

机器学习

Machine Learning

北京航空航天大学计算机学院智能识别与图像处理实验室
IRIP Lab, School of Computer Science and Engineering, Beihang University

黄 迪 刘庆杰

2020年秋季学期
Fall 2020

课前回顾

回归模型

训练集 (x_i, y_i)

学习算法

变量(特征) x → f → 计算 y

$$f(\mathbf{x}) = w_0 + w_1x_1 + \dots + w_mx_m$$

$$= w_0 + \sum_{i=1}^m w_i x_i$$

$$= \sum_{i=0}^m w_i x_i \quad [x_0 = 1]$$

如何表示 f ?

线性回归:
假设函数 f 为输入 x 的线性函数

$$\begin{aligned} f(\mathbf{x}) &= w_0 + w_1x_1 + \dots + w_mx_m \\ &= w_0 + \sum_{j=1}^m w_j x_j \end{aligned}$$

写成向量形式:
增加一维 $x_0=1$, 表示截距项

$$f(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$$

回归模型

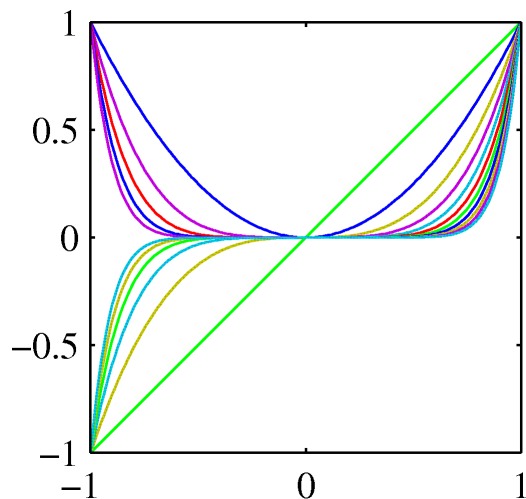
$$y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x})$$

基函数

最简单的情况下: $\phi_j(x) = x_j$

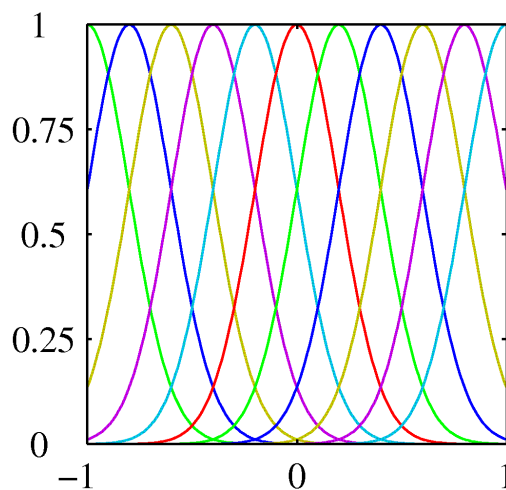
- 多项式基函数

$$\phi_j(x) = x^j$$



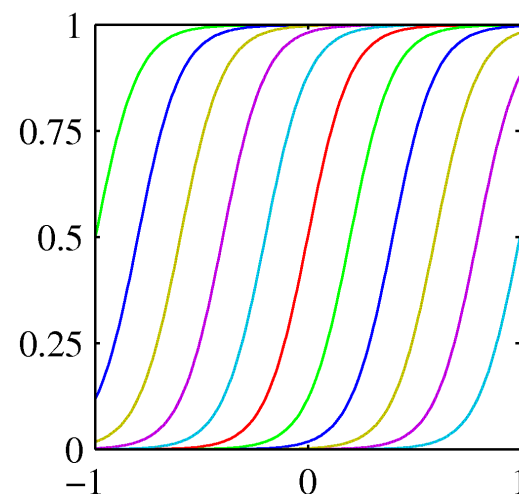
- 高斯基函数

$$\phi_j(x) = \exp\left\{-\frac{(x - \mu_j)^2}{2s^2}\right\}$$



- Sigmoid基函数

$$\phi_j(x) = \sigma\left(\frac{x - \mu_j}{s}\right) \quad \sigma(a) = \frac{1}{1 + \exp(-a)}$$



问题求解

- 问题本质：确定模型中的参数 \mathbf{w}^T
- 基本思想：基于训练集最小化预测值 f 与真实输出值 y 的差异
- 定义目标函数(又叫代价函数Cost Function):

$$J(\mathbf{w}) = \frac{1}{2} \sum_{i=1}^N (f(\mathbf{x}_i) - y_i)^2$$

- 进一步:

$$\hat{\mathbf{w}} = \arg \min_{\mathbf{w}} J(\mathbf{w}) = \arg \min_{\mathbf{w}} \left[\frac{1}{2} \sum_{i=1}^N (f(\mathbf{x}_i) - y_i)^2 \right]$$

梯度下降(Gradient Descent)

- 损失函数 $J(w)$

$$\min_w J(w)$$

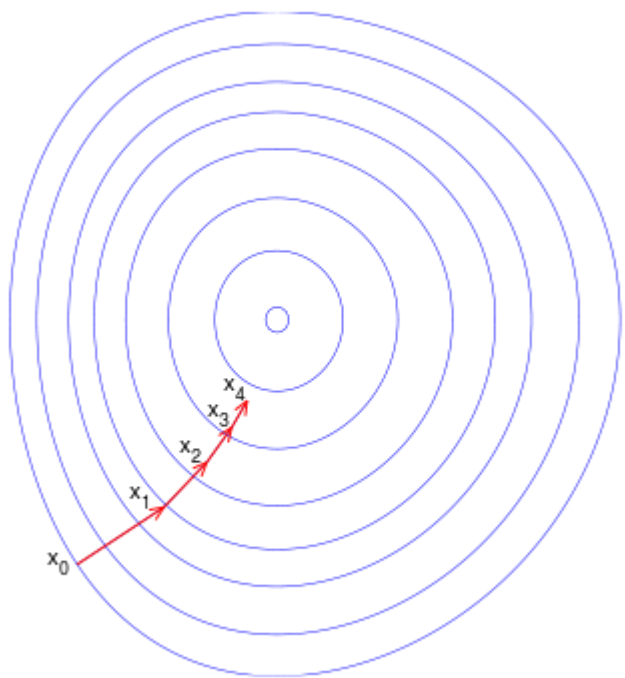
- 一般流程

- (1) 首先对 w 赋值，这个值可以是随机的，也可以是一个全零的向量。
- (2) 改变 w 的值，使得 $J(w)$ 按梯度下降的方向进行减少，直到收敛。

梯度下降(Gradient Descent)

- 梯度下降

梯度下降法是一个最优化算法，是求解无约束优化问题最简单和最基础的方法之一。



以负梯度方向为搜索方向
越接近目标值，步长越小，
前进越慢。

梯度下降(Gradient Descent)

- 梯度下降:

- 给定初始值 \mathbf{w}^0

- 更新 \mathbf{w} 使得 $J(\mathbf{w})$ 越来越小

$$w_j^t = w_j^{t-1} - \alpha \frac{\partial}{\partial w_j} J(\mathbf{w})$$

$$\frac{\partial}{\partial w_j} J(\mathbf{w}) = \sum_{i=1}^N (f(\mathbf{x}_i) - y_i) \cdot \mathbf{x}_{i,j}$$

- 同时更新 \mathbf{w} 的各维 $f(\mathbf{x}_i) = [\mathbf{w}^{t-1}]^T \mathbf{x}_i$

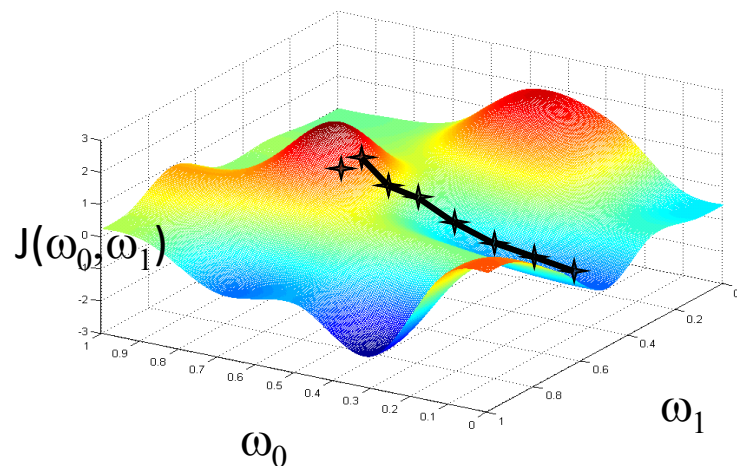
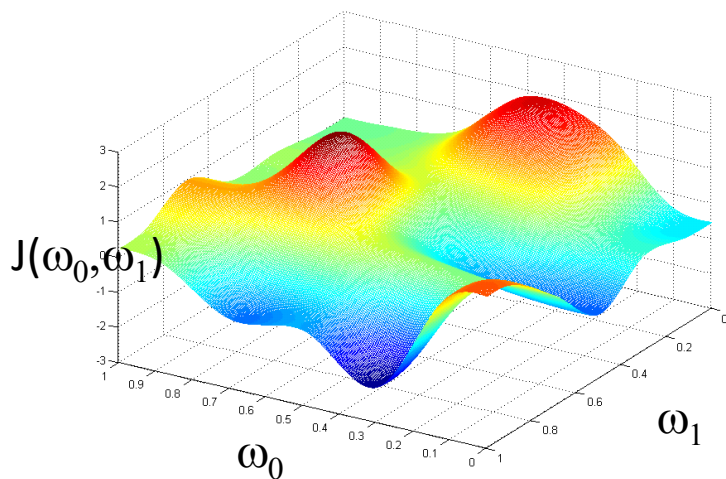
α 为学习率(Learning Rate)或更新步长

批处理梯度下降 (Batch Gradient Descent)

- 在梯度下降方法中，每次更新都利用所有数据，即：

$$w_j^t = w_j^{t-1} - \alpha \sum_{i=1}^N (f(\mathbf{x}_i) - y_i) \cdot x_j$$

在大样本条件下，批处理梯度下降的迭代速度很慢



随机梯度下降

(Stochastic Gradient Descent)

- 基本思想：如果条件对所有样本都成立，则对任一样本也成立

每次只用一个样本 (\mathbf{x}_r, y_r)

$$w_j^t = w_j^{t-1} - \alpha(f(\mathbf{x}_r) - y_r) \cdot \mathbf{x}_{r,j}$$

- 收敛速度较快
- 不太容易陷入局部极值
- 对大样本数据较有效

又称在线学习(Online Learning)

- 变化形式

可离线学习，每次“看”一个样本，对所有样本循环使用多次(一次循环称为一个Epoch)

每次可以看一些样本(Mini-Batch)

标准方程组 (Normal Equations)

- 目标函数：

$$\begin{aligned} J(\mathbf{w}) &= \sum_{i=1}^N (f(\mathbf{x}_i) - y_i)^2 \\ &= \sum_{i=1}^N (\mathbf{w}^T \mathbf{x}_i - y_i)^2 \end{aligned}$$

- 直接求导，并令其等于0，求得极值

标准方程组 (Normal Equations)

- 求导：

$$\begin{aligned}\frac{\partial}{\partial \mathbf{w}} J(\mathbf{w}) &= \frac{\partial}{\partial \mathbf{w}} (X\mathbf{w} - \mathbf{y})^T (X\mathbf{w} - \mathbf{y}) \\ &= 2X^T (X\mathbf{w} - \mathbf{y}) = 0\end{aligned}$$

$$\Rightarrow X^T X \mathbf{w} = X^T \mathbf{y}$$

- 得到模型的参数：

$$\hat{\mathbf{w}} = (X^T X)^{-1} X^T \mathbf{y}$$

梯度下降 vs 标准方程组

● 梯度下降法：

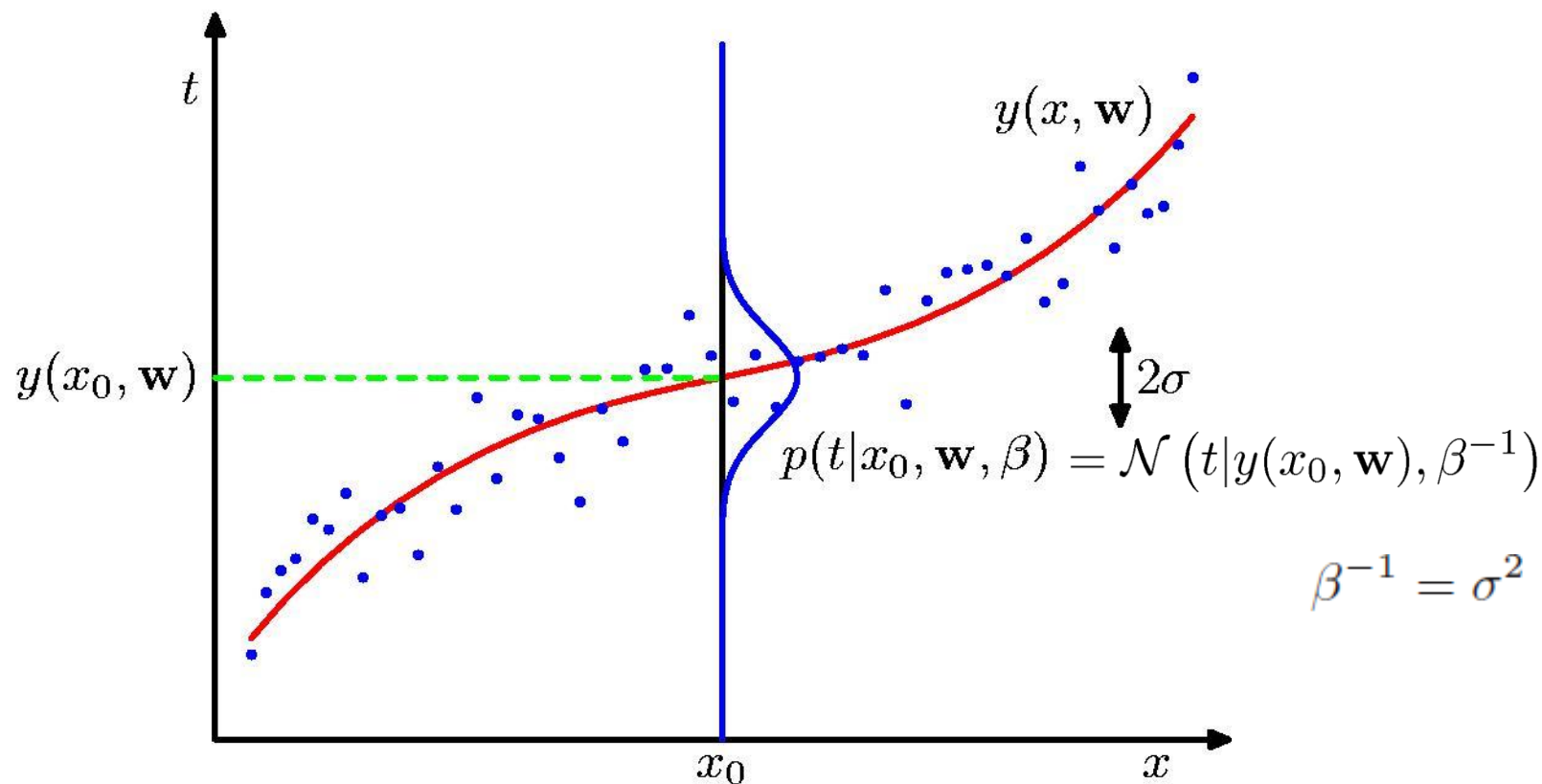
- 需要选择 α
- 需要迭代多次
- 需要数据归一化
- 样本量非常大时也适用

● 标准方程组：

- 不需要选择 α
- 不需要迭代多次
- 无需数据归一化
- 样本量大时不适用
(需要计算 $(X^T X)^{-1}$)

样本量较小时选用标准方程组求解
样本量较大时选用梯度下降法求解

再看曲线拟合



最大似然估计与贝叶斯估计

假定观测数据由确定的函数加高斯噪声组成

$$t = y(\mathbf{x}, \mathbf{w}) + \epsilon \quad p(\epsilon|\beta) = \mathcal{N}(\epsilon|0, \beta^{-1})$$

- **最大似然估计：**

把待估计的参数看做是确定的量，只是其取值未知。最佳估计就是使得产生以观测到的样本的概率最大的那个值。

- **贝叶斯估计：**

把待估计的参数看做是符合某种先验概率分布的随机变量。对样本进行观测的过程，就是把先验概率密度转化为后验概率密度，从而利用样本信息修正了对参数的初始估计值。

模型求解

- 最大似然估计(\mathbf{w} 是确定量)

$$p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = \prod_{n=1}^N \mathcal{N}(t_n|y(x_n, \mathbf{w}), \beta^{-1})$$

- 贝叶斯估计(\mathbf{w} 是随机量)

$$p(t|x, \mathbf{x}, \mathbf{t}) = \int p(t|x, \mathbf{w})p(\mathbf{w}|\mathbf{x}, \mathbf{t}) d\mathbf{w} = \mathcal{N}(t|m(x), s^2(x))$$

最大似然估计

$$p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = \prod_{n=1}^N \mathcal{N}(t_n | y(x_n, \mathbf{w}), \beta^{-1})$$

$$\ln p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) = \boxed{-\frac{\beta}{2} \sum_{n=1}^N \{y(x_n, \mathbf{w}) - t_n\}^2} + \frac{N}{2} \ln \beta - \frac{N}{2} \ln(2\pi)$$

Determine \mathbf{w}_{ML} by minimizing sum-of-squares error, $E(\mathbf{w})$.

$$p(t|x, \mathbf{w}_{\text{ML}}, \beta_{\text{ML}}) = \mathcal{N}(t | y(x, \mathbf{w}_{\text{ML}}), \beta_{\text{ML}}^{-1})$$

MAP: 离贝叶斯更近一步

$$p(\mathbf{w}|\alpha) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha^{-1}\mathbf{I}) = \left(\frac{\alpha}{2\pi}\right)^{(M+1)/2} \exp\left\{-\frac{\alpha}{2}\mathbf{w}^T\mathbf{w}\right\}$$

$$p(\mathbf{w}|\mathbf{x}, \mathbf{t}, \alpha, \beta) \propto p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta)p(\mathbf{w}|\alpha)$$

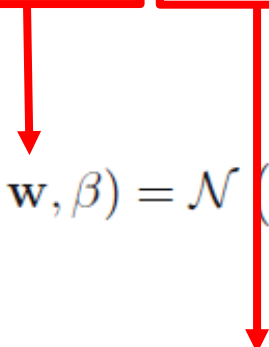
$$\beta\tilde{E}(\mathbf{w}) = \frac{\beta}{2} \sum_{n=1}^N \{y(x_n, \mathbf{w}) - t_n\}^2 + \frac{\alpha}{2}\mathbf{w}^T\mathbf{w}$$

Determine \mathbf{w}_{MAP} by minimizing regularized sum-of-squares error, $\tilde{E}(\mathbf{w})$.

$$\lambda = \alpha/\beta.$$

贝叶斯曲线拟合

$$p(t|x, \mathbf{x}, \mathbf{t}) = \int p(t|x, \mathbf{w}) p(\mathbf{w}|\mathbf{x}, \mathbf{t}) d\mathbf{w}.$$


$$p(t|x, \mathbf{w}, \beta) = \mathcal{N}(t|y(x, \mathbf{w}), \beta^{-1})$$

$$p(\mathbf{w}|\mathbf{x}, \mathbf{t}, \alpha, \beta) \propto p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta) p(\mathbf{w}|\alpha).$$

$$p(t|x, \mathbf{x}, \mathbf{t}) = \mathcal{N}(t|m(x), s^2(x))$$

贝叶斯曲线拟合

$$p(t|x, \mathbf{x}, \mathbf{t}) = \int p(t|x, \mathbf{w})p(\mathbf{w}|\mathbf{x}, \mathbf{t}) d\mathbf{w} = \mathcal{N}(t|m(x), s^2(x))$$

$$p(\mathbf{w}|\mathbf{x}, \mathbf{t}, \alpha, \beta) \propto p(\mathbf{t}|\mathbf{x}, \mathbf{w}, \beta)p(\mathbf{w}|\alpha)$$

$$y(\mathbf{x}, \mathbf{w}) = \sum_{j=0}^{M-1} w_j \phi_j(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x})$$

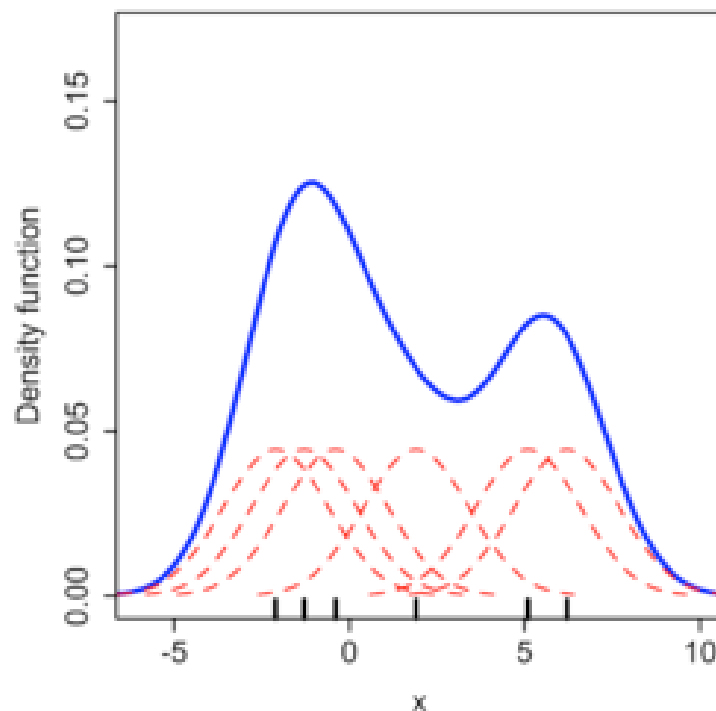
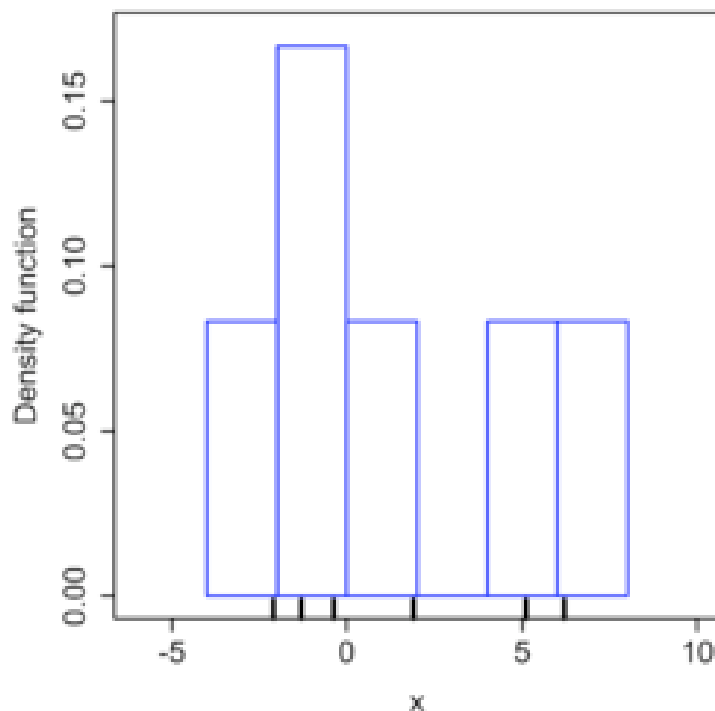
$$m(x) = \beta \boldsymbol{\phi}(x)^T \mathbf{S} \sum_{n=1}^N \boldsymbol{\phi}(x_n) t_n \quad s^2(x) = \beta^{-1} + \boldsymbol{\phi}(x)^T \mathbf{S} \boldsymbol{\phi}(x)$$

$$\mathbf{S}^{-1} = \alpha \mathbf{I} + \beta \sum_{n=1}^N \boldsymbol{\phi}(x_n) \boldsymbol{\phi}(x_n)^T \quad \boldsymbol{\phi}(x_n) = (x_n^0, \dots, x_n^M)^T$$

非参数估计

● 直接用样本估计总体分布

基本思路：要估计 x_i 点的密度 $p(x_i)$ ，可把所有样本在该点上的“贡献”相加近似作为其概率密度，进而得到 $\hat{p}(x)$ 。



非参数估计

● 直接用样本估计总体分布

$p(x)$ 为 x 的总体概率密度函数， N 个样本 $x = \{x_1, x_2, \dots, x_N\}$ 从密度为 $p(x)$ 的总体中独立抽取，估计 $\hat{p}(x)$ 近似 $p(x)$ 。

考虑随机向量 x 落入区域 \mathfrak{R} 的概率 $P_R = \int_{\mathfrak{R}} p(x) dx$

k 个样本落入该区域的概率符合二项分布 $P_k = C_N^k P_R^k (1 - P_R)^{N-k}$

$$E[k] = NP_R \longrightarrow \hat{P}_R \doteq \frac{k}{N}$$

设 $p(x)$ 连续，且区域体积 V 足够小，则有 $P_R = \int_{\mathfrak{R}} p(x) dx = p(x)V$

$$\hat{p}(x) = \frac{k}{NV} \quad \text{与总样本数、区域的体积及落入的样本数有关}$$

非参数估计

$$\hat{p}(x) = \frac{k}{NV} \longrightarrow \hat{p}_N(x) = \frac{k_N}{NV_N}$$

$$\lim_{N \rightarrow \infty} V_N = 0$$

$$\lim_{N \rightarrow \infty} k_N = \infty \longrightarrow \hat{p}_N(x) \text{收敛于 } p(x)$$

$$\lim_{N \rightarrow \infty} \frac{k_N}{N} = 0$$

● Parzen窗估计

使区域体积序列 V_N 以 N 的某个函数的关系不断缩小

同时限制 k_N 和 k_N/N 。

有限的 N , V_I 选择很敏感

● k_n 近邻估计

使落入区域样本数 k_N 为 N 的某个函数

V_N 使区域包含 x 的 k_N 个近邻

动态变化 V 的取值



第三章：混合模型和聚类

Chapter 3: Mixture Models and Clustering

机器学习算法

机器学习主要问题

		<i>Supervised Learning</i>	<i>Unsupervised Learning</i>
<i>Discrete</i>		Classification or Categorization	Clustering
<i>Continuous</i>		Regression	Dimensionality Reduction

聚类

- 一种**无监督**学习方法
- 把**相似**的对象通过静态分类的方法分成不同的**组别**或**子集合**
- 同一类别或子集中的对象具有相似的属性
- 在数据挖掘、模式识别、图像分析、数据分析中具有广泛的应用

聚类应用

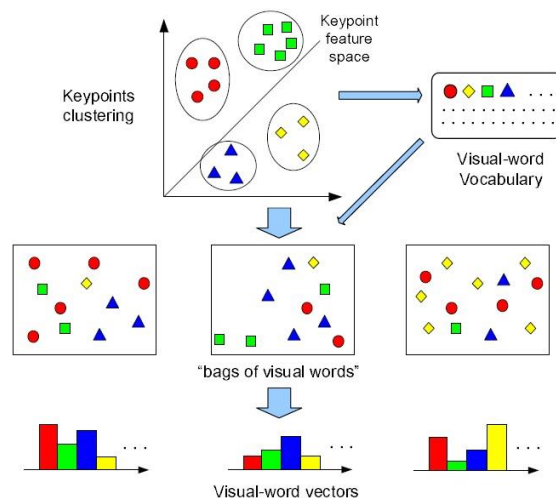
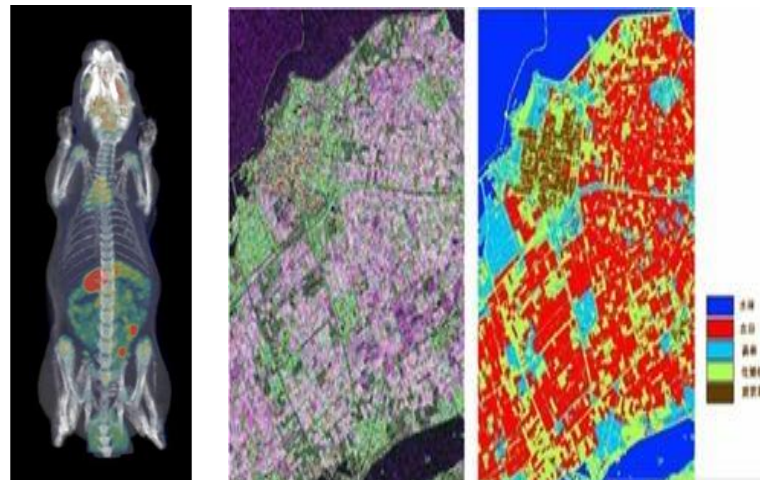
- 聚类的应用：

- 医学图像-组织分类

- 遥感图像-地貌分类

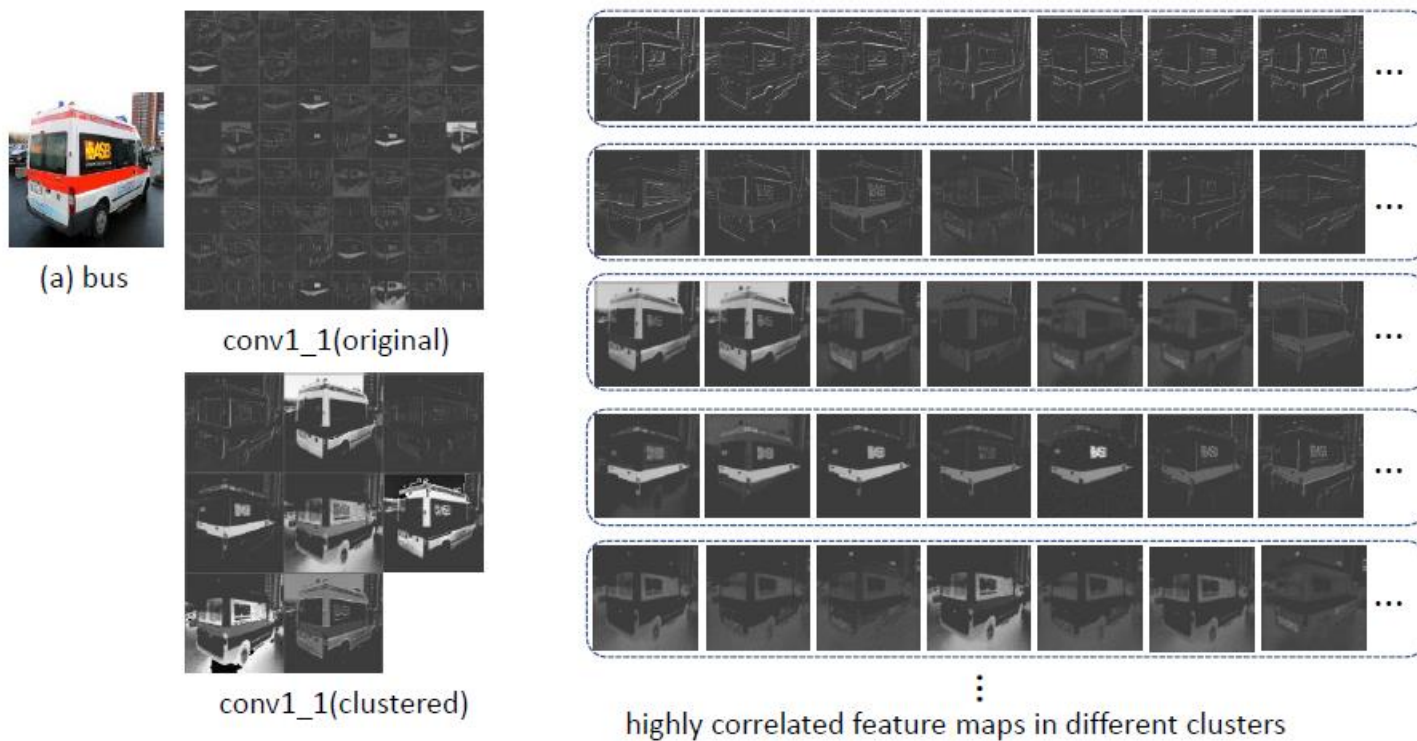
- 降维-BOW模型

- 网络用户分类



聚类应用

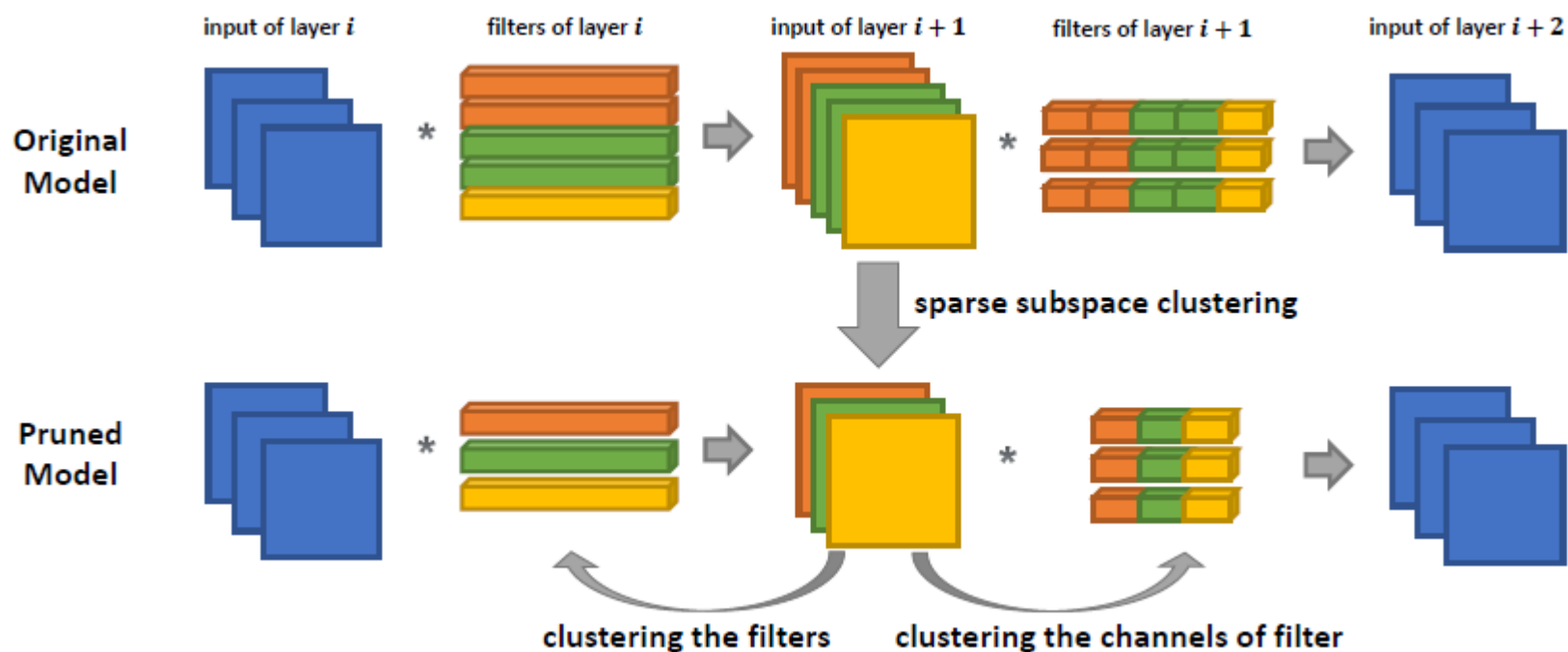
● 神经网络压缩



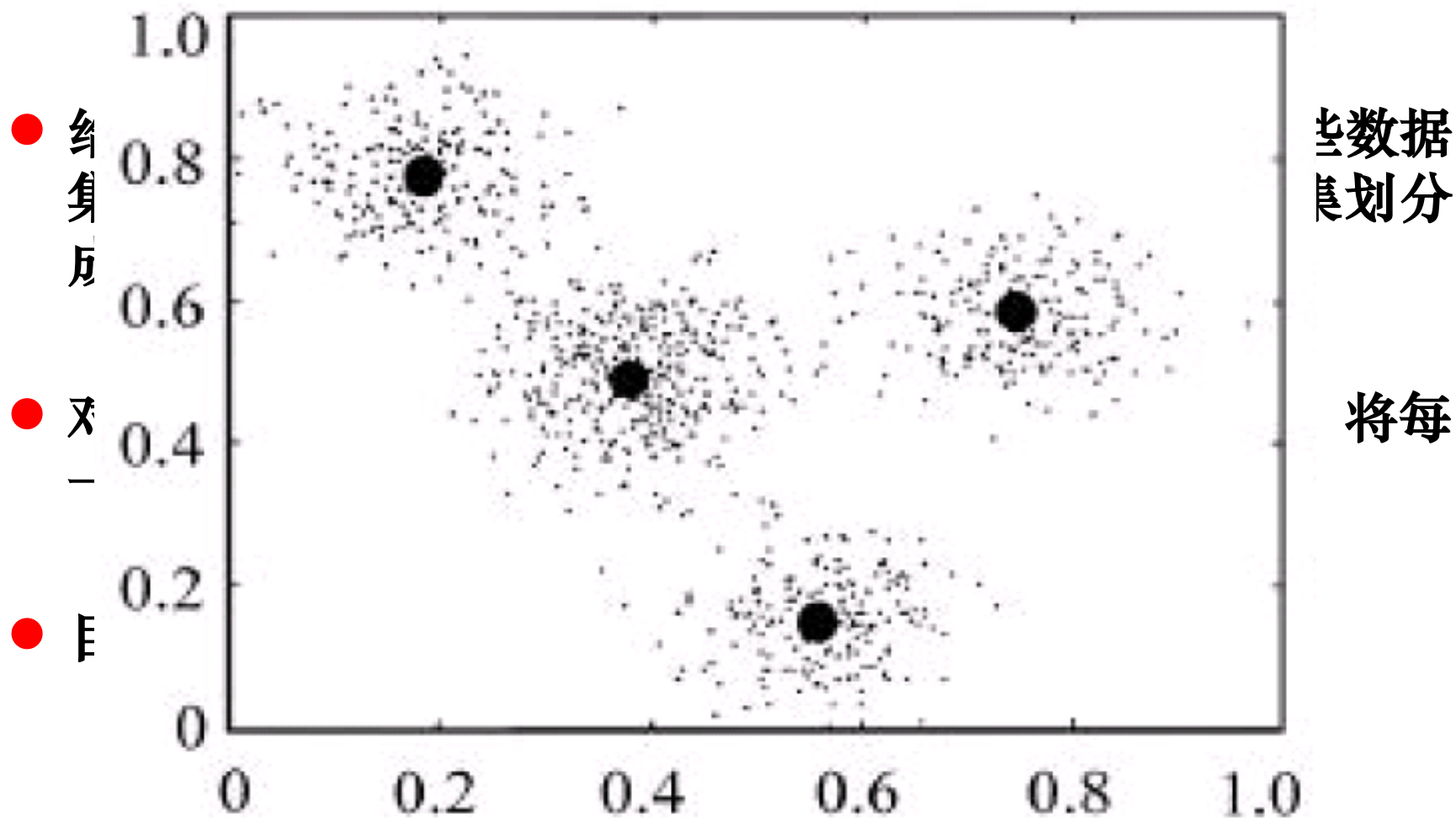
神经网络特征图有很大的冗余性

聚类应用

● 神经网络压缩

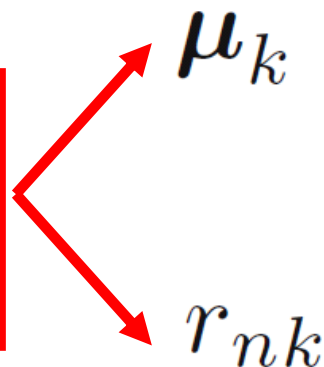


K均值算法



K均值算法

- 对于样本 \mathbf{x}_n ，定义一个聚类标注 r_n $r_n \in \mathbb{R}^K$
- 如果 \mathbf{x}_n 属于第 k 个聚类，则第 k 维为 1，即
 $r_{nk} = 1$, and $r_{nj} = 0$ for $j \neq k$
如 $r_n = [0, 0, 1, 0, 0]^T$ 表示 \mathbf{x}_n 属于 5 类中的第 3 类
- 准则函数：


$$J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \|\mathbf{x}_n - \mu_k\|^2$$


K均值算法

- 两步走策略

- 第一步：初始化 μ_k ，按照最优化准则产生 r_{nk}

$$J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \| \mathbf{x}_n - \mu_k \|^2$$


$$r_{nk} = \begin{cases} 1 & \text{if } k = \arg \min_j \| \mathbf{x}_n - \mu_j \|^2 \\ 0 & \text{otherwise} \end{cases}$$

K均值算法

- 两步走策略

- 第二步：初始化 r_{nk} ，按照最优化准则产生 μ_k

$$J = \sum_{n=1}^N \sum_{k=1}^K r_{nk} \| \mathbf{x}_n - \mu_k \|^2$$



$$2 \sum_{n=1}^N r_{nk} (\mathbf{x}_n - \mu_k) = 0$$



$$\mu_k = \frac{\sum_n r_{nk} \mathbf{x}_n}{\sum_n r_{nk}}$$

K均值算法

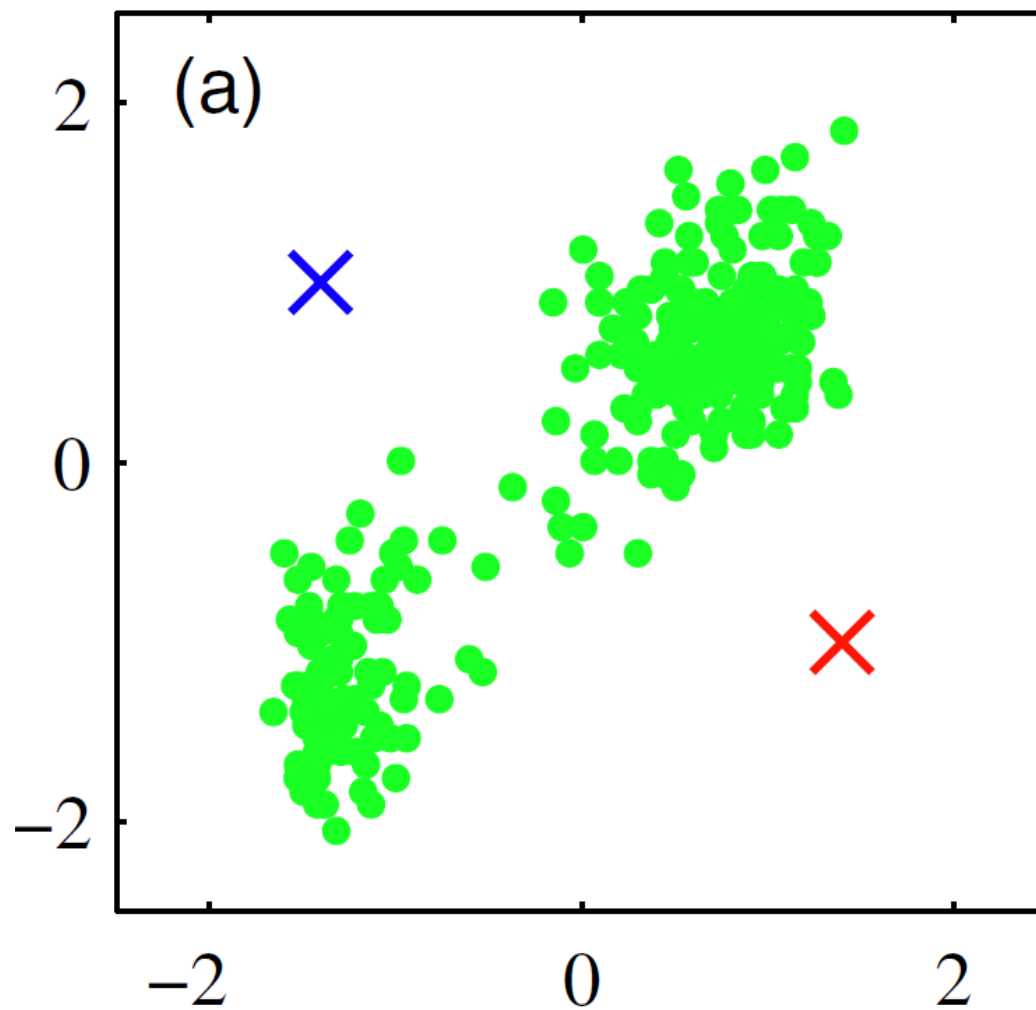
- 两步走策略

- 循环迭代：根据产生的 μ_k ，按照最优准则产生 r_{nk}

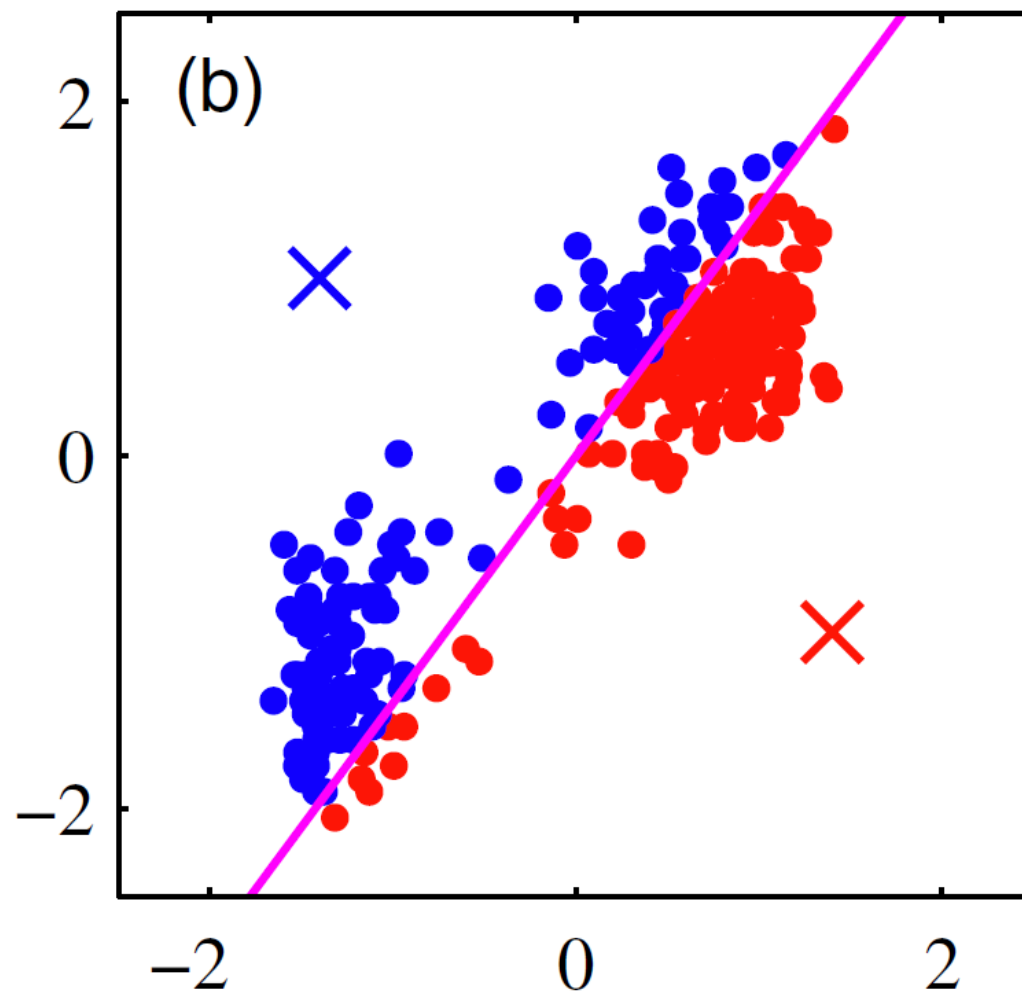
迭代 $r_{nk} \longrightarrow$ Expectation

迭代 $\mu_k \longrightarrow$ Maximization

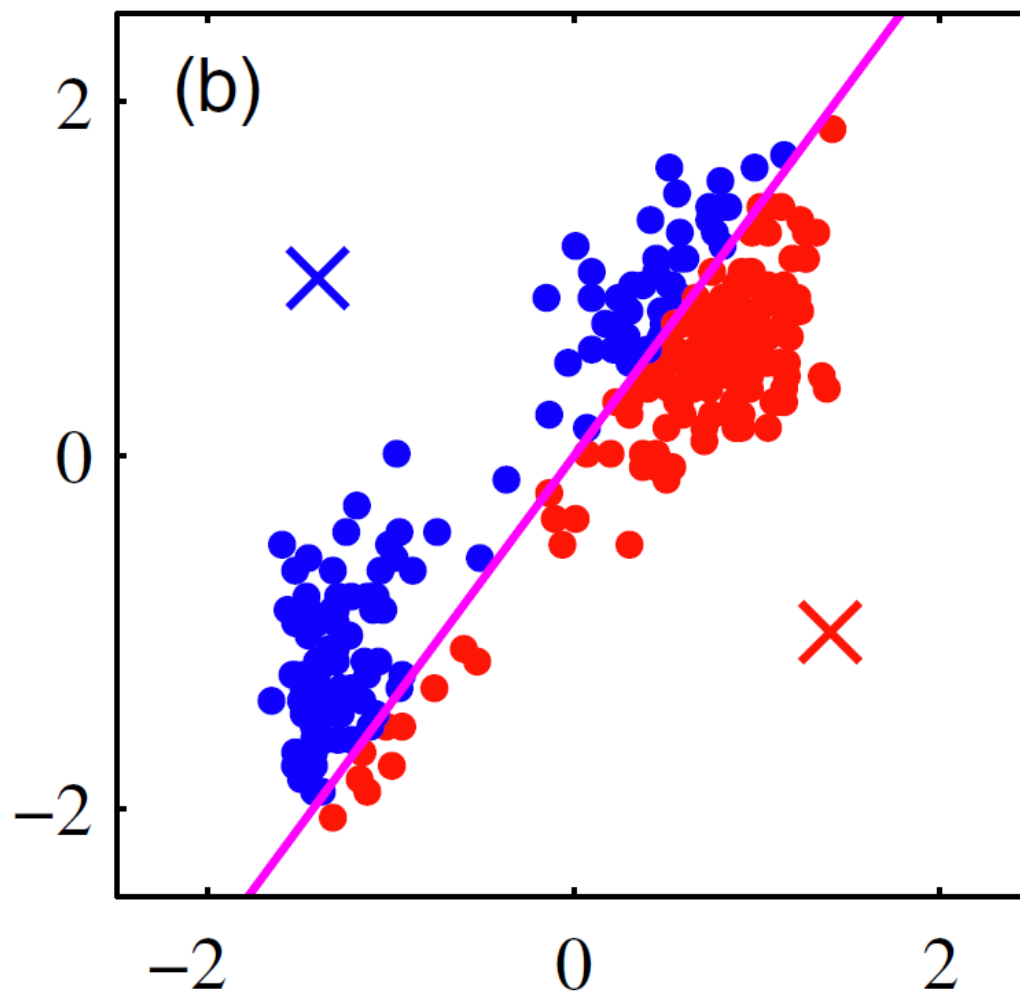
K均值过程示例



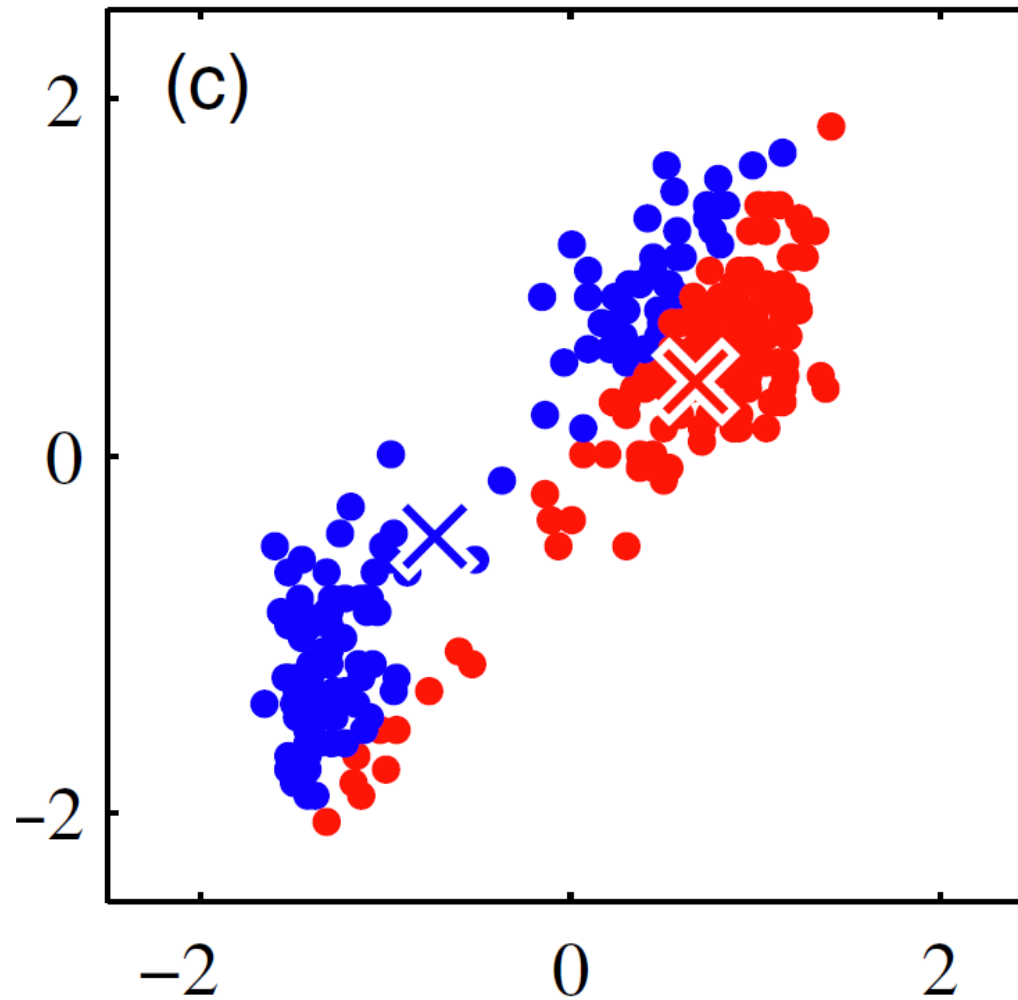
K均值过程示例



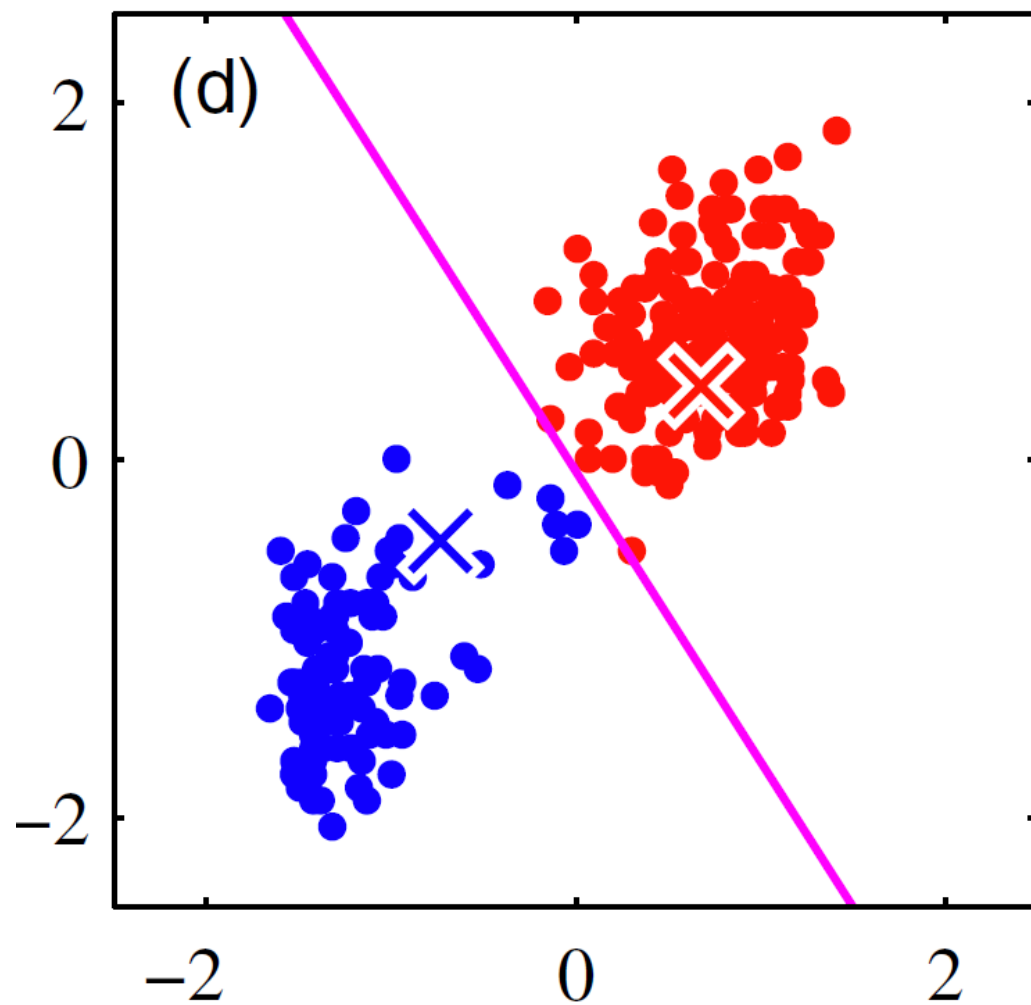
K均值过程示例



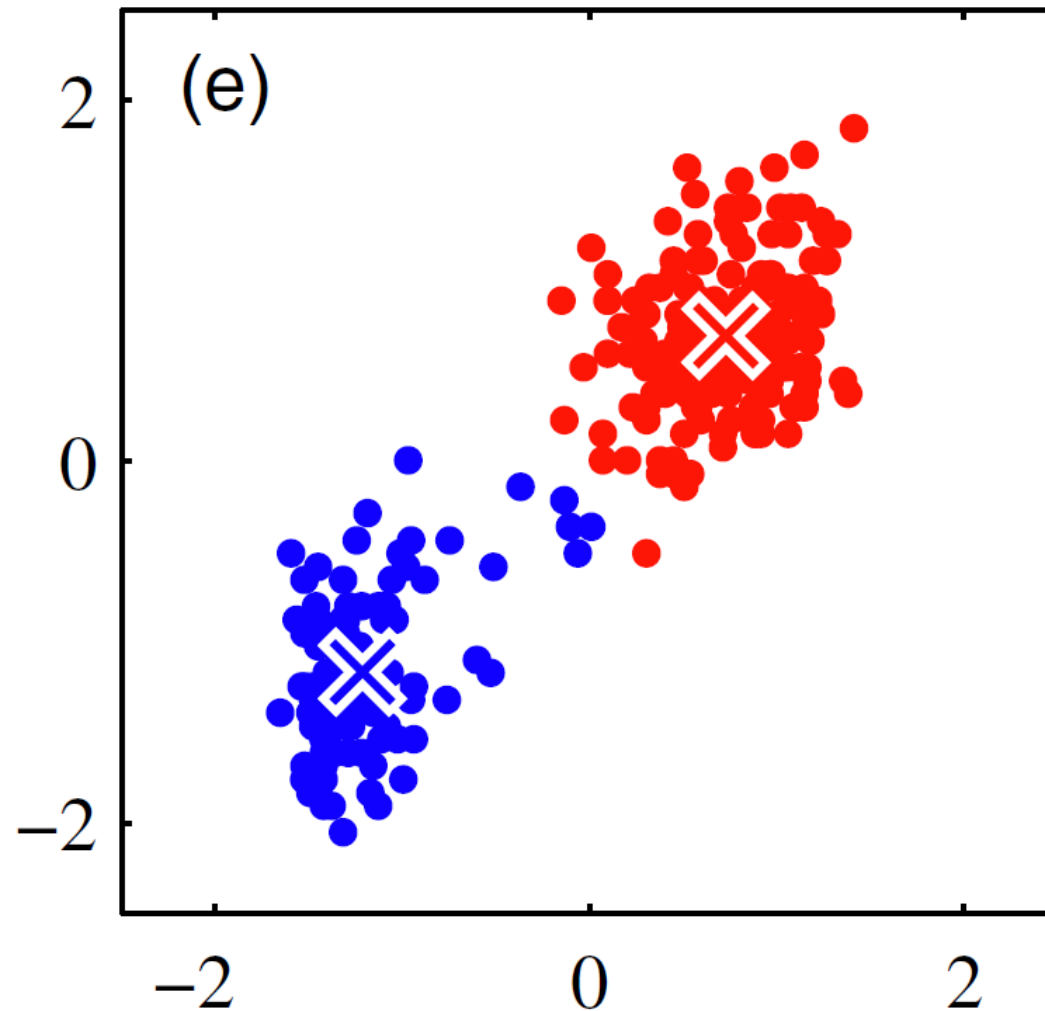
K均值过程示例



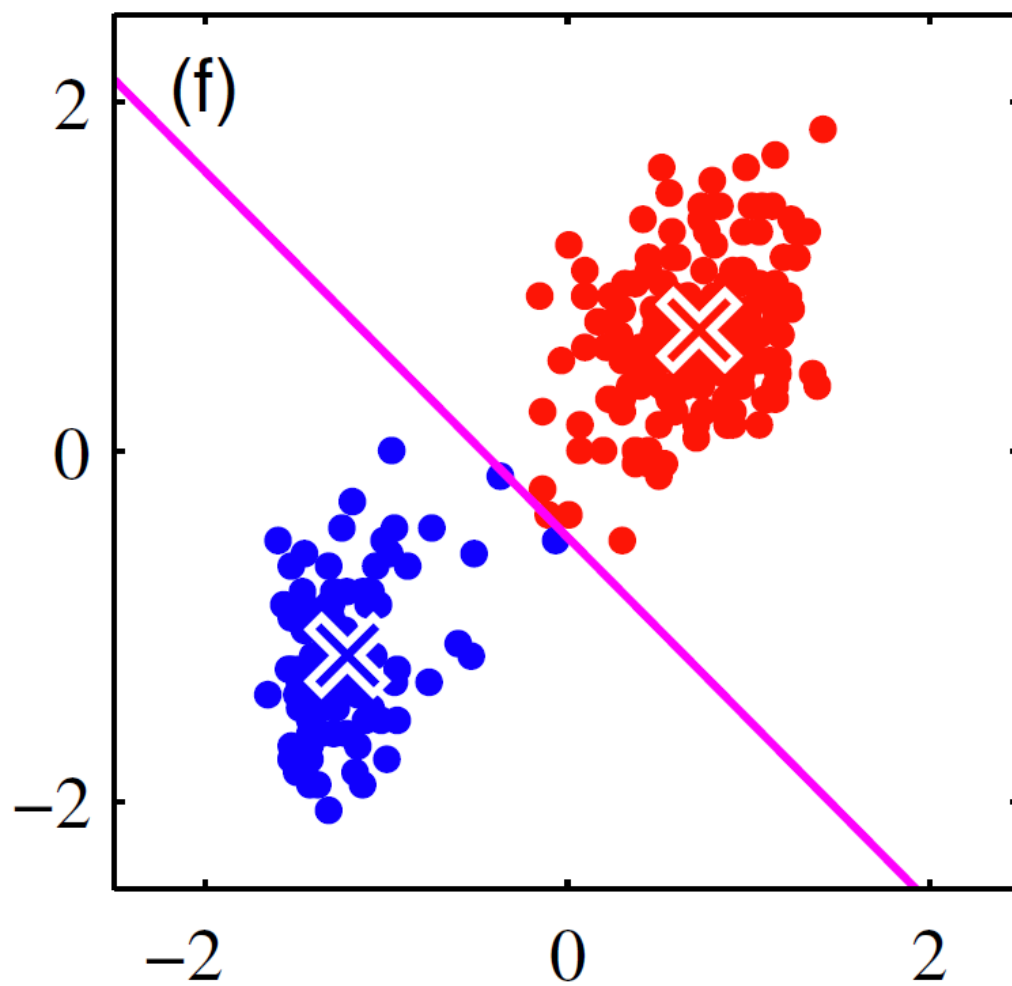
K均值过程示例



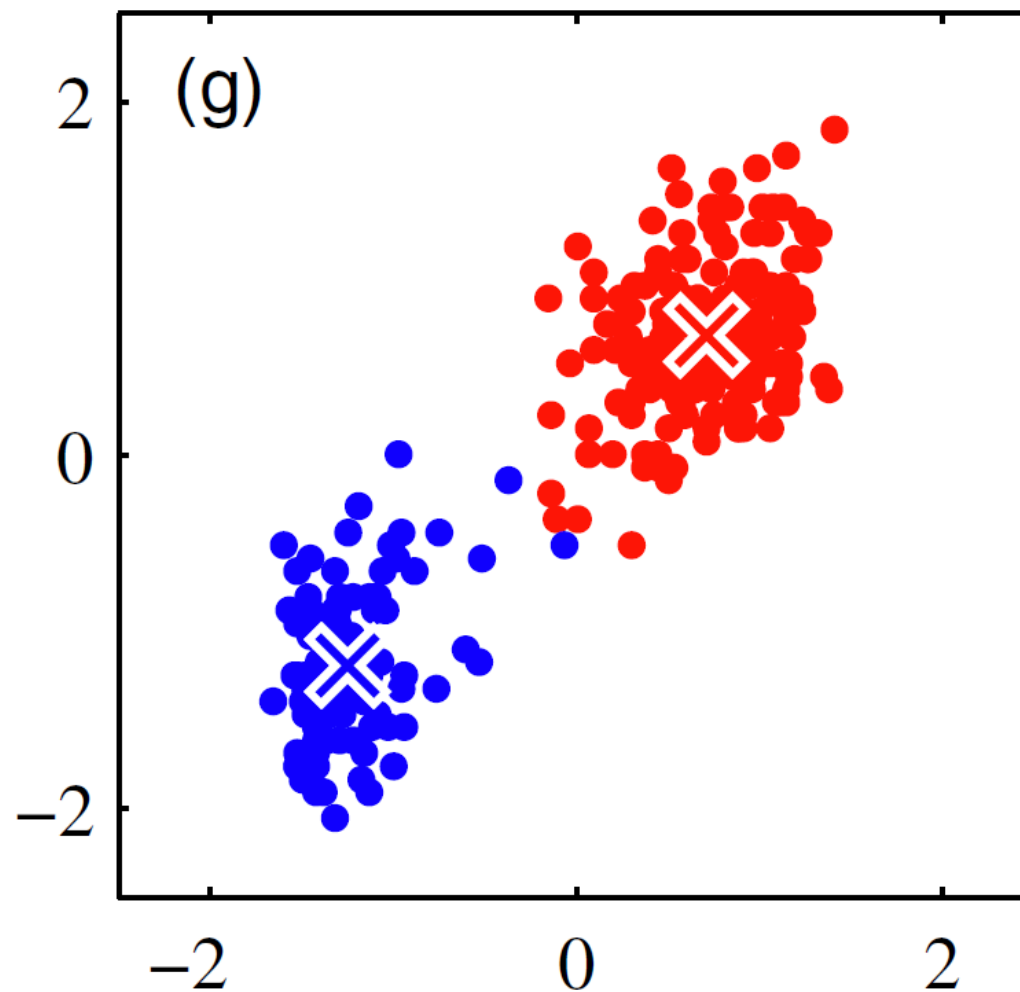
K均值过程示例



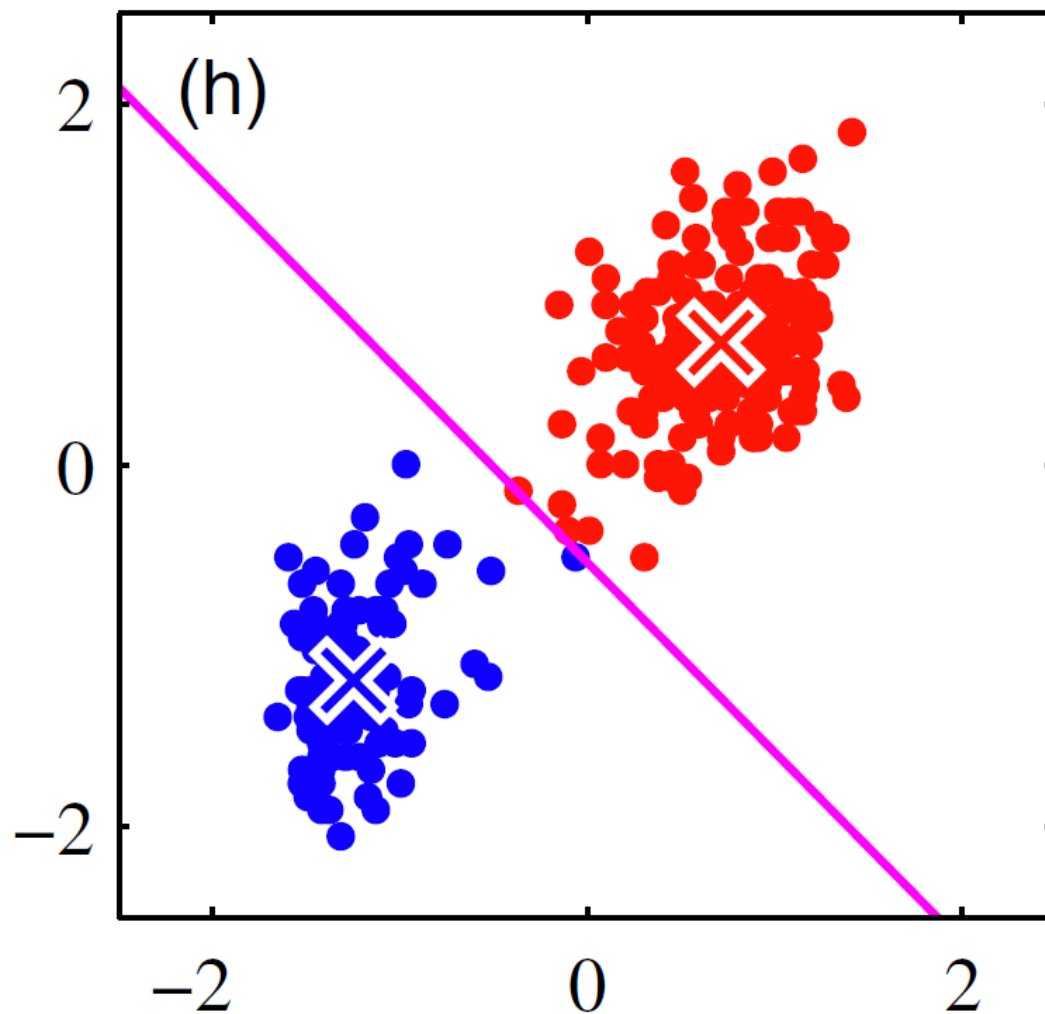
K均值过程示例



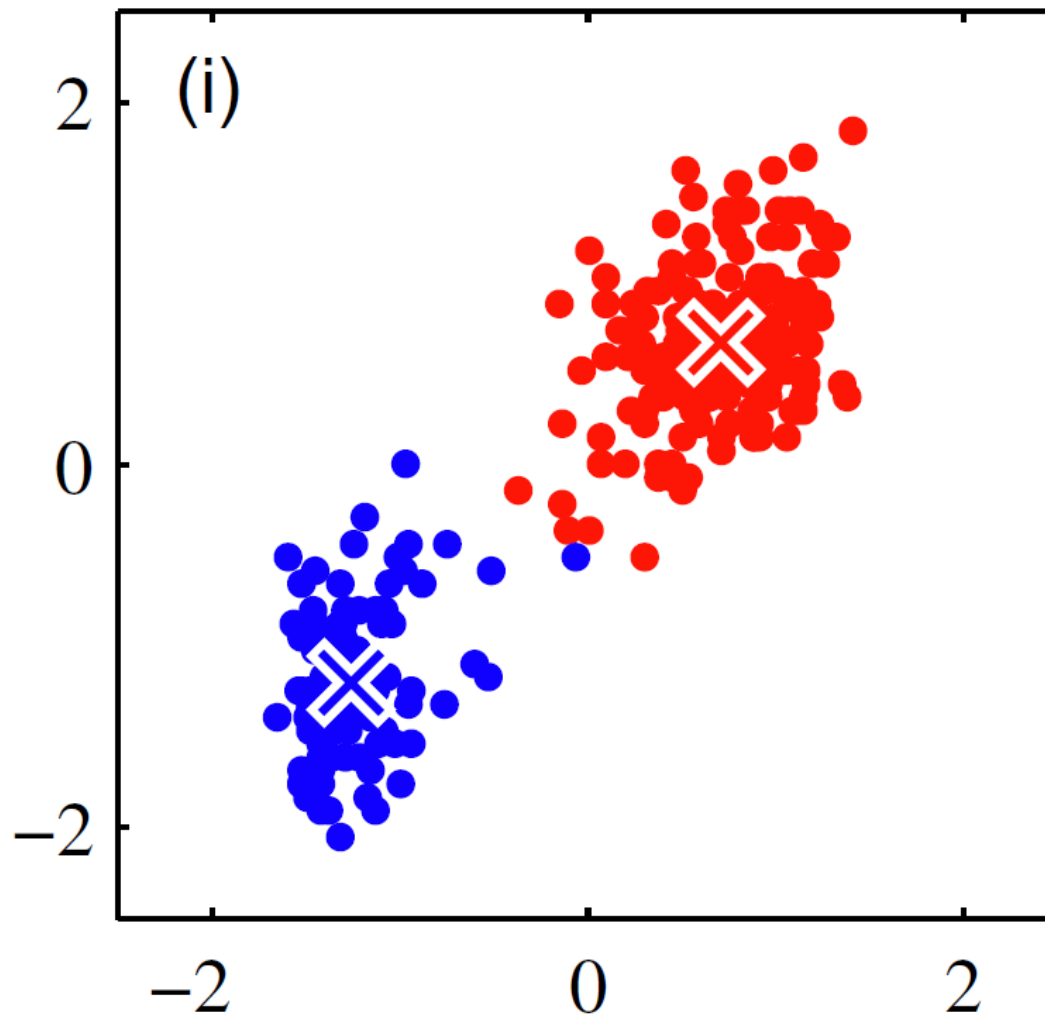
K均值过程示例



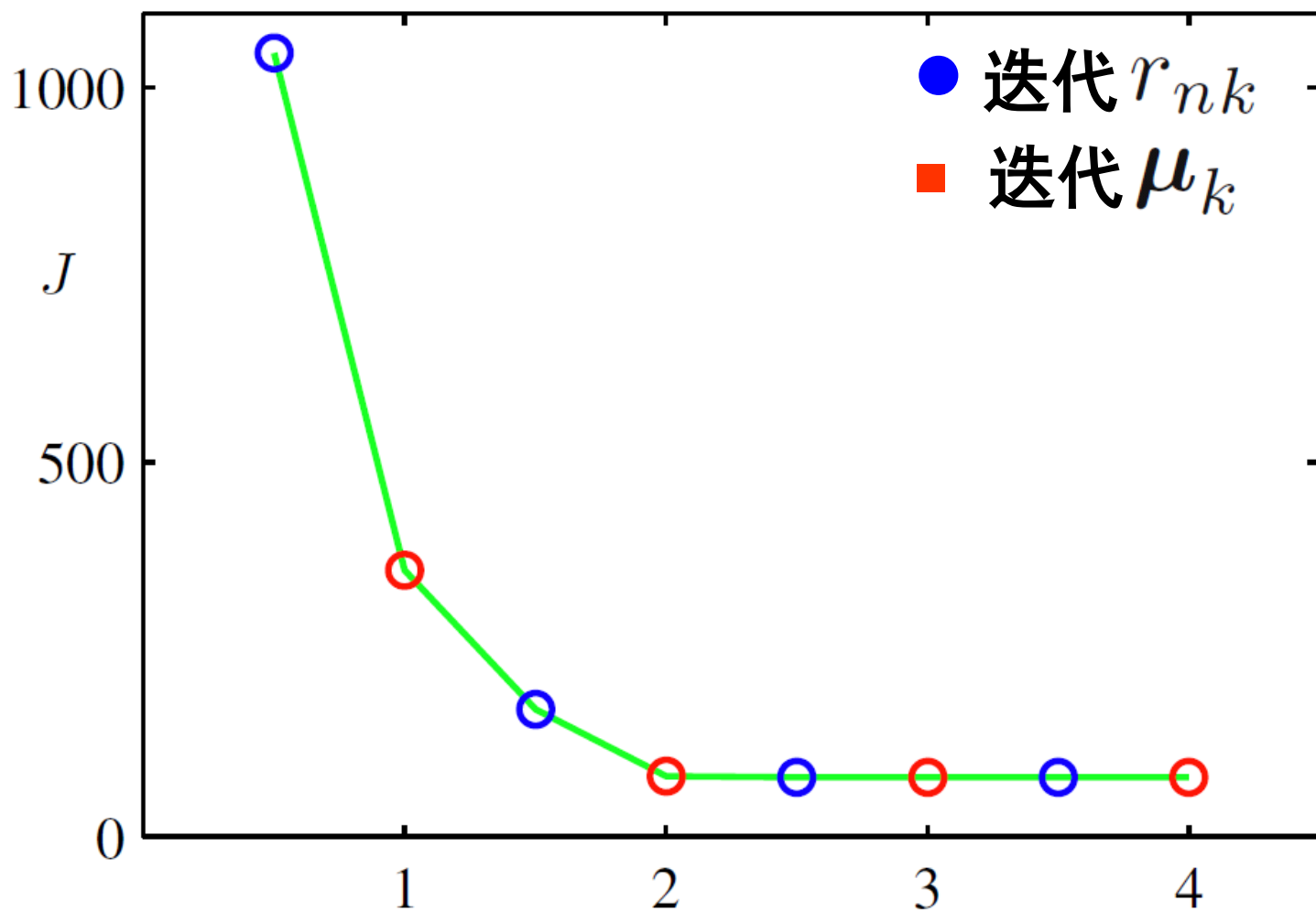
K均值过程示例



K均值过程示例



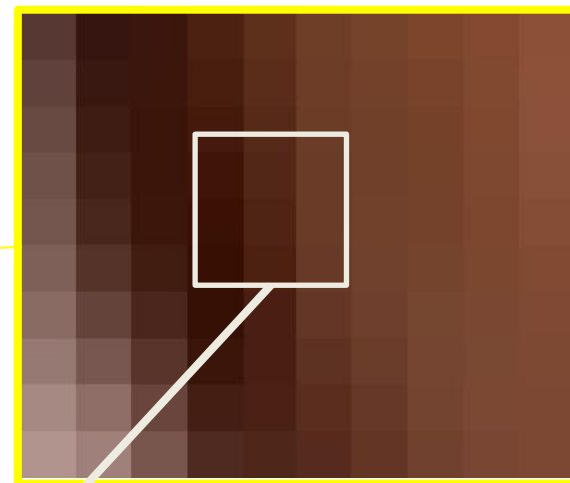
K均值过程示例



实例：基于K均值聚类的图像分割



素矩阵



	163	211		207	14	63		
	166	125		157	156	54		
	146	102		56	165	32		

实例：基于K均值聚类的图像分割

- $K=2$



实例：基于K均值聚类的图像分割

- $K=3$

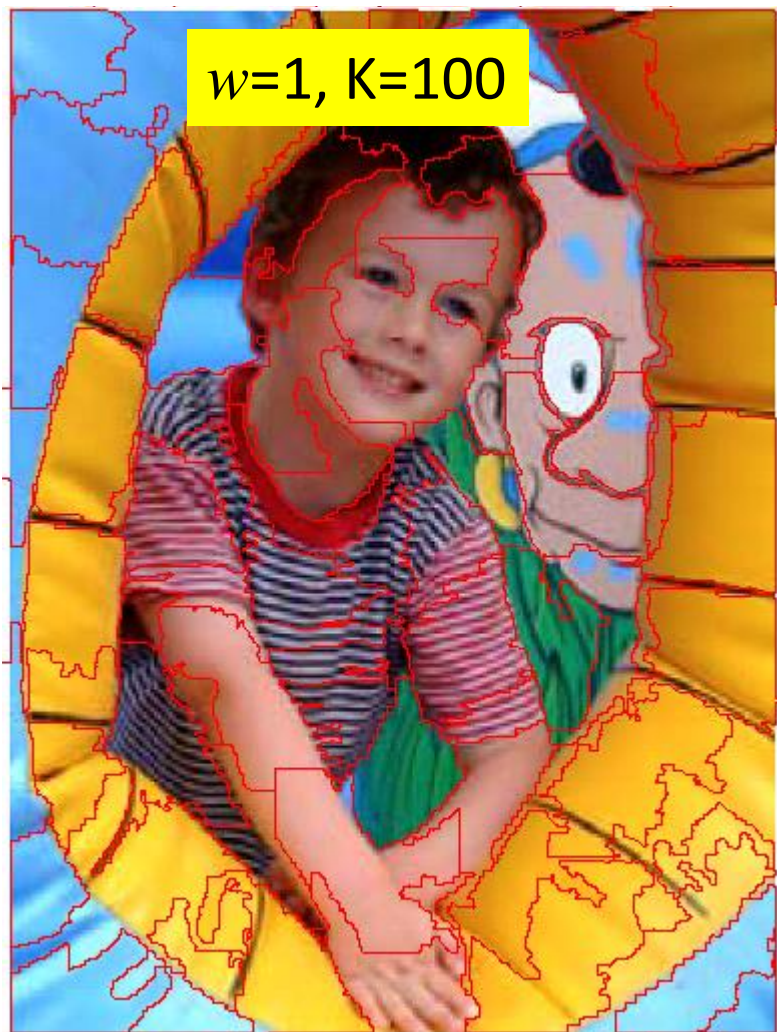


实例：基于K均值聚类的图像分割

- $K=10$



实例：SLIC分割



示信息

$r, b)$ —



高斯混合模型 (Gaussian Mixture Model)

● 一个例子

- 假设学校一个叫小明的老师，要统计学校各院系学生的身高，他从学校的档案馆随机抽取了 N （如 $N=2000$ ）名学生的档案，并从中得到了学生的身高信息。
- 问：如何从中得到各院系的学生身高分布？（假设身高服从高斯分布）以及各院系学生的人数分布。

高斯混合模型 (Gaussian Mixture Model)

➤ 学生的身高（观测值）：

$$\{x_1, x_2, \dots, x_N\} \quad p(x) \sim \sum_z p(z) p(x|z)$$

➤ 每个院系的身高分布：

$$\mathcal{N}(\mu_k, \sigma_k)$$

➤ 每个院系的人数分布（人数不一样，被抽中的概率也不一样）：

$$p(z) = \{\pi_1, \pi_2, \dots, \pi_{21}\}$$

高斯混合模型 (Gaussian Mixture Model)

- z 是一个隐变量, K 维, 且这 K 维当中只有第 k 维的值为 1, 其他维均为 0

$$z = \begin{pmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ \vdots \\ 0 \end{pmatrix} \quad \begin{aligned} p(z_k = 1) &= \pi_k \\ \text{s.t. } 0 &\leq \pi_k \leq 1 \\ \sum_{k=1}^K \pi_k &= 1 \end{aligned}$$

混合高斯模型

- X 从第 k 个高斯采样的概率是

$$p(\mathbf{x} | z_k = 1) = \mathcal{N}(\mathbf{x} | \mu_k, \Sigma_k)$$

- \mathbf{x} 和 \mathbf{z} 的联合分布是

$$p(\mathbf{x}, \mathbf{z}) = p(\mathbf{z}) \cdot p(\mathbf{x} | \mathbf{z})$$



$$p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{z}) p(\mathbf{x} | \mathbf{z}) = \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x} | \mu_k, \Sigma_k)$$

混合高斯模型

- N 个样本的联合分布

$$p(X|\pi, \mu, \Sigma) = \prod_{n=1}^N p(\mathbf{x}_n|\pi, \mu, \Sigma) \longrightarrow \text{似然函数}$$
$$= \prod_{n=1}^N \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n|\mu_k, \Sigma_k)$$

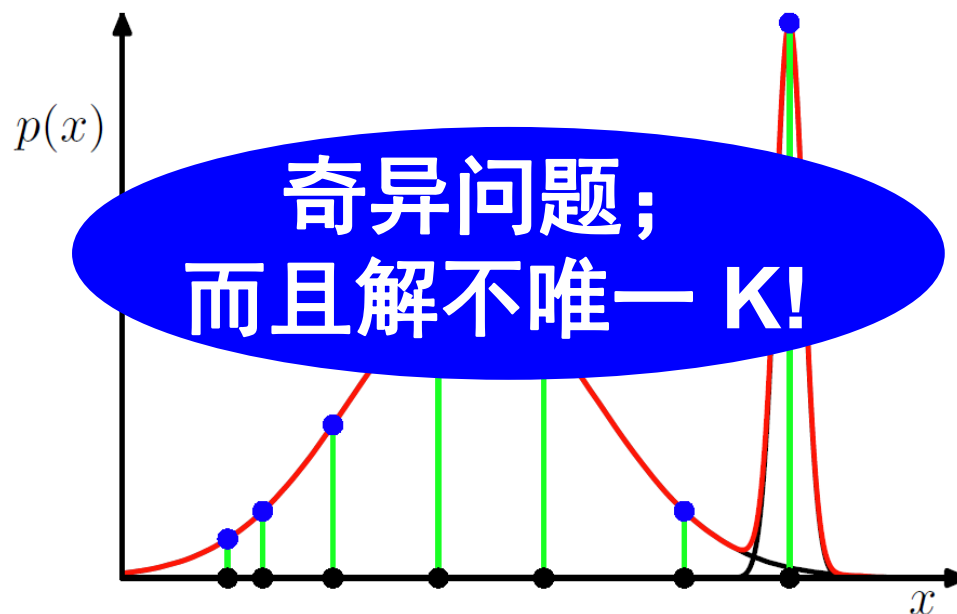
- 两边取对数

$$\ln p(X|\pi, \mu, \Sigma) = \sum_{n=1}^N \ln \left\{ \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n|\mu_k, \Sigma_k) \right\}$$

混合高斯模型的最大似然估计

- 给定 $\{\mathbf{x}_1, \dots, \mathbf{x}_N\}$, 估计混合高斯模型参数

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^N \ln \left\{ \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$




EM算法求解混合高斯问题

$$\ln p(\mathbf{X}|\pi, \mu, \Sigma) = \sum_{n=1}^N \ln \left\{ \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n | \mu_k, \Sigma_k) \right\}$$

直接求解该最大似然估计较难

但它的解一定满足以下条件：

对 μ_k 求导


$$0 = - \sum_{n=1}^N \frac{\pi_k \mathcal{N}(\mathbf{x}_n | \mu_k, \Sigma_k)}{\underbrace{\sum_j \pi_j \mathcal{N}(\mathbf{x}_n | \mu_j, \Sigma_j)}_{\gamma(z_{nk})}} \Sigma_k (\mathbf{x}_n - \mu_k)$$

\mathbf{x}_n 属于第K个高斯的概率


混合高斯模型

$$\begin{aligned}\gamma(z_k) &\equiv p(z_k = 1 | \mathbf{x}) \\ &= \frac{p(z_k = 1)p(\mathbf{x} | z_k = 1)}{\sum_{j=1}^K p(z_j = 1)p(\mathbf{x} | z_j = 1)} \\ &= \frac{\pi_k \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x} | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}\end{aligned}$$

EM算法求解混合高斯问题

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^N \ln \left\{ \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

对 μ_k 求导

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) \mathbf{x}_n \quad N_k = \sum_{n=1}^N \gamma(z_{nk})$$


第 k 个高斯所包含的样本数

EM算法求解混合高斯问题

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \sum_{n=1}^N \ln \left\{ \sum_{k=1}^K \pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \right\}$$

对 $\boldsymbol{\Sigma}_k$ 求导

对 π_k 求导

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T$$

必须考虑一个约束条件

$$\sum_{k=1}^K \pi_k = 1$$

$$\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) + \lambda \left(\sum_{k=1}^K \pi_k - 1 \right) \rightarrow 0 = \sum_{n=1}^N \frac{\mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_j \pi_j \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} + \lambda$$

$$\rightarrow \lambda = -N$$

$$N_k = \sum_{n=1}^N \frac{\pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_j \pi_j \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} \rightarrow \pi_k = \frac{N_k}{N}$$

EM算法求解混合高斯问题

$$\boldsymbol{\mu}_k = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) \mathbf{x}_n \quad N_k = \sum_{n=1}^N \gamma(z_{nk})$$

$$\boldsymbol{\Sigma}_k = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) (\mathbf{x}_n - \boldsymbol{\mu}_k)(\mathbf{x}_n - \boldsymbol{\mu}_k)^T$$

$$\pi_k = \frac{N_k}{N}$$

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_j \pi_j \mathcal{N}(\mathbf{x}_n | \boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

EM算法求解混合高斯问题

EM for Gaussian Mixtures

Given a Gaussian mixture model, the goal is to maximize the likelihood function with respect to the parameters (comprising the means and covariances of the components and the mixing coefficients).

1. Initialize the means μ_k , covariances Σ_k and mixing coefficients π_k , and evaluate the initial value of the log likelihood.
2. **E step.** Evaluate the responsibilities using the current parameter values

$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n | \mu_k, \Sigma_k)}{\sum_{j=1}^K \pi_j \mathcal{N}(\mathbf{x}_n | \mu_j, \Sigma_j)}. \quad (9.23)$$

EM算法求解混合高斯问题

3. **M step.** Re-estimate the parameters using the current responsibilities

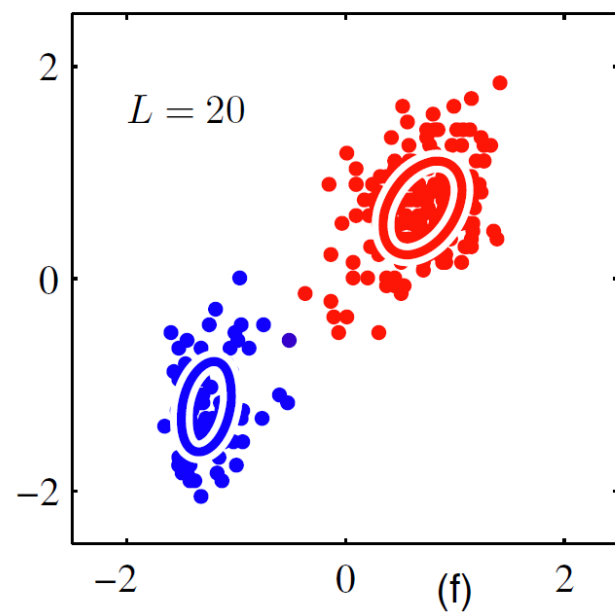
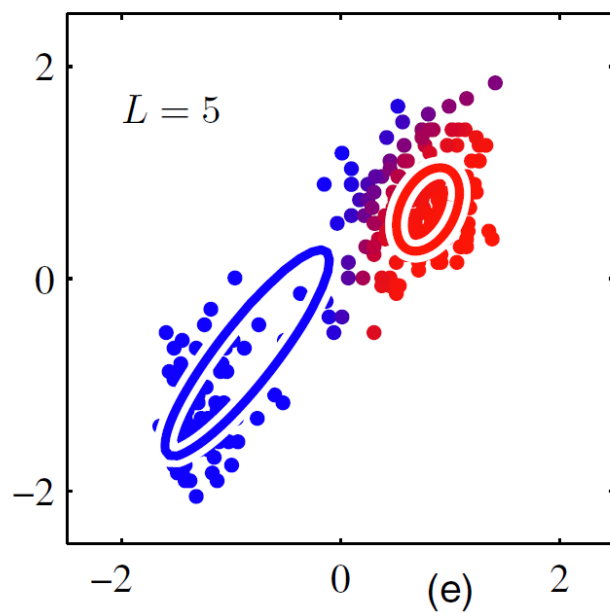
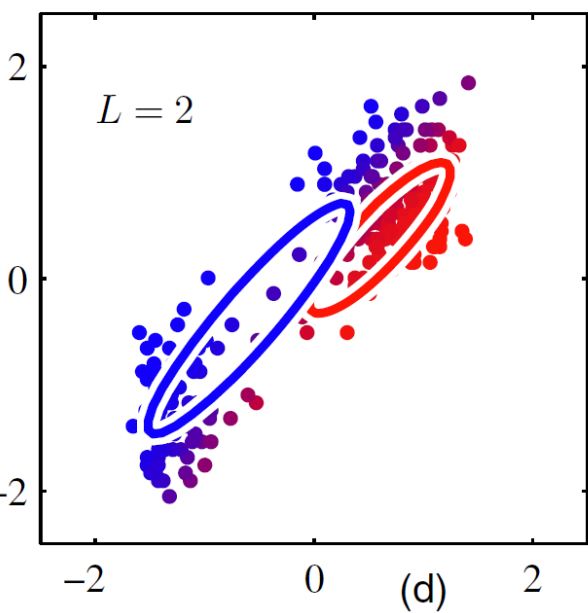
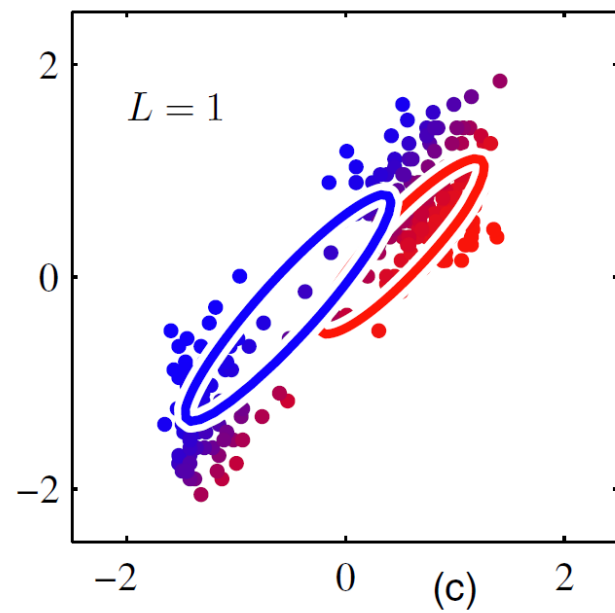
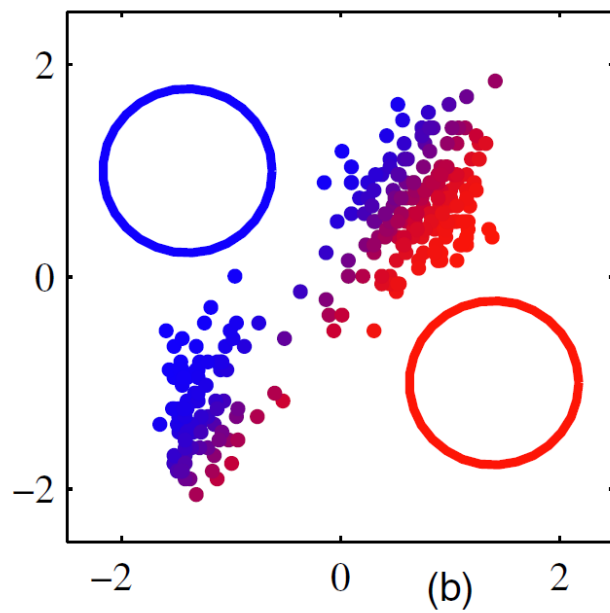
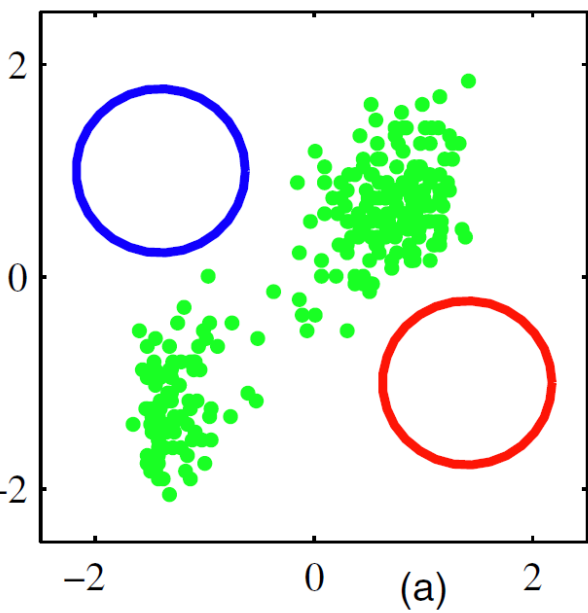
$$\mu_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) \mathbf{x}_n \quad (9.24)$$

$$\Sigma_k^{\text{new}} = \frac{1}{N_k} \sum_{n=1}^N \gamma(z_{nk}) (\mathbf{x}_n - \mu_k^{\text{new}}) (\mathbf{x}_n - \mu_k^{\text{new}})^T \quad (9.25)$$

$$\pi_k^{\text{new}} = \frac{N_k}{N} \quad (9.26)$$

where

$$N_k = \sum_{n=1}^N \gamma(z_{nk}). \quad (9.27)$$



EM算法

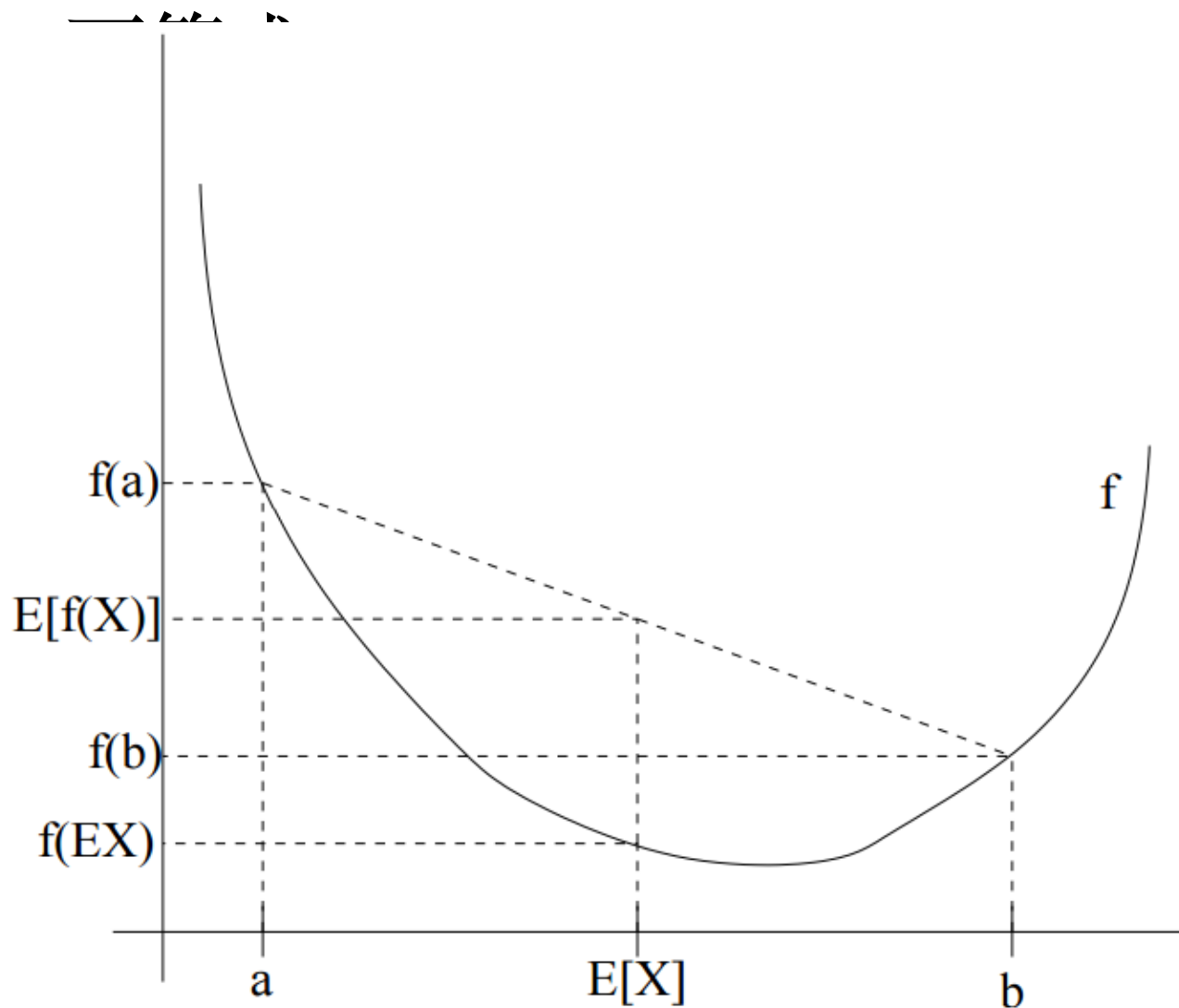
● Jensen

不等式

成立，则

当且

成立



EM算法

- Jensen不等式

如果一个函数 $f(x)$ 为凹函数，即 $f''(x) \leq 0$

$$E[f(x)] \leq f(E[x])$$

当且仅当 $x = E[x]$ 即 x 为常量时，等号成立

EM算法

- 假设有一组数据 $X = \{x_1, \dots, x_N\}$ ，经隐变量 Z 产生，该数据的对数似然函数可以写作：

$$\ell(\theta) = \ln p(X; \theta) = \ln \sum_Z p(X, Z | \theta)$$

$$= \ln \sum_Z p(X|Z, \theta) p(Z|\theta)$$



式中含有和（或积分）的对数，优化困难

EM算法

- 要最大化 $\ell(\theta)$ ，可以考虑利用迭代的方式，每一个新的 θ 能使 $\ell(\theta)$ 变大，即 $\ell(\theta) > \ell(\theta^{(i)})$ 为此，考虑两者的差：

$$\ell(\theta) - \ell(\theta^{(i)}) = \ln \sum_Z p(X|Z, \theta)p(Z|\theta) - \ln p(X; \theta^{(i)})$$

EM算法

● 利用Jensen不等式，得其下界：

$$\begin{aligned}\ell(\theta) - \ell(\theta^{(i)}) &= \ln \sum_Z p(X|Z, \theta)p(Z|\theta) - \ln p(X|\theta^{(i)}) \\&= \ln \left(\sum_Z p(X|Z, \theta^{(i)}) \frac{p(X|Z, \theta)p(Z|\theta)}{p(X|Z, \theta^{(i)})} \right) - \ln p(X|\theta^{(i)}) \\&\geq \sum_Z p(Z|X, \theta^{(i)}) \ln \frac{p(X|Z, \theta)p(Z|\theta)}{p(Z|X, \theta^{(i)})} - \ln p(X|\theta^{(i)}) \\&= \sum_Z p(Z|X, \theta^{(i)}) \ln \frac{p(X|Z, \theta)p(Z|\theta)}{p(Z|X, \theta^{(i)})p(X|\theta^{(i)})}\end{aligned}$$

EM算法

● 令

$$B(\theta, \theta^{(i)}) = \ell(\theta^{(i)}) + \sum_Z p(Z|X, \theta^{(i)}) \ln \frac{p(X|Z, \theta)p(Z|\theta)}{p(Z|X, \theta^{(i)})p(X|\theta^{(i)})}$$

则：

$$\ell(\theta) \geq B(\theta, \theta^{(i)})$$

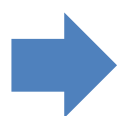
$B(\theta, \theta^{(i)})$ 是 $\ell(\theta)$ 的一个下界

因此，使 $B(\theta, \theta^{(i)})$ 增大的 θ 也可以使 $\ell(\theta)$ 增大

EM算法

- 为了使 $\ell(\theta)$ 尽可能增大，求使 $B(\theta, \theta^{(i)})$ 极大的 θ

$$\theta^{(i+1)} = \arg \max_{\theta} B(\theta, \theta^{(i)})$$


$$\theta^{(i+1)} = \arg \max_{\theta} \left(\ell(\theta^{(i)}) + \sum_Z p(Z|X, \theta^{(i)}) \ln \frac{p(X|Z, \theta)p(Z|\theta)}{p(Z|X, \theta^{(i)})p(X|\theta^{(i)})} \right)$$

$$= \arg \max_{\theta} \left(\sum_Z p(Z|X, \theta^{(i)}) \ln(p(X|Z, \theta)p(Z|\theta)) \right)$$

$$= \arg \max_{\theta} \left(\sum_Z p(Z|X, \theta^{(i)}) \ln p(X, Z|\theta) \right)$$

$$= \arg \max_{\theta} Q(\theta, \theta^{(i)}) \longrightarrow \text{EM算法的一次迭代，即求Q函数及其极大化}$$

EM算法

- 有一组数据 $\{x_1, \dots, x_N\}$ ，经隐变量 z 产生，该数据的对数似然函数可以写作：

$$\begin{aligned}\ell(\theta) &= \sum_{i=1}^N \ln p(x_i; \theta) \\ &= \sum_{i=1}^N \ln \sum_k p(x_i, z_k; \theta)\end{aligned}$$

- 此处 z 为无法直接观测的隐藏变量

EM算法

- 定义 z 的分布为 Q

$$\sum_i \ln p(\mathbf{x}_i; \theta) = \sum_i \ln \sum_k p(\mathbf{x}_i, \mathbf{z}_k; \theta)$$

$$= \sum_i \ln \sum_k Q(\mathbf{z}_k) \frac{p(\mathbf{x}_i, \mathbf{z}_k; \theta)}{Q(\mathbf{z}_k)}$$

Jensen不等式



$$\geq \sum_i \sum_k Q(\mathbf{z}_k) \ln \frac{p(\mathbf{x}_i, \mathbf{z}_k; \theta)}{Q(\mathbf{z}_k)}$$



$\ell(\theta)$ 的下界函数

EM算法

- 根据Jensen不等式要求，当

$$\frac{p(x, z; \theta)}{Q(z)} = c$$

等式成立

移项得到： $p(x, z; \theta) = c \cdot Q(z)$

$$\sum_z p(x, z; \theta) = c \cdot \sum_z Q(z)$$

$$\sum_z p(x, z; \theta) = c$$

EM算法

$$\begin{aligned} Q(z) &= \frac{p(x, z; \theta)}{c} \\ &= \frac{p(x, z; \theta)}{\sum_z p(x, z; \theta)} \\ &= \frac{p(x, z; \theta)}{p(x; \theta)} \end{aligned}$$

z的后验概率分布

$$= p(z|x; \theta)$$

EM算法

Repeat until convergence {

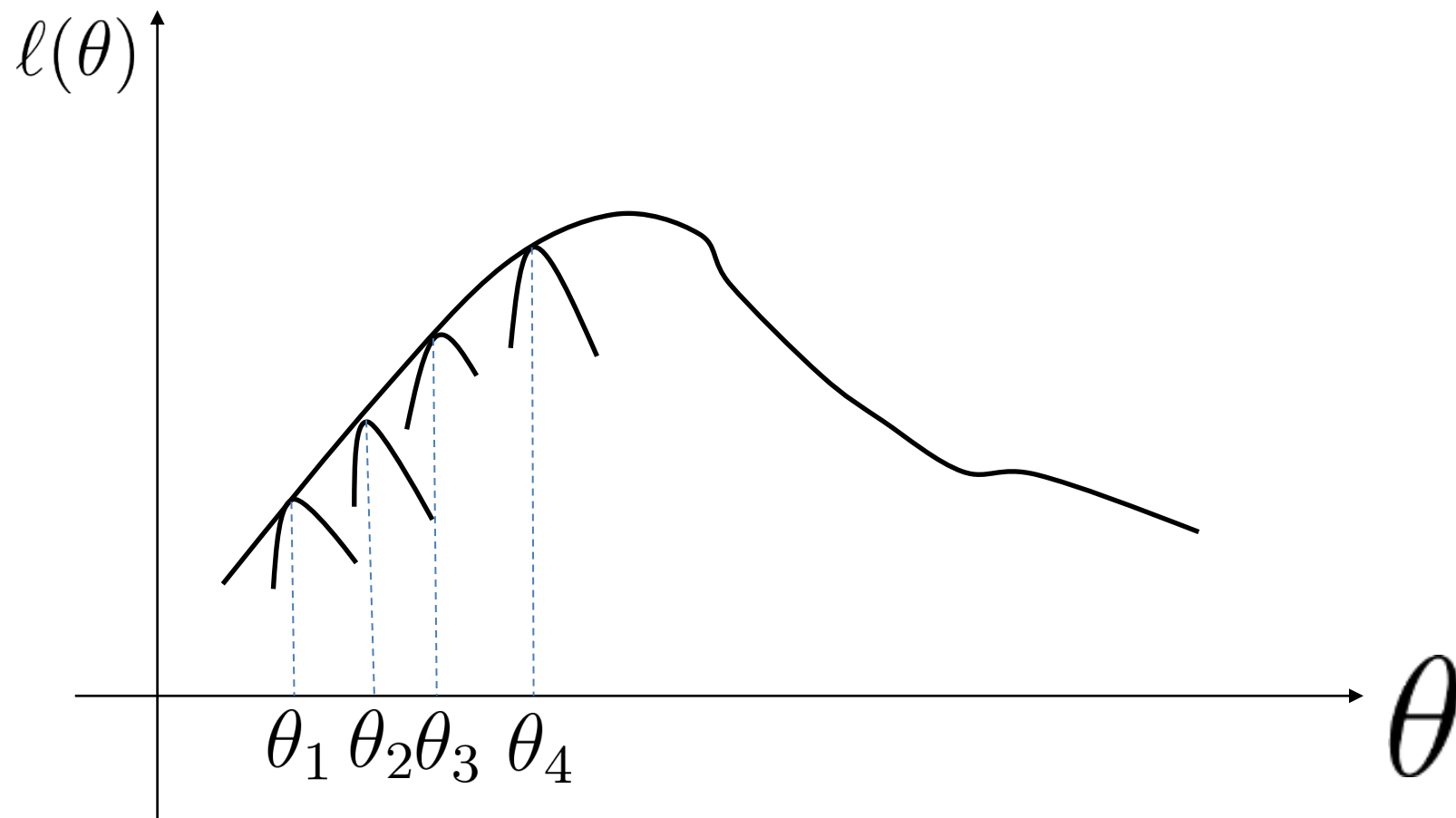
(E-step) For each i , set

$$Q(z) := p(z|x_i; \theta).$$

(M-step) Set

$$\theta := \arg \max_{\theta} \sum_i \sum_k Q(z_k) \ln \frac{p(x_i, z_k; \theta)}{Q(z_k)}$$

}



收敛性证明

- 首先证明 $\ell(\theta^{(t)}) \leq \ell(\theta^{(t+1)})$

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

Jensen不等式中等号成立，即

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

此时 $\theta^{(t+1)}$ 通过最大化上式求得

收敛性证明

- 因此有

$$\ell(\theta^{(t+1)}) \geq \sum_i \sum_{z^{(i)}} Q_i^{(t)}(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta^{(t+1)})}{Q_i^{(t)}(z^{(i)})} \quad (4)$$

似然函数 $\ell(\theta)$ 存在极值，且EM步骤单调收敛，因此必能找到极值

(4)式来自： $\ell(\theta) \geq \sum_i \sum_{z^{(i)}} Q_i(z^{(i)}) \log \frac{p(x^{(i)}, z^{(i)}; \theta)}{Q_i(z^{(i)})}$

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

实例：背景建模



Frame #1



Frame #100



Frame #200



Frame #236



Frame #247



Frame #261



Frame #280



Frame #296

● 如何检测到视频中出现的人？

实例：背景建模

- 针对每一个像素 $I(x, y)$ ，建立一个混合高斯模型

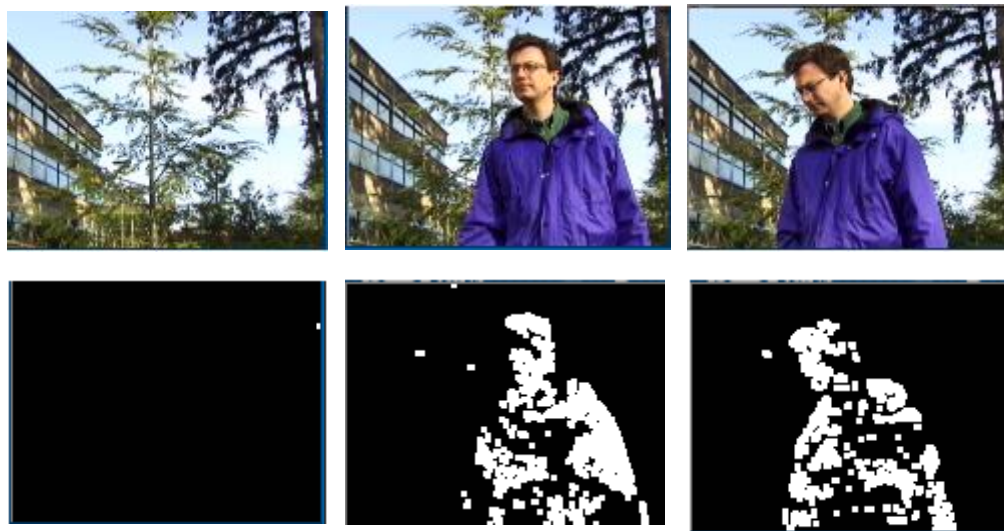
$$I_t(x, y) \sim p(x_t, y_t) = \sum_{k=1}^K \pi_k \cdot \mathcal{N}(x_t, y_t; \mu_k, \Sigma_k), \quad \sum_{k=1}^K \pi_k = 1$$

- π_k 为第 k 个高斯的权重， μ_k ， Σ_k 分别为第 k 个高斯的均值向量和协方差矩阵
- 用前 N 帧图像进行训练，得到GMM的参数

$$\hat{\pi}_{x,y,k}, \quad \hat{\mu}_{x,y,k}, \quad \hat{\Sigma}_{x,y,k}$$

- 如果第 $N+i$ 帧图像中某像素不服从GMM，则该像素为前景

实例：背景建模



- 对阴影、光照敏感
- 无法更新权重