

# 法律声明

□ 本课件包括演示文稿、示例、代码、题库、视频和声音等内容，小象学院和主讲老师拥有完全知识产权的权利；只限于善意学习者在本课程使用，不得在课程范围外向任何第三方散播。任何其他人或机构不得盗版、复制、仿造其中的创意及内容，我们保留一切通过法律手段追究违反者的权利。

□ 课程详情请咨询

■ 微信公众号：小象

■ 新浪微博：ChinaHadoop



# EM算法

---



小象学院  
ChinaHadoop.cn

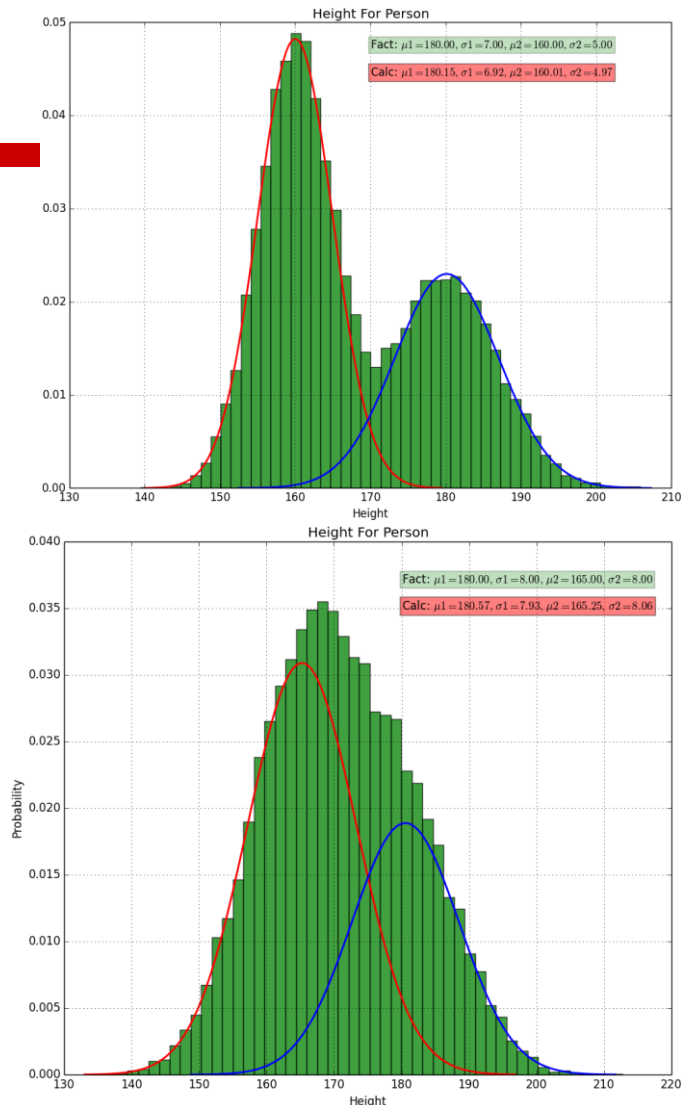
邹博

# 主要内容

---

- 通过实例直观求解高斯混合模型GMM
  - 适合快速掌握GMM, 及编程实现
- 通过最大似然估计详细推导EM算法
  - 适合理论层面的深入理解
  - 用坐标上升理解EM的过程
- 推导GMM的参数 $\phi$ 、 $\mu$ 、 $\sigma$ 
  - 复习多元高斯模型
  - 复习拉格朗日乘子法
- Arthur Dempster, Nan Laird, Donald Rubin, 1977
  - C.F. Jeff Wu(1983)给出了收敛的进一步分析

# EM Code



```
em3.py x
def calcEM(height):
    N = len(height)
    gp = 0.5 #girl probability
    bp = 0.5 #boy probability
    gmu,gsigma = min(height),1 #先验: 直接取最大和最小值
    bmu,bsigma = max(height),1
    ggamma = range(N)
    bgamma = range(N)
    cur = [gp, bp, gmu, gsigma, bmu, bsigma]
    now = []

    times = 0
    while times < 100:
        i = 0
        for x in height:
            ggamma[i] = gp * gauss(x, gmu, gsigma)
            bgamma[i] = bp * gauss(x, bmu, bsigma)
            s = ggamma[i] + bgamma[i]
            ggamma[i] /= s
            bgamma[i] /= s
            i += 1

        gn = sum(ggamma)
        gp = float(gn) / float(N)
        bn = sum(bgamma)
        bp = float(bn) / float(N)
        gmu = averageWeight(height, ggamma, gn)
        gsigma = varianceWeight(height, ggamma, gmu, gn)
        bmu = averageWeight(height, bgamma, bn)
        bsigma = varianceWeight(height, bgamma, bmu, bn)

        now = [gp, bp,gmu,gsigma,bmu,bsigma]
        if isSame(cur, now):
            break
        cur = now
        print "Times:\t", times
        print "Girl mean/gsigma:\t", gmu,gsigma
        print "Boy mean/bsigma:\t", bmu,bsigma
        print "Boy/Girl:\t", bn, gn, bn+gn
        print "\n\n"
        times += 1
    return now
```

# 复习：Jensen不等式：若f是凸函数

## □ 基本Jensen不等式

$$f(\theta x + (1 - \theta)y) \leq \theta f(x) + (1 - \theta)f(y)$$

## □ 若 $\theta_1, \dots, \theta_k \geq 0, \theta_1 + \dots + \theta_k = 1$

## □ 则 $f(\theta_1 x_1 + \dots + \theta_k x_k) \leq \theta_1 f(x_1) + \dots + \theta_k f(x_k)$

## □ 若 $p(x) \geq 0$ on $S \subseteq \text{dom } f, \int_S p(x) dx = 1$

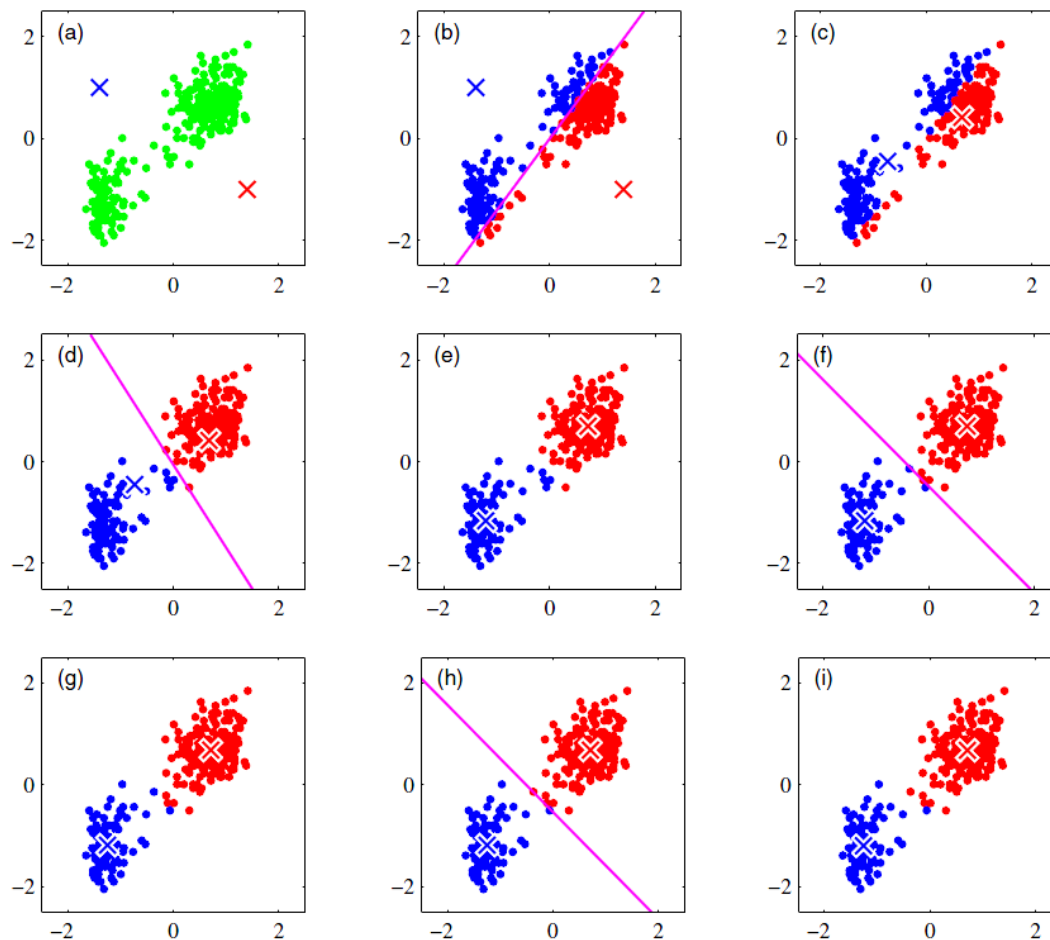
## □ 则 $f\left(\int_S p(x)x dx\right) \leq \int_S f(x)p(x) dx$

$$f(\mathbf{E} x) \leq \mathbf{E} f(x)$$

# 引子：K-means算法

- 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 $S=X_1, X_2, \dots, X_m$ ，则算法步骤为：
  - 选择初始的k个簇中心 $\mu_1 \mu_2 \dots \mu_k$
  - 将样本 $x_i$ 标记为距离簇中心最近的簇： $label_i = \arg \min_{1 \leq j \leq k} \|x_i - \mu_j\|$
  - 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

# K-means过程



# 思考

---

- 经典的K-means聚类方法，能够非常方便的将来标记的样本分成若干簇；
- 但无法给出某个样本属于该簇的后验概率。
- 其他方法可否处理未标记样本呢？



# 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：

$$\begin{aligned} P &= pp(1-p)ppp(1-p)(1-p)pp \\ &= p^7(1-p)^3 \end{aligned}$$

■ 最优解是：  $p=0.7$

# 二项分布的最大似然估计

- 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：

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

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

# 进一步考察

---

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

# 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

$$\begin{aligned}l(x) &= \log \prod_i \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_i - \mu)^2}{2\sigma^2}} \\&= \sum_i \log \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x_i - \mu)^2}{2\sigma^2}} \\&= \left( \sum_i \log \frac{1}{\sqrt{2\pi}\sigma} \right) + \left( \sum_i -\frac{(x_i - \mu)^2}{2\sigma^2} \right) \\&= -\frac{n}{2} \log(2\pi\sigma^2) - \frac{1}{2\sigma^2} \sum_i (x_i - \mu)^2\end{aligned}$$

# 参数估计的结论

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

□ 将目标函数对参数 $\mu, \sigma$ 分别求偏导，很容易得到 $\mu, \sigma$ 的式子：

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

# 符合直观想象

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

- 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。

# 问题：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 $\pi_1\pi_2\cdots\pi_K$ ，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 $x_1, x_2, \dots, x_n$ ，试估计参数 $\pi$ ,  $\mu$ ,  $\Sigma$ 。

# 建立目标函数

---

## □ 对数似然函数

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

- 由于在对数函数里面又有加和，无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：估算数据来自哪个组份

- 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

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

- 上式中的 $\mu$ 和 $\Sigma$ 也是待估计的值，因此采样迭代法：在计算 $\gamma(i, k)$ 时假定 $\mu$ 和 $\Sigma$ 已知；
  - 需要先验给定 $\mu$ 和 $\Sigma$ 。
  - $\gamma(i, k)$ 亦可看成组份 $k$ 在生成数据 $x_i$ 时所做的贡献。

## 第二步：估计每个组份的参数

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

$$\left\{ \begin{array}{l} \mu = \frac{1}{n} \sum_i x_i \\ \sigma^2 = \frac{1}{n} \sum_i (x_i - \mu)^2 \end{array} \right. \quad \left\{ \begin{array}{l} N_k = \sum_{i=1}^N \gamma(i,k) \\ \mu_k = \frac{1}{N_k} \sum_{i=1}^N \gamma(i,k)x_i \\ \Sigma_k = \frac{1}{N_k} \sum_{i=1}^N \gamma(i,k)(x_i - \mu_k)(x_i - \mu_k)^T \\ \pi_k = \frac{N_k}{N} = \frac{1}{N} \sum_{i=1}^N \gamma(i,k) \end{array} \right.$$

# EM算法的提出

---

□ 假定有训练集

$$\{x^{(1)}, x^{(2)}, \dots, x^{(m)}\}$$

□ 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

---

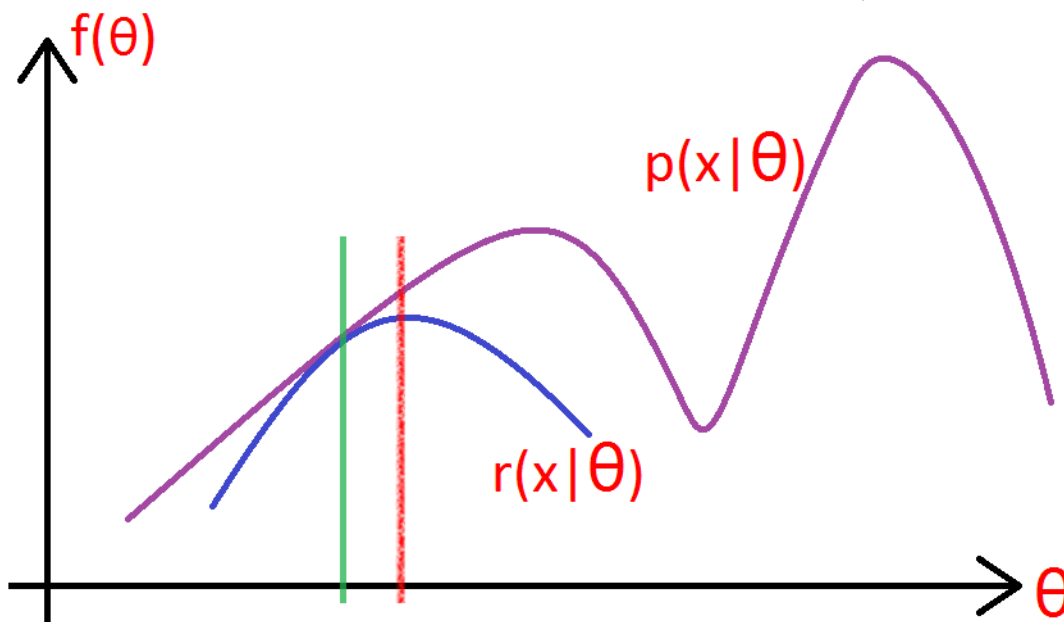
## □ 取对数似然函数

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

# 问题的提出

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

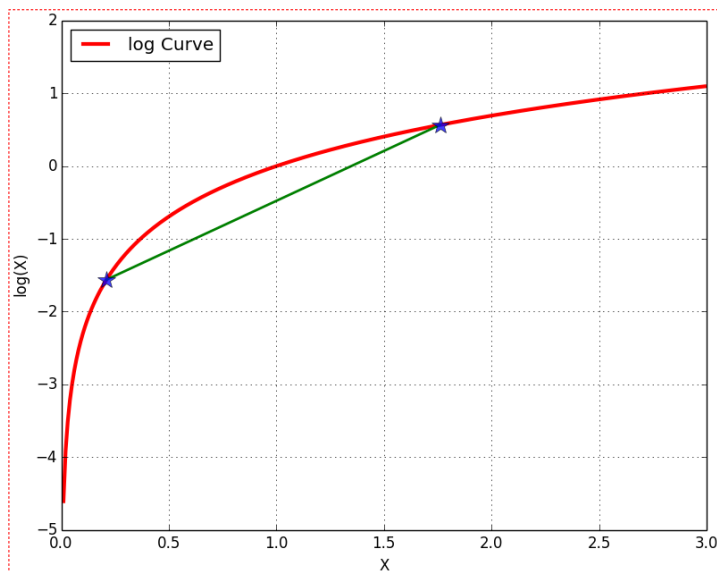
- $z$  是隐随机变量，不方便直接找到参数估计。  
策略：计算  $l(\theta)$  下界，求该下界的最大值；  
重复该过程，直到收敛到局部最大值。



# Jensen不等式

□ 令 $Q_i$ 是 $z$ 的某一个分布,  $Q_i \geq 0$ , 有:

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



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



# 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

---

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

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

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

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

# 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

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

}

# 坐标上升

---

**Remark.** If we define

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

then we know  $\ell(\theta) \geq J(Q, \theta)$  from our previous derivation. The EM can also be viewed as 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  $\theta$ .

# EM的收敛性

---

Expectation-maximization works to improve  $Q(\theta|\theta^{(t)})$  rather than directly improving  $\log p(\mathbf{X}|\theta)$ . Here is shown that improvements to the former imply improvements to the latter.

For any  $\mathbf{Z}$  with non-zero probability  $p(\mathbf{Z}|\mathbf{X}, \theta)$ , we can write

$$\log p(\mathbf{X}|\theta) = \log p(\mathbf{X}, \mathbf{Z}|\theta) - \log p(\mathbf{Z}|\mathbf{X}, \theta).$$

We take the expectation over possible values of the unknown data  $\mathbf{Z}$  under the current parameter estimate  $\theta^{(t)}$  by multiplying both sides by  $p(\mathbf{Z}|\mathbf{X}, \theta^{(t)})$  and summing (or integrating) over  $\mathbf{Z}$ . The left-hand side is the expectation of a constant, so we get:

$$\begin{aligned}\log p(\mathbf{X}|\theta) &= \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \theta^{(t)}) \log p(\mathbf{X}, \mathbf{Z}|\theta) - \sum_{\mathbf{Z}} p(\mathbf{Z}|\mathbf{X}, \theta^{(t)}) \log p(\mathbf{Z}|\mathbf{X}, \theta) \\ &= Q(\theta|\theta^{(t)}) + H(\theta|\theta^{(t)}),\end{aligned}$$

where  $H(\theta|\theta^{(t)})$  is defined by the negated sum it is replacing. This last equation holds for any value of  $\theta$  including  $\theta = \theta^{(t)}$ ,

$$\log p(\mathbf{X}|\theta^{(t)}) = Q(\theta^{(t)}|\theta^{(t)}) + H(\theta^{(t)}|\theta^{(t)}),$$

and subtracting this last equation from the previous equation gives

$$\log p(\mathbf{X}|\theta) - \log p(\mathbf{X}|\theta^{(t)}) = Q(\theta|\theta^{(t)}) - Q(\theta^{(t)}|\theta^{(t)}) + H(\theta|\theta^{(t)}) - H(\theta^{(t)}|\theta^{(t)}),$$

However, [Gibbs' inequality](#) tells us that  $H(\theta|\theta^{(t)}) \geq H(\theta^{(t)}|\theta^{(t)})$ , so we can conclude that

$$\log p(\mathbf{X}|\theta) - \log p(\mathbf{X}|\theta^{(t)}) \geq Q(\theta|\theta^{(t)}) - Q(\theta^{(t)}|\theta^{(t)}).$$

In words, choosing  $\theta$  to improve  $Q(\theta|\theta^{(t)})$  beyond  $Q(\theta^{(t)}|\theta^{(t)})$  can not cause  $\log p(\mathbf{X}|\theta)$  to decrease below  $\log p(\mathbf{X}|\theta^{(t)})$ , and so the marginal likelihood of the data is non-decreasing.

# 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 $x_1, x_2, \dots, x_n$ ，试估计参数 $\varphi$ ， $\mu$ ， $\Sigma$ 。

# 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：

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



# 对均值求偏导

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

# 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 $\phi$ ，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

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

■ 求解的 $\phi_i$ 一定非负，不用考虑 $\phi_i \geq 0$ 这个条件

# 求偏导，等于0

---

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

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

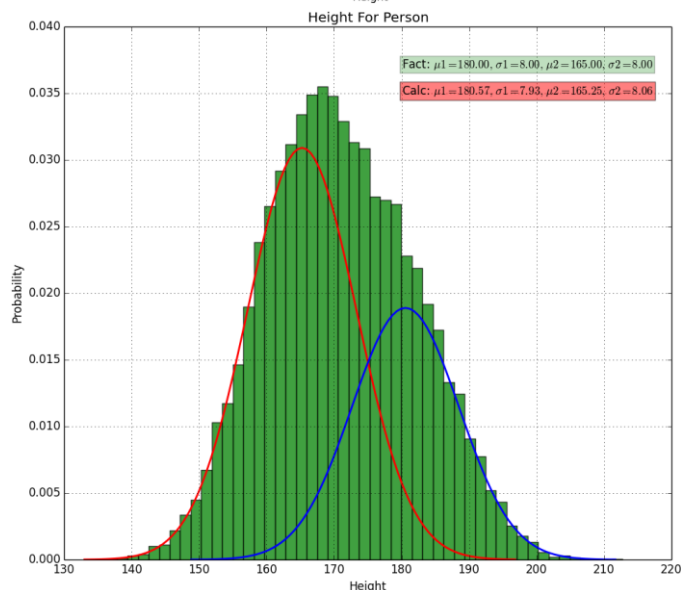
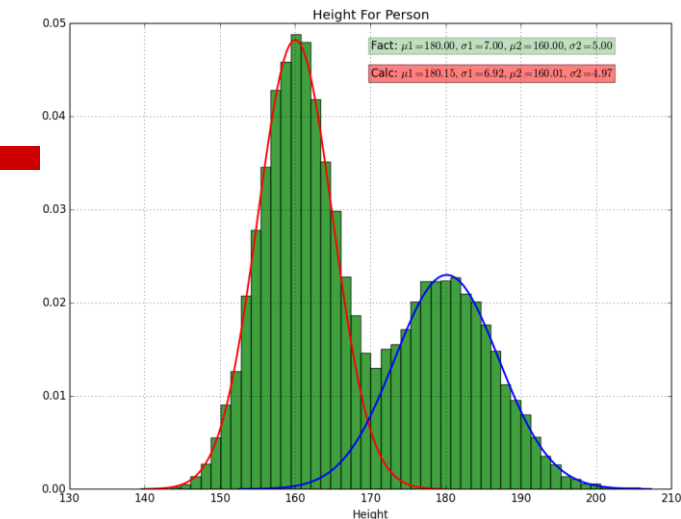
$$\phi_j := \frac{1}{m} \sum_{i=1}^m w_j^{(i)}$$

# 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： $\{\gamma(i,k)x_i \mid i=1,2,\dots,N\}$

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

# EM Code



```
em3.py x
def calcEM(height):
    N = len(height)
    gp = 0.5 #girl probability
    bp = 0.5 #boy probability
    gmu,gsigma = min(height),1 #先验: 直接取最大和最小值
    bmu,bsigma = max(height),1
    ggamma = range(N)
    bgamma = range(N)
    cur = [gp, bp, gmu, gsigma, bmu, bsigma]
    now = []

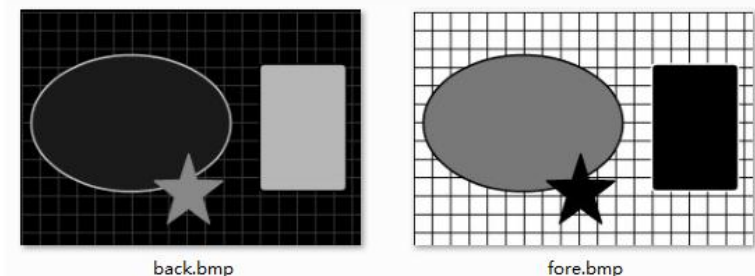
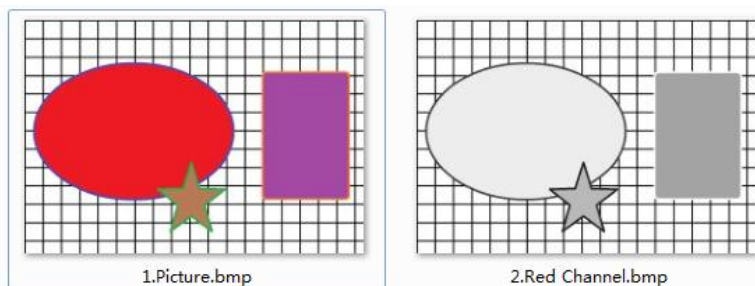
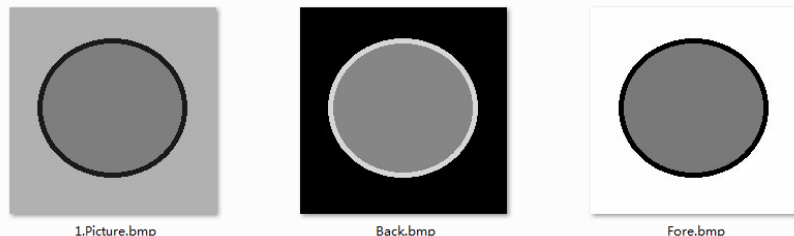
    times = 0
    while times < 100:
        i = 0
        for x in height:
            ggamma[i] = gp * gauss(x, gmu, gsigma)
            bgamma[i] = bp * gauss(x, bmu, bsigma)
            s = ggamma[i] + bgamma[i]
            ggamma[i] /= s
            bgamma[i] /= s
            i += 1

        gn = sum(ggamma)
        gp = float(gn) / float(N)
        bn = sum(bgamma)
        bp = float(bn) / float(N)
        gmu = averageWeight(height, ggamma, gn)
        gsigma = varianceWeight(height, ggamma, gmu, gn)
        bmu = averageWeight(height, bgamma, bn)
        bsigma = varianceWeight(height, bgamma, bmu, bn)

        now = [gp, bp,gmu,gsigma,bmu,bsigma]
        if isSame(cur, now):
            break
        cur = now
        print "Times:\t", times
        print "Girl mean/gsigma:\t", gmu,gsigma
        print "Boy mean/bsigma:\t", bmu,bsigma
        print "Boy/Girl:\t", bn, gn, bn+gn
        print "\n\n"
        times += 1
    return now
```



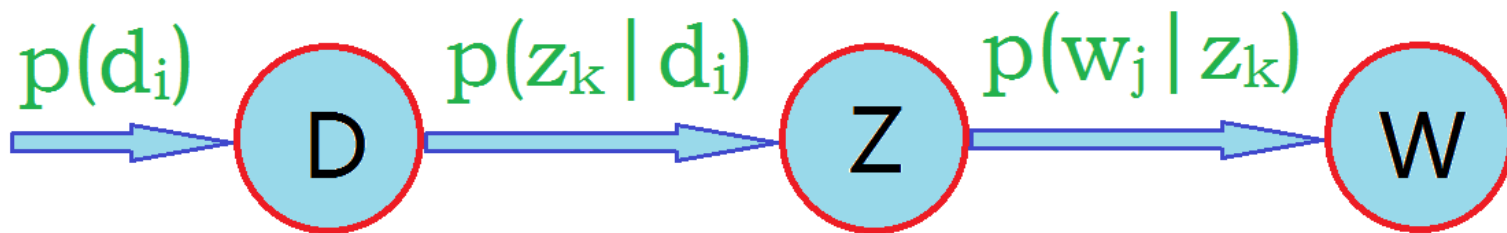
# GMM与图像

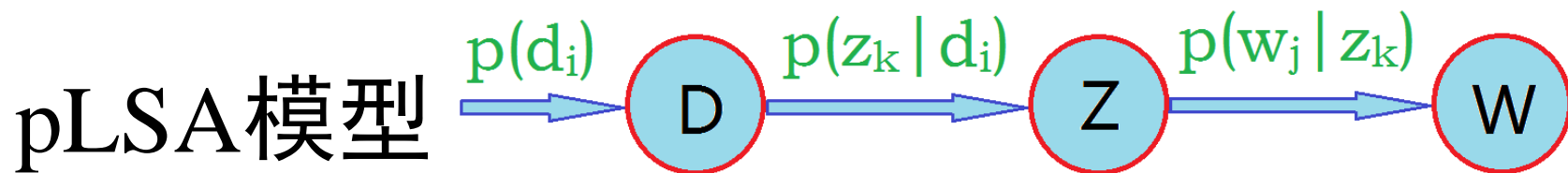


```
def composite(band, parameter):  
    c1 = parameter[0]  
    mu1 = parameter[2]  
    sigma1 = parameter[3]  
    c2 = parameter[1]  
    mu2 = parameter[4]  
    sigma2 = parameter[5]  
  
    p1 = []  
    p2 = []  
    for pixel in band:  
        p1.append(c1 * gauss(pixel, mu1, sigma1))  
        p2.append(c2 * gauss(pixel, mu2, sigma2))  
  
    scale(p1)  #灰度均衡  
    scale(p2)  
    return [p1, p2]  
  
if __name__ == "__main__":  
    im = Image.open('.\\Pic\\test.bmp')  
    print im.format, im.size, im.mode  
  
    im = im.split()[0]  #只处理第一个通道  
    nb = []  #处理后的新通道  
    data = list(im.getdata())  
    parameter = GMM(data)  
    t = composite(data, parameter)  
  
    im1 = Image.new('L', im.size)  
    im1.putdata(t[0])
```

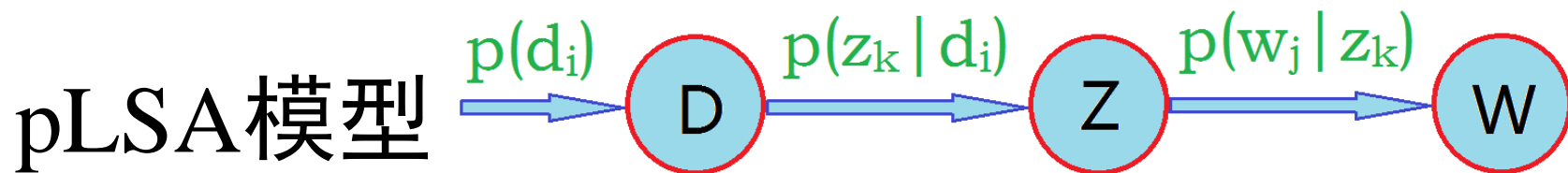
## 附：pLSA模型

- 基于概率统计的pLSA模型(probabilistic Latent Semantic Analysis, 概率隐语义分析), 增加了主题模型, 形成简单的贝叶斯网络, 可以使用EM算法学习模型参数。





- **D**代表文档，**Z**代表主题(隐含类别)，**W**代表单词；
  - $P(d_i)$ 表示文档 $d_i$ 的出现概率，
  - $P(z_k | d_i)$ 表示文档 $d_i$ 中主题 $z_k$ 的出现概率，
  - $P(w_j | z_k)$ 表示给定主题 $z_k$ 出现单词 $w_j$ 的概率。
- 每个主题在所有词项上服从多项分布，每个文档在所有主题上服从多项分布。
- 整个文档的生成过程是这样的：
  - 以 $P(d_i)$ 的概率选中文档 $d_i$ ；
  - 以 $P(z_k | d_i)$ 的概率选中主题 $z_k$ ；
  - 以 $P(w_j | z_k)$ 的概率产生一个单词 $w_j$ 。



□ 观察数据为 $(d_i, w_j)$ 对，主题 $z_k$ 是隐含变量。

□  $(d_i, w_j)$ 的联合分布为

$$P(d_i, w_j) = P(w_j | d_i)P(d_i)$$

$$P(w_j | d_i) = \sum_{k=1}^K P(w_j | z_k)P(z_k | d_i)$$

□ 而  $P(w_j | z_k), P(z_k | d_i)$ 对应了两组多项分布，而计算每个文档的主题分布，就是该模型的任务目标。

最大似然估计： $w_j$ 在 $d_i$ 中出现的次数  $n(d_i, w_j)$

$$L = \prod_{i=1}^N \prod_{j=1}^M P(d_i, w_j) = \prod_i \prod_j P(d_i, w_j)^{n(d_i, w_j)}$$

$$l = \sum_i \sum_j n(d_i, w_j) \log P(d_i, w_j)$$

$$= \sum_i \sum_j n(d_i, w_j) \log P(w_j | d_i) P(d_i)$$

$$P(d_i, w_j) = P(w_j | d_i) P(d_i)$$
$$P(w_j | d_i) = \sum_{k=1}^K P(w_j | z_k) P(z_k | d_i)$$

$$= \sum_i \sum_j n(d_i, w_j) \log \left( \sum_{k=1}^K P(w_j | z_k) P(z_k | d_i) \right) P(d_i)$$

$$= \sum_i \sum_j n(d_i, w_j) \log \left( \sum_{k=1}^K P(w_j | z_k) P(z_k | d_i) P(d_i) \right)$$

# 目标函数分析

- 观察数据为 $(d_i, w_j)$ 对，主题 $z_k$ 是隐含变量。
- 目标函数 
$$l = \sum_i \sum_j n(d_i, w_j) \log \left( \sum_{k=1}^K P(w_j | z_k) P(z_k | d_i) P(d_i) \right)$$
- 未知变量/自变量  $P(w_j | z_k), P(z_k | d_i)$
- 使用逐次逼近的办法：
  - 假定 $P(z_k | d_i), P(w_j | z_k)$ 已知，求隐含变量 $z_k$ 的后验概率；
  - 在 $(d_i, w_j, z_k)$ 已知的前提下，求关于参数 $P(z_k | d_i), P(w_j | z_k)$ 的似然函数期望的最大值，得到最优解 $P(z_k | d_i), P(w_j | z_k)$ ，带入上一步，从而循环迭代；
  - 即：EM算法。

# 求隐含变量主题 $z_k$ 的后验概率

□ 假定 $P(z_k|d_i)$ 、 $P(w_j|z_k)$ 已知，求隐含变量 $z_k$ 的后验概率；

$$P(z_k | d_i, w_j) = \frac{P(w_j | z_k)P(z_k | d_i)}{\sum_{l=1}^K P(w_j | z_l)P(z_l | d_i)}$$

□ 在 $(d_i, w_j, z_k)$ 已知的前提下，求关于参数 $P(z_k|d_i)$ 、 $P(w_j|z_k)$ 的似然函数期望的最大值，得到最优解 $P(z_k|d_i)$ 、 $P(w_j|z_k)$ ，带入上一步，从而循环迭代；

# 分析似然函数期望

- 在 $(d_i, w_j, z_k)$ 已知的前提下，求关于参数 $P(z_k|d_i)$ 、 $P(w_j|z_k)$ 的似然函数期望的最大值，得到最优解 $P(z_k|d_i)$ 、 $P(w_j|z_k)$ ，带入上一步，从而循环迭代；



# 关于参数 $P(z_k|d_i)P(w_j|z_k)$ 的似然函数期望

$$\begin{aligned}l &= \sum_i \sum_j n(d_i, w_j) \log P(d_i, w_j) \\&= \sum_i \sum_j n(d_i, w_j) \log (P(w_j | d_i) P(d_i)) \\&= \sum_i \sum_j n(d_i, w_j) (\log P(w_j | d_i) + \log P(d_i)) \\&= \left( \sum_i \sum_j n(d_i, w_j) \log P(w_j | d_i) \right) + \left( \sum_i \sum_j n(d_i, w_j) \log P(d_i) \right)\end{aligned}$$

$\Rightarrow$

$$\begin{aligned}l_{new} &= \left( \sum_i \sum_j n(d_i, w_j) \log P(w_j | d_i) \right) \\E(l_{new}) &= \sum_i \sum_j n(d_i, w_j) \sum_{k=1}^K P(z_k | d_i, w_j) \log P(w_j, z_k | d_i) \\&= \sum_i \sum_j n(d_i, w_j) \sum_{k=1}^K P(z_k | d_i, w_j) \log P(w_j | z_k) P(z_k | d_i)\end{aligned}$$

# 完成目标函数的建立

- 关于参数 $P(z_k|d_i)$ 、 $P(w_j|z_k)$  的函数 $E$ ，并且，带有概率加和为1的约束条件：

$$E = \sum_i \sum_j n(d_i, w_j) \sum_{k=1}^K P(z_k | d_i, w_j) \log P(w_j | z_k) P(z_k | d_i)$$
$$s.t. \begin{cases} \sum_{j=1}^M P(w_j | z_k) = 1 \\ \sum_{k=1}^K P(z_k | d_i) = 1 \end{cases}$$

- 显然，这是只有等式约束的求极值问题，使用Lagrange乘子法解决。

# 目标函数的求解

□ Lagrange 函数为：

$$\begin{aligned} Lag = & \sum_i \sum_j n(d_i, w_j) \sum_{k=1}^K P(z_k | d_i, w_j) \log P(w_j | z_k) P(z_k | d_i) \\ & + \sum_{k=1}^K \tau_k \left( 1 - \sum_{j=1}^M P(w_j | z_k) \right) + \sum_{i=1}^N \rho_i \left( 1 - \sum_{k=1}^K P(z_k | d_i) \right) \end{aligned}$$

□ 求驻点：

$$\begin{aligned} \frac{\partial Lag}{\partial P(w_j | z_k)} &= \frac{\sum_i n(d_i, w_j) P(z_k | d_i, w_j)}{P(w_j | z_k)} - \tau_k \stackrel{\text{令}}{=} 0 \\ \frac{\partial Lag}{\partial P(z_k | d_i)} &= \frac{\sum_j n(d_i, w_j) P(z_k | d_i, w_j)}{P(z_k | d_i)} - \rho_i \stackrel{\text{令}}{=} 0 \end{aligned}$$

# 分析第一个等式

$$\frac{\partial Lag}{\partial P(w_j | z_k)} = \frac{\sum_i n(d_i, w_j) P(z_k | d_i, w_j)}{P(w_j | z_k)} - \tau_k \stackrel{\text{令}}{=} 0$$

$$\Rightarrow \sum_i n(d_i, w_j) P(z_k | d_i, w_j) = \tau_k P(w_j | z_k)$$

$$\Rightarrow \sum_{m=1}^M \sum_i n(d_i, w_j) P(z_k | d_i, w_j) = \sum_{m=1}^M \tau_k P(w_j | z_k)$$

$$\Rightarrow \sum_{m=1}^M \sum_i n(d_i, w_j) P(z_k | d_i, w_j) = \tau_k \sum_{m=1}^M P(w_j | z_k)$$

$$\Rightarrow \sum_{m=1}^M \sum_i n(d_i, w_j) P(z_k | d_i, w_j) = \tau_k$$

$$\xrightarrow{\text{将 } \tau_k \text{ 代回第二式}} \sum_i n(d_i, w_j) P(z_k | d_i, w_j) = \sum_{m=1}^M \sum_i n(d_i, w_j) P(z_k | d_i, w_j) P(w_j | z_k)$$

$$\Rightarrow P(w_j | z_k) = \frac{\sum_i n(d_i, w_j) P(z_k | d_i, w_j)}{\sum_{m=1}^M \sum_i n(d_i, w_j) P(z_k | d_i, w_j)}$$

# 同理分析第二个等式

□ 求极值时的解——M-Step:

$$\begin{cases} P(w_j | z_k) = \frac{\sum_i n(d_i, w_j) P(z_k | d_i, w_j)}{\sum_{m=1}^M \sum_i n(d_i, w_j) P(z_k | d_i, w_j)} \\ P(z_k | d_i) = \frac{\sum_j n(d_i, w_j) P(z_k | d_i, w_j)}{\sum_{k=1}^K \sum_j n(d_i, w_j) P(z_k | d_i, w_j)} \end{cases}$$

□ 别忘了E-step:  $P(z_k | d_i, w_j) = \frac{P(w_j | z_k) P(z_k | d_i)}{\sum_{l=1}^K P(w_j | z_l) P(z_l | d_i)}$

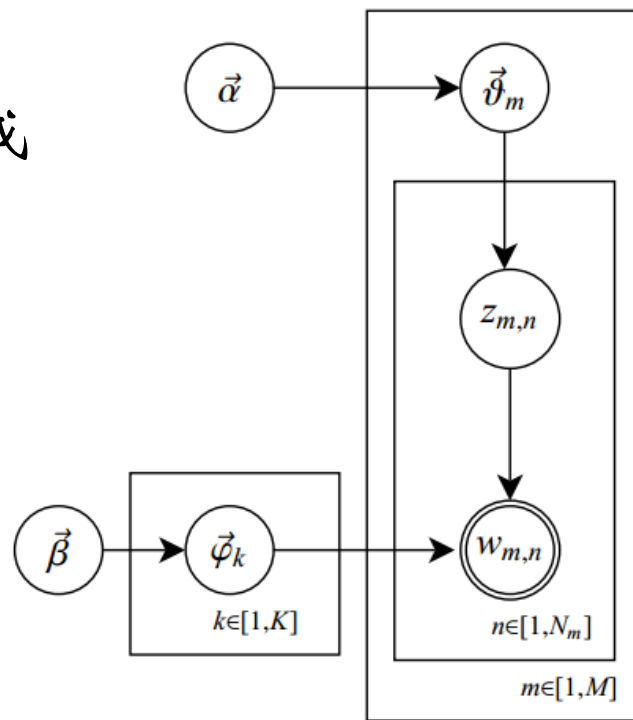
# pLSA的总结

---

- pLSA应用于信息检索、过滤、自然语言处理等领域，pLSA考虑到词分布和主题分布，使用EM算法来学习参数。
- 虽然推导略显复杂，但最终公式简洁清晰，很符合直观理解，需用心琢磨；此外，推导过程使用了EM算法，也是学习EM算法的重要素材。

# pLSA进一步思考 $p(d_i) \rightarrow D \xrightarrow{p(z_k | d_i)} Z \xrightarrow{p(w_j | z_k)} W$

- 相对于“简单”的链状贝叶斯网络，可否给出“词”“主题”“文档”更细致的网络拓扑，形成更具一般性的模型？
- pLSA不需要先验信息即可完成自学习——这是它的优势。如果在特定的要求下，需要有先验知识的影响呢？
- 答：LDA模型；
  - 三层结构的贝叶斯模型
  - 需要超参数



# 参考文献

---

- Prof. Andrew Ng. *Machine Learning*. Stanford University
- [https://en.wikipedia.org/wiki/Expectation-maximization\\_algorithm](https://en.wikipedia.org/wiki/Expectation-maximization_algorithm)



# 我们在这里

□ <http://wenda.ChinaHadoop.cn>

■ 视频/课程/社区

□ 微博

■ @ChinaHadoop

■ @邹博\_机器学习

□ 微信公众号

■ 小象学院

■ 大数据分析挖掘

互联网新技术在线教育领航者

小象问答 搜索标题、用户 全站内容搜索 提问 首页 动态 发现 话题 通知

全部 招聘求职 机器学习 大数据平台技术 DCon 大数据行业应用 NoSQL数据库 数据科学 江湖救急

发现 最新 推荐 热门 等待回复

graphviz has no attribute 'write' 贡献  
邹博 回复了问题 • 2 人关注 • 1 个回复 • 3 次浏览 • 2017-04-09 15:48

sklearn中如何理解Pipeline机制 贡献  
数据分析与数据挖掘 邹博 回复了问题 • 2 人关注 • 1 个回复 • 28 次浏览 • 2017-04-09 15:39

关于9.Logistic回归的ppt中第9页的对数线性函数 贡献  
机器学习 邹博 回复了问题 • 3 人关注 • 3 个回复 • 39 次浏览 • 2017-04-09 15:35

关于“贝叶斯估计中，最大后验概率估计就是结构化风险最小化的例子：当模型是条件概率分布，损失函数为对数损失函数，模型的复杂度由模型的先验概率表示，结构化风险最小化就等价于最大后验概率估计” 贡献  
机器学习 邹博 回复了问题 • 2 人关注 • 1 个回复 • 26 次浏览 • 2017-04-09 15:27

关于连续值的预测 贡献  
咨询 邹博 回复了问题 • 2 人关注 • 1 个回复 • 31 次浏览 • 2017-04-09 15:24

拉格朗日对偶函数为什么一定是凸函数 贡献  
数据科学 邹博 回复了问题 • 2 人关注 • 2 个回复 • 26 次浏览 • 2017-04-09 15:20

梯度下降公式中的斯堪J 是 贡献  
机器学习 邹博 回复了问题 • 2 人关注 • 1 个回复 • 29 次浏览 • 2017-04-09 15:17

深度学习适合做预测吗？ 贡献  
深度学习 邹博 回复了问题 • 2 人关注 • 1 个回复 • 27 次浏览 • 2017-04-09 15:15

关于6.4PCA\_FeatureSelection.py中plt.legend的参数疑问 贡献  
机器学习 邹博 回复了问题 • 2 人关注 • 1 个回复 • 28 次浏览 • 2017-04-09 15:04

@邹博 有哪些可以下载数据源的网站？ 贡献  
数据分析与数据挖掘 邹博 回复了问题 • 4 人关注 • 1 个回复 • 31 次浏览 • 2017-04-09 14:53

LDA主题模型 贡献  
机器学习 邹博 回复了问题 • 2 人关注 • 1 个回复 • 29 次浏览 • 2017-04-09 14:45

代码10.6bagging\_ridged老师提到了采样率设为0.2能够使峰值部分的数据被体现出来。这是为什么呢？ 贡献  
机器学习 邹博 回复了问题 • 2 人关注 • 1 个回复 • 22 次浏览 • 2017-04-09 14:26

GraphViz's executables not found 贡献  
机器学习 邹博 回复了问题 • 3 人关注 • 2 个回复 • 23 次浏览 • 2017-04-09 13:47

决策树中关于feature\_importances代码的问题 贡献  
机器学习 邹博 回复了问题 • 2 人关注 • 1 个回复 • 6 次浏览 • 2017-04-09 13:11

专题  
招聘求职  
大数据行业应用  
数据科学  
系统与编程  
云计算技术

热门话题 更多 >  
机器学习 907 个问题, 230 人关注  
spark 387 个问题, 172 人关注  
hadoop 1059 个问题, 155 人关注  
python数据分析 171 个问题, 28 人关注  
数据分析与数据挖掘 54 个问题, 111 人关注

热门用户 更多 >  
小心巴 14 个问题, 0 次赞同  
又又V 45 个问题, 22 次赞同  
铁甲无声 10 个问题, 0 次赞同  
带刀锦衣卫 13 个问题, 0 次赞同

---

感谢大家！

恳请大家批评指正！

甲文 金文 小篆 隶书 楷书

王 → 福 → 福 → 福 → 福

各式“王”字

Different Forms of the Graph for “King”

小臣簋  
Xiaozhentu Wine Container

宰甫簋  
Zaifu Food Container

戎禹鬲  
Shuyong Vessel

周季子白盘  
Guozijibai Plate

方彝  
Fang Wine Vessel

小盂鼎  
Xiaoyu Vessel

楚王熊渠戈  
King Chu's Yanshang Spear

楚王盨  
King Chu's Vessel

敦敦  
Duo Dun

戊	庚	辛	癸	子	寅	卯	辰
𠩺 𠩺	𠩺 𠩺 𠩺 𠩺	𠩺	𠩺 𠩺	𠩺 𠩺 𠩺	𠩺 𠩺	𠩺 𠩺 𠩺	𠩺 𠩺 𠩺
午	未	申	酉	其	令	卯	辰
𠩺	𠩺 𠩺 𠩺	𠩺 𠩺 𠩺	𠩺 𠩺	𠩺 𠩺 𠩺	𠩺 𠩺	𠩺 𠩺	𠩺 𠩺 𠩺
因	占	羌	豕	貞	于	牛	羊
𠩺 𠩺	𠩺 𠩺	𠩺 𠩺 𠩺	𠩺 𠩺 𠩺	𠩺	𠩺 𠩺	𠩺 𠩺 𠩺	𠩺 𠩺 𠩺
戎	雨	弔	旬	隹	不	牝	巳
𠩺 𠩺	𠩺 𠩺 𠩺	𠩺 𠩺	𠩺	𠩺	𠩺 𠩺	𠩺 𠩺 𠩺	𠩺 𠩺

## ■ 图像识别