Softmax

softmax是个多分类方法

Softmax 回归（或者叫做多项逻辑回归）是逻辑回归在多分类问题上的推广。在逻辑回归中，我们假设标签是二元的：$y^{(i)} \in \{0,1\}$

我们使用一个这样的分类器来对两种手写数字进行区分。Softmax 回归则让我们可以处理 $y^{(i)} \in \{1,\cdots,K\}$ ，其中 K是类的编号。

softmax函数

$$softmax(\mathbf{x})_i = \frac{e^{\mathbf{x}_i}}{\sum_j e^{\mathbf{x}_j}}$$

$$softmax(\mathbf{x}) = softmax(\mathbf{x} + c)$$

代价函数

$$J(\theta) = -\left[\sum_{i=1}^{m}\sum_{k=1}^{K} 1\left\{y^{(i)} = k\right\} \log \frac{\exp(\theta^{(k)\top} x^{(i)})}{\sum_{j=1}^{K} \exp(\theta^{(j)\top} x^{(i)})}\right]$$

Logistic损失函数

$$J(\theta) = -\left[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)}))\right]$$

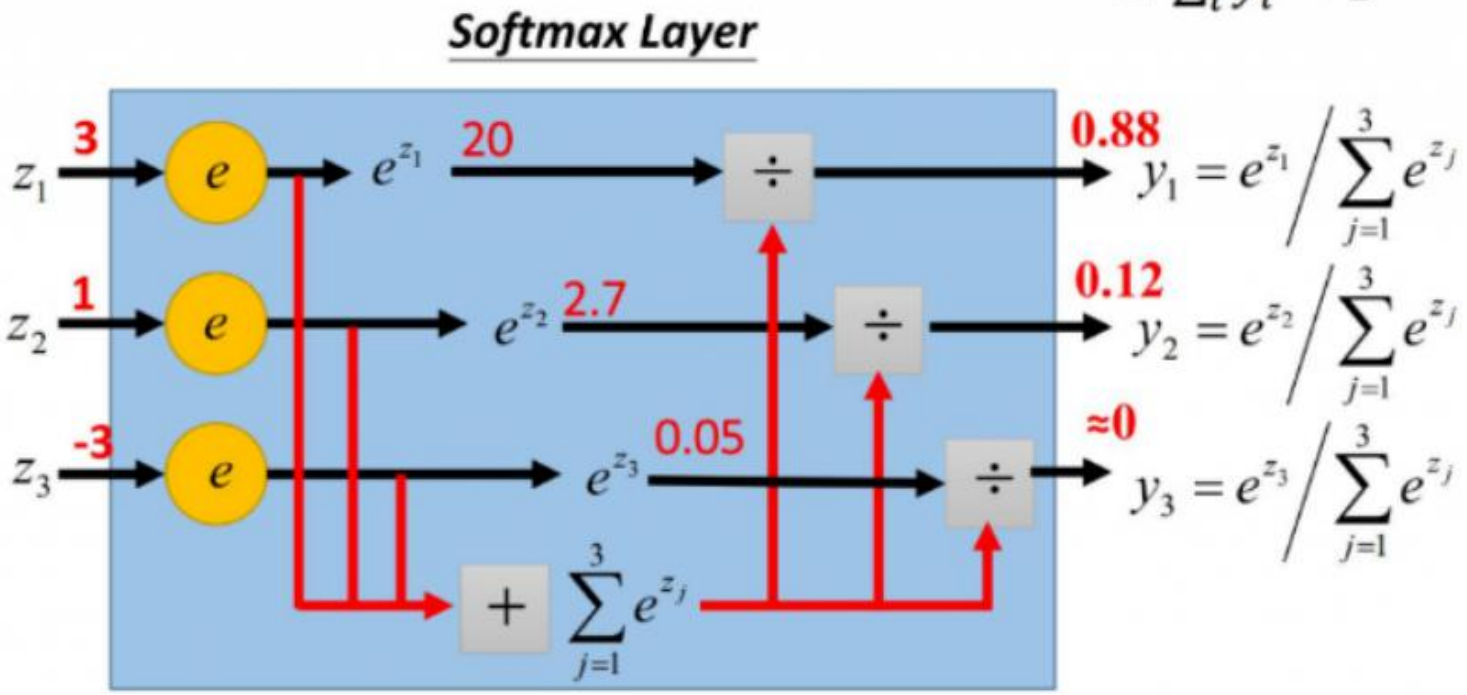
和softmax定义的损失函数是否一致？

形象的图

$$S_i = \frac{e^i}{\sum_j e^j}$$

- Softmax layer as the output layer

Probability:
- $1 > y_i > 0$
- $\sum_i y_i = 1$



梯度推导

$$CE(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_i y_i \log(\hat{y}_i)$$

$$\frac{\partial CE(\mathbf{y}, \hat{\mathbf{y}})}{\partial \theta} = \hat{\mathbf{y}} - \mathbf{y}$$

$$\frac{\partial CE(\mathbf{y}, \hat{\mathbf{y}})}{\partial \theta} = \begin{cases} \hat{y}_i - 1, i = k \\ \hat{y}_i, otherwise \end{cases}$$

$\hat{\mathbf{y}}$ =softmax(XW+b)

θ = XW+b

梯度推导

$\partial CE / \partial W = X^T(\hat{\mathbf{y}} - \mathbf{y})$

$\partial CE / \partial b = \sum(\hat{\mathbf{y}}_i - y_i) / n$

生成模型

生成模型

- **判别方法：**由数据直接学习决策函数Y=f(X)或者条件概率分布P(Y|X)作为预测的模型，即判别模型。基本思想是有限样本条件下建立判别函数，不考虑样本的产生模型，直接研究预测模型。典型的判别模型包括k近邻，感知级，决策树，支持向量机等。

- **生成方法：**由数据学习联合概率密度分布P(X,Y)，然后求出条件概率分布P(Y|X)作为预测的模型，即生成模型：P(Y|X)= P(X,Y)/ P(X)。基本思想是首先建立样本的联合概率概率密度模型P(X,Y)，然后再得到后验概率P(Y|X)，再利用它进行分类。

生成模型和判别模型的优缺点

- 生成方法的特点：生成方法学习联合概率密度分布P(X,Y)，所以就可以从统计的角度表示数据的分布情况，能够反映同类数据本身的相似度。但它不关心到底划分各类的那个分类边界在哪。当存在隐变量时，仍可以用生成方法学习。此时判别方法就不能用。

- 判别方法的特点：判别方法直接学习的是决策函数Y=f(X)或者条件概率分布P(Y|X)。不能反映训练数据本身的特性。但它寻找不同类别之间的最优分类面，反映的是异类数据之间的差异。直接面对预测，往往学习的准确率更高。由于直接学习P(Y|X)或P(X)，可以对数据进行各种程度上的抽象、定义特征并使用特征，因此可以简化学习问题。

相对优势

生成模型：需要求解的东西最多，因为它涉及到寻找在x和C_k上的联合概率分布。对于许多应用，x的维度很高，这会导致我们需要大量的训练数据才能在合理的精度下确定类条件概率密度。注意，先验概率$p(C_k)$经常能够根据训练数据集里的每个类别的数据点所占的比例简单地估计出来。但是，生成方法的一个优点是，它能够通过公式求出数据的边缘概率密度$p(x)$。这对于检测模型中具有低概率的新数据点很有用，对于这些点，模型的预测准确率可能会很低。这种技术被称为离群点检测（outlier detection）或者异常检测（novelty detection）

然而，如果我们只想进行分类的决策，那么这种方法会浪费计算资源。并且，实际上我们只是想求出后验概率$p(C_k|x)$（可以直接通过判别方法求出），但是为了求出它，这种方法需要大量的数据来寻找联合概率$p(x; C_k)$。事实上，类条件密度可能包含很多对于后验概率几乎没有影响的结构，如图1.27所示。

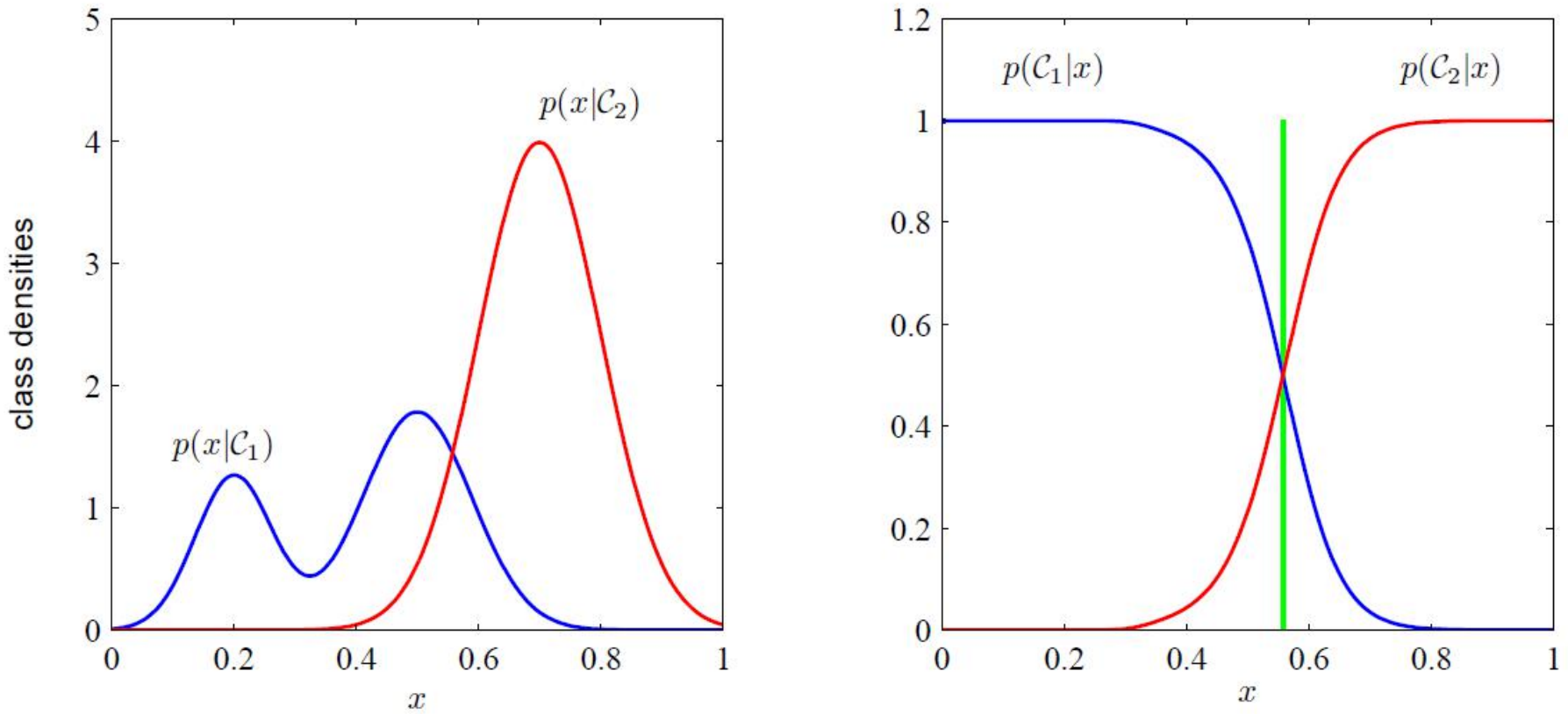


图 1.27: 具有一元输入变量x的两个类别的类条件概率密度（左图）以及对应的后验概率密度（右图）。注意，左图中，蓝色曲线表示类条件概率密度$p(\boldsymbol{x} \mid \mathcal{C}_1)$，它的峰值对于后验概率分布没有影响。右图中的垂直绿色直线表示给出最小误分类率的x的决策边界。我们假设先验概率分布$p(\mathcal{C}_1)$和$p(\mathcal{C}_2)$是相等的。

生成模型和判别模型的联系

- 由生成模型可以得到判别模型，但由判别模型得不到生成模型。
- 生成模型学习联合概率分布p(x,y)，而判别模型学习条件概率分布p(y|x)。
- 生成算法尝试去找到底这个数据是怎么生成的（产生的），然后再对一个信号进行分类。基于你的生成假设，那么那个类别最有可能产生这个信号，这个信号就属于那个类别。判别模型不关心数据是怎么生成的，它只关心信号之间的差别，然后用差别来简单对给定的一个信号进行分类。

高斯判别分析

包括QDA（二次）和LDA（线性）两种情形

多元高斯分布

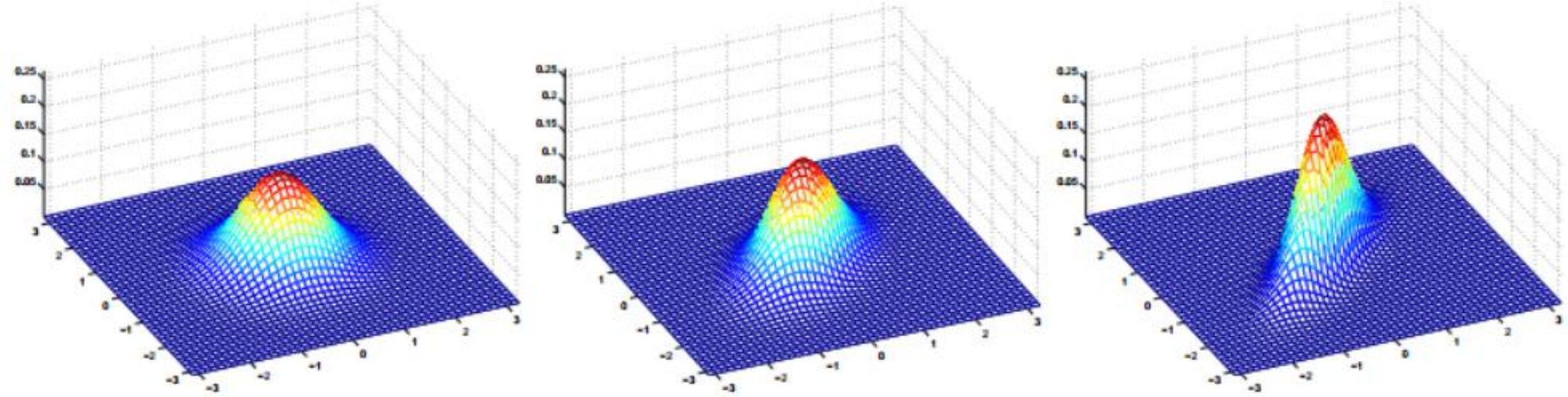
$$p(x;\mu,\Sigma)=\frac{1}{(2\pi)^{n/2}\,|\Sigma|^{1/2}}\exp(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu))$$

$$E[X]=\int_x xp(x;\mu,\Sigma)dx=\mu.$$

$$Cov(Z)=E[(Z-E[Z])(Z-E[Z])^T]$$

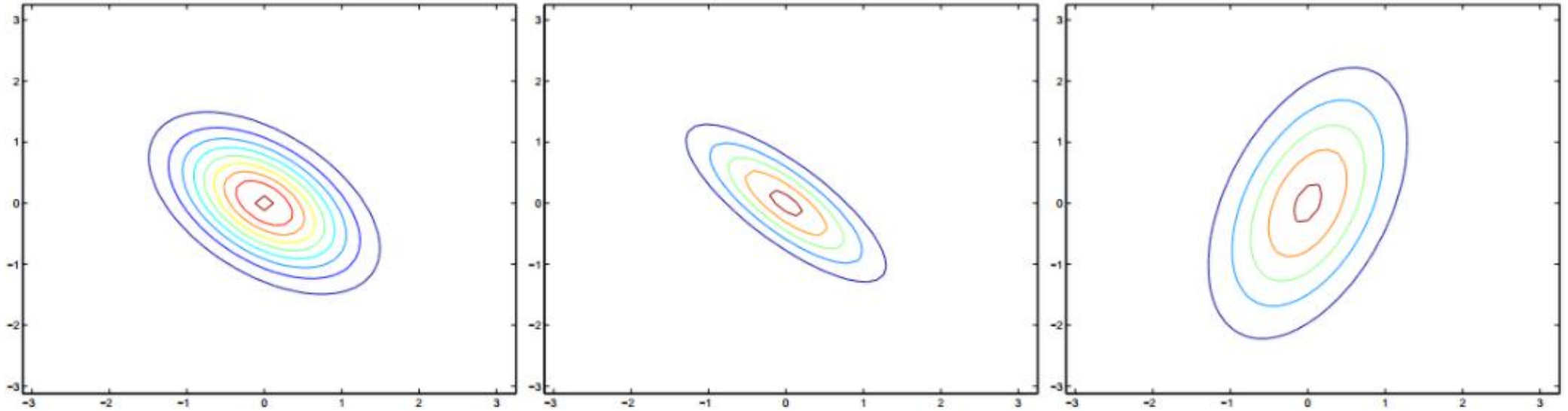
$$Cov(X)=\Sigma.$$

可视化图



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

可视化图



$$\Sigma = \begin{bmatrix} 1 & -0.5 \\ -0.5 & 1 \end{bmatrix}; \Sigma = \begin{bmatrix} 1 & -0.8 \\ -0.8 & 1 \end{bmatrix}; \Sigma = \begin{bmatrix} 3 & 0.8 \\ 0.8 & 1 \end{bmatrix}.$$

Quadratic discriminant analysis(QDA)

$$p(\mathbf{x}|y=c,\boldsymbol{\theta}) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_c, \boldsymbol{\Sigma}_c)$$

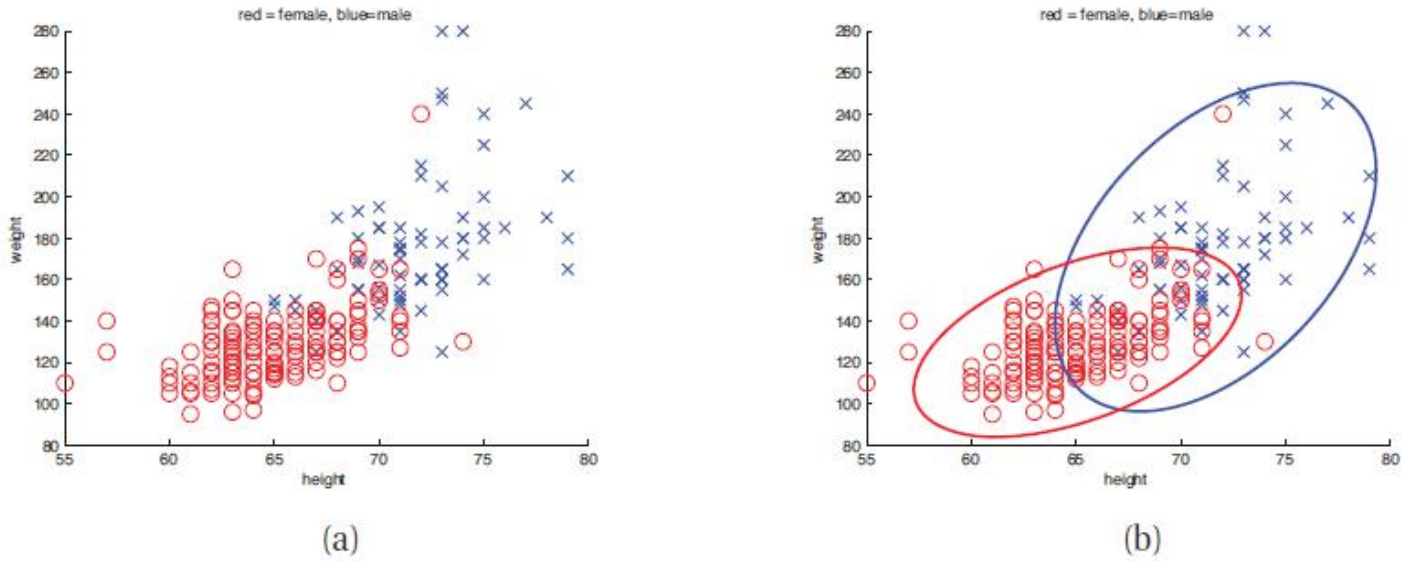
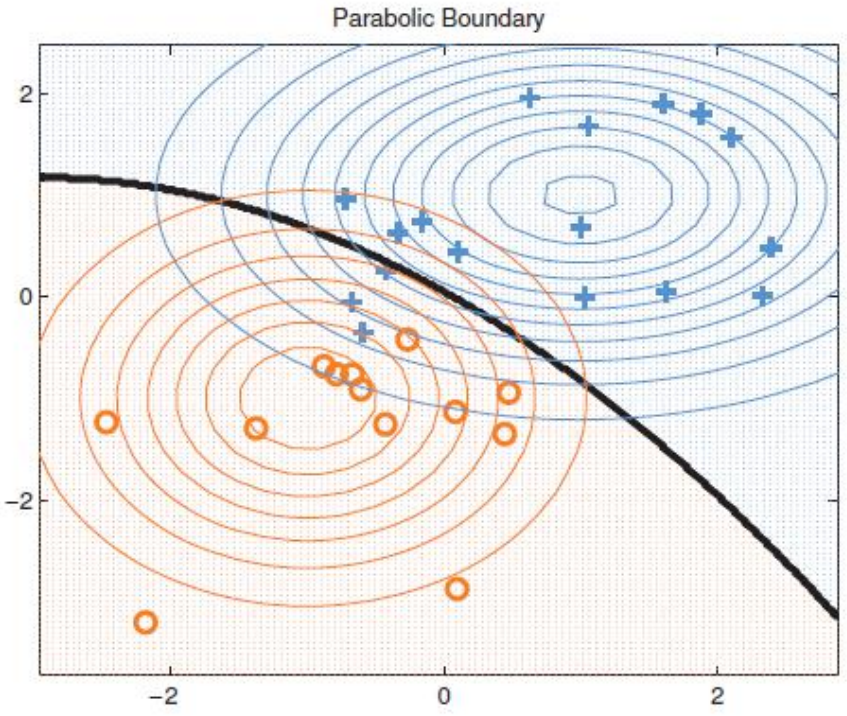


Figure 4.2 (a) Height/weight data. (b) Visualization of 2d Gaussians fit to each class. 95% of the probability mass is inside the ellipse. Figure generated by `gaussHeightWeight`.

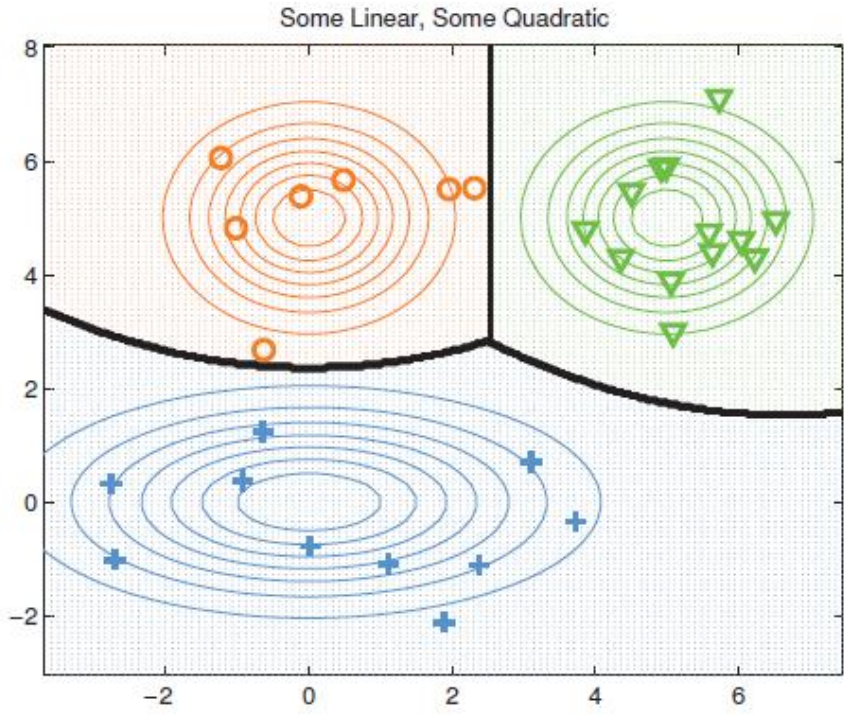
二次的来由

$$p(y=c|\mathbf{x},\boldsymbol{\theta}) = \frac{\pi_c|2\pi\boldsymbol{\Sigma}_c|^{-\frac{1}{2}}\exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_c)^T\boldsymbol{\Sigma}_c^{-1}(\mathbf{x}-\boldsymbol{\mu}_c)\right]}{\sum_{c'}\pi_{c'}|2\pi\boldsymbol{\Sigma}_{c'}|^{-\frac{1}{2}}\exp\left[-\frac{1}{2}(\mathbf{x}-\boldsymbol{\mu}_{c'})^T\boldsymbol{\Sigma}_{c'}^{-1}(\mathbf{x}-\boldsymbol{\mu}_{c'})\right]}$$

Thresholding this results in a quadratic function of $\mathbf{x}$. The result is known as **quadratic discriminant analysis** (QDA). Figure 4.3 gives some examples of what the decision boundaries look like in 2D.


Parabolic Boundary
2
0
−2
−2
0
2

(a)


Some Linear, Some Quadratic
8
6
4
2
0
−2
−2
0
2
4
6

(b)

线性判别分析(LDA)

We now consider a special case in which the covariance matrices are **tied** or **shared** across classes, $\boldsymbol{\Sigma}_c = \boldsymbol{\Sigma}$. In this case, we can simplify Equation 4.33 as follows:

$$
\begin{aligned}
p(y=c|\mathbf{x},\boldsymbol{\theta}) &\propto \pi_c \exp\left[\boldsymbol{\mu}_c^T\boldsymbol{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}\mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}\boldsymbol{\mu}_c^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_c\right] \\
&= \exp\left[\boldsymbol{\mu}_c^T\boldsymbol{\Sigma}^{-1}\mathbf{x} - \frac{1}{2}\boldsymbol{\mu}_c^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_c + \log\pi_c\right]\exp[-\frac{1}{2}\mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x}]
\end{aligned}
$$

公式化成熟悉的形式

Since the quadratic term $\mathbf{x}^T\boldsymbol{\Sigma}^{-1}\mathbf{x}$ is independent of c, it will cancel out in the numerator and denominator. If we define

$$\begin{aligned}\gamma_c &= -\frac{1}{2}\boldsymbol{\mu}_c^T\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_c + \log \pi_c \\ \boldsymbol{\beta}_c &= \boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_c\end{aligned}$$

then we can write

$$p(y=c|\mathbf{x},\boldsymbol{\theta}) = \frac{e^{\boldsymbol{\beta}_c^T\mathbf{x}+\gamma_c}}{\sum_{c'} e^{\boldsymbol{\beta}_{c'}^T\mathbf{x}+\gamma_{c'}}} = \mathcal{S}(\boldsymbol{\eta})_c$$

where $\boldsymbol{\eta} = [\boldsymbol{\beta}_1^T\mathbf{x}+\gamma_1, \ldots, \boldsymbol{\beta}_C^T\mathbf{x}+\gamma_C]$, and $\mathcal{S}$ is the **softmax** function, defined as follows:

$$\mathcal{S}(\boldsymbol{\eta})_c = \frac{e^{\eta_c}}{\sum_{c'=1}^{C} e^{\eta_{c'}}}$$

线性的来由

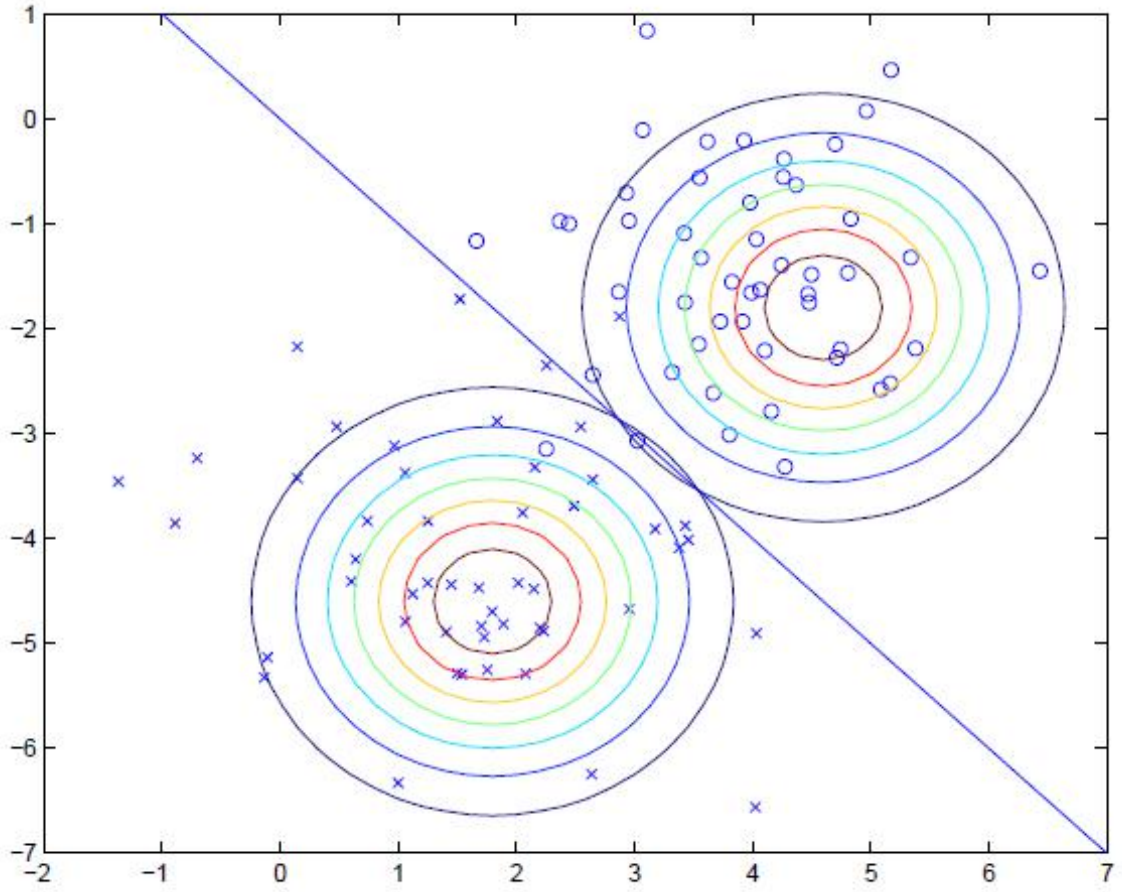
$$
\begin{aligned}
p(y = c|\mathbf{x}, \boldsymbol{\theta}) &= p(y = c'|\mathbf{x}, \boldsymbol{\theta}) \\
\boldsymbol{\beta}_c^T \mathbf{x} + \gamma_c &= \boldsymbol{\beta}_{c'}^T \mathbf{x} + \gamma_{c'} \\
\mathbf{x}^T (\boldsymbol{\beta}_{c'} - \boldsymbol{\beta}) &= \gamma_{c'} - \gamma_c
\end{aligned}
$$

Two-class LDA

Logistic的形式

$$
\begin{aligned}
p(y=1|\mathbf{x},\boldsymbol{\theta}) &= \frac{e^{\boldsymbol{\beta}_1^T\mathbf{x}+\gamma_1}}{e^{\boldsymbol{\beta}_1^T\mathbf{x}+\gamma_1}+e^{\boldsymbol{\beta}_0^T\mathbf{x}+\gamma_0}} \\
&= \frac{1}{1+e^{(\boldsymbol{\beta}_0-\boldsymbol{\beta}_1)^T\mathbf{x}+(\gamma_0-\gamma_1)}} = \mathrm{sigm}\left((\boldsymbol{\beta}_1-\boldsymbol{\beta}_0)^T\mathbf{x}+(\gamma_1-\gamma_0)\right)
\end{aligned}
$$

LDA一个例子



LDA一个例子

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

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

MLE(最大似然估计)

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

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

GDA and logistic regression

We just argued that if $p(x|y)$ is multivariate gaussian (with shared Σ), then $p(y|x)$ necessarily follows a logistic function. The converse, however, is not true; i.e., $p(y|x)$ being a logistic function does not imply $p(x|y)$ is multivariate gaussian. This shows that GDA makes *stronger* modeling assumptions about the data than does logistic regression. It turns out that when these modeling assumptions are correct, then GDA will find better fits to the data, and is a better model. Specifically, when $p(x|y)$ is indeed gaussian (with shared Σ), then GDA is **asymptotically efficient**. Informally, this means that in the limit of very large training sets (large m), there is no algorithm that is strictly better than GDA (in terms of, say, how accurately they estimate $p(y|x)$). In particular, it can be shown that in this setting, GDA will be a better algorithm than logistic regression; and more generally, even for small training set sizes, we would generally expect GDA to better.

GDA and logistic regression

In contrast, by making significantly weaker assumptions, logistic regression is also more *robust* and less sensitive to incorrect modeling assumptions. There are many different sets of assumptions that would lead to $p(y|x)$ taking the form of a logistic function. For example, if $x|y = 0 \sim \text{Poisson}(\lambda_0)$, and $x|y = 1 \sim \text{Poisson}(\lambda_1)$, then $p(y|x)$ will be logistic. Logistic regression will also work well on Poisson data like this. But if we were to use GDA on such data—and fit Gaussian distributions to such non-Gaussian data—then the results will be less predictable, and GDA may (or may not) do well.

GDA and logistic regression

To summarize: GDA makes stronger modeling assumptions, and is more data efficient (i.e., requires less training data to learn "well") when the modeling assumptions are correct or at least approximately correct. Logistic regression makes weaker assumptions, and is significantly more robust to deviations from modeling assumptions. Specifically, when the data is indeed non-Gaussian, then in the limit of large datasets, logistic regression will almost always do better than GDA. For this reason, in practice logistic regression is used more often than GDA.

高斯混合模型（GMM）

回顾K-means

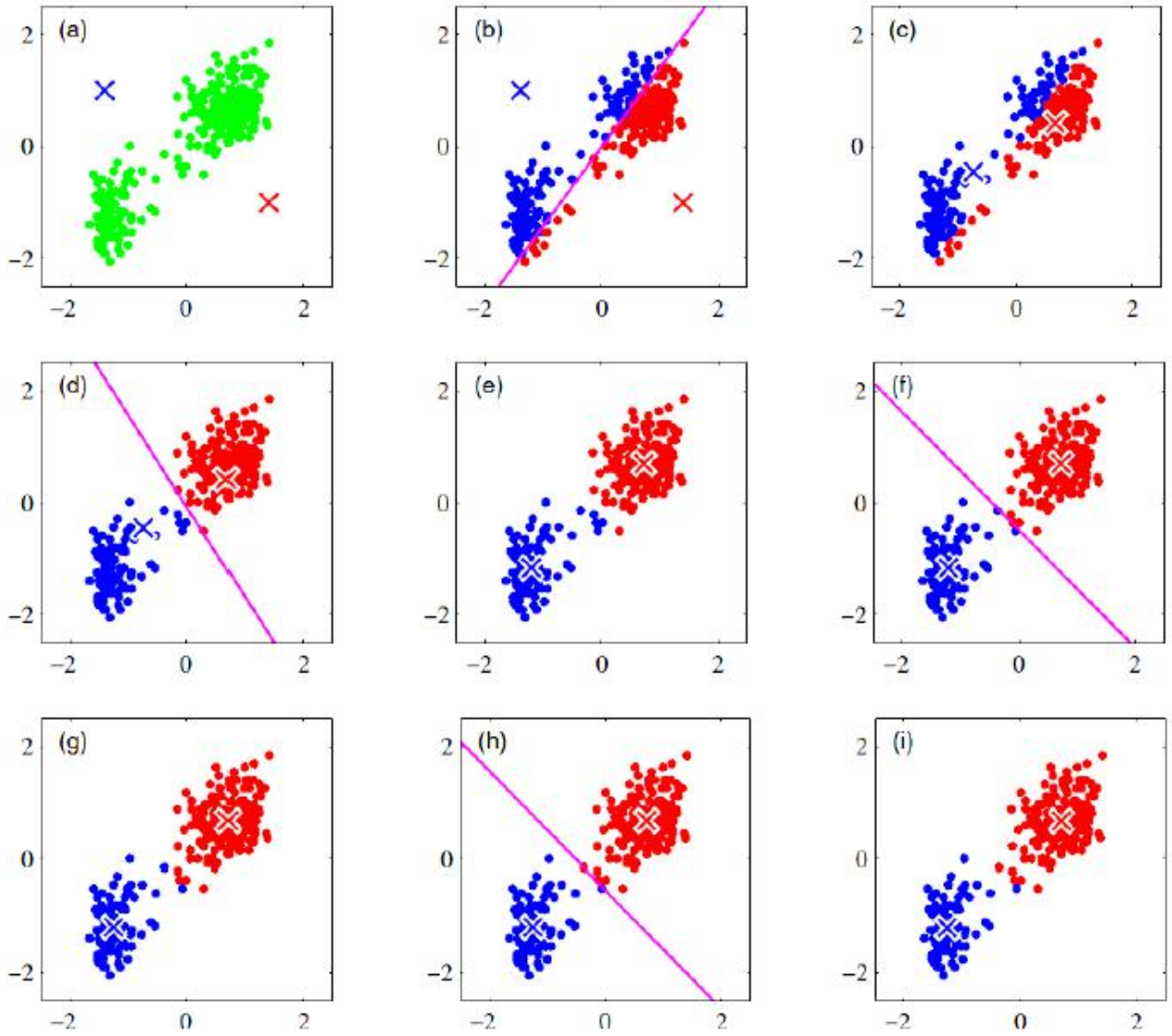
K-means算法，也被称为k-平均或k-均值，是一种广泛使用的聚类算法，或者成为其他聚类算法的基础。

假定输入样本为$S=x_1,x_2,\ldots,x_m$，则算法步骤为：

- 选择初始的k个簇中心$\mu_1\mu_2\ldots\mu_k$
- 将样本x_i标记为距离簇中心最近的簇：$label_i = \arg\min_{1\le j\le k}\|x_i-\mu_j\|$
- 更新簇中心：$\mu_j = \frac{1}{|c_j|}\sum_{i\in c_j} x_i$
- 重复最后两步，直到满足终止条件。

中止条件：迭代次数/簇中心变化率/最小平方误差MSE

Kmeans 过程



思考

- 无法给出某个样本属于该簇的后验概率（硬分类）
- 欧式距离作为评判标准，没有考虑数据点本身的形状和密度

最大似然估计

找出与样本的分布最接近的概率分布模型。

简单的例子

- 10次抛硬币的结果是：正正反正正正反反正正

假设p是每次抛硬币结果为正的概率。则：

得到这样的实验结果的概率是：

$$P = pp(1-p)ppp(1-p)(1-p)pp$$
$$= p^7(1-p)^3$$

- 最优解是：p=0.7

二项分布的最大似然

投硬币试验中，进行N次独立试验，n次朝上，N-n次朝下。

假定朝上的概率为p，使用对数似然函数作为目标函数：

$$f(n \mid p) = \log\left(p^n (1-p)^{N-n}\right) \xrightarrow{\Delta} h(p)$$

$$\frac{\partial h(p)}{\partial p} = \frac{n}{p} - \frac{N-n}{1-p} \xrightarrow{\Delta} 0 \quad \Rightarrow p = \frac{n}{N}$$

进一步考察

若给定一组样本$x_1,x_2\ldots x_n$，已知它们来自于高斯分布$N(\mu,\sigma)$，试估计参数μ,σ。

按照MLE的过程分析

高斯分布的概率密度函数：

$$f(x)=\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

将X_i的样本值x_i带入，得到：

$$L(x)=\prod_{i=1}^{n}\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

化简对数似然函数

$$l(x) = \log \prod_i \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

$$= \sum_i \log \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

$$= \left(\sum_i \log \frac{1}{\sqrt{2\pi}\sigma} \right) + \left(\sum_i - \frac{(x_i-\mu)^2}{2\sigma^2} \right)$$

$$= -\frac{n}{2} \log\left(2\pi\sigma^2\right) - \frac{1}{2\sigma^2} \sum_i (x_i-\mu)^2$$

参数估计的结论

目标函数 $$l(x)=-\frac{n}{2}\log\left(2\pi\sigma^2\right)-\frac{1}{2\sigma^2}\sum_i\left(x_i-\mu\right)^2$$

将目标函数对参数μ,σ分别求偏导，很容易得到μ,σ的式子：

$$\begin{cases}\mu=\dfrac{1}{n}\sum_i x_i\\ \sigma^2=\dfrac{1}{n}\sum_i\left(x_i-\mu\right)^2\end{cases}$$

符合直观想象

$$\begin{cases} \mu = \dfrac{1}{n}\sum_i x_i \\ \sigma^2 = \dfrac{1}{n}\sum_i (x_i - \mu)^2 \end{cases}$$

上述结论和矩估计的结果是一致的，并且意义非常直观：样本的均值即高斯分布的均值，样本的伪方差即高斯分布的方差。

- 该结论将作为下面分析的基础。

随机变量无法直接（完全）观察到

随机挑选10000位志愿者，测量他们的身高：
若样本中存在男性和女性，身高分别服从
$N(\mu_1,\sigma_1)$和$N(\mu_2,\sigma_2)$的分布，试估计$\mu_1,\sigma_1,\mu_2,\sigma_2$。
给定一幅图像，将图像的前景背景分开

无监督分类：聚类/EM

从直观理解猜测GMM的参数估计

随机变量X是有K个高斯分布混合而成，取各个高斯分布的概率为$\pi_1 \pi_2 \ldots \pi_K$，第i个高斯分布的均值为μ_i，方差为Σ_i。若观测到随机变量X的一系列样本$x_1, x_2, \ldots, x_n$，试估计参数π，μ，Σ。

建立目标函数

对数似然函数

$$l_{\pi,\mu,\Sigma}(x)=\sum_{i=1}^{N}\log\left(\sum_{k=1}^{K}\pi_k N(x_i \mid \mu_k,\Sigma_k)\right)$$

目标函数

由于在对数函数里面又有加和，我们没法直接用求导解方程的办法直接求得最大值。为了解决这个问题，我们分成两步。

第一步 估计数据来自哪个组分

估计数据由每个组份生成的概率：对于每个样本x_i，它由第k个组份生成的概率为

$$\gamma(i,k)=\frac{\pi_k N(x_i \mid \mu_k,\Sigma_k)}{\sum_{j=1}^{K}\pi_j N(x_i \mid \mu_j,\Sigma_j)}$$

上式中的μ和Σ也是待估计的值，因此采样迭代法：在计算γ(i,k)时假定μ和Σ已知；

- 需要先验给定μ和Σ。
- γ(i,k) 亦可看成组份k在生成数据x_i时所做的贡献。

第二步 估计每个组分的参数

对于所有的样本点，对于组份k而言，可看做生成了 $\{\gamma(i,k)x_i \mid i=1,2,\cdots N\}$ 这些点。组份k是一个标准的高斯分布，利用上面的结论：

$$\begin{cases} \mu = \dfrac{1}{n}\sum_i x_i \\ \sigma^2 = \dfrac{1}{n}\sum_i (x_i-\mu)^2 \end{cases}$$

$$\begin{cases} N_k = \sum_{i=1}^{N} \gamma(i,k) \\ \mu_k = \dfrac{1}{N_k}\sum_{i=1}^{N} \gamma(i,k)x_i \\ \Sigma_k = \dfrac{1}{N_k}\sum_{i=1}^{N} \gamma(i,k)(x_i-\mu_k)(x_i-\mu_k)^T \\ \pi_k = \dfrac{N_k}{N} = \dfrac{1}{N}\sum_{i=1}^{N}\gamma(i,k) \end{cases}$$

EM算法

EM算法的提出

假定有训练集

$$\left\{x^{(1)}, x^{(2)}, \cdots, x^{(m)}\right\}$$

包含m个独立样本，希望从中找到该组数据的模型p(x,z)的参数。

通过最大似然估计建立目标函数

取对数似然函数

$$l(\theta)=\sum_{i=1}^{m}\log p(x;\theta)$$

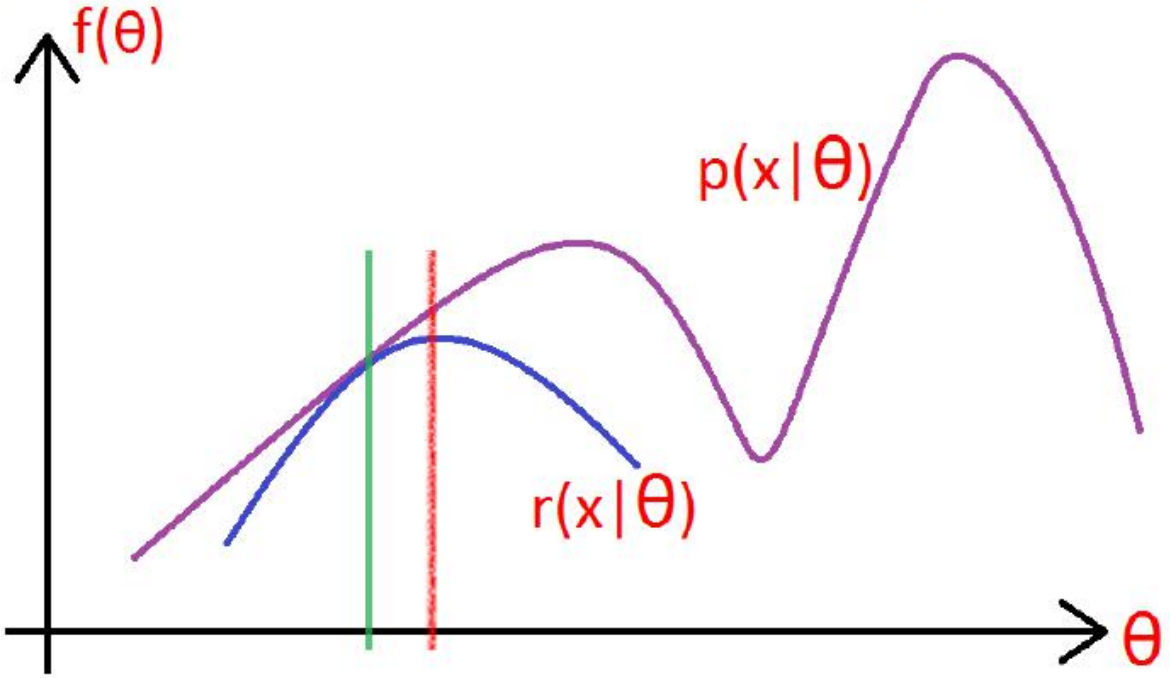
$$=\sum_{i=1}^{m}\log\sum_{z}p(x,z;\theta)$$

问题的提出

z是隐随机变量，不方便直接找到参数估计。

策略：计算l(θ)下界，求该下界的最大值；

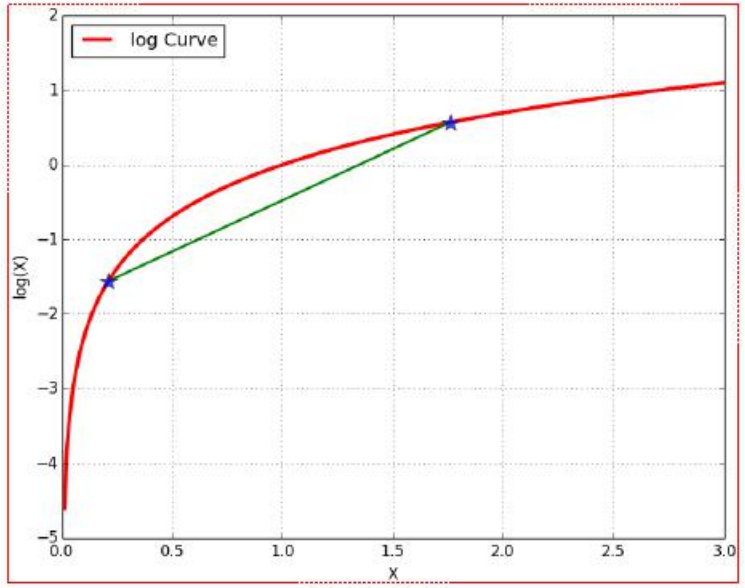
重复该过程，直到收敛到局部最大值。



Jessen不等式

令Qi是z的某一个分布，Qi≥0，有：

$$l(\theta)=\sum_{i=1}^{m}\log\sum_{z}p(x,z;\theta)=\sum_{i=1}^{m}\log\sum_{z^{(i)}}p(x^{(i)},z^{(i)};\theta)$$



$$=\sum_{i=1}^{m}\log\sum_{z^{(i)}}Q_i(z^{(i)})\frac{p(x^{(i)},z^{(i)};\theta)}{Q_i(z^{(i)})}$$

$$\geq\sum_{i=1}^{m}\sum_{z^{(i)}}Q_i(z^{(i)})\log\frac{p(x^{(i)},z^{(i)};\theta)}{Q_i(z^{(i)})}$$

The last step of this derivation used Jensen's inequality. Specifically, $f(x) = \log x$ is a concave function, since $f''(x) = -1/x^2 < 0$ over its domain $x \in \mathbb{R}^+$. Also, the term

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

in the summation is just an expectation of the quantity $\left[p(x^{(i)}, z^{(i)}; \theta)/Q_i(z^{(i)})\right.$ with respect to $z^{(i)}$ drawn according to the distribution given by Q_i. By Jensen's inequality, we have

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

where the "$z^{(i)} \sim Q_i$" subscripts above indicate that the expectations are with respect to $z^{(i)}$ drawn from Q_i.

寻找尽量紧的下界

为了使等号成立

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

进一步分析

$$Q_i\left(z^{(i)}\right) \propto p\left(x^{(i)}, z^{(i)}; \theta\right) \qquad \sum_z Q_i\left(z^{(i)}\right) = 1$$

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

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

$$= p(z^{(i)} \mid x^{(i)}; \theta)$$

EM算法整体框架

Repeat until convergence {

(E-step) For each i, set

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

(M-step) Set

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

}

EM的收敛性

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

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

坐标上升

Remark. If we define

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

the we know $\ell(\theta) \geq J(Q, \theta)$ from our previous derivation. The EM can also be viewed a coordinate ascent on J, in which the E-step maximizes it with respect to Q, and the M-step maximizes it with respect to θ.

从理论公式推导GMM

随机变量X是有K个高斯分布混合而成，取各个高斯分布的概率为$\varphi_1 \varphi_2 \cdots \varphi_K$，第i个高斯分布的均值为$\mu_i$，方差为$\Sigma_i$。若观测到随机变量X的一系列样本$x_1, x_2, \ldots, x_n$，试估计参数$\varphi$，$\boldsymbol{\mu}$，$\boldsymbol{\Sigma}$。

E-step

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

M-step

将多项分布和高斯分布的参数带入：

$$\sum_{i=1}^{m}\sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \phi, \mu, \Sigma)}{Q_i(z^{(i)})}$$

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

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

对均值求偏导

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

$$= -\nabla_{\mu_l} \sum_{i=1}^{m} \sum_{j=1}^{k} w_j^{(i)} \frac{1}{2}(x^{(i)} - \mu_j)^T \Sigma_j^{-1}(x^{(i)} - \mu_j)$$

$$= \frac{1}{2} \sum_{i=1}^{m} w_l^{(i)} \nabla_{\mu_l} 2\mu_l^T \Sigma_l^{-1} x^{(i)} - \mu_l^T \Sigma_l^{-1} \mu_l$$

$$= \sum_{i=1}^{m} w_l^{(i)} \left(\Sigma_l^{-1} x^{(i)} - \Sigma_l^{-1} \mu_l\right)$$

高斯分布的均值

令上式等于0，解的均值：

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

高斯分布的方差：求偏导等于0

$$\Sigma_j = \frac{\sum_{i=1}^{m} w_j^{(i)} (x^{(i)} - \mu_j)(x^{(i)} - \mu_j)^T}{\sum_{i=1}^{m} w_j^{(i)}}$$

多项分布的参数

考察M-step的目标函数，对于φ，删除常数项

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

得到

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

拉格朗日乘子法

由于多项分布的概率和为1，建立拉格朗日方程

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

- 求解的φ_i一定非负，不用考虑$\varphi_i \geq 0$这个条件

求偏导，等于0

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

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

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

总结

对于所有的数据点，可以看作组份k生成了这些点。组份k是一个标准的高斯分布，利用上面的结论：$\{\gamma(i,k)x_i \mid i=1,2,\cdots N\}$

$$
\begin{cases}
\mu_k = \dfrac{1}{N_k}\sum_{i=1}^{N}\gamma(i,k)x_i \\
\Sigma_k = \dfrac{1}{N_k}\sum_{i=1}^{N}\gamma(i,k)(x_i-\mu_k)(x_i-\mu_k)^T \\
\pi_k = \dfrac{1}{N}\sum_{i=1}^{N}\gamma(i,k) \\
N_k = N\cdot\pi_k
\end{cases}
$$