AI300学院

机器学习之无监督学习

聚类算法-GMM

倪冰冰
上海交通大学

课程脉络

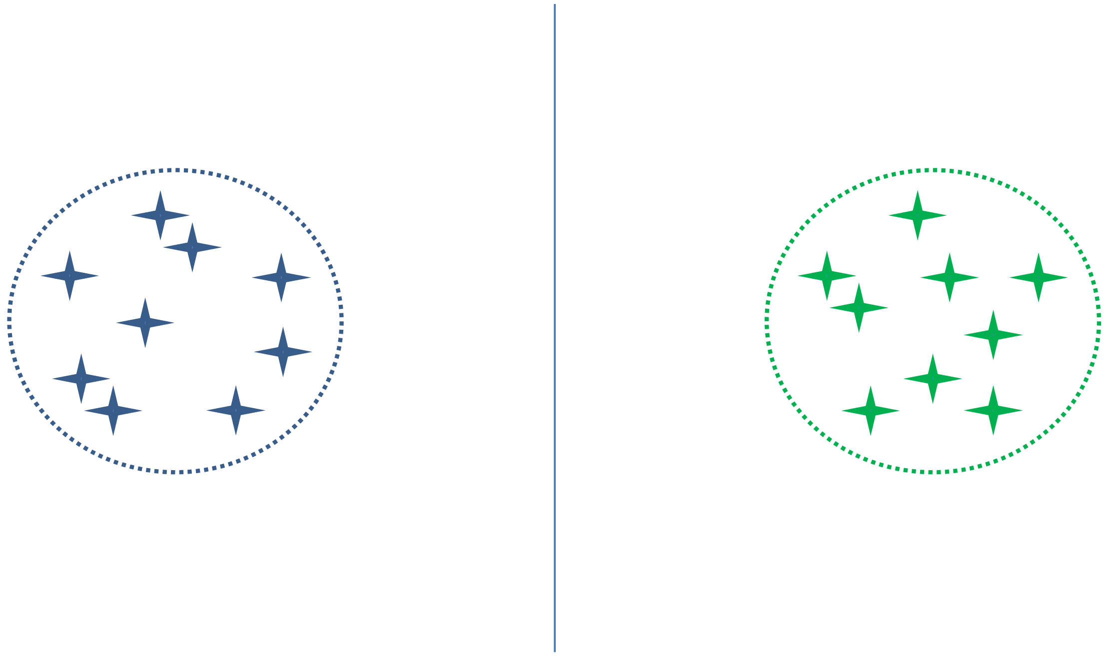
聚类分析

- 聚合聚类(Agglomerative Clustering)
- K-均值聚类(K-Means)
- 层次化聚类(H-KMeans)
- 高斯混合模型（GMM）
- Expectation-Maximization
- 谱聚类(Spectral Methods)

K-Means

- K-means算法的缺陷

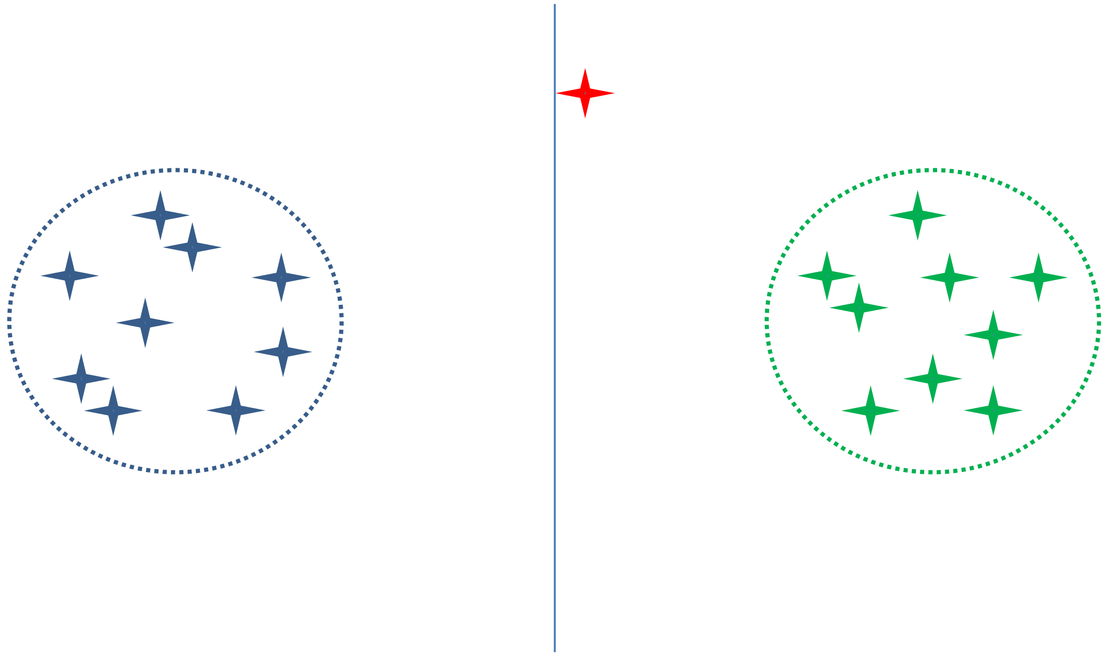
  因为对所有数据进行明确的分类，因此如果样本数据发生很小的扰动，那么样本的分类结果容易发生明显的改变。(*GMM*)



K-Means

- K-means算法的缺陷

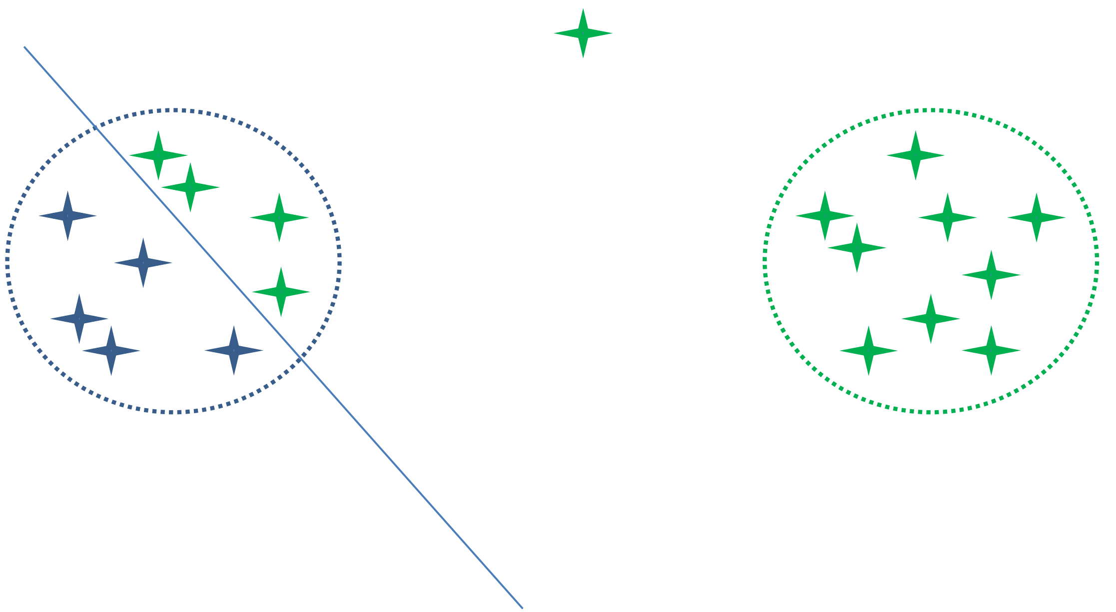
  因为对所有数据进行明确的分类，因此如果样本数据发生很小的扰动，那么样本的分类结果容易发生明显的改变。(*GMM*)



K-Means

- K-means算法的缺陷

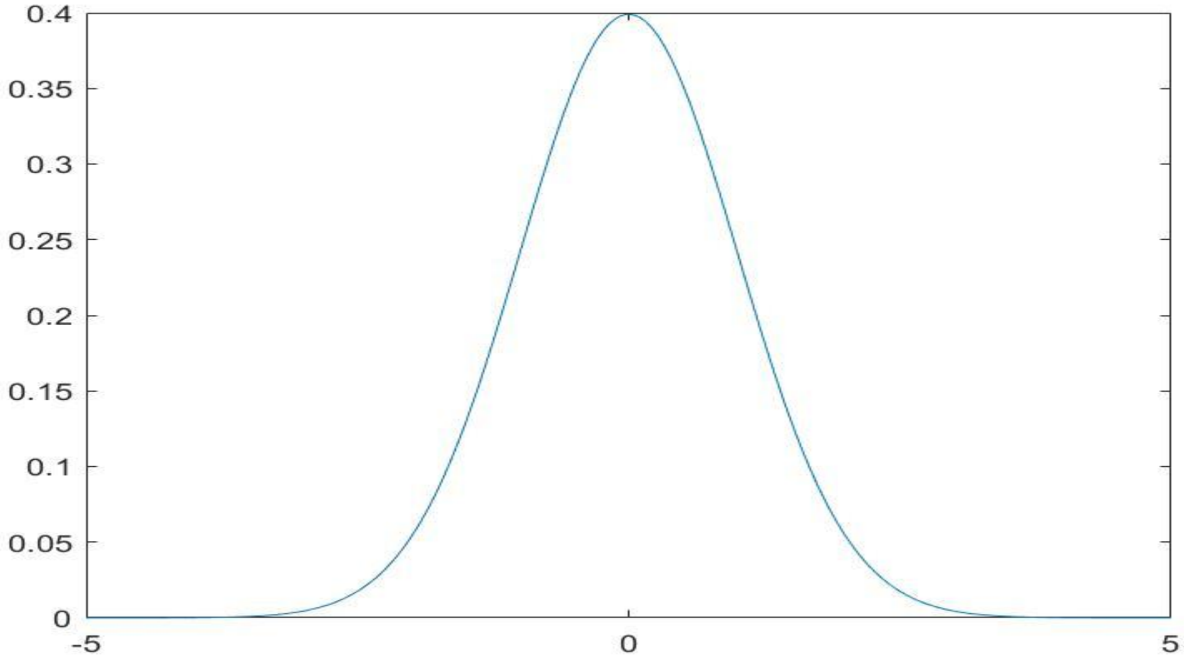
  因为对所有数据进行明确的分类，因此如果样本数据发生很小的扰动，那么样本的分类结果容易发生明显的改变。(*GMM*)



Gaussian Mixture Model

- 什么是高斯混合模型（GMM）？

一维高斯分布：$f(x) = \frac{1}{\delta\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\delta^2}}$ $N \sim (\mu, \sigma^2)$

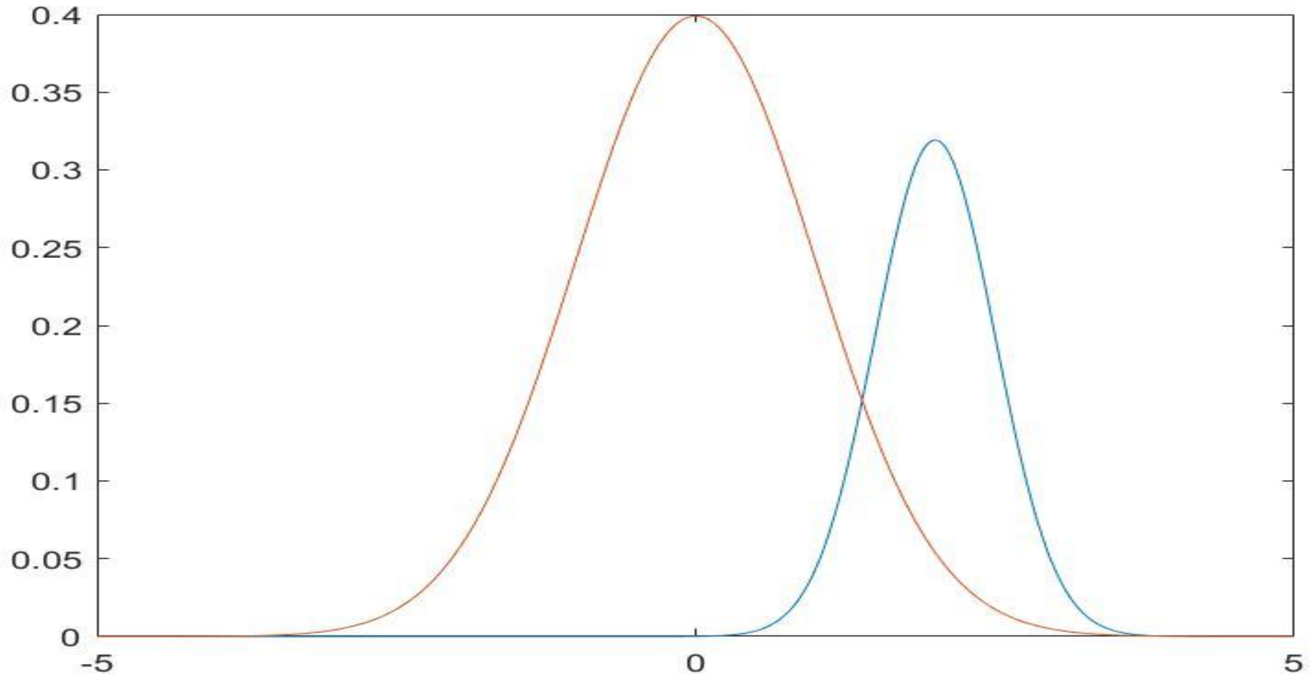


AI300学院

Gaussian Mixture Model

- 什么是高斯混合模型（GMM）？

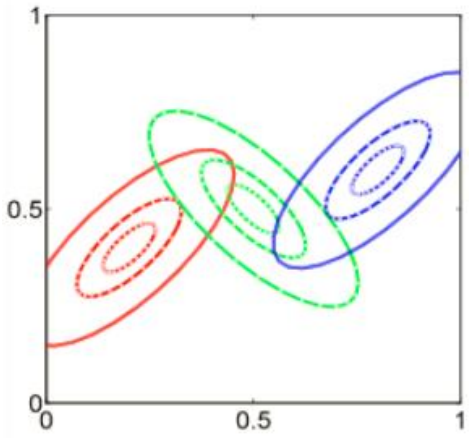
混合高斯分布： $p(x) = \sum_{k=1}^{K} p(k)p(x|k) = \sum_{k=1}^{K} \pi_k N(x|\mu_k, \Sigma_k)$



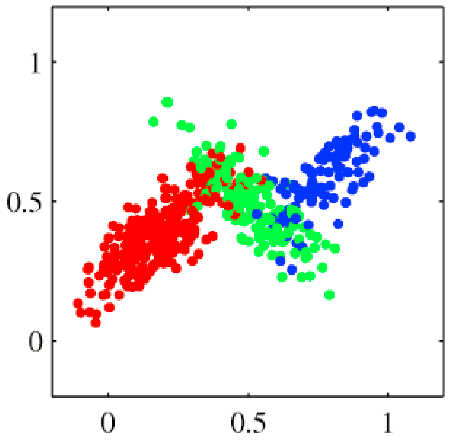
Gaussian Mixture Model

- 每一类都对应于一个高斯分布(K models)
- 数据的生成过程可以表示为:
  - 以概率π_k随机选择一个聚类k
  - 从第k个高斯模型中采样
- 概率密度函数表示为:

$$p(\mathbf{x}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \qquad \text{with} \qquad 0 \leqslant \pi_k \leqslant 1 \qquad \sum_{k=1}^{K} \pi_k = 1$$



拟合 $\Longleftrightarrow$



Gaussian Mixture Model

- 极大似然估计（MLE）

  1. 已知观测数据：$D = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \dots, \boldsymbol{x}_N\}$
  2. $Max_{\boldsymbol{\theta}} \prod_{i=1}^{N} p(\boldsymbol{x}_i|\boldsymbol{\theta})$

- 损失函数为**对数似然函数**（**log likelihood**）

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

- 训练过程困难

  1. 属于非凸问题，且高度非线性。
  2. 对于不同组成部分的求和操作在log函数内，因此将所有参数进行了复杂耦合。

  在上述情况下，简单地求导置零取值方法不可行

Gaussian Mixture Model

- 引入隐变量

1. 为每个样本$\boldsymbol{x}$定义一个K 维向量$\boldsymbol{z}$:

$$z_k \in \{0, 1\} \qquad \sum_k z_k = 1$$

2. 定义如下概率:

$$p(z_k = 1) = \pi_k \quad \Longleftrightarrow \quad p(\mathbf{z}) = \prod_{k=1}^{K} \pi_k^{z_k}$$

Likelihood: $p(\mathbf{x}|z_k = 1) = \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k) \Longleftrightarrow p(\mathbf{x}|\mathbf{z}) = \prod_{k=1}^{K} \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)^{z_k}$

$$\boxed{P(\boldsymbol{x}) = \sum_{\boldsymbol{z}} P(\boldsymbol{x}, \boldsymbol{z})}$$

$$\boxed{p(\mathbf{x}) = \sum_{\mathbf{z}} p(\mathbf{z}) p(\mathbf{x}|\mathbf{z}) = \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}$$

3. 后验概率

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

$$\boxed{p(z|\mathbf{x}) = \frac{p(z, \mathbf{x})}{p(\mathbf{x})} = \frac{p(z, \mathbf{x})}{\sum_z p(z, \mathbf{x})}}$$

AI300学院

Gaussian Mixture Model

- 将$\frac{\partial L}{\partial \boldsymbol{\mu}_k}$置零：

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

$$0 = -\sum_{n=1}^{N} \underbrace{\frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_j \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}}_{\gamma(z_{nk})} \boldsymbol{\Sigma}_k(\mathbf{x}_n - \boldsymbol{\mu}_k)$$

求得：

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

- 类似的，将$\frac{\partial L}{\partial \boldsymbol{\Sigma}_k}$置零，我们得到：

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

Gaussian Mixture Model

- 求导可得：

$$\frac{\partial L}{\partial \boldsymbol{\mu}_k} = \frac{\ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\Sigma}, \boldsymbol{\mu})}{\partial \boldsymbol{\mu}_k}$$

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

$$= \sum_{i=1}^{N} \frac{\pi_k \frac{\partial \mathcal{N}(\mathbf{x}_i|\boldsymbol{\Sigma}_k, \boldsymbol{\mu}_k)}{\partial \boldsymbol{\mu}_k}}{\sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_i|\boldsymbol{\Sigma}_k, \boldsymbol{\mu}_k)}$$

$$\boxed{\frac{\partial \mathcal{N}(\mathbf{x}_i|\boldsymbol{\Sigma}_k, \boldsymbol{\mu}_k)}{\partial \boldsymbol{\mu}_k} = \mathcal{N}(\mathbf{x}_i|\boldsymbol{\Sigma}_k, \boldsymbol{\mu}_k)\Sigma_k(\mathbf{x}_i - \boldsymbol{\mu}_k)}$$

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

$$\boxed{\mathcal{N}(\mathbf{x}_i|\boldsymbol{\Sigma}_k, \boldsymbol{\mu}_k) = \frac{1}{(\sqrt{2\pi|\Sigma_k|})^D} \exp\left(-\frac{1}{2}(\mathbf{x}_i-\boldsymbol{\mu}_k)^T \Sigma_k^{-1}(\mathbf{x}_i-\boldsymbol{\mu}_k)\right)}$$

Gaussian Mixture Model

- 对于$\boldsymbol{\pi}$ 的优化需要一些数学技巧
- 思路: 使用拉格朗日乘数法 $L(\boldsymbol{\theta}) = \ln p(\mathbf{X}|\boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi})$

$$Q(\boldsymbol{\theta}, \lambda) = \ln p(\mathbf{X}|\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma}) + \lambda \left(\sum_{k=1}^{K} \pi_k - 1 \right)$$

- 通过令 $\frac{\partial Q}{\partial \pi_k} = 0$, 我们可以得到

$$0 = \sum_{n=1}^{N} \frac{\mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_j \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)} + \lambda$$

$$\Downarrow$$

对于全部的 π_k: $\sum_{n=1}^{N} \gamma(z_{nk}) + \lambda \pi_k = 0, \forall k = 1, 2, \cdots, K$

$$\Leftrightarrow$$

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

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

Expectation-Maximization算法

- 模型求参流程：
  EM算法：
  1. 初始化权重$\gamma(z_{nk})$ 和参数$\boldsymbol{\pi}$, $\boldsymbol{\mu}$, $\boldsymbol{\Sigma}$
  2. 运行如下步骤知道似然函数$L(\boldsymbol{\theta})$ 收敛
     E-step：固定参数，重新计算权重：

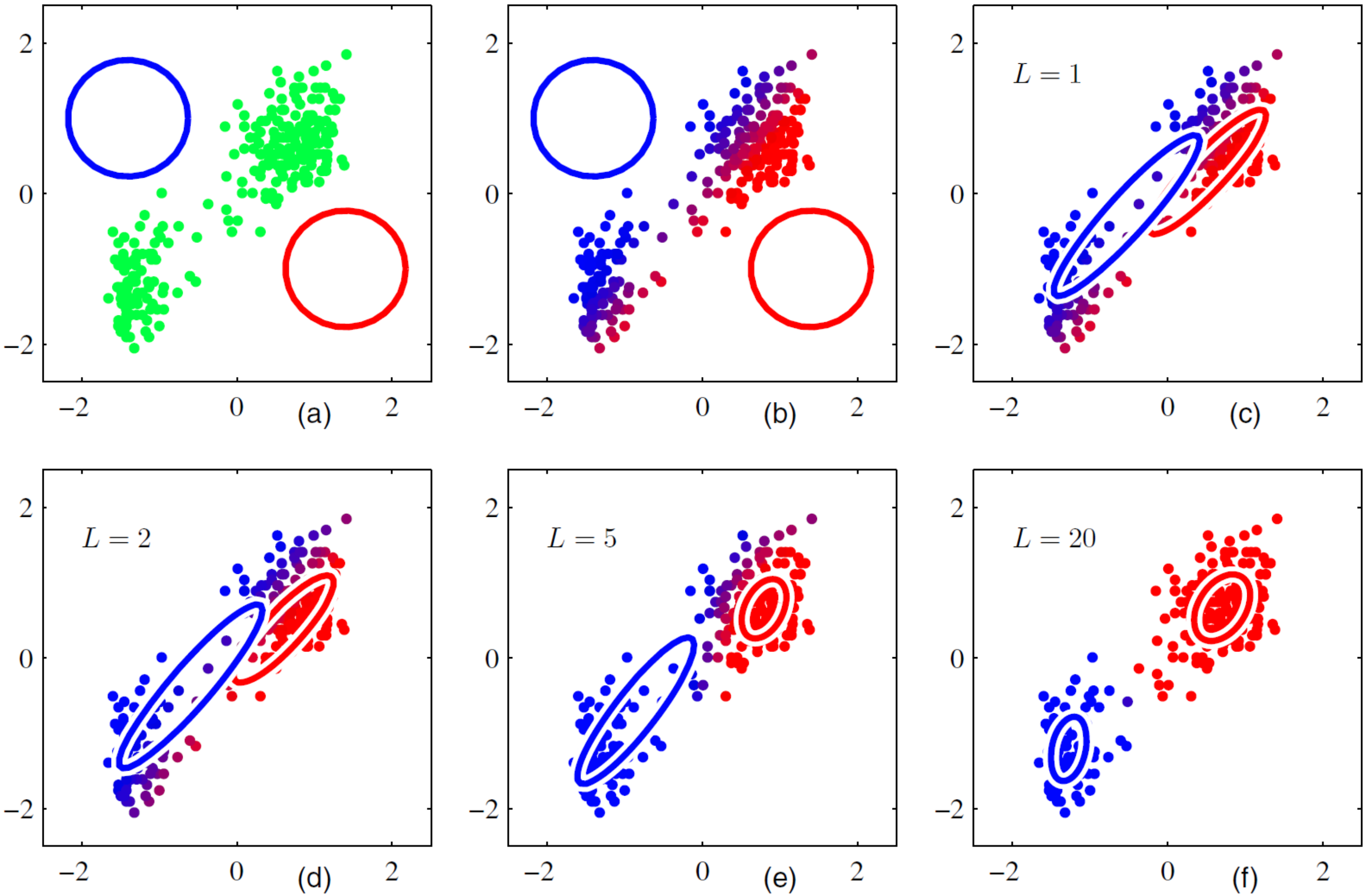
$$\gamma(z_{nk}) = \frac{\pi_k \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)}{\sum_{j=1}^{K} \pi_j \mathcal{N}(\mathbf{x}_n|\boldsymbol{\mu}_j, \boldsymbol{\Sigma}_j)}$$

     M-step：固定权重，重新计算参数：

$$\begin{aligned}
\boldsymbol{\mu}_k^{\text{new}} &= \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \mathbf{x}_n \\
\boldsymbol{\Sigma}_k^{\text{new}} &= \frac{1}{N_k} \sum_{n=1}^{N} \gamma(z_{nk}) \left(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}\right) \left(\mathbf{x}_n - \boldsymbol{\mu}_k^{\text{new}}\right)^{\mathrm{T}} \\
\pi_k^{\text{new}} &= \frac{N_k}{N}
\end{aligned}$$

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

Expectation-Maximization算法



模型运行示例

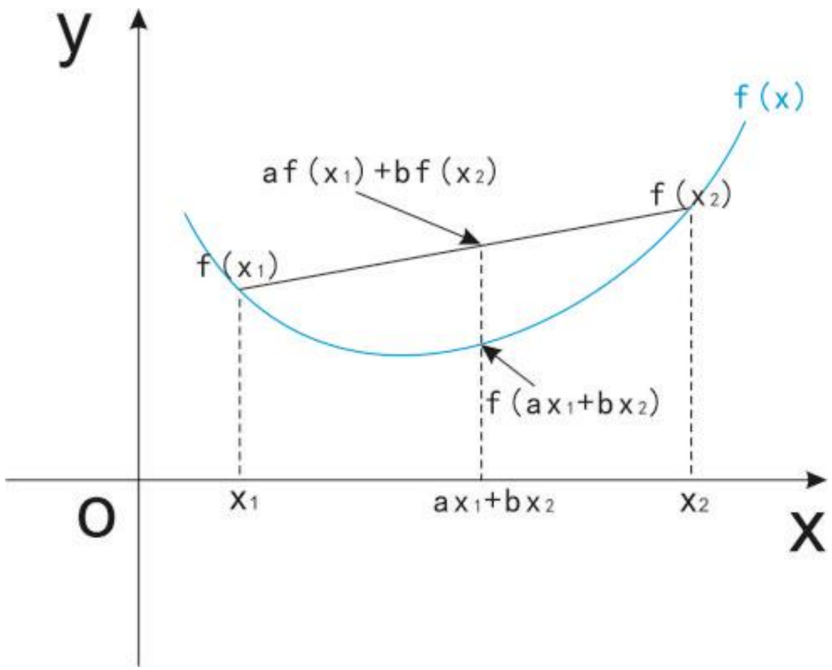
Expectation-Maximization算法

- Jensen’s Inequality(简森不等式)

1.$f(\boldsymbol{x})$是凸函数;
2.$\lambda_1, \lambda_2, \dots, \lambda_N \geq 0, \sum_i \lambda_i = 1$;

$$\Rightarrow \boxed{f(\sum_i \lambda_i \boldsymbol{x}_i) \leq \sum_i \lambda_i f(\boldsymbol{x}_i)}$$

等号成立条件：$\boldsymbol{x}_1 = \boldsymbol{x}_2 = \cdots = \boldsymbol{x}_N$



举例:
$p(x_1) + p(x_2) + \cdots + p(x_N) = 1$ 概率函数

$$\Rightarrow f\left(\sum_i p(x_i)x_i\right) \leq \sum_i p(x_i) f(\boldsymbol{x}_i)$$

$$\Rightarrow f(E(x)) \leq E(f(x))$$

Expectation-Maximization算法

- 目标：最大似然

$$\boxed{log\,P(X|\theta) = log\sum_{Z} P(X,Z|\theta)}$$

联合概率

$$\Downarrow$$

$$log\sum_{Z} P(X,Z|\theta) = log\sum_{Z} q(Z)\frac{P(X,Z|\theta)}{q(Z)}$$

$$\Downarrow$$

Jensen不等式

$$\boxed{log\sum_j \lambda_j y_j \geq \sum_j \lambda_j log y_j}$$

$$log\sum_Z P(X,Z|\theta) = log\sum_Z q(Z)\frac{P(X,Z|\theta)}{q(Z)} \geq \sum_Z q(Z) log\frac{P(X,Z|\theta)}{q(Z)}$$

Expectation-Maximization算法

- 优化下界

$$\begin{aligned} log\sum_Z P(X,Z|\theta) &= log\sum_Z q(Z)\frac{P(X,Z|\theta)}{q(Z)} \\ &\geq \sum_Z q(Z) log\frac{P(X,Z|\theta)}{q(Z)} \end{aligned}$$

Idea:交替优化$q(z)$以及θ

1. 假设θ给定，优化$q(z)$
2. 假设$q(z)$给定，优化θ
3. 迭代以上两步骤

Expectation-Maximization算法

- 优化下界

$$\boxed{\text{最大化 } \sum_Z q(Z)log\frac{P(X,Z|\theta)}{q(Z)}}$$

1. 假设θ给定，优化$q(z)$

不等式取等号 $\Rightarrow$ $\frac{P(X,Z|\theta)}{q(Z)} = C$

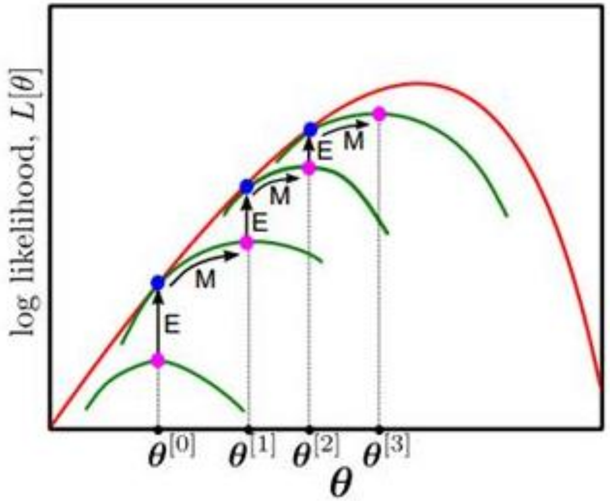
后验概率

$$\Rightarrow q(z) = \frac{P(x,z|\theta)}{\sum_z P(x,z|\theta)} = P(z|x,\theta)$$

$$p(z=k|x,\theta) = \frac{\pi_k p(x|\theta_k)}{\sum_k \pi_k p(x|\theta_k)}$$

1. 假设$q(z)$给定，优化θ

$$\theta = argmax \sum_z q(z) log P(x,z|\theta)$$

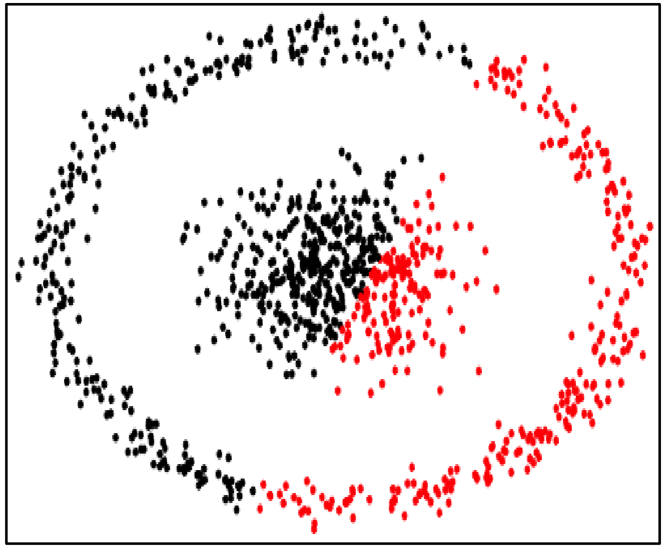


Gaussian Mixture Model

- GMM与K-Means对比：
  1. GMM可以认为是一种平滑过的K-Means方法

$$\gamma(z_{nk}) = \frac{\pi_k \exp\left\{-\|\mathbf{x}_n - \boldsymbol{\mu}_k\|^2/2\epsilon\right\}}{\sum_j \pi_j \exp\left\{-\|\mathbf{x}_n - \boldsymbol{\mu}_j\|^2/2\epsilon\right\}} \xrightarrow{\epsilon \to 0} \{0,1\}$$

  2. 同样的，两种方法都只能处理“凸性”数据



GMM实战

GMM scikit-learn的python实现

【对scikit-learn中random forest概述】

SKlearn库GaussianMixture类是EM算法在混合高斯分布的实现

参数分析：

- n_components：混合高斯模型个数，默认为1
- covariance_type：协方差类型，包括{‘full’,‘tied’, ‘diag’, ‘spherical’}四种
- random_state :随机数发生器
- max_iter，n_init：最大迭代次数，默认100；初始化次数，默认1
- reg_covar:协方差对角非负正则化，保证协方差矩阵均为正，默认为0
- init_params： {‘kmeans’, ‘random’}

GMM实战

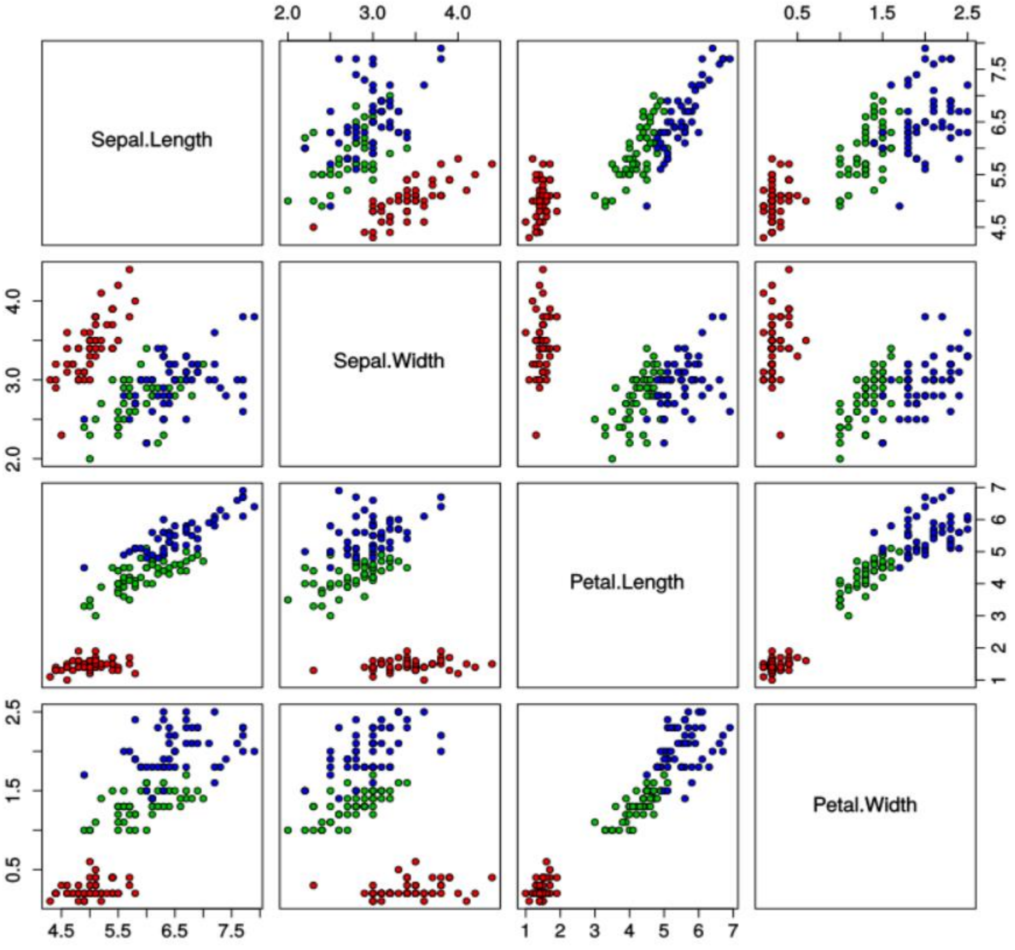
【GMM在iris数据集上的无监督分类】

iris数据集包含150个样本，对应数据集的每行数据。每行数据包含每个样本的四个特征和样本的类别信息，所以iris数据集是一个150行5列的二维表。

Iris数据集包含四个特征（花萼长度、花萼宽度、花瓣长度、花瓣宽度），三类样本（山鸢尾、变色鸢尾还是维吉尼亚鸢尾）。

GMM scikit-learn的python实现

Iris Data (red=setosa,green=versicolor,blue=virginica)



GMM实战

GMM scikit-learn的python实现

【GMM分类】

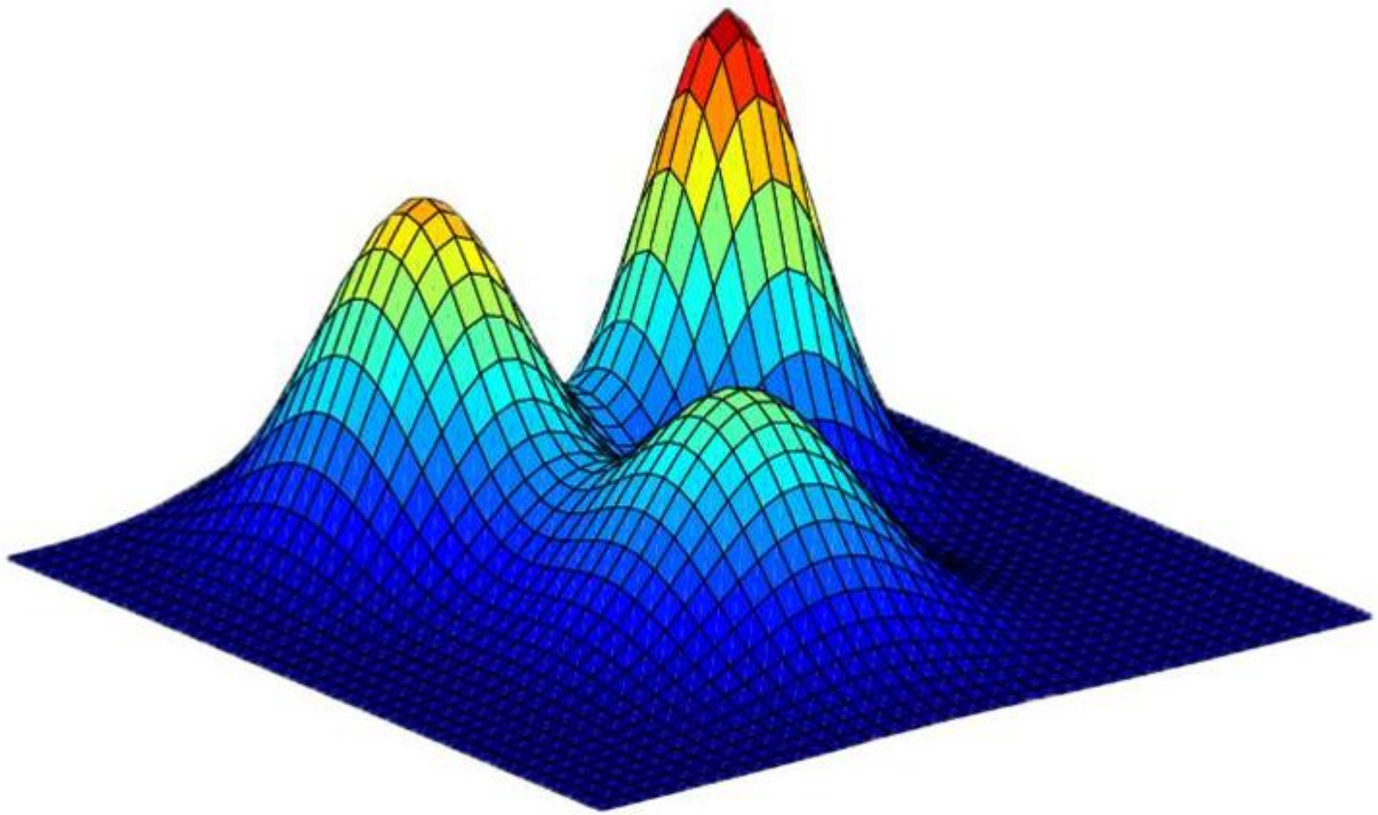
```
import numpy as np
from sklearn import datasets
from sklearn.mixture import GaussianMixture
#读取数据
iris=datasets.load_iris()
x=iris.data[:,:2]
y=iris.target
mu = np.array([np.mean(x[y == i], axis=0) for i in range(3)])
print '实际均值 = \n', mu
gmm=GaussianMixture(n_components=3,covariance_type='full', random_state=0)
gmm.fit(x)
print 'GMM均值 = \n', gmm.means_
y_hat2=gmm.predict(x)
y_hat2[y_hat2==1]=3
y_hat2[y_hat2==2]=1
y_hat2[y_hat2==3]=2
print '分类正确率为',np.mean(y_hat2==y)
```

```
实际均值 =[[5.006  3.418]
           [ 5.936 2.77 ]
           [ 6.588 2.974]]
GMM均值 =[[5.01494511  3.44040237]
          [ 6.69225795 3.03018616]
          [ 5.90652226 2.74740414]]
分类正确率为 0.786666666667
```

GMM实战

GMM scikit-learn的python实现

【GMM分类结果】



AI300学院