + Nw_0 - \sum_{i=1}^{N} y^{(i)} = w^T N\mu + Nw_0 - \left(N_1 \frac{N}{N_1} - N_2 \frac{N}{N_2}\right) = 0$$

消元后，得

$$w_0 = -w^T \mu$$

$$\mu = \frac{1}{\mathrm{N}} \sum_{i=1}^{N} x^{(i)} = \frac{1}{\mathrm{N}} (N_1 \mu_1 + N_2 \mu_2)$$

可以证明第二个式子展开后和下面的公式等价

$$\left(S_w + \frac{N_1 N_2}{N} S_B\right) w = N(\mu_1 - \mu_2)$$

其中S_w和S_B与二值分类中的公式一样。

由于$S_B \mathrm{w} = (\mu_1 - \mu_2) * \lambda_{\mathrm{w}}$

因此，最后结果仍然是

$$w = \mathrm{S}_w^{-1}(\mu_1 - \mu_2)$$

这个过程从几何意义上去理解也就是变形后的线性回归（将类标签重新定义），线性回归后的直线方向就是二值分类中 LDA 求得的直线方向 w。

好了，我们从改变后的 y 的定义可以看出 y>0 属于类C_1，y<0 属于类C_2。因此我们可以选取 y0=0，即如果$\mathrm{y(x)} = w^T x + w_0 > 0$，就是类$C_1$，否则是类$C_2$。

写了好多，挺杂的，还有个 topic 模型也叫做 LDA，不过名字叫做 Latent Dirichlet Allocation，第二作者就是 Andrew Ng 大牛，最后一个他导师 Jordan 泰斗了，什么时候拜读后再写篇总结发上来吧。

因子分析（Factor Analysis）

JerryLead
csxulijie@gmail.com
2011 年 5 月 11 日

1 问题

之前我们考虑的训练数据中样例$x^{(i)}$的个数 m 都远远大于其特征个数 n，这样不管是进行回归、聚类等都没有太大的问题。然而当训练样例个数 m 太小，甚至 m<<n 的时候，使用梯度下降法进行回归时，如果初值不同，得到的参数结果会有很大偏差（因为方程数小于参数个数）。另外，如果使用多元高斯分布(Multivariate Gaussian distribution)对数据进行拟合时，也会有问题。让我们来演算一下，看看会有什么问题：

多元高斯分布的参数估计公式如下：

$$\mu = \frac{1}{m}\sum_{i=1}^{m} x^{(i)}$$

$$\Sigma = \frac{1}{m}\sum_{i=1}^{m}(x^{(i)} - \mu)(x^{(i)} - \mu)^T$$

分别是求 mean 和协方差的公式，$x^{(i)}$表示样例，共有 m 个，每个样例 n 个特征，因此μ是 n 维向量，Σ是 n*n 协方差矩阵。

当 m<<n 时，我们会发现Σ是奇异阵（$|\Sigma| = 0$），也就是说Σ^{-1}不存在，没办法拟合出多元高斯分布了，确切的说是我们估计不出来Σ。

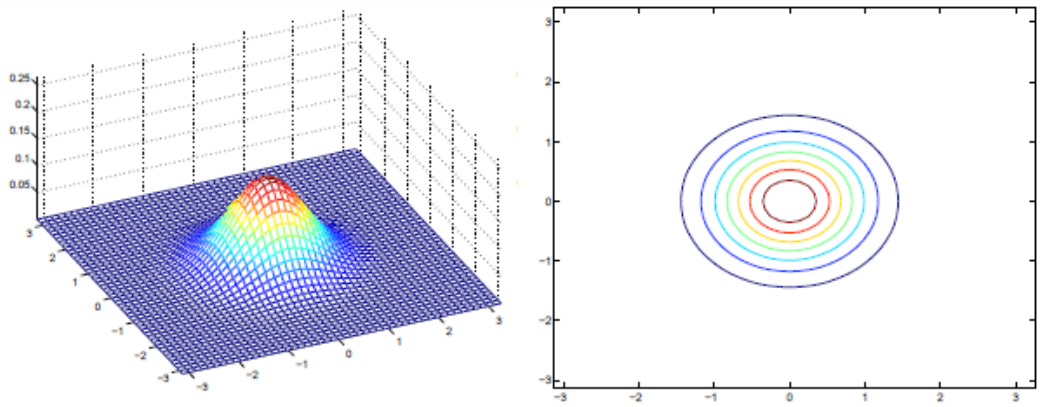
如果我们仍然想用多元高斯分布来估计样本，那怎么办呢？

2 限制协方差矩阵

当没有足够的数据去估计Σ时，那么只能对模型参数进行一定假设，之前我们想估计出完全的Σ（矩阵中的全部元素），现在我们假设Σ就是对角阵（各特征间相互独立），那么我们只需要计算每个特征的方差即可，最后的Σ只有对角线上的元素不为 0

$$\Sigma_{jj} = \frac{1}{m}\sum_{i=1}^{m}(x_j^{(i)} - \mu_j)^2$$

回想我们之前讨论过的二维多元高斯分布的几何特性，在平面上的投影是个椭圆，中心点由μ决定，椭圆的形状由Σ决定。Σ如果变成对角阵，就意味着椭圆的两个轴都和坐标轴平行了。



如果我们想对Σ进一步限制的话，可以假设对角线上的元素都是等值的。

$$\Sigma = \sigma^2 I$$

其中

$$\sigma^2 = \frac{1}{mn}\sum_{j=1}^{n}\sum_{i=1}^{m}(x_j^{(i)} - \mu_j)^2$$

也就是上一步对角线上元素的均值，反映到二维高斯分布图上就是椭圆变成圆。

当我们要估计出完整的Σ时，我们需要 m>=n+1 才能保证在最大似然估计下得出的Σ是非奇异的。然而在上面的任何一种假设限定条件下，只要 m>=2 都可以估计出限定的Σ。

这样做的缺点也是显然易见的，我们认为特征间独立，这个假设太强。接下来，我们给出一种称为因子分析的方法，使用更多的参数来分析特征间的关系，并且不需要计算一个完整的Σ。

3 边缘和条件高斯分布

在讨论因子分析之前，先看看多元高斯分布中，条件和边缘高斯分布的求法。这个在后面因子分析的 EM 推导中有用。

假设 x 是有两个随机向量组成（可以看作是将之前的$x^{(i)}$分成了两部分）

$$\mathrm{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

其中$x_1 \in \mathbb{R}^r$，$x_2 \in \mathbb{R}^s$，那么$\mathrm{x} \in \mathbb{R}^{r+s}$。假设 x 服从多元高斯分布$\mathrm{x} \sim N(\mu, \Sigma)$，其中

$$\mu = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix}, \quad \Sigma = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}.$$

其中$\mu_1 \in \mathbb{R}^r$，$\mu_2 \in \mathbb{R}^s$，那么$\Sigma_{11} \in \mathbb{R}^{r\times r}$，$\Sigma_{12} \in \mathbb{R}^{r\times s}$，由于协方差矩阵是对称阵，因此$\Sigma_{12} = \Sigma_{21}^{T}$。

整体看来x_1和x_2联合分布符合多元高斯分布。

那么只知道联合分布的情况下，如何求得x_1的边缘分布呢？从上面的μ和Σ可以看出，$\mathrm{E}[x_1] = \mu_1, \mathrm{Cov}(x_1) = E[(x_1 - \mu_1)(x_1 - \mu_1)^T] = \Sigma_{11}$，下面我们验证第二个结果

$$\begin{aligned}
\mathrm{Cov}(x) &= \Sigma \\
&= \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix} \\
&= \mathrm{E}[(x-\mu)(x-\mu)^T] \\
&= \mathrm{E}\left[\begin{pmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{pmatrix} \begin{pmatrix} x_1 - \mu_1 \\ x_2 - \mu_2 \end{pmatrix}^T \right] \\
&= \mathrm{E}\begin{bmatrix} (x_1-\mu_1)(x_1-\mu_1)^T & (x_1-\mu_1)(x_2-\mu_2)^T \\ (x_2-\mu_2)(x_1-\mu_1)^T & (x_2-\mu_2)(x_2-\mu_2)^T \end{bmatrix}.
\end{aligned}$$

由此可见，多元高斯分布的边缘分布仍然是多元高斯分布。也就是说$\mathrm{x}_1 \sim \mathrm{N}(\mu_1, \Sigma_{11})$。

上面 Cov(x)里面有趣的是Σ_{12}，这个与之前计算协方差的效果不同。之前的协方差矩阵都是针对一个随机变量（多维向量）来说的，而Σ_{12}评价的是两个随机向量之间的关系。比如x_1={身高，体重}，x_2={性别，收入}，那么Σ_{11}求的是身高与身高，身高与体重，体重与体重的协方差。而Σ_{12}求的是身高与性别，身高与收入，体重与性别，体重与收入的协方差，看起来与之前的大不一样，比较诡异的求法。

上面求的是边缘分布，让我们考虑一下条件分布的问题，也就是$\mathrm{x}_1|\mathrm{x}_2$的问题。根据多元高斯分布的定义，$\mathrm{x}_1|\mathrm{x}_2 \sim \mathrm{N}(\mu_{1|2}, \Sigma_{1|2})$。

且

$$\begin{aligned}
\mu_{1|2} &= \mu_1 + \Sigma_{12}\Sigma_{22}^{-1}(x_2 - \mu_2), && (1) \\
\Sigma_{1|2} &= \Sigma_{11} - \Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}. && (2)
\end{aligned}$$

这是我们接下来计算时需要的公式，这两个公式直接给出，没有推导过程。如果想了解具体的推导过程，可以参见 Chuong B. Do 写的《Gaussian processes》。

4 因子分析例子

下面通过一个简单例子，来引出因子分析背后的思想。

因子分析的实质是认为 m 个 n 维特征的训练样例$\mathrm{x}^{(i)}(x_1^{(i)}, x_2^{(i)}, \ldots, x_n^{(i)})$的产生过程如下：

1、 首先在一个 k 维的空间中按照多元高斯分布生成 m 个$\mathrm{z}^{(i)}$（k 维向量），即

$$z^{(i)} \sim N(0, I)$$

2、 然后存在一个变换矩阵$\Lambda \in \mathbb{R}^{\mathrm{n}\times\mathrm{k}}$，将$\mathrm{z}^{(i)}$映射到 n 维空间中，即

$$\Lambda z^{(i)}$$

因为$\mathrm{z}^{(i)}$的均值是 0，映射后仍然是 0。

3、 然后将$\Lambda z^{(i)}$加上一个均值μ（n 维），即

$$\mu + \Lambda z^{(i)}$$

对应的意义是将变换后的$\Lambda z^{(i)}$（n 维向量）移动到样本$x^{(i)}$的中心点μ。

4、 由于真实样例$x^{(i)}$与上述模型生成的有误差，因此我们继续加上误差ϵ（n 维向量），而且ϵ符合多元高斯分布，即

$$\epsilon \sim N(0, \Psi)$$

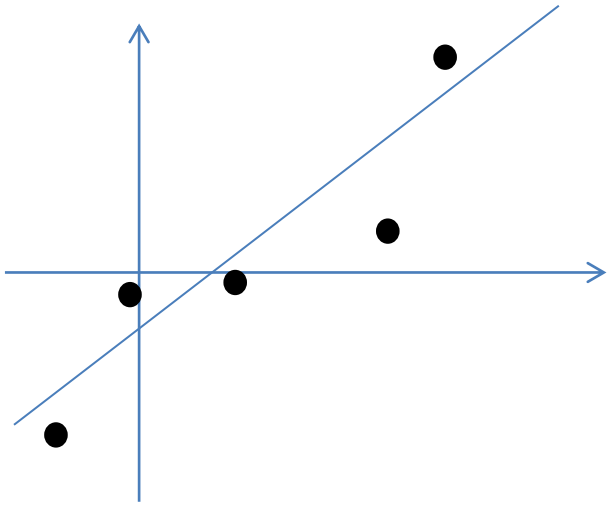
$$\mu + \Lambda z^{(i)} + \epsilon$$

5、 最后的结果认为是真实的训练样例$x^{(i)}$的生成公式

$$x^{(i)} = \mu + \Lambda z^{(i)} + \epsilon$$

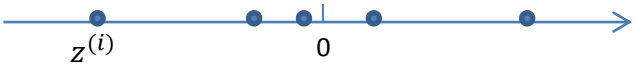
让我们使用一种直观方法来解释上述过程：

假设我们有 m=5 个 2 维的样本点$x^{(i)}$（两个特征），如下：



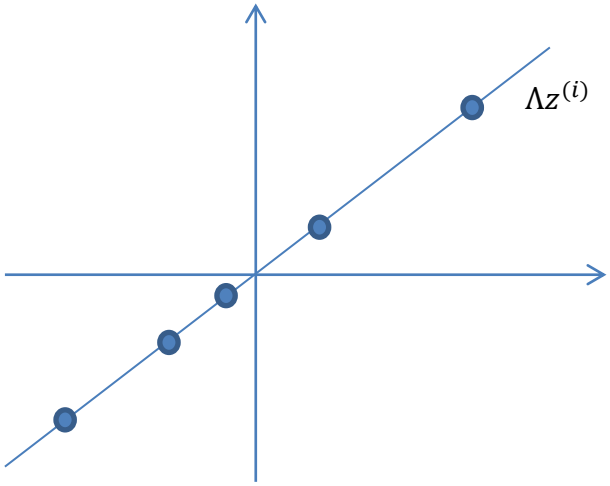
那么按照因子分析的理解，样本点的生成过程如下：

1、 我们首先认为在 1 维空间（这里 k=1），存在着按正态分布生成的 m 个点$z^{(i)}$，如下

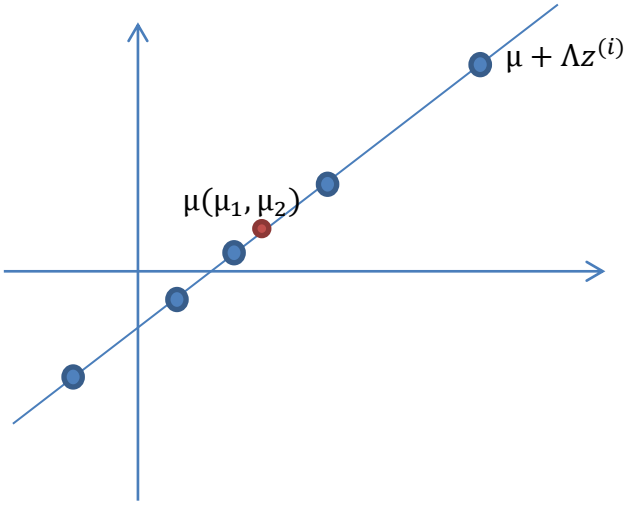


均值为 0，方差为 1。

2、 然后使用某个$\Lambda = (a, b)^T$将一维的 z 映射到 2 维，图形表示如下：

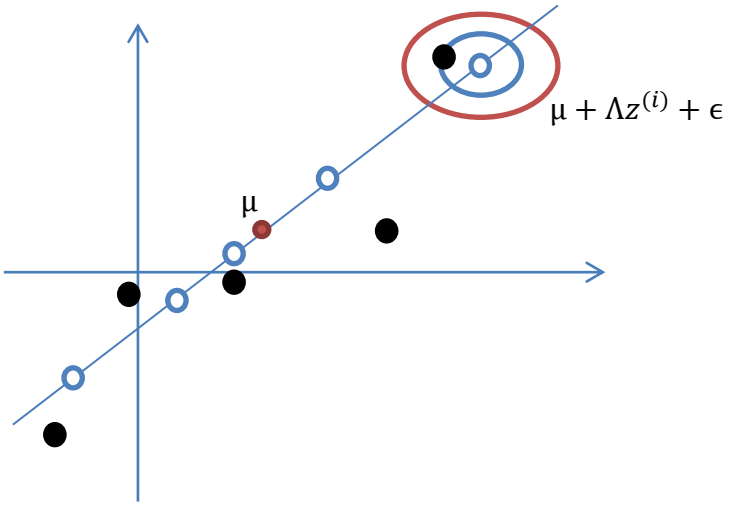


3、 之后加上$\mu\ (\mu_1, \mu_2)^T$，即将所有点的横坐标移动μ_1，纵坐标移动μ_2，将直线移到一个位置，使得直线过点μ，原始左边轴的原点现在为μ（红色点）。



然而，样本点不可能这么规则，在模型上会有一定偏差，因此我们需要将上步生成的点做一些扰动（误差），扰动$\epsilon \sim N(0, \Psi)$。

4、 加入扰动后，我们得到黑色样本$\mathrm{x}^{(i)}$如下：



5、 其中由于 z 和ϵ的均值都为 0，因此μ也是原始样本点（黑色点）的均值。

由以上的直观分析，我们知道了因子分析其实就是认为高维样本点实际上是由低维样本点经过高斯分布、线性变换、误差扰动生成的，因此高维数据可以使用低维来表示。

5 因子分析模型

上面的过程是从隐含随机变量 z 经过变换和误差扰动来得到观测到的样本点。其中 z 被称为因子，是低维的。

我们将式子再列一遍如下：

$$z \sim N(0, I)$$
$$\epsilon \sim N(0, \Psi)$$
$$x = \mu + \Lambda z + \epsilon$$

其中误差ϵ和 z 是独立的。

下面使用的因子分析表示方法是矩阵表示法，在参考资料中给出了一些其他的表示方法，如果不明白矩阵表示法，可以参考其他资料。

矩阵表示法认为 z 和 x 联合符合多元高斯分布，如下

$$\begin{bmatrix} z \\ x \end{bmatrix} \sim N(\mu_{zx}, \Sigma)$$

求μ_{zx}之前需要求 E[x]

$$E[x] = E[\mu + \Lambda z + \epsilon]$$
$$= \mu + \Lambda E[z] + E[\epsilon]$$

$= \mu$

我们已知 E[z]=0，因此

$$\mu_{zx} = \begin{bmatrix} \vec{0} \\ \mu \end{bmatrix}$$

下一步是计算Σ，

其中$\Sigma_{zz} = Cov(z) = I$

接着求Σ_{zx}

$$\begin{aligned} \mathrm{E}[(z - \mathrm{E}[z])(x - \mathrm{E}[x])^T] &= \mathrm{E}[z(\mu + \Lambda z + \epsilon - \mu)^T] \\ &= \mathrm{E}[zz^T]\Lambda^T + \mathrm{E}[z\epsilon^T] \\ &= \Lambda^T. \end{aligned}$$

这个过程中利用了 z 和ϵ独立假设（$\mathrm{E}[z\epsilon^T] = E[z]E[\epsilon^T] = 0$）。并将Λ看作已知变量。

接着求Σ_{xx}

$$\begin{aligned} \mathrm{E}[(x - \mathrm{E}[x])(x - \mathrm{E}[x])^T] &= \mathrm{E}[(\mu + \Lambda z + \epsilon - \mu)(\mu + \Lambda z + \epsilon - \mu)^T] \\ &= \mathrm{E}[\Lambda zz^T\Lambda^T + \epsilon z^T\Lambda^T + \Lambda z\epsilon^T + \epsilon\epsilon^T] \\ &= \Lambda\mathrm{E}[zz^T]\Lambda^T + \mathrm{E}[\epsilon\epsilon^T] \\ &= \Lambda\Lambda^T + \Psi. \end{aligned}$$

然后得出联合分布的最终形式

$$\begin{bmatrix} z \\ x \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \vec{0} \\ \mu \end{bmatrix}, \begin{bmatrix} I & \Lambda^T \\ \Lambda & \Lambda\Lambda^T + \Psi \end{bmatrix} \right).$$

从上式中可以看出 x 的边缘分布$x \sim N(\mu, \Lambda\Lambda^T + \Psi)$

那么对样本$\{x^{(i)}; i = 1, \ldots, m\}$进行最大似然估计

$$\ell(\mu, \Lambda, \Psi) = \log \prod_{i=1}^{m} \frac{1}{(2\pi)^{n/2}|\Lambda\Lambda^T + \Psi|} \exp\left(-\frac{1}{2}(x^{(i)} - \mu)^T(\Lambda\Lambda^T + \Psi)^{-1}(x^{(i)} - \mu) \right).$$

然后对各个参数求偏导数不就得到各个参数的值了么？

可惜我们得不到 closed-form。想想也是，如果能得到，还干嘛将 z 和 x 放在一起求联合分布呢。根据之前对参数估计的理解，在有隐含变量 z 时，我们可以考虑使用 EM 来进行估计。

6 因子分析的 EM 估计

我们先来明确一下各个参数，z 是隐含变量，μ, Λ, Ψ是待估参数。

回想 EM 两个步骤：

循环重复直到收敛 {

（E 步）对于每一个 i，计算

$$Q_i\left(z^{(i)}\right) \coloneqq p\left(z^{(i)}|x^{(i)};\theta\right)$$

（M 步）计算

$$\theta := \arg\max_{\theta} \sum_i \sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)};\theta)}{Q_i(z^{(i)})}.$$

}

我们套用一下：

（E 步）：

$$Q_i\left(z^{(i)}\right) \coloneqq p\left(z^{(i)}|x^{(i)};\mu,\Lambda,\Psi\right)$$

根据第 3 节的条件分布讨论，

$$z^{(i)}|x^{(i)};\mu,\Lambda,\Psi \sim N(\mu_{z^{(i)}|x^{(i)}}, \Sigma_{z^{(i)}|x^{(i)}})$$

因此

$$\begin{aligned}
\mu_{z^{(i)}|x^{(i)}} &= \Lambda^T(\Lambda\Lambda^T+\Psi)^{-1}(x^{(i)}-\mu), \\
\Sigma_{z^{(i)}|x^{(i)}} &= I-\Lambda^T(\Lambda\Lambda^T+\Psi)^{-1}\Lambda.
\end{aligned}$$

那么根据多元高斯分布公式，得到

$$Q_i(z^{(i)}) = \frac{1}{(2\pi)^{k/2}|\Sigma_{z^{(i)}|x^{(i)}}|^{1/2}} \exp\left(-\frac{1}{2}(z^{(i)}-\mu_{z^{(i)}|x^{(i)}})^T\Sigma_{z^{(i)}|x^{(i)}}^{-1}(z^{(i)}-\mu_{z^{(i)}|x^{(i)}})\right).$$

（M 步）：

直接写要最大化的目标是

$$\sum_{i=1}^{m}\int_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)};\mu,\Lambda,\Psi)}{Q_i(z^{(i)})} dz^{(i)}$$

其中待估参数是μ,Λ,Ψ

下面我们重点求Λ的估计公式

首先将上式简化为：

$$\sum_{i=1}^{m}\int_{z^{(i)}} Q_i(z^{(i)}) \left[\log p(x^{(i)}|z^{(i)};\mu,\Lambda,\Psi) + \log p(z^{(i)}) - \log Q_i(z^{(i)})\right] dz^{(i)} \quad (5)$$

$$= \sum_{i=1}^{m} \mathrm{E}_{z^{(i)}\sim Q_i}\left[\log p(x^{(i)}|z^{(i)};\mu,\Lambda,\Psi) + \log p(z^{(i)}) - \log Q_i(z^{(i)})\right] \quad (6)$$

这里$z^{(i)}$~Q_i表示$z^{(i)}$服从Q_i分布。然后去掉与Λ不相关的项（后两项），得

$$\sum_{i=1}^{m} \mathrm{E}\left[\log p(x^{(i)}|z^{(i)};\mu,\Lambda,\Psi)\right]$$
$$= \sum_{i=1}^{m} \mathrm{E}\left[\log \frac{1}{(2\pi)^{n/2}|\Psi|^{1/2}} \exp\left(-\frac{1}{2}(x^{(i)}-\mu-\Lambda z^{(i)})^T\Psi^{-1}(x^{(i)}-\mu-\Lambda z^{(i)})\right)\right]$$
$$= \sum_{i=1}^{m} \mathrm{E}\left[-\frac{1}{2}\log|\Psi| - \frac{n}{2}\log(2\pi) - \frac{1}{2}(x^{(i)}-\mu-\Lambda z^{(i)})^T\Psi^{-1}(x^{(i)}-\mu-\Lambda z^{(i)})\right]$$

去掉不相关的前两项后，对Λ进行导，

$$\nabla_\Lambda \sum_{i=1}^{m} -\mathrm{E}\left[\frac{1}{2}(x^{(i)}-\mu-\Lambda z^{(i)})^T\Psi^{-1}(x^{(i)}-\mu-\Lambda z^{(i)})\right]$$
$$= \sum_{i=1}^{m} \nabla_\Lambda \mathrm{E}\left[-\mathrm{tr}\frac{1}{2}z^{(i)T}\Lambda^T\Psi^{-1}\Lambda z^{(i)} + \mathrm{tr}z^{(i)T}\Lambda^T\Psi^{-1}(x^{(i)}-\mu)\right]$$
$$= \sum_{i=1}^{m} \nabla_\Lambda \mathrm{E}\left[-\mathrm{tr}\frac{1}{2}\Lambda^T\Psi^{-1}\Lambda z^{(i)}z^{(i)T} + \mathrm{tr}\Lambda^T\Psi^{-1}(x^{(i)}-\mu)z^{(i)T}\right]$$
$$= \sum_{i=1}^{m} \mathrm{E}\left[-\Psi^{-1}\Lambda z^{(i)}z^{(i)T} + \Psi^{-1}(x^{(i)}-\mu)z^{(i)T}\right]$$

第一步到第二步利用了 tr a = a（a 是实数时）和 tr AB = tr BA。最后一步利用了

$$\nabla_A \mathrm{tr}ABA^TC = CAB + C^TAB^T$$

tr 就是求一个矩阵对角线上元素和。

最后让其值为 0，并且化简得

$$\sum_{i=1}^{m} \Lambda \mathrm{E}_{z^{(i)}\sim Q_i}\left[z^{(i)}z^{(i)T}\right] = \sum_{i=1}^{m}(x^{(i)}-\mu)\mathrm{E}_{z^{(i)}\sim Q_i}\left[z^{(i)T}\right].$$

然后得到

$$\Lambda = \left(\sum_{i=1}^{m}(x^{(i)}-\mu)\mathrm{E}_{z^{(i)}\sim Q_i}\left[z^{(i)T}\right]\right)\left(\sum_{i=1}^{m}\mathrm{E}_{z^{(i)}\sim Q_i}\left[z^{(i)}z^{(i)T}\right]\right)^{-1}. \quad (7)$$

到这里我们发现，这个公式有点眼熟，与之前回归中的最小二乘法矩阵形式类似

$$\text{“}\theta^T = (y^TX)(X^TX)^{-1}.\text{”}$$

这里解释一下两者的相似性，我们这里的 x 是 z 的线性函数（包含了一定的噪声）。在 E 步得到 z 的估计后，我们找寻的Λ实际上是 x 和 z 的线性关系。而最小二乘法也是去找特征和结果直接的线性关系。

到这还没完，我们需要求得括号里面的值

根据我们之前对 z|x 的定义，我们知道

$$\begin{aligned}\mathrm{E}_{z^{(i)}\sim Q_i}\left[z^{(i)^T}\right] &= \mu_{z^{(i)}|x^{(i)}}^T \\ \mathrm{E}_{z^{(i)}\sim Q_i}\left[z^{(i)}z^{(i)^T}\right] &= \mu_{z^{(i)}|x^{(i)}}\mu_{z^{(i)}|x^{(i)}}^T + \Sigma_{z^{(i)}|x^{(i)}}.\end{aligned}$$

第一步根据 z 的条件分布得到，第二步根据$\mathrm{Cov}(Y) = \mathrm{E}[YY^T] - E[Y]E[Y]^T$得到
将上面的结果代入（7）中得到

$$\Lambda = \left(\sum_{i=1}^{m}(x^{(i)} - \mu)\mu_{z^{(i)}|x^{(i)}}^T\right)\left(\sum_{i=1}^{m}\mu_{z^{(i)}|x^{(i)}}\mu_{z^{(i)}|x^{(i)}}^T + \Sigma_{z^{(i)}|x^{(i)}}\right)^{-1}. \quad (8)$$

至此，我们得到了Λ，注意一点是 E[z]和$\mathrm{E}[zz^T]$的不同，后者需要求 z 的协方差。

其他参数的迭代公式如下：

$$\mu = \frac{1}{m}\sum_{i=1}^{m}x^{(i)}.$$

均值μ在迭代过程中值不变。

$$\Phi = \frac{1}{m}\sum_{i=1}^{m}x^{(i)}x^{(i)^T} - x^{(i)}\mu_{z^{(i)}|x^{(i)}}^T\Lambda^T - \Lambda\mu_{z^{(i)}|x^{(i)}}x^{(i)^T} + \Lambda(\mu_{z^{(i)}|x^{(i)}}\mu_{z^{(i)}|x^{(i)}}^T + \Sigma_{z^{(i)}|x^{(i)}})\Lambda^T$$

然后将Φ上的对角线上元素抽取出来放到对应的Ψ中，就得到了Ψ。

7 总结

根据上面的 EM 的过程，要对样本 X 进行因子分析，只需知道要分解的因子数（z 的维度）即可。通过 EM，我们能够得到转换矩阵Λ和误差协方差Ψ。

因子分析实际上是降维，在得到各个参数后，可以求得 z。但是 z 的各个参数含义需要自己去琢磨。

下面从一个 ppt 中摘抄几段话来进一步解释因子分析。

因子分析(factor analysis)是一种数据简化的技术。它通过研究众多变量之间的内部依赖关系，探求观测数据中的基本结构，并用少数几个假想变量来表示其基本的数据结构。这几个假想变量能够反映原来众多变量的主要信息。原始的变量是可观测的显在变量，而假想变量是不可观测的潜在变量，称为因子。

例如，在企业形象或品牌形象的研究中，消费者可以通过一个有 24 个指标构成的评价体系，评价百货商场的 24 个方面的优劣。

但消费者主要关心的是三个方面，即商店的环境、商店的服务和商品的价格。因子分析方法可以通过 24 个变量，找出反映商店环境、商店服务水平和商品价格的三个潜在的因子，对商店进行综合评价。而这三个公共因子可以表示为：

$$x_i = \mu_i + \alpha_{i1}F_1 + \alpha_{i2}F_2 + \alpha_{i3}F_3 + \varepsilon_i \quad i = 1, \cdots, 24$$

这里的x_i就是样例 x 的第 i 个分量，μ_i就是μ的第 i 个分量，α_{ij}就是Λ的第 i 行第 j 列元素，F_i是 z 的第 i 个分量，ε_i是$\varepsilon^{(i)}$。

称F_i是不可观测的潜在因子。24 个变量共享这三个因子，但是每个变量又有自己的个性，不被包含的部分ε_i，称为特殊因子。

注：

因子分析与回归分析不同，因子分析中的因子是一个比较抽象的概念，而回归因子有非常明确的实际意义；

主成分分析分析与因子分析也有不同，主成分分析仅仅是变量变换，而因子分析需要构造因子模型。

主成分分析:原始变量的线性组合表示新的综合变量，即主成分；

因子分析：潜在的假想变量和随机影响变量的线性组合表示原始变量。

PPT 地址

http://www.math.zju.edu.cn/webpagenew/uploadfiles/attachfiles/2008123195228555.ppt

其他值得参考的文献

An Introduction to Probabilistic Graphical Models by Jordan Chapter 14

增强学习（Reinforcement Learning and Control）

JerryLead
csxulijie@gmail.com

在之前的讨论中，我们总是给定一个样本 x，然后给或者不给 label y。之后对样本进行拟合、分类、聚类或者降维等操作。然而对于很多序列决策或者控制问题，很难有这么规则的样本。比如，四足机器人的控制问题，刚开始都不知道应该让其动那条腿，在移动过程中，也不知道怎么让机器人自动找到合适的前进方向。

另外如要设计一个下象棋的 AI，每走一步实际上也是一个决策过程，虽然对于简单的棋有 A*的启发式方法，但在局势复杂时，仍然要让机器向后面多考虑几步后才能决定走哪一步比较好，因此需要更好的决策方法。

对于这种控制决策问题，有这么一种解决思路。我们设计一个回报函数（reward function），如果 learning agent（如上面的四足机器人、象棋 AI 程序）在决定一步后，获得了较好的结果，那么我们给 agent 一些回报（比如回报函数结果为正），得到较差的结果，那么回报函数为负。比如，四足机器人，如果他向前走了一步（接近目标），那么回报函数为正，后退为负。如果我们能够对每一步进行评价，得到相应的回报函数，那么就好办了，我们只需要找到一条回报值最大的路径（每步的回报之和最大），就认为是最佳的路径。

增强学习在很多领域已经获得成功应用，比如自动直升机，机器人控制，手机网络路由，市场决策，工业控制，高效网页索引等。

接下来，先介绍一下马尔科夫决策过程（MDP，Markov decision processes）。

1. 马尔科夫决策过程

一个马尔科夫决策过程由一个五元组构成$(S, A, \{P_{sa}\}, \gamma, R)$

- S 表示状态集（states）。（比如，在自动直升机系统中，直升机当前位置坐标组成状态集）
- A 表示一组动作（actions）。（比如，使用控制杆操纵的直升机飞行方向，让其向前，向后等）
- P_{sa}是状态转移概率。S 中的一个状态到另一个状态的转变，需要 A 来参与。P_{sa}表示的是在当前$s \in S$状态下，经过$a \in A$作用后，会转移到的其他状态的概率分布情况（当前状态执行 a 后可能跳转到很多状态）。
- $\gamma \in [0,1)$是阻尼系数（discount factor）
- $R: S \times A \mapsto \mathbb{R}$，R 是回报函数（reward function），回报函数经常写作 S 的函数（只与 S 有关），这样的话，R 重新写作$R: S \mapsto \mathbb{R}$。

MDP 的动态过程如下：某个 agent 的初始状态为s_0，然后从 A 中挑选一个动作a_0执行，执行后，agent 按P_{sa}概率随机转移到了下一个s_1状态，$s_1 \in P_{s_0 a_0}$。然后再执行一个动作a_1，就转移到了s_2，接下来再执行a_2…，我们可以用下面的图表示整个过程

$$s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} s_2 \xrightarrow{a_2} s_3 \xrightarrow{a_3} \dots$$

如果对 HMM 有了解的话，理解起来比较轻松。

我们定义经过上面转移路径后，得到的回报函数之和如下

$$R(s_0, a_0) + \gamma R(s_1, a_1) + \gamma^2 R(s_2, a_2) + \cdots .$$

如果 R 只和 S 有关，那么上式可以写作

$$R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \cdots .$$

我们的目标是选择一组最佳的 action，使得全部的回报加权和期望最大。

$$\mathrm{E}\left[R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \cdots\right]$$

从上式可以发现，在 t 时刻的回报值被打了γ^t的折扣，是一个逐步衰减的过程，越靠后的状态对回报和影响越小。最大化期望值也就是要将大的R(s_i)尽量放到前面，小的尽量放到后面。

已经处于某个状态 s 时，我们会以一定策略π来选择下一个动作 a 执行，然后转换到另一个状态 s′。我们将这个动作的选择过程称为策略（policy），每一个 policy 其实就是一个状态到动作的映射函数π：S ⟼ A。给定π也就给定了a = π(s)，也就是说，知道了π就知道了每个状态下一步应该执行的动作。

我们为了区分不同π的好坏，并定义在当前状态下，执行某个策略π后，出现的结果的好坏，需要定义值函数（value function）也叫折算累积回报（discounted cumulative reward）

$$V^{\pi}(s) = \mathrm{E}\left[R(s_0) + \gamma R(s_1) + \gamma^2 R(s_2) + \cdots \,\middle|\, s_0 = s, \pi\right].$$

可以看到，在当前状态 s 下，选择好 policy 后，值函数是回报加权和期望。这个其实很容易理解，给定π也就给定了一条未来的行动方案，这个行动方案会经过一个个的状态，而到达每个状态都会有一定回报值，距离当前状态越近的其他状态对方案的影响越大，权重越高。这和下象棋差不多，在当前棋局s_0下，不同的走子方案是π，我们评价每个方案依靠对未来局势（R(s_1), R(s_2),…）的判断。一般情况下，我们会在头脑中多考虑几步，但是我们会更看重下一步的局势。

从递推的角度上考虑，当期状态 s 的值函数 V，其实可以看作是当前状态的回报 R(s)和下一状态的值函数 V′之和，也就是将上式变为：

$$V^{\pi}(s) = R(s_0) + \gamma(E[R(s_1) + \gamma R(s_2) + \gamma^2 R(s_3) + \cdots]) = R(s_0) + \gamma V^{\pi}(s')$$

然而，我们需要注意的是虽然给定π后，在给定状态 s 下，a 是唯一的，但A ⟼ S可能不是多到一的映射。比如你选择 a 为向前投掷一个骰子，那么下一个状态可能有 6 种。再由 Bellman 等式，从上式得到

$$V^{\pi}(s) = R(s) + \gamma \sum_{s' \in S} P_{s\pi(s)}(s') V^{\pi}(s').$$

s′表示下一个状态。

前面的 R(s)称为立即回报（immediate reward），就是 R(当前状态)。第二项也可以写作

$E_{s\prime\sim P_{s\pi(s)}}[V^{\pi}(s')]$，是下一状态值函数的期望值，下一状态 s'符合$P_{s\pi(s)}$分布。

可以想象，当状态个数有限时，我们可以通过上式来求出每一个 s 的 V（终结状态没有第二项 V(s')）。如果列出线性方程组的话，也就是|S|个方程，|S|个未知数，直接求解即可。

当然，我们求 V 的目的就是想找到一个当前状态 s 下，最优的行动策略π，定义最优的 V*如下：

$$V^{*}(s) = \max_{\pi} V^{\pi}(s)$$

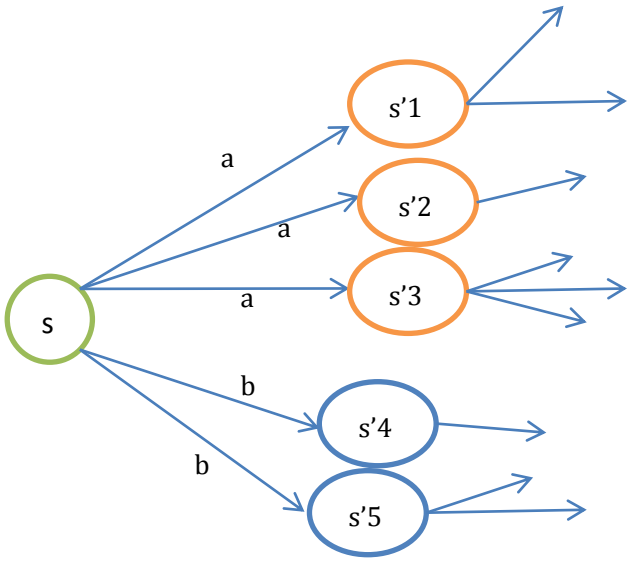
就是从可选的策略π中挑选一个最优的策略（discounted rewards 最大）。

上式的 Bellman 等式形式如下：

$$V^{*}(s) = R(s) + \max_{a\in A} \gamma \sum_{s'\in S} P_{sa}(s')V^{*}(s'). \qquad (2)$$

第一项与π无关，所以不变。第二项是一个π就决定了每个状态 s 的下一步动作 a，执行 a 后，s'按概率分布的回报概率和的期望。

如果上式还不好理解的话，可以参考下图：



定义了最优的 V*，我们再定义最优的策略π^{*}: S ⟼ A 如下：

$$\pi^{*}(s) = \arg\max_{a\in A} \sum_{s'\in S} P_{sa}(s')V^{*}(s'). \qquad (3)$$

选择最优的π^{*}，也就确定了每个状态 s 的下一步最优动作 a。

根据以上式子，我们可以知道

$$V^{*}(s) = V^{\pi^{*}}(s) \geq V^{\pi}(s).$$

解释一下就是当前状态的最优的值函数 V*，是由采用最优执行策略π^{*}的情况下得出的，采用最优执行方案的回报显然要比采用其他的执行策略π要好。

这里需要注意的是，如果我们能够求得每个 s 下最优的 a，那么从全局来看，S ⟼ A的映射即可生成，而生成的这个映射是最优映射，称为π^*。π^*针对全局的 s，确定了每一个 s 的下一个行动 a，不会因为初始状态 s 选取的不同而不同。

2. 值迭代和策略迭代法

上节我们给出了迭代公式和优化目标，这节讨论两种求解有限状态 MDP 具体策略的有效算法。这里，我们只针对 MDP 是有限状态、有限动作的情况，$|S| < \infty, |A| < \infty$。

- 值迭代法

1、 将每一个 s 的 V(s)初始化为 0
2、 循环直到收敛 {
对于每一个状态 s，对 V(s)做更新

$$V(s) := R(s) + \max_{a \in A} \gamma \sum_{s'} P_{sa}(s')V(s')$$

}

值迭代策略利用了上节中公式（2）
内循环的实现有两种策略：

1、 同步迭代法
拿初始化后的第一次迭代来说吧，初始状态所有的 V(s)都为 0。然后对所有的 s 都计算新的 V(s)=R(s)+0=R(s)。在计算每一个状态时，得到新的 V(s)后，先存下来，不立即更新。待所有的 s 的新值 V(s)都计算完毕后，再统一更新。
这样，第一次迭代后，V(s)=R(s)。

2、 异步迭代法
与同步迭代对应的就是异步迭代了，对每一个状态 s，得到新的 V(s)后，不存储，直接更新。这样，第一次迭代后，大部分 V(s)>R(s)。

不管使用这两种的哪一种，最终 V(s)会收敛到 V*(s)。知道了 V*后，我们再使用公式（3）来求出相应的最优策略π^*，当然π^*可以在求 V*的过程中求出。

- 策略迭代法

值迭代法使 V 值收敛到 V*，而策略迭代法关注π，使π收敛到π^*。

1、 将随机指定一个 S 到 A 的映射π。
2、 循环直到收敛 {
（a） 令$V := V^{\pi}$
（b） 对于每一个状态 s，对π(s)做更新

$$\pi(s) := arg \max_{a \in A} \sum_{s'} P_{sa}(s')V(s')$$

}

(a)步中的 V 可以通过之前的 Bellman 等式求得

$$V^{\pi}(s) = R(s) + \gamma \sum_{s' \in S} P_{s\pi(s)}(s') V^{\pi}(s').$$

这一步会求出所有状态 s 的V^{π}(s)。

(b)步实际上就是根据(a)步的结果挑选出当前状态 s 下，最优的 a，然后对π(s)做更新。

对于值迭代和策略迭代很难说哪种方法好，哪种不好。对于规模比较小的 MDP 来说，策略一般能够更快地收敛。但是对于规模很大（状态很多）的 MDP 来说，值迭代比较容易（不用求线性方程组）。

3. MDP 中的参数估计

在之前讨论的 MDP 中，我们是已知状态转移概率P_{sa}和回报函数 R(s)的。但在很多实际问题中，这些参数不能显式得到，我们需要从数据中估计出这些参数（通常 S、A 和γ是已知的）。

假设我们已知很多条状态转移路径如下：

$$s_0^{(1)} \xrightarrow{a_0^{(1)}} s_1^{(1)} \xrightarrow{a_1^{(1)}} s_2^{(1)} \xrightarrow{a_2^{(1)}} s_3^{(1)} \xrightarrow{a_3^{(1)}} \dots$$
$$s_0^{(2)} \xrightarrow{a_0^{(2)}} s_1^{(2)} \xrightarrow{a_1^{(2)}} s_2^{(2)} \xrightarrow{a_2^{(2)}} s_3^{(2)} \xrightarrow{a_3^{(2)}} \dots$$
$$\dots$$

其中，$s_i^{(j)}$是 i 时刻，第 j 条转移路径对应的状态，$a_i^{(j)}$是$s_i^{(j)}$状态时要执行的动作。每个转移路径中状态数是有限的，在实际操作过程中，每个转移链要么进入终结状态，要么达到规定的步数就会终结。

如果我们获得了很多上面类似的转移链（相当于有了样本），那么我们就可以使用最大似然估计来估计状态转移概率。

$$P_{sa}(s') = \frac{\#\text{times took we action } a \text{ in state } s \text{ and got to } s'}{\#\text{times we took action a in state } s} \qquad (4)$$

分子是从 s 状态执行动作 a 后到达 s′的次数，分母是在状态 s 时，执行 a 的次数。两者相除就是在 s 状态下执行 a 后，会转移到 s′的概率。

为了避免分母为 0 的情况，我们需要做平滑。如果分母为 0，则令$P_{sa}(s') = 1/|S|$，也就是说当样本中没有出现过在 s 状态下执行 a 的样例时，我们认为转移概率均分。

上面这种估计方法是从历史数据中估计，这个公式同样适用于在线更新。比如我们新得到了一些转移路径，那么对上面的公式进行分子分母的修正（加上新得到的 count）即可。修正过后，转移概率有所改变，按照改变后的概率，可能出现更多的新的转移路径，这样P_{sa}会越来越准。

同样，如果回报函数未知，那么我们认为 R(s)为在 s 状态下已经观测到的回报均值。

当转移概率和回报函数估计出之后，我们可以使用值迭代或者策略迭代来解决 MDP 问

题。比如，我们将参数估计和值迭代结合起来（在不知道状态转移概率情况下）的流程如下：

1、 随机初始化π
2、 循环直到收敛 {
    (a) 在样本上统计π中每个状态转移次数，用来更新P_{sa}和 R
    (b) 使用估计到的参数来更新 V（使用上节的值迭代方法）
    (c) 根据更新的 V 来重新得出π
}

在(b)步中我们要做值更新，也是一个循环迭代的过程，在上节中，我们通过将 V 初始化为 0，然后进行迭代来求解 V。嵌套到上面的过程后，如果每次初始化 V 为 0，然后迭代更新，就会很慢。一个加快速度的方法是每次将 V 初始化为上一次大循环中得到的 V。也就是说 V 的初值衔接了上次的结果。

4. 总结

首先我们这里讨论的 MDP 是非确定的马尔科夫决策过程，也就是回报函数和动作转换函数是有概率的。在状态 s 下，采取动作 a 后的转移到的下一状态 s’也是有概率的。再次，在增强学习里有一个重要的概念是 Q 学习，本质是将与状态 s 有关的 V(s)转换为与 a 有关的 Q。强烈推荐 Tom Mitchell 的《机器学习》最后一章，里面介绍了 Q 学习和更多的内容。最后，里面提到了 Bellman 等式，在《算法导论》中有 Bellman-Ford 的动态规划算法，可以用来求解带负权重的图的最短路径，里面最值得探讨的是收敛性的证明，非常有价值。有学者仔细分析了增强学习和动态规划的关系。

这篇是 ng 讲义中最后一篇了，还差一篇 learning theory，暂时不打算写了，感觉对 learning 的认识还不深。等到学习完图模型和在线学习等内容后，再回过头来写 learning theory 吧。另外，ng 的讲义中还有一些数学基础方面的讲义比如概率论、线性代数、凸优化、高斯过程、HMM 等，都值得看一下。

典型关联分析（Canonical Correlation Analysis）

JerryLead
csxulijie@gmail.com

1. 问题

在线性回归中，我们使用直线来拟合样本点，寻找 n 维特征向量 X 和输出结果（或者叫做 label）Y 之间的线性关系。其中$X \in R^n$，$Y \in R$。然而当 Y 也是多维时，或者说 Y 也有多个特征时，我们希望分析出 X 和 Y 的关系。

当然我们仍然可以使用回归的方法来分析，做法如下：

假设$X \in R^n$，$Y \in R^m$，那么可以建立等式 Y=AX 如下

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1n} \\ w_{21} & w_{22} & \cdots & w_{2n} \\ \vdots & \vdots & \vdots & \vdots \\ w_{m1} & w_{m2} & \cdots & w_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

其中$y_i = w_i^T x$，形式和线性回归一样，需要训练 m 次得到 m 个w_i。

这样做的一个缺点是，Y 中的每个特征都与 X 的所有特征关联，Y 中的特征之间没有什么联系。

我们想换一种思路来看这个问题，如果将 X 和 Y 都看成整体，考察这两个整体之间的关系。我们将整体表示成 X 和 Y 各自特征间的线性组合，也就是考察$a^T x$和$b^T y$之间的关系。

这样的应用其实很多，举个简单的例子。我们想考察一个人解题能力 X（解题速度x_1，解题正确率x_2）与他/她的阅读能力 Y（阅读速度y_1，理解程度y_2）之间的关系，那么形式化为：

$u = a_1 x_1 + a_2 x_2$ 和 $v = b_1 y_1 + b_2 y_2$

然后使用 Pearson 相关系数

$$\rho_{X,Y} = \operatorname{corr}(X,Y) = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y},$$

来度量 u 和 v 的关系，我们期望寻求一组最优的解 a 和 b，使得 Corr(u, v)最大，这样得到的 a 和 b 就是使得 u 和 v 就有最大关联的权重。

到这里，基本上介绍了典型相关分析的目的。

2. CCA 表示与求解

给定两组向量x_1和x_2（替换之前的 x 为x_1，y 为x_2），x_1维度为p_1，x_2维度为p_2，默认$p_1 \le p_2$。形式化表示如下：

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \quad E[x] = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} \quad \Sigma = Var(x) = \begin{bmatrix} \Sigma_{11} & \Sigma_{12} \\ \Sigma_{21} & \Sigma_{22} \end{bmatrix}$$

Σ是 x 的协方差矩阵；左上角是x_1自己的协方差矩阵；右上角是$\mathrm{Cov}(x_1, x_2)$；左下角是$\mathrm{Cov}(x_2, x_1)$，也是Σ_{12}的转置；右下角是x_2的协方差矩阵。

与之前一样，我们从x_1和x_2的整体入手，定义

$$u = a^T x_1 \quad v = b^T x_2$$

我们可以算出 u 和 v 的方差和协方差：

$$\mathrm{Var}(u) = a^T \Sigma_{11} a \qquad \mathrm{Var}(v) = b^T \Sigma_{22} b \qquad \mathrm{Cov}(u, v) = a^T \Sigma_{12} b$$

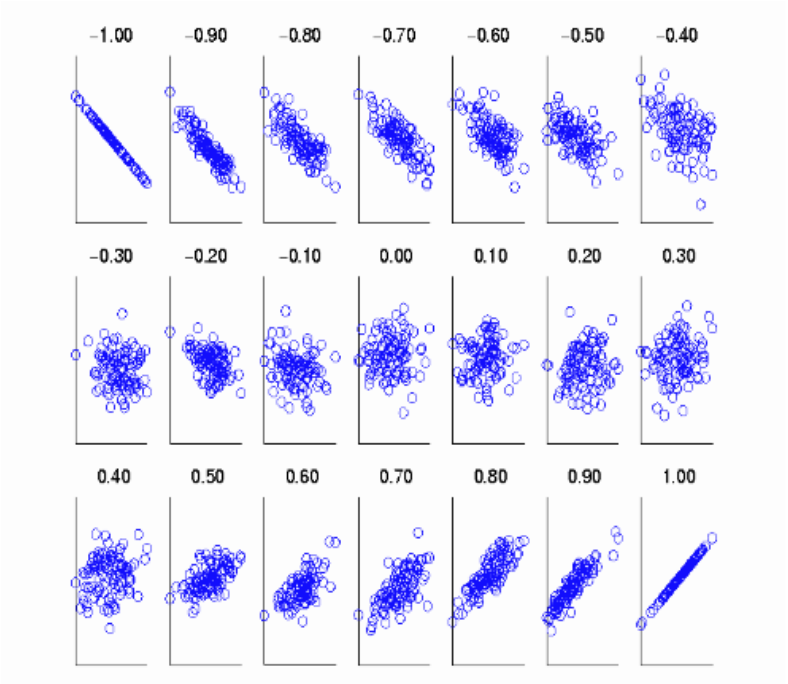
上面的结果其实很好算，推导一下第一个吧：

$$\mathrm{Var}(u) = \mathrm{Var}(a^T x_1) = \frac{1}{N}\sum_{i=1}^{N}(a^T x_1 - a^T \mu_1)^2 = a^T \frac{1}{N}\sum_{i=1}^{N}(x_1 - \mu_1)^2 \, a = a^T \Sigma_{11} a$$

最后，我们需要算 Corr(u,v)了

$$\mathrm{Corr}(u, v) = \frac{a^T \Sigma_{12} b}{\sqrt{a^T \Sigma_{11} a}\sqrt{b^T \Sigma_{22} b}}$$

我们期望 Corr(u,v)越大越好，关于 Pearson 相关系数，《数据挖掘导论》给出了一个很好的图来说明：



Scatter plots showing the similarity from –1 to 1.

横轴是 u，纵轴是 v，这里我们期望通过调整 a 和 b 使得 u 和 v 的关系越像最后一个图越好。其实第一个图和最后一个图有联系的，我们可以调整 a 和 b 的符号，使得从第一个图变为最后一个。

接下来我们求解 a 和 b。

回想在 LDA 中，也得到了类似 Corr(u,v)的公式，我们在求解时固定了分母，来求分子（避免 a 和 b 同时扩大 n 倍仍然符号解条件的情况出现）。这里我们同样这么做。

这个优化问题的条件是：

Maximize $a^T\Sigma_{12}b$
Subject to: $a^T\Sigma_{11}a = 1, b^T\Sigma_{22}b = 1$

求解方法是构造 Lagrangian 等式，这里我简单推导如下：

$$\mathcal{L} = a^T\Sigma_{12}b - \frac{\lambda}{2}(a^T\Sigma_{11}a - 1) - \frac{\theta}{2}(b^T\Sigma_{22}b - 1)$$

求导，得

$$\frac{\partial\mathcal{L}}{\partial a} = \Sigma_{12}b - \lambda\Sigma_{11}a$$
$$\frac{\partial\mathcal{L}}{\partial b} = \Sigma_{21}a - \theta\Sigma_{22}b$$

令导数为 0 后，得到方程组：

$$\Sigma_{12}b - \lambda\Sigma_{11}a = 0$$
$$\Sigma_{21}a - \theta\Sigma_{22}b = 0$$

第一个等式左乘a^T，第二个左乘b^T，再根据$a^T\Sigma_{11}a = 1, b^T\Sigma_{22}b = 1$，得到

$$\lambda = \theta = a^T\Sigma_{12}b$$

也就是说求出的λ即是 Corr(u,v)，只需找最大λ即可。

让我们把上面的方程组进一步简化，并写成矩阵形式，得到

$$\Sigma_{11}^{-1}\Sigma_{12}b = \lambda a$$
$$\Sigma_{22}^{-1}\Sigma_{21}a = \lambda b$$

写成矩阵形式

$$\begin{bmatrix}\Sigma_{11}^{-1} & 0\\ 0 & \Sigma_{22}^{-1}\end{bmatrix}\begin{bmatrix}0 & \Sigma_{12}\\ \Sigma_{21} & 0\end{bmatrix}\begin{bmatrix}a\\ b\end{bmatrix} = \lambda\begin{bmatrix}a\\ b\end{bmatrix}$$

令

$$B = \begin{bmatrix}\Sigma_{11} & 0\\ 0 & \Sigma_{22}\end{bmatrix}, \mathrm{A} = \begin{bmatrix}0 & \Sigma_{12}\\ \Sigma_{21} & 0\end{bmatrix}, \mathrm{w} = \begin{bmatrix}a\\ b\end{bmatrix}$$

那么上式可以写作：

$$B^{-1}\mathrm{A}w = \lambda w$$

显然，又回到了求特征值的老路上了，只要求得B^{-1}A的最大特征值$\lambda_{\max}$，那么 Corr(u,v) 和 a 和 b 都可以求出。

在上面的推导过程中，我们假设了Σ_{11}和Σ_{22}均可逆。一般情况下都是可逆的，只有存在特征间线性相关时会出现不可逆的情况，在本文最后会提到不可逆的处理办法。

再次审视一下，如果直接去计算B^{-1}A的特征值，复杂度有点高。我们将第二个式子代入第一个，得

$$\Sigma_{11}^{-1}\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}a = \lambda^2 a$$

这样先对$\Sigma_{11}^{-1}\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$求特征值$\lambda^2$和特征向量$a$，然后根据第二个式子求得 b。

待会举个例子说明求解过程。

假设按照上述过程，得到了λ最大时的a_1和b_1。那么a_1和b_1称为典型变量（canonical variates），λ即是 u 和 v 的相关系数。

最后，我们得到 u 和 v 的等式为：

$$\mathrm{u} = a_1^{\mathrm{T}}\mathrm{x}_1 \quad \mathrm{v} = b_1^T\mathrm{x}_2$$

我们也可以接着去寻找第二组典型变量对，其最优化条件是

Maximize $a_2^{\mathrm{T}}\Sigma_{12}b_2$
Subject to: $a_2^{\mathrm{T}}\Sigma_{11}a_2 = 1, b_2^T\Sigma_{22}b_2 = 1$
$a_2^{\mathrm{T}}\Sigma_{11}a_1 = 0, b_2^T\Sigma_{22}b_1 = 0$

其实第二组约束条件就是$\mathrm{Cov}(u_2, u_1) = 0, \mathrm{Cov}(v_2, v_1) = 0$。

计算步骤同第一组计算方法，只不过是λ取$\Sigma_{11}^{-1}\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$的第二大特征值。
得到的a_2和b_2其实也满足

$$a_2^{\mathrm{T}}\Sigma_{12}b_1 = 0, b_2^T\Sigma_{21}a_1 = 0 \text{ 即 } \mathrm{Cov}(u_2, v_1) = 0, \mathrm{Cov}(v_2, u_1) = 0$$

总结一下，i 和 j 分别表示λ_i和λ_j得到结果

$$\mathrm{Corr}(u_i, v_i) = \lambda_i \quad \mathrm{Corr}(u_i, u_j) = 0$$

$$\mathrm{Corr}(v_i, v_j) = 0 \quad \mathrm{Corr}(u_i, v_j) = 0 (i \neq j)$$

3. CCA 计算例子

我们回到之前的评价一个人解题和其阅读能力的关系的例子。假设我们通过对样本计算协方差矩阵得到如下结果：

$$\Sigma = \begin{bmatrix} 1 & .4 & .5 & .6 \\ .4 & 1 & .3 & .4 \\ .5 & .3 & 1 & .2 \\ .6 & .4 & .2 & 1 \end{bmatrix}$$

$$\Sigma_{11} = \begin{bmatrix} 1 & .4 \\ .4 & 1 \end{bmatrix} \quad \Sigma_{12} = \begin{bmatrix} .5 & .6 \\ .3 & .4 \end{bmatrix} \quad \Sigma_{21} = \begin{bmatrix} .5 & .3 \\ .6 & .4 \end{bmatrix} \quad \Sigma_{22} = \begin{bmatrix} 1 & .2 \\ .2 & 1 \end{bmatrix}$$

然后求$\Sigma_{11}^{-1}\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$，得

$$\mathrm{A} = \Sigma_{11}^{-1}\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21} = \begin{bmatrix} .452 & .289 \\ .146 & .495 \end{bmatrix}$$

这里的 A 和前面的$B^{-1}\mathrm{A}w = \lambda w$中的 A 不是一回事（这里符号有点乱，不好意思）。
然后对 A 求特征值和特征向量，得到

$$\lambda_1^2 = .5457 \quad \lambda_2^2 = .0009 \quad \mathrm{Vec}\mathbf{A} = \begin{bmatrix} .951 & -.540 \\ .309 & .842 \end{bmatrix}$$

然后求 b，之前我们说的方法是根据$\Sigma_{22}^{-1}\Sigma_{21}a = \lambda b$ 求 b，这里，我们也可以采用类似求 a 的方法来求 b。
回想之前的等式

$$\Sigma_{11}^{-1}\Sigma_{12}b = \lambda a$$
$$\Sigma_{22}^{-1}\Sigma_{21}a = \lambda b$$

我们将上面的式子代入下面的，得

$$\Sigma_{22}^{-1}\Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}b = \lambda^2 b$$

然后直接对$\Sigma_{22}^{-1}\Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}$求特征向量即可，注意$\Sigma_{22}^{-1}\Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12}$和$\Sigma_{11}^{-1}\Sigma_{12}\Sigma_{22}^{-1}\Sigma_{21}$的特征值相同，这个可以自己证明下。

不管使用哪种方法，

$$\mathrm{B} = \Sigma_{22}^{-1}\Sigma_{21}\Sigma_{11}^{-1}\Sigma_{12} = \begin{bmatrix} .206 & .251 \\ .278 & .340 \end{bmatrix}$$

$$\mathrm{Vec}\mathbf{B} = \begin{bmatrix} .595 & -.774 \\ .804 & .633 \end{bmatrix}$$

这里我们得到 a 和 b 的两组向量，到这还没完，我们需要让它们满足之前的约束条件

$$a_i^{\mathrm{T}}\Sigma_{11}a_i = 1, b_i^T\Sigma_{22}b_i = 1$$

这里的a_i应该是我们之前得到的 VecA 中的列向量的 m 倍，我们只需要求得 m，然后将 VecA 中的列向量乘以 m 即可。

$$\mathrm{m}^2 a'^{\mathrm{T}}_i \mathrm{R}_{11} a'_i = 1$$

这里的a'_i是 VecA 的列向量。

$$\mathbf{A} = \mathbf{VecA}\begin{pmatrix} 1.23 & 0 \\ 0 & .636 \end{pmatrix}^{-\frac{1}{2}} \text{ and } \quad \mathbf{B} = \mathbf{VecB}\begin{pmatrix} 1.19 & 0 \\ 0 & .804 \end{pmatrix}^{-\frac{1}{2}}$$

因此最后的 a 和 b 为：

$$\mathbf{A} = \begin{bmatrix} .856 & \text{-}.677 \\ .278 & 1.055 \end{bmatrix} \text{ and } \quad \mathbf{B} = \begin{bmatrix} .545 & \text{-}.863 \\ .737 & .706 \end{bmatrix}$$

第一组典型变量为

$$\mathrm{u}_1 = .856z_1 + .278z_2 \qquad v_1 = .545z_3 + .737z_4$$

相关系数

$$\mathrm{Corr}(u_1, v_1) = \sqrt{\lambda_1^2} = \sqrt{.5457} = .74$$

第二组典型变量为

$$\mathrm{u}_2 = -.677z_1 + 1.055z_2 \qquad v_1 = -.863z_3 + .706z_4$$

相关系数

$$\mathrm{Corr}(u_2, v_2) = \sqrt{\lambda_2^2} = \sqrt{.0009} = .03$$

这里的z_1（解题速度），z_2（解题正确率），z_3（阅读速度），z_4（阅读理解程度）。他们前面的系数意思不是特征对单个 u 或 v 的贡献比重，而是从 u 和 v 整体关系看，当两者关系最密切时，特征计算时的权重。

4. Kernel Canonical Correlation Analysis（KCCA）

通常当我们发现特征的线性组合效果不够好或者两组集合关系是非线性的时候，我们会尝试核函数方法，这里我们继续介绍 Kernel CCA。

在《支持向量机-核函数》那一篇中，大致介绍了一下核函数，这里再简单提一下：

当我们对两个向量作内积的时候

$$\langle x, y\rangle = \sum x_i\, y_i$$

我们可以使用$\Phi(x)$，$\Psi(y)$来替代x和y，比如原来的x特征向量为$(x_1, x_2, x_3)^T$，那么

我们可以定义

$$\phi(x) = \begin{bmatrix} x_1x_1 \\ x_1x_2 \\ x_1x_3 \\ x_2x_1 \\ x_2x_2 \\ x_2x_3 \\ x_3x_1 \\ x_3x_2 \\ x_3x_3 \end{bmatrix}.$$

如果$\Psi(y)$与$\Phi(x)$的构造一样，那么

$$\langle \Phi(x), \Phi(y)\rangle = \sum_{i=1}^{n}\sum_{j=1}^{n}(x_ix_j)(y_iy_j) = \sum_{i=1}^{n}\sum_{j=1}^{n}x_iy_ix_jy_j = \sum_{i=1}^{n}(x_i\, y_i)\sum_{j=1}^{n}(x_jy_j)$$

$$= (x^Ty)^2 = K\langle x, y\rangle$$

这样，仅通过计算 x 和 y 的内积的平方就可以达到在高维空间（这里为n^2）中计算$\Phi(x)$和$\Phi(y)$内积的效果。

由核函数，我们可以得到核矩阵 K，其中

$$\mathrm{K}_{i,j} = K(x^{(i)}, y^{(i)})$$

即第i行第j列的元素是第i个和第j个样例在核函数下的内积。

一个很好的核函数定义：

$$\phi : \mathbf{x} = (\mathbf{x}_1, \ldots \mathbf{x}_n) \mapsto \phi(\mathbf{x}) = (\phi_1(\mathbf{x}), \ldots, \phi_N(\mathbf{x}))\ (n < N)$$

其中样例 x 有 n 个特征，经过$\Phi(x)$变换后，从 n 维特征上升到了 N 维特征，其中每一个特征是$\phi_i(x) = f(x_1, x_2, \ldots, x_n)$。

回到 CCA，我们在使用核函数之前

$$\mathrm{u} = a^Tx \qquad \mathrm{v} = b^Ty$$

这里假设 x 和 y 都是 n 维的，引入核函数后，$\phi_x(x)$和$\phi_y(y)$变为了 N 维。

使用核函数后，u 和 v 的公式为：

$$\mathrm{u} = c^T\phi_x(x) \qquad \mathrm{v} = d^T\phi_y(y)$$

这里的 c 和 d 都是 N 维向量。
现在我们有样本$\{(x_i, y_i)\}_{i=1}^{M}$，这里的x_i表示样本 x 的第 i 个样例，是 n 维向量。
根据前面说过的相关系数，构造拉格朗日公式如下：

$$\begin{aligned} L_0 \quad = \quad & \mathrm{E}[(u-\mathrm{E}[u])(v-\mathrm{E}[v])] \\ & -\frac{\lambda_1}{2}\mathrm{E}[(u-\mathrm{E}[u])^2] \\ & -\frac{\lambda_2}{2}\mathrm{E}[(v-\mathrm{E}[v])^2]. \end{aligned} \tag{7}$$

其中

$$\mathrm{E[u]} = \frac{1}{M}\sum_i c^T \phi_x(x_i)$$

$$\mathrm{E[uv]} = \frac{1}{M}\sum_i c^T \phi_x(x_i)\, d^T \phi_y(y_i)$$

然后让 L 对 a 求导，令导数等于 0，得到（这一步我没有验证，待会从宏观上解释一下）

$$c = \sum_i \alpha_i \phi_x(x_i)$$

同样对 b 求导，令导数等于 0，得到

$$d = \sum_i \beta_i \phi_y(y_i)$$

求出 c 和 d 干嘛呢？c 和 d 只是ϕ的系数而已，按照原始的 CCA 做法去做就行了呗，为了再引入α和β？

回答这个问题要从核函数的意义上来说明。核函数初衷是希望在式子中有$\phi^T(x)\phi(y)$，然后用 K 替换之，根本没有打算去计算出实际的ϕ。因此即是按照原始 CCA 的方式计算出了 c 和 d，也是没用的，因为根本有没有实际的ϕ让我们去做$c^T\phi(x)$。另一个原因是核函数比如高斯径向基核函数可以上升到无限维，N 是无穷的，因此 c 和 d 也是无穷维的，根本没办法直接计算出来。我们的思路是在原始的空间中构造出权重α和β，然后利用ϕ将α和β上升到高维，他们在高维对应的权重就是 c 和 d。

虽然α和β是在原始空间中（维度为样例个数 M），但其作用点不是在原始特征上，而是原始样例上。看上面得出的 c 和 d 的公式就知道。α通过控制每个高维样例的权重，来控制 c。

好了，接下来我们看看使用α和β后，u 和 v 的变化

$$\mathrm{u} = \langle \mathrm{c}, \phi(x) \rangle = \sum_i \alpha_i \langle \phi_x(x_i), \phi_x(x) \rangle$$

$$\mathrm{v} = \langle \mathrm{d}, \phi(y) \rangle = \sum_i \beta_i \langle \phi_y(y_i), \phi_y(y) \rangle$$

$\phi_x(x_i)$表示可以将第 i 个样例上升到的 N 维向量，$\phi_x(x)$意义可以类比原始 CCA 的 x。鉴于这样表示接下来会越来越复杂，改用矩阵形式表示。

$$\begin{bmatrix} \mathrm{c}_1 \\ \mathrm{c}_2 \\ \vdots \\ \mathrm{c}_N \end{bmatrix} = \begin{bmatrix} | & | & | & | \\ \phi_x(x_1) & \phi_x(x_2) & \cdots & \phi_x(x_m) \\ | & | & | & | \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_m \end{bmatrix}$$

简写为

$$\mathrm{c} = \mathrm{X}^T \alpha$$

其中 X（M×N）为

$$\begin{bmatrix} - & \phi_x^T(x_1) & - \\ - & \phi_x^T(x_2) & - \\ - & \vdots & - \\ - & \phi_x^T(x_m) & - \end{bmatrix}$$

我们发现

$$K_x = XX^T$$

我们可以算出 u 和 v 的方差和协方差（这里实际上事先对样本x和y做了均值归 0 处理）：

$$\mathrm{Var(u)} = c^T Var\big(\phi_x(x)\big)c = c^T X^T Xc = \alpha^T XX^T XX^T \alpha = \alpha^T K_x K_x \alpha$$

$$\mathrm{Var(v)} = \beta^T \mathrm{K}_y \mathrm{K}_y \beta$$

$$\mathrm{Cov(u,v)} = c^T Cov\left(\phi_x(x), \phi_y(y)\right) d = c^T X^T Y d = \alpha^T XX^T YY^T \beta = \alpha^T K_x \mathrm{K}_y \beta$$

这里$\phi_x(x)$和$\phi_y(y)$维度可以不一样。

最后，我们得到 Corr(u,v)

$$\mathrm{Corr(u,v)} = \frac{\alpha^{\mathrm{T}} \mathrm{K}_{\mathrm{x}} K_y \beta}{\sqrt{\alpha^{\mathrm{T}} \mathrm{K}_{\mathrm{x}} \mathrm{K}_{\mathrm{x}} \alpha} \sqrt{\beta^{\mathrm{T}} K_y K_y \beta}}$$

可以看到，在将x_1和x_2处理成$\mathrm{E}[x_1] = 0$，$\mathrm{E}[x_2] = 0$后，得到的结果和之前形式基本一样，只是将Σ替换成了两个 K 乘积。

因此，得到的结果也是一样的，之前是

$$B^{-1} \mathrm{A} w = \lambda w$$

其中

$$B = \begin{bmatrix} \Sigma_{11} & 0 \\ 0 & \Sigma_{22} \end{bmatrix}, \mathrm{A} = \begin{bmatrix} 0 & \Sigma_{12} \\ \Sigma_{21} & 0 \end{bmatrix}, \mathrm{w} = \begin{bmatrix} a \\ b \end{bmatrix}$$

引入核函数后，得到

$$B^{-1} \mathrm{A} w = \lambda w$$

其中

$$B = \begin{bmatrix} K_x K_x & 0 \\ 0 & K_y K_y \end{bmatrix}, \mathrm{A} = \begin{bmatrix} 0 & K_x K_y \\ K_y K_x & 0 \end{bmatrix}, \mathrm{w} = \begin{bmatrix} \alpha \\ \beta \end{bmatrix}$$

注意这里的两个 w 有点区别，前面的a维度和 x 的特征数相同，b维度和 y 的特征数相同。后面的α维度和 x 的样例数相同，β维度和 y 的样例数相同，严格来说“α维度=β维度”。

5. 其他话题

1、当协方差矩阵不可逆时，怎么办？

要进行 regularization。

一种方法是将前面的 KCCA 中的拉格朗日等式加上二次正则化项，即：

$$L = L_0 + \frac{\eta}{2}(\|c\|^2 + \|d\|^2)$$

这样求导后得到的等式中，等式右边的矩阵一定是正定矩阵。

第二种方法是在 Pearson 系数的分母上加入正则化项，同样结果也一定可逆。

$$\begin{aligned}\rho &= \max_{\alpha,\beta}\frac{\alpha' K_x K_y \beta}{\sqrt{(\alpha' K_x^2 \alpha + \kappa\|\mathbf{w}_x\|^2)\cdot(\beta' K_y^2 \beta + \kappa\|\mathbf{w}_y\|^2))}} \\ &= \max_{\alpha,\beta}\frac{\alpha' K_x K_y \beta}{\sqrt{(\alpha' K_x^2 \alpha + \kappa\alpha' K_x \alpha)\cdot(\beta' K_y^2 \beta + \kappa\beta' K_y \beta)}}.\end{aligned}$$

2、求 Kernel 矩阵效率不高怎么办？

使用 Cholesky decomposition 压缩法或者部分 Gram-Schmidt 正交化法，。

3、怎么使用 CCA 用来做预测？

先找出 X 和 Y 的典型相关系数，新来一个样例 Xnew，在 X 中使用 KNN，然后找到在 Y 中对应的 N 个样例，求均值或者带权重均值等预测 Ynew。

4、如果有多个集合怎么办？X、Y、Z…？怎么衡量多个样本集的关系？

这个称为 Generalization of the Canonical Correlation。方法是使得两两集合的距离差之和最小。可以参考文献 2。

6. 参考文献

1、 http://www.stat.tamu.edu/~rrhocking/stat636/LEC-9.636.pdf

2、 **Canonical correlation analysis: An overview with application to learning methods**. David R. Hardoon , Sandor Szedmak and John Shawe-Taylor

3、 **A kernel method for canonical correlation analysis**. Shotaro Akaho

4、 **Canonical Correlation a Tutorial.** Magnus Borga

5、 **Kernel Canonical Correlation Analysis**. Max Welling

偏最小二乘法回归（Partial Least Squares Regression）

JerryLead
csxulijie@gmail.com
2011 年 8 月 20 日星期六

1. 问题

这节我们请出最后的有关成分分析和回归的神器 PLSR。PLSR 感觉已经把成分分析和回归发挥到极致了，下面主要介绍其思想而非完整的教程。让我们回顾一下最早的 Linear Regression 的缺点：如果样例数 m 相比特征数 n 少（m<n）或者特征间线性相关时，由于X^TX（n*n 矩阵）的秩小于特征个数（即X^TX不可逆）。因此最小二乘法$\theta = (X^TX)^{-1}X^T\vec{y}$ 就会失效。

为了解决这个问题，我们会使用 PCA 对样本 X（m*n 矩阵）进行降维，不妨称降维后的 X 为 X'（m*r 矩阵，一般加了'就表示转置，这里临时改变下），那么 X'的秩为 r（列不相关）。

2. PCA Revisited

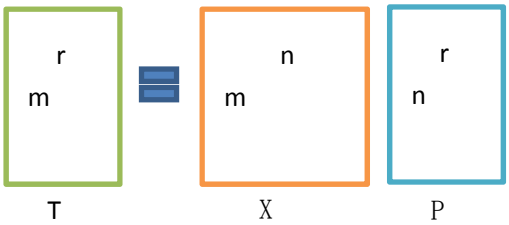
所谓磨刀不误砍柴工，这里先回顾下 PCA。

令 X 表示样本，含有 m 个样例$\{x^{(1)}, x^{(2)}, \ldots, x^{(m)}\}$，每个样例特征维度为 n，$x^{(i)} = \{x_1^{(i)}, x_2^{(i)}, \ldots x_n^{(i)}\}$。假设我们已经做了每个特征均值为 0 处理。

如果 X 的秩小于 n，那么 X 的协方差矩阵$\frac{1}{m}X^TX$的秩小于 n，因此直接使用线性回归的话不能使用最小二乘法来求解出唯一的θ，我们想使用 PCA 来使得X^TX可逆，这样就可以用最小二乘法来进行回归了，这样的回归称为主元回归（PCR）。

PCA 的一种表示形式：

$T = XP$



其中 X 是样本矩阵，P 是 X 的协方差矩阵的特征向量（当然是按照特征值排序后选取的前 r 个特征向量），T 是 X 在由 P 形成的新的正交子空间上的投影（也是样本 X 降维后的新矩阵）。

在线性代数里面我们知道，实对称阵 A 一定存在正交阵 P，使得$P^{-1}AP$为对角阵。因此可以让X^TX的特征向量矩阵 P 是正交的。

其实 T 的列向量也是正交的，不太严谨的证明如下：

$T^TT = (XP)^T(XP) = P^TX^TXP = P^T(P\Lambda P^T)P = P^TP\Lambda P^TP = \Lambda$

其中利用了$X^TX = P\Lambda P^T$，这是求 P 的过程，Λ是对角阵，对角线上元素就是特征值λ。这里对 P 做了单位化，即$P^TP = I$。这就说明了 T 也是正交的， P 是X^TX的特征向量矩阵，更进一步，T 是XX^T的特征向量矩阵（$XX^TT = XX^TXP = XP\Lambda P^TP = T\Lambda$）。

这样经过 PCA 以后，我们新的样本矩阵 T（m*r）是满秩的，而且列向量正交，因此直接代入最小二乘法公式，就能得到回归系数θ。

PCA 的另一种表示：

$X = M_1 + M_2 + M_3 + \cdots + M_n = t_1p_1^T + t_2p_2^T + t_3p_3^T + \cdots + t_np_n^T = TP^T$（假设 X 秩为 n）

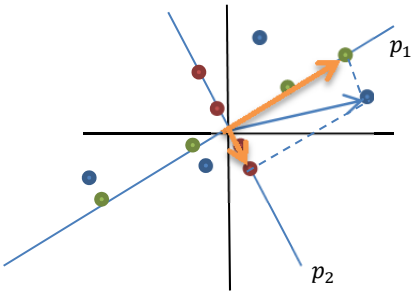
这个公式其实和上面的表示方式$T = XP$没什么区别。

$T = XP \rightarrow TP^T = XPP^T \rightarrow X = TP^T$ （当然我们认为 P 是 n*n 的，因此$P^T = P^{-1}$）

如果 P 是 n*r 的，也就是舍弃了特征值较小的特征向量，那么上面的加法式子就变成了

$X = M_1 + M_2 + M_3 + \cdots + M_r + E = t_1p_1^T + t_2p_2^T + t_3p_3^T + \cdots + t_rp_r^T + E = TP^T + E$

这里的 E 是残差矩阵。其实这个式子有着很强的几何意义，p_i是X^TX第i大特征值对应的归一化后的特征向量，t_i就是 X 在p_i上的投影。$t_ip_i^T$就是 X 先投影到p_i上，还以原始坐标系得到的 X'。下面这个图可以帮助理解：



黑色线条表示原始坐标系，蓝色的点是原始的 4 个 2 维的样本点，做完 PCA 后，得到两个正交的特征向量坐标p_1和p_2。绿色点是样本点在p_1上的投影（具有最大方差），红色点是在p_2上的投影。t_1的每个分量是绿色点在p_1上的截距，t_2是红色点在p_2上的截距。$t_ip_i^T$中的每个分量都可以看做是方向为p_i，截距为t_i相应分量大小的向量，如那个p_1上的橘色箭头。$t_ip_i^T$就得到了 X 在p_i的所有投影向量，由于p_1和p_2正交，因此$t_1p_1^T + t_2p_2^T$就相当于每个点的橘色箭头的加和，可想而知，得到了原始样本点。

如果舍弃了一些特征向量如p_2，那么通过$t_1p_1^T$只能还原出原始点的部分信息（得到的绿色点，丢失了蓝色点在另一维度上的信息）。另外，P 有个名字叫做 loading 矩阵，T 叫做 score 矩阵。

3. PLSR 思想及步骤

我们还需要回味一下 CCA 来引出 PLSR。在 CCA 中，我们将 X 和 Y 分别投影到直线得到 u 和 v，然后计算 u 和 v 的 Pearson 系数（也就是 Corr(u,v)），认为相关度越大越好。形式化表示：

Maximize $a^TCov(x,y)b$
Subject to: $a^TVar(x)a = 1, b^TVar(y)b = 1$

其中 a 和 b 就是要求的投影方向。

想想 CCA 的缺点：对特征的处理方式比较粗糙，用的是线性回归来表示 u 和 x 的关系，u 也是 x 在某条线上的投影，因此会存在线性回归的一些缺点。我们想把 PCA 的成分提取技术引入 CCA，使得 u 和 v 尽可能携带样本的最主要信息。还有一个更重要的问题，CCA 是寻找 X 和 Y 投影后 u 和 v 的关系，显然不能通过该关系来还原出 X 和 Y，也就是找不到 X 到 Y 的直接映射。这也是使用 CCA 预测时大多配上 KNN 的原因。

而 PLSR 更加聪明，同时兼顾 PCA 和 CCA，并且解决了 X 和 Y 的映射问题。看 PCA Revisited 的那张图，假设对于 CCA，X 的投影直线是p_1，那么 CCA 只考虑了 X 的绿色点与 Y 在某条直线上投影结果的相关性，丢弃了 X 和 Y 在其他维度上的信息，因此不存在 X 和 Y 的映射。而 PLSR 会在 CCA 的基础上再做一步，由于原始蓝色点可以认为是绿色点和红色点的叠加，因此先使用 X 的绿色点t_1对 Y 做回归（$Y = t_1 r_1^T + F$，样子有点怪，两边都乘以r_1就明白了，这里的 Y 类似于线性回归里的X，t_1类似y），然后用 X 的红色点t_2对 Y 的剩余部分 F 做回归（得到r_2，$F = t_2 r_2^T + F'$）。这样 Y 就是两部分回归的叠加。当新来一个 x 时，投影一下得到其绿色点t_1和红色点t_2，然后通过 r 就可以还原出 Y，实现了 X 到 Y 的映射。当然这只是几何上的思想描述，跟下面的细节有些出入。

下面正式介绍 PLSR：

1） 设 X 和 Y 都已经过标准化（包括减均值、除标准差等）。

2） 设 X 的第一个主成分为p_1，Y 的第一个主成分为q_1，两者都经过了单位化。（这里的主成分并不是通过 PCA 得出的主成分）

3） $u_1 = Xp_1$，$v_1 = Yq_1$，这一步看起来和 CCA 是一样的，但是这里的 p 和 q 都有主成分的性质，因此有下面 4）和 5）的期望条件。

4） $Var(u_1) \to max, Var(v_1) \to max$，即在主成分上的投影，我们期望是方差最大化。

5） $Corr(u_1, v_1) \to max$，这个跟 CCA 的思路一致。

6） 综合 4）和 5），得到优化目标$Cov(u_1, v_1) = \sqrt{Var(u_1)Var(v_1)}Corr(u_1, v_1) \to max$。

形式化一点：

Maximize $\langle Xp_1, Yq_1 \rangle$

Subject to: $\|p_1\| = 1, \|q_1\| = 1$

看起来比 CCA 还要简单一些，其实不然，CCA 做完一次优化问题就完了。但这里的p_1和q_1对 PLSR 来说只是一个主成分，还有其他成分呢，那些信息也要计算的。

先看该优化问题的求解吧：

引入拉格朗日乘子

$$\mathcal{L} = p_1^T X^T Y q_1 - \frac{\lambda}{2}(p_1^T p_1 - 1) - \frac{\theta}{2}(q_1^T q_1 - 1)$$

分别对p_1, q_1求偏导，得

$$\frac{\partial \mathcal{L}}{\partial p_1} = X^T Y q_1 - \lambda p_1 = 0$$

$$\frac{\partial \mathcal{L}}{\partial q_1} = Y^T X p_1 - \theta q_1 = 0$$

从上面可以看出$\lambda = \theta$（两边都乘以 p 或 q，再利用=1 的约束）

下式代入上式得到

$$X^TYY^TXp_1 = \lambda^2 p_1$$

上式代入下式得到

$$Y^TXX^TYq_1 = \lambda^2 q_1$$

目标函数$\langle Xp_1, Yq_1\rangle \rightarrow p_1^T X^T Y q_1 \rightarrow\ p_1^T(\lambda p_1) \rightarrow\ \lambda$，要求最大。

因此p_1就是对称阵X^TYY^TX的最大特征值对应的单位特征向量，q_1就是Y^TXX^TY最大特征值对应的单位特征向量。

可见p_1和q_1是投影方差最大和两者相关性最大上的权衡，而 CCA 只是相关性上最大化。

求得了p_1和q_1，即可得到

$$u_1 = Xp_1$$
$$v_1 = Yq_1$$

这里得到的u_1和v_1类似于上图中的绿色点，只是在绿色点上找到了 X 和 Y 的关系。如果就此结束，会出现与 CCA 一样的不能由 X 到 Y 映射的问题。

利用我们在 PCA Revisited 里面的第二种表达形式，我们可以继续做下去，建立回归方程：

$$X = u_1 c_1^T + E$$
$$Y = v_1 d_1^T + G$$

这里的 c 和 d 不同于 p 和 q，但是它们之间有一定联系，待会证明。E 和 G 是残差矩阵。

我们进行 PLSR 的下面几个步骤：

1） $Y = u_1 r_1^T + F$，使用u_1对 Y 进行回归，原因已经解释过，先利用 X 的主成分对 Y 进行回归。

2） 使用最小二乘法，计算 c，d，r 分别为：

$$c_1 = \frac{X^T u_1}{\|u_1\|^2}$$
$$d_1 = \frac{Y^T v_1}{\|v_1\|^2}$$
$$r_1 = \frac{Y^T u_1}{\|u_1\|^2}$$

实际上这一步计算出了各个投影向量。

p_1和c_1的关系如下：

$$p_1^T c_1 = p_1^T \frac{X^T u_1}{\|u_1\|^2} = \frac{u_1^T u_1}{\|u_1\|^2} = 1$$

再谈谈p_1和c_1的关系，虽然这里将c_1替换成p_1可以满足等式要求和几何要求，而且p_1就是 X 投影出u_1的方向向量。但这里我们想做的是回归（让 E 尽可能小），因此根据最小二乘法得到的c_1一般与p_1不同。

3） 将剩余的 E 当做新的 X，剩余的 F 当做新的 Y，然后按照前面的步骤求出p_2和q_2，得到：

$$u_2 = Ep_2$$
$$v_2 = Fq_2$$

目标函数$\langle Ep_2, Fq_2\rangle \rightarrow p_2^T E^T F q_2 \rightarrow\ p_2^T(\lambda p_2) \rightarrow\ \lambda$，这个与前面一样，$p_2$和$q_2$分别是新的$E^TFF^TE$和$F^TEE^TF$的最大特征值对应的单位特征向量。

4） 计算得到第二组回归系数：

$$c_2 = \frac{E^T u_2}{\|u_2\|^2}$$

$$r_2 = \frac{F^T u_2}{\|u_2\|^2}$$

这里的u_2和之前的u_1是正交的，证明如下：

$$u_1^T u_2 = u_1^T E p_2 = u_1^T (X - u_1 c_1^T) p_2 = \left[u_1^T X - u_1^T u_1 \frac{u_1^T X}{\|u_1\|^2} \right] p_2 = 0$$

其实u_i和不同的u_j都是相互正交的。

同样p_i和不同的p_j也是正交的。

$$p_1^T p_2 = p_1^T \frac{1}{\lambda} E^T F q_2 = p_1^T \frac{1}{\lambda} E^T v_2 = \frac{1}{\lambda} p_1^T (X - u_1 c_1^T)^T v_2$$

$$= \frac{1}{\lambda} (X p_1 - u_1 c_1^T p_1)^T v_2 = \frac{1}{\lambda} (u_1 - u_1)^T v_2 = 0$$

但c_i和不同的c_j一般不是正交的。

5） 从上一步得到回归方程：

$$E = u_2 c_2^T + E'$$
$$F = u_2 r_2^T + F'$$

如果还有残差矩阵的话，可以继续计算下去。

6） 如此计算下去，最终得到：

$$X = u_1 c_1^T + u_2 c_2^T + u_3 c_3^T + \cdots + u_n c_n^T + E$$
$$Y = u_1 r_1^T + u_2 r_2^T + u_3 r_3^T + \cdots + u_n r_n^T + F$$

与 PCA 中表达式不一样的是这里的c_i和不同的c_j之间一般不是正交的。

其实这里不必一直计算到 n，可以采用类似于 PCA 的截尾技术，计算到合适的 r 即可。关于 r 数目的选取可以使用交叉验证方法，这与 PCA 里面的问题类似。

另外，p_i和c_j的关系是$p_i^T c_j = 1(i = j), p_i^T c_j = 0(i \neq j)$

上面的公式如果写成矩阵形式如下：

$$X = UC^T + E$$
$$Y = UR^T + F = XPR^T + F = XB + F$$

这就是$X \rightarrow Y$的回归方程，其中$B = PR^T$。

在计算过程中，收集一下 P 和 R 的值即可。

7） 使用 PLSR 来预测。

从 6)中可以发现 Y 其实是多个回归的叠加(其实$u_1 r_1^T$已经回归出 Y 的最主要信息)。

我们在计算模型的过程中，得到了 p 和 r。那么新来一个 x，首先计算 u（这里的 u 变成了实数，而不是向量了），得到

$u_1 = x^T p_1,\ u_2 = x^T p_2,\ u_3 = x^T p_3 \ldots$

然后代入 Y 的式子即可求出预测的 y 向量，或者直接代入$y^T = x^T B$

8） 至此，PLSR 的主要步骤结束。

4. PLSR 相关问题

1） 其实不需要计算 v 和 q，因为我们使用 u 去做 Y 的回归时认为了$u_i = cv_i$，其中 c 是常数。之所以这样是因为前面提到过的 Y 可以首先在 X 的主要成分上做回归，然后将 Y 的残差矩阵在 X 的残差矩阵的主要成分上做回归。最后 X 的各个成分回归之和就是 Y。

2） 一般使用的 PLSR 求解方法是迭代化的求解方法，称之为 NIPALS，还有简化方法 SIMPLS，这些方法在一般论文或参考文献中提供的网址里都有，这里就不再贴了。

3） PLSR 里面还有很多高级话题，比如非线性的 Kernel PLSR，异常值检测，带有缺失值的处理方法，参数选择，数据转换，扩展的层次化模型等等。可以参考更多的论文有针对性的研究。

5. 一些感悟

本文试图将 PCA、CCA、PLSR 综合起来对比、概述和讨论，不免对符号的使用稍微都点混乱，思路也有穿插混淆。还是以推导出的公式为主进行理解吧。另外，本文有很多个人理解在里面，难免有误，还望批评指正。提供 PDF 版本，只是为了格式好看些。

之前也陆陆续续地关注了一些概率图模型和时间序列分析，以后可能会转向介绍这两方面的内容，也会穿插一些其他的内容。说实话，自学挺吃力的，尤其对我这样一个不是专业搞 ML 的人来说，也需要花大量时间。感叹国外的资料多，lecture 多，视频多，可惜因为我这的网速和 GFW 原因，看不了教学视频，真是遗憾。

6. 参考文献：

1. PARTIAL LEAST-SQUARES REGRESSION: A TUTORIAL. Paul Geladi and Bruce R. Kowalski
2. 王惠文－偏最小二乘回归方法及应用
3. Partial Least Squares (PLS) Regression.
4. A Beginner's Guide to Partial Least Squares Analysis
5. Nonlinear Partial Least Squares: An Overview
6. http://www.statsoft.com/textbook/partial-least-squares/
7. Canonical Correlation a Tutorial
8. Pattern Recognition And Machine Learning