基于高斯混合模型的 EM 算法改进与优化

Improvement and Optimization of EM Algorithm Based on Gaussian Mixture Model

王凯南　金立左　（东南大学自动化学院，江苏 南京 210096）

摘要：作为聚类分析的一种重要模型，高斯混合模型在模式识别领域得到了广泛的应用。高斯混合模型的参数通常使用 EM 算法迭代训练获得。然而，传统的 EM 算法具有稳定性低，容易陷入局部极小值等缺点。针对传统 EM 算法的不足，改进了相关算法，在迭代过程中引入了自适应模型合并和模型分裂的策略。通过计算各高斯模型的熵，合并权值过低的模型，分裂熵过大的模型。此外，还优化了算法计算过程的相关步骤。相应实验结果表明，与传统 EM 算法相比，改进后的算法具有更强的适应性和更好的性能。

关键词：GMM 模型，EM 算法，信息熵，合并与分裂

Abstract:As an important model in the field of clustering analysis,Gaussian Mixture Model (GMM) has been widely used in pattern recognition.Usually,the iterative EM algorithm is applied to estimate GMM parameters.The traditional EM algorithm has a disadvantage of low stability,and it is easy to fall into the local minimum.In this paper,an improved EM algorithm is proposed to improve the stability of the algorithm by cutting off the low-weight component,and splitting the component with largest information entropy to reinitialize the empty mode.In addition,several steps of the calculation process are optimized.

Keywords:GMM model,EM Algorithm,entropy,merging and splitting

聚类分析是统计学习的一个重要分支，在许多领域内都被广泛应用，包括数据挖掘、模式识别、图像分析、信息检索等。由于能够很好地近似描述任何分布，高斯混合模型的聚类方法在聚类分析中得到了广泛的应用。高斯混合模型假定每个聚类成分是一个高斯分布，把模型看作是这几个高斯分布的加权叠加。相比于 K-means 聚类方法，高斯混合模型不仅为每个聚类成分分配了中心，还分配了每个聚类中心的方差。这些参数可以通过训练样本的参数估计的方法求得。

对于高斯混合模型，其参数估计方法通常使用迭代的 EM (Expectation-Maximization)算法[1]训练获得。该算法于 1977 年由 Dempster 等人总结提出，用于含有隐变量的概率参数模型的极大似然估计。EM 算法的每次迭代由两步组成：E 步，求期望；M 步，求极大值，所以这一算法称为期望极大化算法。

针对传统的 EM 算法训练过程中模型容易收敛于局部极值等问题，本文提出了基于模型熵的分裂与合并策略[2]以改进该算法。

1　高斯混合模型与 EM 算法

1.1 高斯混合模型

高斯混合模型是一种基于概率的聚类模型。该模型将每个点出现的概率看作几个高斯模型混合的结果。一个高斯混合模型由 K 个高斯模型组成，每个高斯模型称为一个成分，这些成分线性加成在一起就组成了高斯混合模型的概率密度函数：

$$p(x|\Theta)=\sum_{k=1}^{K}\pi_k p(x_i|\mu_k,\Sigma_k) \tag{1}$$

$$p(x|\mu_k,\Sigma_k)=\frac{1}{\sqrt{(2\pi)^d det\Sigma_k}}exp[-\frac{1}{2}(x-\mu_k)^T\Sigma_k^{-1}(x-\mu_k)] \tag{2}$$

其中 μ_k、Σ_k 分别为各高斯成分的均值和方差，π_k 为各成分的权重。

从高斯混合模型分布中随机采样一个点实际上可以分为两步。第一步，随机地在这 K 个高斯模型之中选择一个模型，每个模型被选中的概率是 π_k。第二步，选中了一个模型之后，考虑单独从这个被选中的高斯模型选取一个点，此时概率即公式(2)。

通过数据集训练高斯混合模型通常可以采用最大化似然函数的方法。对于高斯混合模型，其似然函数为：

$$l(\Theta|X)=\frac{1}{n}\sum_{i=1}^{n}log\sum_{k=1}^{K}\pi_k p(x_i|\mu_k,\Sigma_k) \tag{3}$$

由于直接求解(3)的最值比较困难，一般采取 EM 算法迭代训练获得。

1.2 EM 算法训练高斯混合模型

训练高斯混合模型的困难在于点到各高斯模式的概率是不可观的，因此必须视为隐变量[3]。EM 算法是一种近似计算含有隐变量模型的极大似然估计的方法，可以有效解决分配概率未知的问题。

考虑一般分布 $p(x|\Theta)=\int p(x,h|\Theta)dh$，其中 x 是测量值，h 是隐变量，$\Theta$ 是模型参数。通过在隐变量上引入辅助分布 q(h|x)，EM 算法可在给定当前参数的情况下更新隐变量辅助分布 $q(h|x)=p(h|x,\Theta_t)$(E-Step)，然后通过最大化对数似然函数下限来更新模型参数 Θ_t(M-Step)。这样通过反复迭代，使模型的对数似然函数收敛到局部最优。

对于高斯混合模型，隐变量是点归属于聚类中心的概率，即：

$$q(k_i|x_i)=q_{ik}=\frac{\pi_k p(x_i|\mu_k,\Sigma_k)}{\sum_{l=1}^{K}\pi_l p(x_i|\mu_l,\Sigma_l)} \tag{4}$$

此时 EM 算法[4]如下：

E-Step，更新隐变量 q_{ik}：

$$q_{ik}=\frac{\pi_k p(x_i|\mu_k,\Sigma_k)}{\sum_{l=1}^{K}\pi_l p(x_i|\mu_l,\Sigma_l)} \tag{5}$$

其中 $p(x|u_k,\Sigma_k)$ 为第 k 个高斯模型的高斯密度函数，π_k 为该高斯模型的先验概率。

M-Step，更新模型参数 π_k、μ_k、Σ_k：

$$\pi_k=\frac{\sum_{i=1}^{n} q_{ik}}{\sum_{i=1}^{n}\sum_{l=1}^{n} q_{il}} \tag{6}$$

$$\mu_k=\frac{\sum_{i=1}^{n} q_{ik}x_i}{\sum_{i=1}^{n} q_{ik}} \tag{7}$$

$$\Sigma_k=\frac{\sum_{i=1}^{n} q_{ik}(x_i-\mu_k)(x_i-\mu_k)^T}{\sum_{i=1}^{n} q_{ik}} \tag{8}$$

反复执行 E–Step 和 M–Step，直到达到终止条件，例如对数似然函数收敛到某一值。

2 EM 算法的改进与优化

2.1 EM 算法的缺陷

EM 算法优点在于简单性和普适性，通过反复执行 E 步和 M 部，逐步逼近最佳的参数，从而简化了参数估计的计算过程。

然而传统 EM 算法存在一些不足。比如，在迭代过程中，如果出现部分模型的权重过低，算法将不能收敛到理想结果，而陷入局部最优。针对传统 EM 算法的不足，本文提出了改进和优化的方法，在 M 步中增加自适应模型合并分裂的策略，合并权值过低的模型，并将“过大”的模型分裂为两个新模型。

2.2 基于模型熵的模型合并分裂策略

本文采取自适应模型合并分裂的策略，当混合模型中出现权值过小的模型时，寻找对对数似然函数贡献最大的高斯模型，将权值过小的模型合并到后者中，并将后者分裂为两个新模型。对数似然函数的期望如下：

$$E[\log(f(x)]=H(f)=-\int f(x)\log(f(x))dx \tag{9}$$

其中，$f(x)=\sum_{k=1}^{K} p_k g_k(x)$ (10)

在默认各高斯分布不相交的情况下混合模型的熵可以认为是每个高斯模型的贡献的累加，由此可以得到如下近似：

$$H(f)=\sum_{k=1}^{K} -p_k\log(p_k)+p_k H(g_k) \tag{11}$$

其中 $H(g_k)$ 是单独考虑的第 k 个高斯成分的信息熵：

$$H(g_k)=\frac{D}{2}(1+\log(2\pi)+\frac{1}{2}\sum_{i=1}^{D}\log(\Sigma_i^2)) \tag{12}$$

每次迭代过程中，找出对 H(f) 贡献最大的模型。寻找其方差最大的维度，以该维度上的高斯均值为基准，按照样本值大于或小于均值，将训练样本重新分配到两个新模型中。然后根据式(6)~(8)，采取最大化算法，更新重分配后的模型的参数。

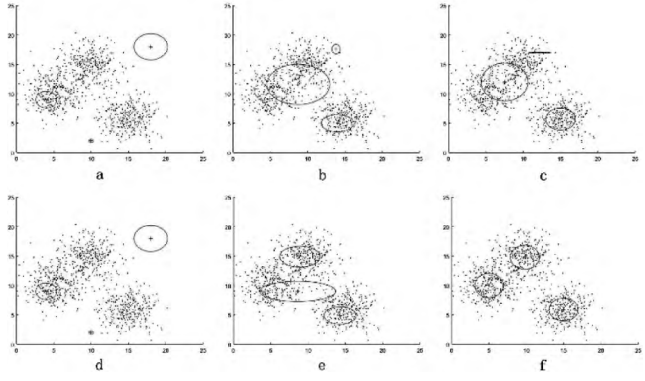

图 1 聚类过程

两种不同策略在训练过程中的效果如图 1 所示。其中，a 和 d 为初始值，b 为采用传统算法的一次迭代后的值，e 为采用改进算法的一次迭代后的值，c 和 f 为各自最终收敛的值。可以发现，改进后的算法可以更容易获取最佳效果，传统的算法陷入了局部最优。

此外，在计算过程中采用如下技巧可以减少 EM 算法计算量，并提高算法精度：

1）分配概率取对数。对于 EM 算法，采用(4)式计算点的分配概率，这个值一般很小。为了保证计算精度，可对其取对数计算，以抑制浮点数下溢；

2）用上次迭代的 μ_k 计算 Σ_k。在每次迭代中，通过(8)式更新 Σ_k 时需要知道 μ_k 的值，这样先后计算(7)式和(8)式会导致两次遍历数据。可以在计算 Σ_k 时采用上一次迭代时的 μ_k，这样(7)式和(8)式的计算同时进行，只需要在最终结果里修正得到的 Σ_k 即可，避免了重复遍历数据。

3 实验结果和分析

实验数据集来自 MNIST 数据库[5]，MNIST 数据库是 Google 实验室的 Corinna Cortes 和纽约大学柯朗研究所的 Yann LeCun 建立的一个手写数字数据库。训练库有 60,000 张手写数字图像。实验计算机配置为：Intel Core i5–2400 CPU 3.1GHz，4GB 内存。

由于 EM 算法对初始值敏感，需要多次训练取最优结果。表 1 为在对应相同初值条件下，采取传统 EM 算法和改进后的 EM 算法的训练结果对比。

表 1 传统 EM 算法和改进后 EM 算法的对比

实验序号	采用 EM 算法	迭代次数	迭代时间/s	对数似然函数值
1	传统算法	50	19	-3.52e6
	改进算法	78	26	-3.44e6
2	传统算法	50	19	-3.53e6
	改进算法	43	18	-3.48e6
3	传统算法	83	27	-3.52e6
	改进算法	36	17	-3.48e6

从表 1 可以看到，在相同初始条件下，改进后的 EM 算法有效避免了模型收敛于局部最优，可以得到更好的聚类效果。

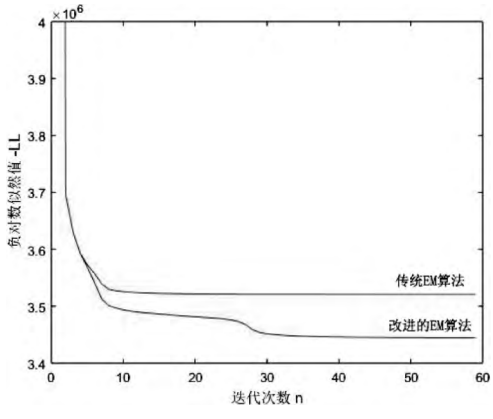

图 2 训练中负对数似然的下降曲线

图 2 为其中实验 1 训练过程中负对数似然函数值的下降曲线。可以发现，改进后算法的负对数似然函数值下降得更快。

今后的研究工作包括大规模样本下的模型训练及程序的进一步优化。

参考文献

[1]Dempster A P, Laird N M, Rubin D B. Maximum–likelihood from incomplete data via the em algorithm [J].Royal Statist. Soc. Ser. B, 1977,39(1):1–38

（下转第 118 页）

本环境的配置，因为每一条脚本可能对应着不同的测试环境。②数据服务器模块。该模块的主要功能是将对象、属性、业务数据等数据存放在数据池中，如操作登录用户名和密码等数据。③测试用例管理模块。该模块主要存放测试用例脚本，用例脚本分为数据驱动的脚本和共享的脚本，因为有些脚本会被多个测试用例使用，被其他脚本调用，故为了减少测试人员的工作量，将这些脚本单独提出形成共享脚本。数据驱动的脚本则是数据和流程控制分离的脚本，通过读入数据文件来驱动流程进行的脚本。④测试用例执行模块。该模块主要进行用例的执行。⑤测试结果分析模块。该功能主要是将用例执行后的结果与数据基线进行对比分析，得出测试结论。

3 Web 自动化测试框架(WATF)流程介绍

本次自动化测试的流程如图 3 所示。

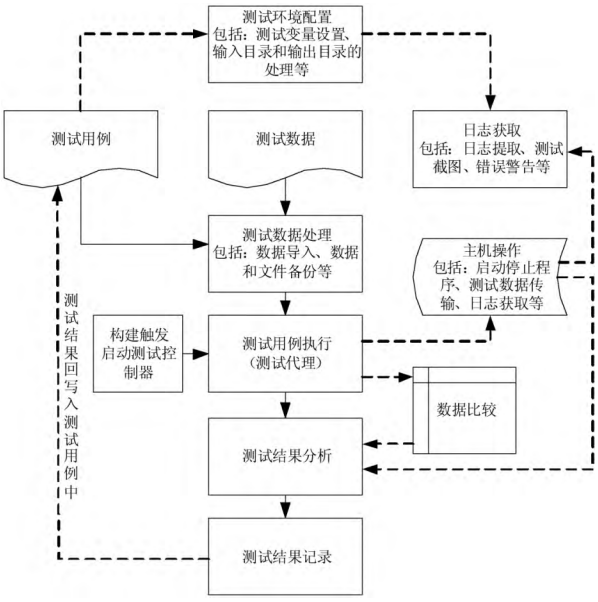

图 3 自动化测试流程图

1)测试前期准备：包括测试用例和测试数据准备。测试人员编写好测试用例，把准备好的测试用例和测试数据存放到指定的目录中。本次项目框架采用了脚本模块化思想，就是将测试用例分为共享用例模块和非共享用例模块，这样有利于测试用例的可维护性；测试数据采用了数据驱动框架的思想，将测试数据以 Excel 格式存放在测试控制器中，测试数据与测试脚本分离。

2)构建触发：即指启动框架进行自动化测试的外界因素。一般外界因素是多样化的，其中包括监测代码变更、定时启动、新版本等等。本次项目主要采用的是新版本监测方式，因此本次项目开发模式是敏捷开发，该模式的最大特点就是开发过程中，需求会随时改变，故而导致版本不断更新，因此就会不断产生新版本，使得框架更及时地进行自动化测试。

3)启动测试控制器：产生新版本则框架会启动测试控制器，测试控制器第一步调用测试环境，测试环境主要是针对不同的测试用例提供相应的测试环境，以便测试用例进行测试；第二步启动测试代理，本次项目框架特点之一就是实现并发式测试，提高测试效率，故而引入测试代理，其主要就是将大量的测试任务分配给不同的测试代理同时进行测试，分配任务主要就是通过测试代理的 IP 地址进行分配，测试代理的主要任务就是进行测试。

4)执行测试用例：测试代理执行测试用例，具体测试过程如图 4 所示。

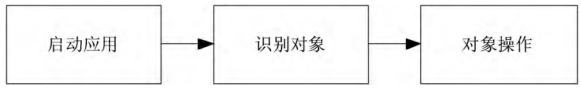

图 4 测试执行过程图

当有新的版本产生，则触发测试管理器启动测试代理执行测试，一般第一步是启动应用一般是打开网页或者调用接口两种方式，前者适用于 Web 系统，后者适用于传统软件进行测试；接着识别对象，其主要是指对页面的控件进行自动识别；最后对象操作，其是指在页面产生模拟的键盘鼠标进行自动操作、测试。

5)测试结果分析：测试用例执行后的测试结果将与之前测试人员设定的基线值进行对比，判断该条用例是否执行成功，如果用例执行成功，则用例结果为 pass，如果用例没有执行成功，则用例结果为 Fail，并且自动生成失败原因、分析结果、操作截屏，根据结果校验生成测试报告。

6)测试结果记录：当测试用例执行测试之后，系统将测试分析之后的结果和日志回记录到测试用例中。

4 结束语

本文以实际的工业云的 Web 页面为背景，结合已有的自动化测试框架的思想，着重研究基于 Visual Studio 2015 平台下的自动化测试框架，在新框架下测试人员只需要关注测试用例，而不是去重复执行非智力的测试，对于测试的效率得到显著的提升。同时，还可以自动生成测试结果并且发送到测试人员邮箱，降低测试时间。总之，该框架的提出实现了一种针对工业云 Web 界面的自动化测试框架。

参考文献

[1]朱少民.软件测试方法和技术[M].北京：清华大学出版社，2005
[2]赖利峰，刘强.Web 应用程序的一种功能自动化测试模型与实现[J].计算机工程，2006，32(17)：42-44
[3]贺平.软件测试教程[M].北京：电子工业出版社，2005
[4]李晓会.Web 系统自动化功能测试框架研究与实践[D].北京：北京邮电大学，2011
[5]曾北溟.自动化测试框架的研究与实现[D].武汉：武汉大学，2004
[6]王光源.Web 应用的自动化测试[D].济南：山东大学，2006
[7]朱菊，王志坚，杨雪.基于数据驱动的软件自动化测试框架[J].计算机技术与发展，2006，16(5)：68-70
[8]Mike K.Choosing a Test Automation Framework [S].IBM Developer

[收稿日期：2017.1.8]

(上接第 116 页)

[2]Lloyd S. Least squares quantization in PCM[J]. IEEE Transactions on Information Theory, 1982, 28(2):129-137
[3]Sundberg R.Maximum likelihood theory for incomplete data from an exponential family[J]. Scandinavian Journal of Statistics, 1974, 1(2):49-58
[4]Bilmes J A,Bilmes J A.A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models[J].International Computer Science Institute, 2000(4):2-7
[5]Lecun Y, Bottou L, Bengio Y, et al. Gradient-based learning applied to document recognition [J].Proceedings of the IEEE, 1998, 86(11):2278-2324

[收稿日期：2016.11.18]