基于协方差的高斯混合模型参数学习算法

廖晓锋[1,2] 范修斌[3] 姜青山[2]
(南昌大学信息工程学院 南昌 330031)[1] (中国科学院深圳先进技术研究院 深圳 550085)[2]
(中国科学院软件研究所 北京 100190)[3]

摘 要 对混合高斯模型参数估计问题的算法通常是基于期望最大(Expectation Maximization)给出的。在混合高斯模型的因素协方差矩阵已知、因素各分量独立的前提下,给出了基于协方差矩阵的机器学习算法,简称CVB(Covariance Based)算法,并进行了一定的数学分析。最后给出了与期望最大算法的实验结果比较。实验结果表明,在该条件下,基于协方差的算法优于期望最大算法。

关键词 混合高斯模型,期望最大化,协方差,CVB算法

Covariance Based Learning Algorithm for Gaussian Mixture Model

LIAO Xiao-feng[1,2] FAN Xiu-bin[3] JIANG Qing-shan[2]
(Information Engineering School, Nanchang University, Nanchang 330031, China)[1]
(Shenzhen Institute of Advanced Technology, Chinese Academy of Sciences, Shenzhen 550085, China)[2]
(Institute of Software, Chinese Academy of Sciences, Beijing 100190, China)[3]

Abstract Expectation maximization is commonly used for parameter estimation in Gaussian mixture model. This paper presented a machine learning algorithm based on covariance(CVB) for solving the Gaussian mixture model with the specific constrain that covariance is already known. Experiments show that the CVB algorithm has better performance than the EM algorithm with regard to the specific constraint.

Keywords Gaussian mixture model, Expectation maximization, Covariance based, CVB algorithm

1 引言

高斯混合模型(Gaussian Mixture Model, GMM)指由多个高斯分布混合而成的模型。设:

$$p(X=x)=\sum_{i=1}^{K}\pi_i g(X_i=x|\mu_i,\Sigma_i)$$

式中,X 是 D 维连续或离散随机变量,π_i 是混合权重,满足 $\sum_{i=1}^{K}\pi_i=1$。其中 $g(X_i=x|\mu_i,\Sigma_i)$ 是第 i 个分布的密度函数,满足:

$$g(X_i=x|\mu_i,\Sigma_i)=\frac{\exp\{-\frac{1}{2}(x-\mu_i)'\Sigma_i^{-1}(x-\mu_i)\}}{(2\pi)^{D/2}|\Sigma_i|^{1/2}}$$

式中,μ_i,Σ_i 是第 i 个分布的均值向量和协方差矩阵。高斯混合模型参数分离的目的是求出参数集合 $\theta=\{\mu_i,\Sigma_i,\pi_i\}$,这是社会实践中常见的问题,例如人脸识别[1,2]、图像分割[3-7]、语音识别[8-10]等。

常用的高斯混合模型参数分离算法是期望最大(Expectation Maximization)算法[11-13]。EM算法是实现最大似然参数估计的一种算法,尤其适用于存在隐含变量时的情况。在高斯混合模型中,通常不知道观测数据来自于混合分布中的哪一个,用变量 z_{ij} 来表示样本 x_i 来自于第 j 个高斯分布。隐藏变量 z_{ij} 的存在,使得EM算法成为高斯混合模型的参数估计中常用的机器学习算法。EM的名字在Dempster、Laird和Rubin于1977年发表的文章中给出[14],由两个步骤组成,一是期望步(Expectation Step),二是最大化步(Maximization Step)。

令 $Y=X\cup Z$ 代表全体数据,h 代表参数 θ 的假设值,h' 代表在EM算法的每次迭代中修改的假设值。EM算法通过搜寻使 $E[\ln P(Y|h')]$ 最大的 h' 来寻找极大似然假设。

定义一个函数 $Q(h'|h)$,在 $\theta=h$ 和观察到的部分 X 的假定之下,它将 $E[\ln P(Y|h')|h,X]$ 作为 h' 的一个函数给出,我们可令:

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

在EM算法中,重复以下两个步骤直至收敛。

步骤1(估计期望) 使用当前假设 h 和观察到的数据 X 来估计似然函数的期望。

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

步骤2(最大化期望) 将假设 h 替换为使 Q 函数最大化的假设 h'。

$h\leftarrow \arg\max Q(h'|h)$

当函数 Q 连续时,EM算法收敛到似然函数 $P(Y|h')$ 的一个不动点。若此似然函数只有单个的最大值时,EM算法

本文受深圳市战略性新兴产业发展专项资金基础研究重点项目:海量恶意软件鉴别关键技术及其在钓鱼网站检测中的应用(JCYJ20120617120716224),江西省教育厅青年科学基金项目:双模态概率主题模型及基于DOT的并行扩展研究(GJJ13013)资助。

廖晓锋(1981—),男,博士,讲师,主要研究方向为主题模型、机器学习,E-mail:xfliao@ncu.edu.cn;范修斌(1962—),男,博士,研究员,主要研究方向为密码学、信息安全;姜青山(1962—),博士,研究员,主要研究方向为数据挖掘、信息安全。

可以收敛到这个全局的极大似然。否则，它只保证收敛到一个局部最大值[15]。

EM算法在GMM中的应用简介如下：

不妨设随机变量 $X\in R^d$ 是 k 个高斯分布的混合，如果其密度函数如下：

$$f(x|\theta)=\sum_{j=1}^{k}\pi_j\frac{1}{\sqrt{(2\pi)^d|\Sigma_j|}}\times\exp\{-\frac{1}{2}(x-\mu_j)^T\Sigma_j^{-1}(x-\mu_j)\}$$

其中参数集 $\theta=\{\pi_j,\mu_j,\Sigma_j\}_{j=1}^k$，满足：

1. $\pi_j>0,\sum_{j=1}^{k}\pi_j=1$；

2. $\mu_j\in R^d,\Sigma_j$ 是 $d\times d$ 维矩阵。

在给定数据 $x_1,\cdots,x_n$ 时，θ 的最大似然估计是 $\theta_{ML}=\arg\max f(x_1,\cdots,x_n|\theta)$。

GMM中需要估计的参数一般有 π_j,μ_j,Σ_j，或其中的部分参数。GMM的EM算法基本步骤：

第1步(Expectation step)：

计算

$$w_{ij}=\frac{\pi_j f(x_i|\mu_j,\Sigma_j)}{\sum_{i=1}^{K}\pi_i f(x_t|\mu_i,\Sigma_i)},j=1,\cdots,k,t=1,\cdots,n。$$

第2步(Maximization step)：

计算

$$\pi_j=\frac{1}{n}\sum_{t=1}^{n}w_{tj},\mu_j=\frac{\sum_{t=1}^{n}w_{tj}x_t}{\sum_{t=1}^{n}w_{tj}}$$

$$\Sigma_j=\frac{\sum_{t=1}^{n}w_{tj}(x_t-\mu_j)(x_t-\mu_j)^T}{\sum_{t=1}^{n}w_{tj}}$$

本文对高斯混合模型中因素协方差矩阵已知、因素各分量独立前提下的期望参数分离问题进行研究。这个条件问题的研究是具有理论意义和实践意义的，在社会实践中比比皆是，例如标准正态分布的混合模型中的期望参数分离等。EM算法是一般条件下的机器学习算法，在特殊条件下，通常都存在特殊的机器学习方法。本文给出了上述条件下的基于协方差矩阵的机器学习算法，并且给出了其数学分析。

本文最后给出了在该条件下的基于期望最大算法以及基于协方差矩阵算法的实验结果比较。实验结果表明，在上述条件下，基于协方差的算法优于期望最大算法。

2 基于协方差的高斯混合模型学习算法

不妨设 $X_i=(X_{i1},\cdots,X_{it}),i=1,\cdots,s$ 都是概率空间(Ω,F,p)上独立的 t 维正态分布随机变量，并且 $\forall i,1\leqslant i\leqslant s,X_{i1},X_{i2},\cdots,X_{it}$ 也是相互独立的。设 $X=X_1\cup X_2\cup\cdots\cup X_s$ 是 s 个 t 维正态分布随机变量的混合随机变量，假设我们已知各个随机变量的协方差矩阵：$\Sigma_i=(E((X_{ij}-\mu_{ij})(X_{ik}-\mu_{ik})))_{t\times t}$。我们要依据 X,Σ_i 求 $\mu_i=(\mu_{i1},\mu_{i2},\cdots,\mu_{it})$，其中 $\mu_{ij}=E(X_{ij})$，是分量的数学期望。

根据上述条件的要求，易得如下结论：

性质1 设 $D(X_{ij})=\sigma_{ij}^2,i=1,2,\cdots,s,j=1,2,\cdots,t,X_{i1},\cdots,X_{it}$ 相互独立，则：

$$\Sigma_i=\begin{pmatrix}\sigma_{i1}^2&0&\cdots&0\\0&\sigma_{i2}^2&\cdots&0\\&&\vdots&\\0&0&\cdots&\sigma_{it}^2\end{pmatrix}$$

证明：当 $j=k$ 时，

$E((X_{ij}-\mu_{ij})(X_{ik}-\mu_{ik}))=E(X_{ij}-\mu_{ij})^2=\sigma_{ij}^2$

当 $j\neq k$ 时，因 X_{ij},X_{ik} 独立，故：

$E((X_{ij}-\mu_{ij})(X_{ik}-\mu_{ik}))=E(X_{ij}-\mu_{ij})E(X_{ij}-\mu_{ij})=0$

我们的研究问题为：已知 $X=X_1\cup\cdots\cup X_s$ 和协方差矩阵 $\Sigma_i,i=1,\cdots s,X_i$ 的各分量独立，且满足正态分布，求 $\mu_i=(\mu_{i1},\cdots,\mu_{it}),i=1,\cdots,s$。

我们给出似然函数如下：

定义1(似然函数)

$$L=\sum_{x_{ik}}\sum_{x_{ij}}\sum_{j\neq k}\sum_{l}\sum_{i}(x_{ij}-\mu_{lj})(x_{ik}-\mu_{lk})\pi(i,j,k,l)F(X=(\cdots,x_{ij},\cdots,x_{ik},\cdots))+\sum_{x_{ij}}\sum_{j}\sum_{i}\sum_{l}(x_{ij}-\mu_{lj})^2\pi(i,j,j,l)F(X=(\cdots,x_{ij},\cdots))$$

其中，$i=1,\cdots,n,j,k=1,\cdots,t,l=1,\cdots,s,\pi(i,j,k,l)=\frac{p(X_{ij}=x_{ij},X_{ik}=x_{ik}|\mu_l)}{\sum_{l=1}^{s}p(X_{ij}=x_{ij},X_{ik}=x_{ik}|\mu_l)}$ 为边缘分布函数。

为描述方便，简记：

$F_2(x_{ij},x_{ik})=F(X=(\cdots,x_{ij},\cdots,x_{ik},\cdots))$

$F_1(x_{ij})=F(X=(\cdots,x_{ij},\cdots))$

性质2 $\pi(i,j,k,l)=\pi(i,k,j,l)$。

证明：由

$$\pi(i,j,k,l)=\frac{p(X_{ij}=x_{ij},X_{ik}=x_{ik}|\mu_l)}{\sum_{l=1}^{s}p(X_{ij}=x_{ij},X_{ik}=x_{ik}|\mu_l)}$$

易知性质成立。

当 $j\neq k$ 时，记

$$\theta(i,j,k,l)\triangleq\int_{x_{ij}-\frac{1}{2}\Delta}^{x_{ij}+\frac{1}{2}\Delta}\int_{x_{ik}-\frac{1}{2}\Delta}^{x_{ik}+\frac{1}{2}\Delta}e^{\frac{(x_{ij}-\mu_{lj})^2}{2\sigma_{lj}^2}+\frac{(x_{ik}-\mu_{lk})^2}{2\sigma_{lk}^2}}dx_{ij}\,dx_{ik}$$

其中，Δ 为我们算法的计算精度。当 $j=k$ 时，记 $\theta'(i,j,l)\triangleq\int_{x_{ij}-\frac{1}{2}\Delta}^{x_{ij}+\frac{1}{2}\Delta}e^{\frac{(x_{ij}-\mu_{lj})^2}{2\sigma_{lj}^2}}dx_j$

上述定义中，在 μ_{ij} 初始值给定的前提下，易得如下性质：

性质3

$$j\neq k,\pi(i,j,k,l)=\frac{\theta(i,j,k,l)}{\sum_{l=1}^{s}\theta(i,j,k,l)}$$

$$j=k,\pi(i,j,j,l)=\frac{\theta'(i,j,l)}{\sum_{l=1}^{s}\theta'(i,j.l)}$$

为了使用似然估计方法，我们近似认为 $F_2(x_{ij},\cdots,x_{ik})$ 及 $\pi(i,j,k,l)$ 是关于 $\mu_l,l=1,2,\cdots,s$ 的常数。在这个近似假定下，对 L 求偏导如下：

$$\frac{\partial L}{\partial\mu_{lj}}=-\sum_{i}\sum_{x_{ik}}\sum_{j\neq k}(x_{ik}-\mu_{lk})\pi(i,j,k,l)F_2(x_{ij},x_{ik})-2\sum_{i}(x_{ij}-\mu_{lj})\pi(i,j,j,l)F_1(x_{ij})$$

可得：

$$\frac{\partial L}{\partial\mu_{lj}}=0$$

$$\Leftrightarrow-\sum_{i}\sum_{x_{ik}}\sum_{j\neq k}(x_{ik}-\mu_{lk})\pi(i,j,k,l)F_2(x_{ij},x_{ik})=2\sum_{i}(x_{ij}-\mu_{lj})\pi(i,j,j,l)F_2(x_{ij},x_{ik})$$

$$\Leftrightarrow2\mu_{lj}\sum_{i}\pi(i,j,j,l)F_1(x_{ij})=2\sum_{i}x_{ij}\pi(i,j,j,l)F_1(x_{ij})+\sum_{i}\sum_{x_{ik}}\sum_{j\neq k}(x_{ik}-\mu_{lk})\pi(i,j,k,l)F_2(x_{ij},x_{ik})$$

$$\Leftrightarrow\mu_{lj}\sum_{i}\pi(i,j,j,l)F_1(x_{ij})=\sum_{i}x_{ij}\pi(i,j,j,l)F_1(x_{ij})+\frac{1}{2}\sum_{i}\sum_{x_{ik}}\sum_{j\neq k}(x_{ik}-\mu_{lk})\pi(i,j,k,l)F_2(x_{ij},x_{ik})$$

可得：

$$\mu_{ij}'=\frac{(\sum_i x_{ij}\pi(i,j,j,l)F_1(x_{ij})+\frac{1}{2}\sum_i\sum_{x_{ik}}\sum_{j\neq k}(x_{ik}-\mu_{ik})\pi(i,j,k,l)F_2(x_{ij},x_{ik}))}{\sum_i\pi(i,j,j,l)F_1(x_{ij})} \tag{1}$$

以上 μ_{ij}' 作为似然估计后的新值。

算法 1 CVB(Covariance Based)算法：

(1)给定初值向量 μ_i，i=1,2,…,s。

(2)利用式(1)求向量 μ_i'，i=1,2,…,s。

$$\mu_{uv}'=\frac{(\sum_{x_{uv}}x_{uv}\pi(u,v,v)F_1(x_{uv})+\sum_{x_{uv}}\sum_{x_{uk}}\sum_{v\neq k}(x_{uk}-\mu_{uk})\pi(u,v,k)F_2(x_{uv},x_{uk}))}{\sum_{x_{uv}}\pi(u,v,v)F_1(x_{uv})}$$

其中

$$\pi(u,v,v)=\frac{\theta'(u,v)}{\sum_{u=1}^{s}\theta'(u,v)}=\frac{\int_{x_{uv}-\frac{1}{2}\Delta}^{x_{uv}+\frac{1}{2}\Delta}e^{\frac{(x_v-\mu_{uv})^2}{2\sigma_{uv}^2}}dx_v}{\sum_{u=1}^{s}\int_{x_{uv}-\frac{1}{2}\Delta}^{x_{uv}+\frac{1}{2}\Delta}e^{\frac{(x_v-\mu_{uv})^2}{2\sigma_{uv}^2}}dx_v}$$

$$=\frac{\int_{x_{uv}-\frac{1}{2}\Delta}^{x_{uv}+\frac{1}{2}\Delta}e^{\frac{(x_v-\mu_{uv})^2}{2\sigma_{uv}^2}}dx_v}{s\int_{x_{uv}-\frac{1}{2}\Delta}^{x_{uv}+\frac{1}{2}\Delta}e^{\frac{(x_v-\mu_{uv})^2}{2\sigma_{uv}^2}}dx_v}=\frac{1}{s}$$

$$\pi(u,v,k)=\frac{\theta(u,v,k)}{\sum_{i=1}^{s}\theta(u,v,k)}$$

$$=\frac{\int_{x_{uv}-\frac{1}{2}\Delta}^{x_{uv}+\frac{1}{2}\Delta}\int_{x_{uk}-\frac{1}{2}\Delta}^{x_{uk}+\frac{1}{2}\Delta}e^{\frac{(x_v-\mu_{uv})^2}{2\sigma_{uv}^2}+\frac{(x_v-\mu_{uk})^2}{2\sigma_{uk}^2}}dx_vdx_k}{\sum_{i=1}^{s}\int_{x_{uv}-\frac{1}{2}\Delta}^{x_{uv}+\frac{1}{2}\Delta}\int_{x_{uk}-\frac{1}{2}\Delta}^{x_{uk}+\frac{1}{2}\Delta}e^{\frac{(x_v-\mu_{uv})^2}{2\sigma_{uv}^2}+\frac{(x_v-\mu_{uk})^2}{2\sigma_{uk}^2}}dx_vdx_k}$$

$$=\frac{1}{s}$$

所以有：

$$\mu_{uv}'=\frac{\frac{1}{s}\sum_{x_{uv}}^{x_{uv}}{}_{F_1(x_{uv})}+\frac{1}{s}\sum_{x_{uv}}\sum_{x_{uk}}\sum_{v\neq k}^{(x_{uk}-\mu_{uk})}{}_{F_2(x_{uv},x_{uk})}}{\frac{1}{s}\sum_{x_{uv}}F_1(x_{uv})}$$

$$=\frac{\sum_{x_{uv}}^{x_{uv}}{}_{F_1(x_{uv})}+\sum_{x_{uv}}\sum_{x_{uk}}\sum_{v\neq k}^{(x_{uk}-\mu_{uk})}{}_{F_1(x_{uv})F_1(x_{uk})}}{\sum_{x_{uv}}F_1(x_{uv})}$$

$$=\mu_{uv}+\sum_{x_{uk}}(x_{uk}-\mu_{uk})F_1(x_{uk})$$

$$=\mu_{uv}+\mu_{uk}-\mu_{uk}$$

$$=\mu_{uv}$$

3 基于方差的一维高斯混合模型算法

作为多维情况的特例，可以对一维高斯混合模型做出同样的改进。当随机变量只有一维时，协方差就不存在了，我们的算法也相应地退化成基于方差。注意到，一维高斯混合模型又称为 K 均值问题，是一类经典的机器学习问题。下面给出我们对一维高斯混合模型的基于方差的机器学习算法。

3.1 K 均值问题

K 均值问题，是为了估计 k 个正态分布的均值 $\theta=\langle\mu_1,\cdots,\mu_k\rangle$。数据 D 是一个实例集合，它由 k 个正态分布的混合而成的分布生成。每个实例由一个两步骤过程生成。首先，随机选择 k 个正态分布中的一个；其次，实例按照被选中的分布生成。

这一问题框架如图 1 所示。简单起见，不妨设 $k=2$，实例为沿着 x 轴分布的点。学习的任务是输出一个假设 $h=\langle\mu_1,\cdots,\mu_k\rangle$，它描述了 k 个分布中每一个分布的均值。我们希望对这些均值找到一个极大似然假设，即一个使 $L(D|h)$ 最大的假设 h，其中 D 代表实例数据。该问题中，已有的数据为观察到的 $X=\langle x_i\rangle$。隐藏变量为 $Z=\langle z_{i1},\cdots,z_{ik}\rangle$，表示第 k 个分布生成 x_i。单个正态分布的选择基于均匀的概率进行，k 个正态分布有相同的方差 σ^2，且方差已知。全部数据为三元组 $\langle x_i,z_{i1},z_{i2}\rangle$，其中 x_i 表示第 i 个实例的观测值，z_{i1},z_{i2} 表示两个正态分布中哪个被用于产生值 x_i。确切地讲，x_i 由第 j 个正态分布产生时，z_{ij} 值为 1，否则为 0。

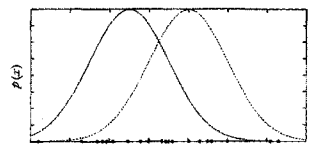


图 1 两个同方差正态分布混合生成的实例[16]

3.2 K 均值问题的 EM 算法

由于 z_{i1},z_{i2} 未知，无法直接使用最大似然法来求均值 μ_1 和 μ_2，Mitchell 在文献[16]中使用 EM 估计两个一维高斯分布的均值。EM 算法根据当前假设 $h=\langle\mu_1,\cdots,\mu_k\rangle$ 不断地再估计隐藏变量 z_{ij} 的期望值。然后用这些隐藏变量的期望值重新计算极大似然假设。

全部数据为 $\langle x_i,z_{i1},z_{i2}\rangle$，其中只有 x_i 可以观察到。令 X 代表观测到的数据，Z 代表未观察到的数据，$Y=X\cup Z$ 代表全体数据。$\theta=\langle\mu_1,\mu_2\rangle$ 代表参数 θ 的当前假设值，θ' 代表每次迭代中的新值。

首先推导出可用于 K 均值问题的表达式 $Q(h'|h)$。每个实例 $y_i=\langle x_i,z_{i1},z_{i2}\rangle$ 的概率 $p(y_i|h')$ 可被写作：

$$p(y|h')=p(x_i,z_{i1},z_{i2}|h')=\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{1}{2\sigma^2}\sum_{j=1}^{2}z_{ij}(x_i-\mu_j)^2}$$

其中只有一个 z_{ij} 值为 1，其他的为 0。所有 m 个实例的概率的对数似然为

$$\ln p(Y|h')=\ln\prod_{i=1}^{m}p(y_i|h')=\sum_{i=1}^{m}\ln p(y_i|h')$$

$$=\sum_{i=1}^{m}(\ln\frac{1}{\sqrt{2\pi\sigma^2}}-\frac{1}{2\sigma^2}\sum_{j=1}^{2}z_{ij}(x_i-\mu_j)^2)$$

最后，计算此对数似然的均值。对 z 的线性函数 $f(z)$，有下面的等式成立：

$$E[f(z)]=f(E[z])$$

可得：

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

$$=\sum_{i=1}^{m}(\ln\frac{1}{\sqrt{2\pi\sigma^2}}-\frac{1}{2\sigma^2}\sum_{j=1}^{2}E[z_{ij}](x_i-\mu_j)^2)$$

也即，Q 函数为：

(3)将第(2)步结果代入第(1)步，直到收敛。

性质 4 初始均值一样，迭代马上进入不动点。

证明：

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

其中，$h'=\langle\mu_1',\mu_2'\rangle$，而 $E[z_{ij}]$表示实例 x_i 由第 j 个正态分布生成的概率，可基于当前假设 $h=\langle\mu_1,\mu_2\rangle$计算得出：

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

在求出了 Q 函数之后，完成了 EM 算法的第 1 步(E)，第 2 步(M)接着寻找使此 Q 函数最大的 $h'=\langle\mu_1',\mu_2'\rangle$。

$$\begin{aligned}\operatorname{argmax} Q(h'|h)&=\arg\max\sum_{i=1}^{m}(\ln\frac{1}{\sqrt{2\pi\sigma^2}}-\frac{1}{2\sigma^2}\sum_{j=1}^{2}E[z_{ij}](x_i-\mu_j)^2)\\&=\operatorname{argmin}\sum_{i=1}^{m}\sum_{j=1}^{2}E[z_{ij}](x_i-\mu_j)^2\end{aligned}$$

由上式可得

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

3.3 基于方差的算法

接下来，我们给出基于方差的算法。不妨设 $X_1,X_2,\cdots,X_n$ 都是概率空间(Ω,F,p)上的独立离散型随机变量，且满足 $X_1\sim N(\mu_1,1),\cdots,X_n\sim N(\mu_n,1)$。我们已知随机变量 $X=X_1\parallel X_2\parallel\cdots\parallel X_n$，即 $X_1,X_2,\cdots,X_n$ 的混合。给出极大似然估计参数分离方法，即求 $\mu_1,\mu_2,\cdots,\mu_n$。设：

$p(X=a)>0,\forall i<a,p(X=i)=0$

$p(X=b)>0,\forall i>b,p(X=i)=0$

即在 $a\leqslant X\leqslant b$ 时有概率值。先构造似然函数如下：

定义 2(似然函数)

$$L=\sum_{i=a}^{b}(\sum_{s=1}^{n}(i-\mu_s)^2\frac{p(X=i)\int_{i-1}^{i}e^{-\frac{(x-\mu_s)^2}{2}}dx}{\sum_{s=1}^{n}\int_{i-1}^{i}e^{-\frac{(x-\mu_s)^2}{2}}dx})$$

记

$$\pi_s=\frac{\int_{i-1}^{i}e^{-\frac{(x-\mu_s)^2}{2}}dx}{\sum_{s=1}^{n}\int_{i-1}^{i}e^{-\frac{(x-\mu_s)^2}{2}}dx}$$

则

$$\begin{aligned}L&=\sum_{i=a}^{b}(\sum_{s=1}^{n}(i-\mu_s)^2p(X=i)\pi_s)\\&=\sum_{i=a}^{b}p(X=i)(\sum_{s=1}^{n}(i-\mu_s)^2\pi_s)\end{aligned}$$

$$\mu_w'=\frac{(\sum x_w\pi(u,v,v)F_1(x_w)+\sum\sum\sum_{x_w x_{uk}v\neq k}(x_{uk}-\mu_{uk})\pi(u,v,k)F_2(x_w,x_{uk}))}{\sum_{x_w}\pi(u,v,v)F_1(x_w)}$$

$$=\frac{\sum_{x_w}x_w\pi(u,v,v)F_1(x_w)}{\sum_{x_w}\pi(u,v,v)F_1(x_w)}=\frac{1}{n}\sum_{x_w}x_w$$

可见，均值为实例的加权平均。易知当这个高斯分布为一维时，结论也成立。

4 实验

本节使用两个实验对本文提出的方法进行验证。第一个实验使用来自两个一维高斯分布的合成数据来检验我们提出的方法处理低维随机变量的能力。第二个实验的目的是检验处理多维数据的能力，使用的是来自两个二维高斯分布的合成数据。简单起见，在试验中，假设各个分布的混合权值相等，也即取均匀分布，另外假设分布的方差、协方差和相关系数已知。所求的参数为均值(向量)。

4.1 K 均值实验

设有随机变量 $X_1\sim N(\sigma_1,\mu_1),X_2\sim N(\sigma_2,\mu_2),X_1\in[0,4],X_2\in[4,8]$。随机变量 X 由 X_1 和 X_2 混合生成，首先，随机选择两个正态分布中的一个，其次，按照选择的分布生成 $X\in[0,8]$。生成一组数据，如图 2 所示。简单起见，假设单

同样，简单起见，我们近似认为 π_s 是关于 $\mu_s,s=1,2,\cdots,n$ 的常数。在这个近似假定下，我们研究基于方差的机器学习算法。

求似然函数的偏导数：

$$\frac{\partial L}{\partial\mu_s}=-2\sum_{i=a}^{b}p(X=i)((i-\mu_s)\pi_s)=0$$

从而求出：

$$\begin{aligned}&\sum_{i=a}^{b}(i\pi_s p(X=i)-\mu_s\pi_s p(X=i))=0\\&\Rightarrow\sum_{i=a}^{b}(i\pi_s p(X=i))=\sum_{i=a}^{b}(\mu_s\pi_s p(X=i))\\&\Rightarrow\sum_{i=a}^{b}(i\pi_s p(X=i))=\mu_s\sum_{i=a}^{b}(\pi_s p(X=i))\\&\Rightarrow\mu_s'=\frac{\sum_{i=a}^{b}(i\pi_s p(X=i))}{\sum_{i=a}^{b}(\pi_s p(X=i))}\end{aligned}\tag{2}$$

算法 2

(1)任意取定(u_1,u_2)，但 $u_1\neq u_2$，否则第(1)步迭代，就进入不动点$(0.5,0.5)$。

(2)由式(2)求新的(u_1,u_2)。

(3)将第(2)步结果代入第(1)步，直至收敛。

性质 5 初始均值一样，马上进入不动点。

证明：已知 $\mu_1=\mu_2=\cdots=\mu_n$，由 π_s 的定义可知，

$$\pi_s=\frac{\int_{i-1}^{i}e^{-\frac{(x-\mu_s)^2}{2}}dx}{\sum_{s=1}^{n}\int_{i-1}^{i}e^{-\frac{(x-\mu_s)^2}{2}}dx}=\frac{1}{n}$$

因此，

$$\mu_s'=\frac{\sum_{i=a}^{b}(i\pi_s p(X=i))}{\sum_{i=a}^{b}(\pi_s p(X=i))}=\frac{\sum_{i=a}^{b}(ip(X=i))}{\sum_{i=a}^{b}(p(X=i))}=\frac{E(x)}{1}=\mu_s$$

推论 1 混合模型中只有一个高斯分布时，基于协方差/方差的算法比 EM 算法更合理。

证明：只有一个高斯分布，也即实例独立同分布抽取于一个高斯分布。当这个高斯分布为多维时，

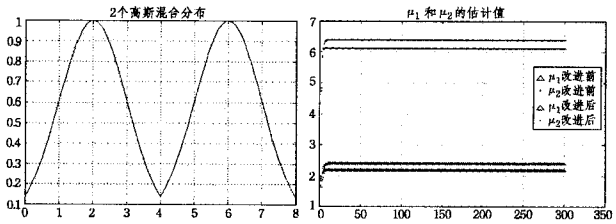
$$w_{tj}=\frac{1}{k}$$

$$\mu_j=\frac{\sum_{t=1}^{n}w_{tj}x_t}{\sum_{t=1}^{n}w_{tj}}=\frac{\sum_{t=1}^{n}\frac{1}{k}x_t}{\sum_{t=1}^{n}\frac{1}{k}}=\frac{1}{n}\sum_{t=1}^{n}x_t$$

为实例 x 的加权平均值。而

个正态分布的选择基于均匀概率进行，并且假设两个方差已知，学习任务是输出一个假设 $h=\langle\mu_1,\mu_2\rangle$，它描述了两个分布中每一个分布的均值。

从图 2 可以看出，CVB 算法比 EM 算法更快收敛，并且更靠近真实的均值($\mu_1=2,\mu_2=6$)。CVB 算法得到的估计参数收敛值为($\mu_1=2.1816,\mu_2=6.1101$)，EM 得到的估计参数收敛值为($\mu_1=2.3941,\mu_2=6.3816$)。相比 EM 算法，CVB 算法所得的两个估计值的精度分别提高 8.87%和 4.25%。



红色为 EM 的结果，蓝色为 CVB 的结果

图 2　两高斯分布混合所生成的实例(左)及参数估计结果(右)

4.2　二维高斯混合合成数据

使用来自两个二维高斯分布的合成数据。假设每个高斯分布中的两个分量各自独立。数据生成过程中，两个高斯分布以相同的概率被选取，另外假设分布的方差已知，需要估计的参数是两个分布的均值向量。

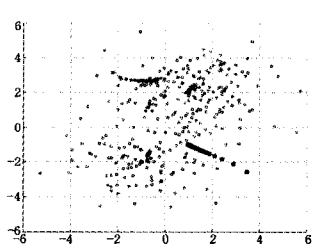
首先生成一个总体，其中样本数量为 300，分别以相同的概率从两个二维高斯分布中独立生成。假设两个二维高斯分布的均值和协方差矩阵均已知，

$$\mu_1=\langle-1,-2\rangle,\mu_2=\langle1,2\rangle$$

$$\Sigma_1=\begin{pmatrix}3&0\\0&2\end{pmatrix},\Sigma_2=\begin{pmatrix}2&0\\0&1\end{pmatrix}$$

我们的目标是在有观察数据 $X_i=(X_{i1},X_{i2},\cdots,X_{it})$，$i=1,2,\cdots,300$，$t=2$，以及已知协方差矩阵的情况下估计均值向量。

实验结果如图 3 所示，其中蓝色点所示为生成数据时所使用的均值真值，红色轨迹为使用 CVB 算法得出的均值估计值的迭代曲线，绿色为使用 EM 算法得出的均值估计值的迭代曲线。从图中看出绿色轨迹似乎比红色轨迹更接近真值。实际情况为，下方的绿色轨迹为对上方蓝色均值点的估计，上方绿色轨迹为对下方蓝色均值点的估计。可以看出 CVB 算法的起始估计值比 EM 算法的起始估计值更接近真值。



红色为 CVB 迭代曲线，绿色为 EM 迭代曲线

图 3　EM 和 CVB 在二维高斯混合合成数据的迭代轨迹

表 1 对比了 EM 算法和 CVB 算法得出的最终估计值和原始均值点的欧式距离，可以看出我们的算法比 EM 算法得出的结果更接近真实值。

表 1　CVB 和 EM 结果与真值的欧式距离

	原始均值	估计值	距离
EM	$\mu_1=\langle-1,-2\rangle$ $\mu_2=\langle1,2\rangle$	$\mu_1=\langle0.9723,-1.0152\rangle$, $\mu_2=\langle-0.3589,2.7041\rangle$	38.3149
CVB		$\mu_1=\langle-0.6591,-1.4822\rangle$, $\mu_2=\langle-1.1607,2.3322\rangle$	31.6317

结束语　本文针对混合高斯模型参数估计问题提出一种基于协方差矩阵 CVB(Covariance Based)的参数学习算法。该算法的适用条件是混合高斯模型的因素协方差矩阵已知，因素各分量独立。本文进行了一定的数学分析，并且通过实验将其与常用的期望最大算法进行了对比分析。实验结果表明，在该条件下，基于协方差的算法优于期望最大算法。

参 考 文 献

[1] Martinez B, Binefa X, Pantic M. Facial component detection in thermal imagery[C]// Proceedings of the Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference. June 2010: 48-54

[2] McKenna S J, Gong Shao-gang, Raja Y. Modelling Facial Colour and Identity with Gaussian Mixtures[J]. Pattern Recognition, 1998, 31(12): 1883-1892

[3] Figueiredo M. Bayesian Image Segmentation Using Gaussian Field Priors[M]// Rangarajan A, Vemuri B, Yuille A. Energy Minimization Methods in Computer Vision and Pattern Recognition. Berlin, Heidelberg: Springer, 2005: 74-89

[4] Chad C, Serge B, Hayit G, et al. Blobworld: Image Segmentation Using Expectation-Maximization and Its Application to Image Querying[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2002, 24: 1026-1038

[5] Greenspan H, Ruf A, Goldberger J. Constrained Gaussian mixture model framework for automatic segmentation of MR brain images[J]. IEEE Transactions on Medical Imaging, 2006, 25(9): 1233-1245

[6] 向日华，王润生. 一种基于高斯混合模型的距离图像分割算法[J]. 软件学报，2003，14(7)：1250-1257

[7] 陈允杰，张建伟，韦志辉，等. 基于高斯混合模型的活动轮廓模型脑 MRI 分割[J]. 计算机研究与发展，2007，9：1595-1603

[8] Reynolds D A, Rose R C. Robust text-independent speaker identification using Gaussian mixture speaker models[J]. IEEE Transactions on Speech and Audio Processing, 1995, 3(1): 72-83

[9] Reynolds D A, Quatieri T F, Dunn R B. Speaker Verification Using Adapted Gaussian Mixture Models[J]. Digital Signal Processing, 2000, 10(1-3): 19-41

[10] 张怡颖，朱小燕，张钹. 与文本无关的说话人自适应确认方法[J]. 软件学报，2000，11(6)：799-803

[11] Zivkovic Z, van der Heijden F. Recursive unsupervised learning of finite mixture models[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2004, 26(5): 651-656

[12] Figueiredo M, Leitão J, Jain A. On Fitting Mixture Models [M]// Hancock E, Pelillo M. Energy Minimization Methods in Computer Vision and Pattern Recognition. Berlin/Heidelberg: Springer, 1999: 732-732

[13] 王平波，蔡志明，刘旺锁. 混合高斯概率密度模型参数的期望最大化估计[J]. 声学技术，2007，26(3)：5

[14] Dempster A P, Laird N M, Rubin D B. Maximun Likelihood from Incomplete Data via the EM Algorithm[J]. Journal of the Royal Statistical Society, 1977, 39(1): 1-38

[15] Xu Lei, Jordan M I. On Convergence Properties of the EM Algorithm for Gaussian Mixtures[J]. Neural Computation, 1996, 8: 129-151

[16] Mitchell T M. Machine Learning[M]. McGraw-Hill, 1997