EM算法求高斯混合模型参数预计——Python实现

EM算法一般表述：

当有部分数据缺失或者无法观察到时，EM算法提供了一个高效的迭代程序用来计算这些数据的最大似然预计。在每一步迭代分为两个步骤：期望（Expectation）步骤和最大化（Maximization）步骤，因此称为EM算法。

如果所有数据Z是由可观测到的样本X={X1, X2，……, Xn}和不可观测到的样本Z={Z1, Z2，……, Zn}组成的，则Y = X∪Z。EM算法通过搜寻使所有数据的似然函数Log(L(Z; h))的期望值最大来寻找极大似然预计，注意此处的h不是一个变量，而是多个变量组成的参数集合。此期望值是在Z所遵循的概率分布上计算，此分布由未知参数h确定。然而Z所遵循的分布是未知的。EM算法使用其当前的如果h`取代实际参数h，以预计Z的分布。

$$Q(h`| h) = E [ \ln P(Y|h`) | h, X]$$

EM算法反复下面两个步骤直至收敛。

步骤1：预计（E）步骤：使用当前如果h和观察到的数据X来预计Y上的概率分布以计算Q(h` | h)。

$$Q(h` | h ) \leftarrow E[ \ln P(Y|h`) | h, X]$$

步骤2：最大化（M）步骤：将如果h替换为使Q函数最大化的如果h`:

$$h \leftarrow \mathrm{argmax} Q(h` | h)$$

高斯混合模型参数预计问题：

简单起见，本问题研究两个高斯混合模型参数预计k=2。

问题描写叙述：如果X是由k个高斯分布均匀混合而成的，这k个高斯分布的均值不同，可是具有同样的方差。设样本值为x1, x2, ……, xn。xi能够表示为一个K+1元组< xi, zi1, zi2, …, zik>。当中仅仅有一个取1，其余的为0。此处的zi1到zik为隐藏变量。是未知的。且随意zij被选择的概率相等，即

$$P（zij = 1）=1/k\ (j=1,2,3.....k)$$

EM算法求解过程推导例如以下：

"将oci位置改成client中OCI的位置"--------这个是什么意思？跟第二步中有区别吗？

--kevin_cnblogs

3. Re:初步了解更新锁（U）与排它锁（X）

全是复制的，翻来覆去就这几篇文章，我也是服

--花溪的小石头

4. Re:关于离职证明和竞业条款

为什么？？？？

--小草OI

5. Re:Qt跨平台的一个例程

Qt的跨平台属于代码级别的跨平台. 同一个项目, 需要在win,Mac,linux下面,使用对应的编译环境,进行编译. 而java是在虚拟机里面运行的, 所以只编译一次. 可是必须运行环境中安装虚拟机......

--[0]

阅读排行榜

1. C++中对字符串进行插入、替换、删除操作(4316)
2. Google翻译PDF文档(2641)
3. Mybatis+0+null，小问题引发的血案(2162)
4. Opencv 图像读取与保存问题(1620)
5. EM算法求高斯混合模型参数预计——Python实现(1478)

评论排行榜

1. 简单工厂模式和策略模式的差别(1)
2. Qt跨平台的一个例程(1)
3. Navicat 提示Cannot create oci environment 解决方式(1)
4. 初步了解更新锁（U）与排它锁（X）(1)
5. 关于离职证明和竞业条款(1)

推荐排行榜

1. Android4.4 Telephony流程分析——彩信(MMS)发送过程(1)
2. Shuttle ESB实现消息推送(1)

一. 求 $E[\ln P(Y|h')]$

$$\ln P(Y|h') = \ln \prod_{i=1}^{n} P(y_i|h')$$
$$= \sum_{i=1}^{n} \ln P(y_i|h')$$
$$= \sum_{i=1}^{n}\left(\ln \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2\sigma^2}\sum_{j=1}^{k} z_{ij}(x_i-\mu_j')^2\right)$$

其中: $P(y_i|h') = P(x_i, z_{i1}, \cdots z_{ik}|h') = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}\sum_{j=1}^{k} z_{ij}(x_i-\mu_j')^2}$

$$E[\ln P(Y|h')] = E\left[\sum_{i=1}^{n}\left(\ln \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2\sigma^2}\sum_{j=1}^{k} z_{ij}(x_i-\mu_j')^2\right)\right]$$
$$= \sum_{i=1}^{n}\left(\ln \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2\sigma^2}\sum_{j=1}^{k} E[z_{ij}](x_i-\mu_j')^2\right)$$

即: $Q(h'|h) = \sum_{i=1}^{n}\left(\ln \frac{1}{\sqrt{2\pi\sigma^2}} - \frac{1}{2\sigma^2}\sum_{j=1}^{k} E[z_{ij}](x_i-\mu_j')^2\right)$

$$E[z_{ij}] = \frac{P(X=x_i|\mu=\mu_j)}{P(X=x_i)} = \frac{e^{-\frac{1}{2\sigma^2}(x_i-\mu_j)^2}}{\sum_{n=1}^{k} e^{-\frac{1}{2\sigma^2}(x_i-\mu_n)^2}}$$

二. 求使 $E[\ln P(Y|h')]$ 期望最大的假设 $h' = <\mu_1', \mu_2', \cdots \mu_k'>$

$$\frac{\partial(E[\ln P(Y|h')])}{\partial \mu_j'} = \sum_{i=1}^{n} \frac{1}{\sigma^2} E[z_{ij}](x_i-\mu_j')$$
$$= \sum_{i=1}^{n} \frac{1}{\sigma^2} E[z_{ij}]x_i - \sum_{i=1}^{n} \frac{1}{\sigma^2} E[z_{ij}]\mu_j'$$

令 $\frac{\partial(E[\ln P(Y|h')])}{\partial \mu_j'} = 0$

可得 $\mu_j' = \frac{\sum_{i=1}^{n} E[z_{ij}]x_i}{\sum_{i=1}^{n} E[z_{ij}]}$

Python实现（模拟2个正态分布的均值预计）：

```
#coding:gbk
import math
import copy
import numpy as np
import matplotlib.pyplot as plt

isdebug = False

# 指定k个高斯分布参数。这里指定k=2。

注意2个高斯分布具有同样均方差Sigma。分别为Mu1,Mu2。


def ini_data(Sigma,Mu1,Mu2,k,N):
    global X
    global Mu
    global Expectations
    X = np.zeros((1,N))
    Mu = np.random.random(2)
    Expectations = np.zeros((N,k))
    for i in xrange(0,N):
        if np.random.random(1) > 0.5:
            X[0,i] = np.random.normal()*Sigma + Mu1
```

```
        else:
            X[0,i] = np.random.normal()*Sigma + Mu2
    if isdebug:
        print "***********"
        print u"初始观测数据X："
        print X
# EM算法：步骤1。计算E[zij]
def e_step(Sigma,k,N):
    global Expectations
    global Mu
    global X
    for i in xrange(0,N):
        Denom = 0
        for j in xrange(0,k):
            Denom += math.exp((-1/(2*(float(Sigma**2))))*(float(X[0,i]-Mu[j]))**2)
        for j in xrange(0,k):
            Numer = math.exp((-1/(2*(float(Sigma**2))))*(float(X[0,i]-Mu[j]))**2)
            Expectations[i,j] = Numer / Denom
    if isdebug:
        print "***********"
        print u"隐藏变量E（Z）："
        print Expectations
# EM算法：步骤2。求最大化E[zij]的参数Mu
def m_step(k,N):
    global Expectations
    global X
    for j in xrange(0,k):
        Numer = 0
        Denom = 0
        for i in xrange(0,N):
            Numer += Expectations[i,j]*X[0,i]
            Denom +=Expectations[i,j]
        Mu[j] = Numer / Denom
# 算法迭代iter_num次，或达到精度Epsilon停止迭代
def run(Sigma,Mu1,Mu2,k,N,iter_num,Epsilon):
    ini_data(Sigma,Mu1,Mu2,k,N)
    print u"初始<u1,u2>:", Mu
    for i in range(iter_num):
        Old_Mu = copy.deepcopy(Mu)
        e_step(Sigma,k,N)
        m_step(k,N)
        print i,Mu
        if sum(abs(Mu-Old_Mu)) < Epsilon:
            break
if __name__ == '__main__':
   run(6,40,20,2,1000,1000,0.0001)
   plt.hist(X[0,:],50)
   plt.show()
```

本代码用于模拟k=2个正态分布的均值预计。当中ini_data(Sigma,Mu1,Mu2,k,N)函数用于生成训练样本，此训练样本时从两个高斯分布中随机生成的，当中高斯分布a均值Mu1=40、均方差Sigma=6，高斯分布b均值Mu2=20、均方差Sigma=6，生成的样本分布例如以下图所看到

的。因为本问题中实现无法直接冲样本数据中获知两个高斯分布参数。因此须要使用EM算法估算出详细Mu1、Mu2取值。

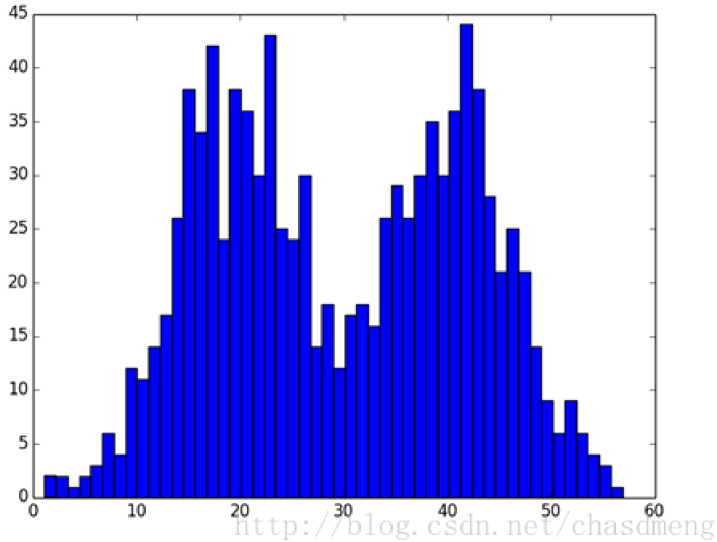

图 1 样本数据分布

在图1的样本数据下，在第11步时，迭代终止，EM预计结果为:

Mu=[40.55261688 19.34252468]

附:

极大似然预计

假定随机变量X服从某一个参数为θ的分布，其概率密度为$P(x;\theta)$，$\theta\in\Theta$，其中θ为带估参数，Θ是θ的可能取值范围。设$X_1, X_2, \ldots\ldots, X_n$是来自X的样本，则$X_1, X_2, \ldots\ldots, X_n$的联合分布概率为

$$\prod_{i=1}^{n} p(x_i;\theta)。$$

又设$x_1, x_2, \ldots\ldots, x_n$是相应于样本$X_1, X_2, \ldots\ldots, X_n$的一个样本值。则事件$\{X_1=x_1, X_2=x_2, \ldots\ldots, X_n=x_n\}$发生的概率为

$$L(\theta)=L(x_1,x_2,\ldots\ldots,x_n;\theta)=\prod_{i=1}^{n} p(x_i;\theta),\theta\in\Theta$$

这一概率随θ的取值变化而变化，称$L(\theta)$为样本的似然函数。

最大似然估计法就是根据固定样本观察值$x_1, x_2, \ldots\ldots, x_n$，在$\theta$可能的取值范围$\Theta$内挑选使似然函数$L(\theta)$达到最大的参数$\hat{\theta}$，作为参数$\theta$的估计值。即取$\hat{\theta}$使

$$L(x_1,x_2,\ldots\ldots,x_n;\hat{\theta})=\max_{\theta\in\Theta} L(x_1,x_2,\ldots\ldots,x_n;\theta)$$

称$Log(L(\theta))$为对数似然估计。

对于最大似然估计的求解一般通过微分学中的极值问题求解。$\hat{\theta}$一般可从方程

$$\frac{d}{d\theta}\ln(L(\theta))=0$$

参考文献：机器学习TomM.Mitchell P.137

« 上一篇：Missing styles. Is the correct theme chosen for this layout? Use the Theme combo box above the layou
» 下一篇：struts2基础梳理（二）