

文章编号:1001-9081(2008)06-1546-03

基于混合微粒群算法的说话人识别

许允喜¹, 陈方²

(1. 湖州师范学院 信息工程学院, 浙江 湖州 313000; 2. 南京航空航天大学 自动化学院, 南京 210016)
(xuyunxi@hute.zj.cn)

摘要:为了解决传统高斯混合模型(GMM)对初值敏感,在实际训练中极易得到局部最优参数的问题,提出了一种采用微粒群算法优化GMM参数的新方法。该方法将最大似然估计融入到微粒群算法迭代过程中,形成了新的混合算法。它利用微粒群算法的全局优化性及最大似然估计的局部寻优性求解高斯混合模型的参数,以提高参数精度。说话人辨认实验表明,与传统的方法相比,新方法可以得到更优的模型参数,使得系统的识别率进一步提高。

关键词:说话人识别;微粒群算法;高斯混合模型

中图分类号: TN912.3; TP391.4 **文献标志码:**A

Speaker recognition based on hybrid particle swarm optimization algorithm

XU Yun-xi¹, CHEN Fang²

(1. Institute of Information Engineering, Huzhou Teacher's College, Huzhou Zhejiang 313000, China;
2. College of Automation Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing Jiangsu 210016, China)

Abstract: The traditional training methods of Gaussian Mixture Model (GMM) are sensitive to the initial model parameters, which often leads to a local optimal parameter in practice. To resolve this problem, a new GMM optimization method was proposed based on Particle Swarm Optimization (PSO). It utilized Maximum Likelihood (ML) algorithm in the PSO iteration and provided a new architecture of hybrid algorithm. Because of the global optimization characteristic of the particle swarm optimizer method and the strong local searching capacity of ML, it can obtain model parameters with high precision. Experiment for text-independent speaker identification shows that this method can obtain more optimum GMM parameters and better results than the traditional method.

Key words: speaker identification; Particle Swarm Optimization (PSO); Gaussian Mixture Model (GMM)

0 引言

说话人识别通过对说话人语音信号的个性特征进行分析来识别说话人的身份,是生物特征识别研究领域的重要内容,具有很高的普适性、可接受性和输入设备简易性等特点。说话人识别技术在公安侦察、声控系统、医疗诊断、电子金融业务等方面有着广泛的应用前景。近年来,基于高斯混合模型(Gaussian Mixture Model, GMM)的说话人识别方法成为现代说话人识别技术的重要方法之一^[1]。

微粒群优化算法(Particle Swarm Optimization, PSO)算法^[2]借鉴了鸟群或鱼群捕食过程的社会行为,是一种启发式计算算法。PSO 的概念和参数调整都很简单,而且容易编程实现,它既保持了传统进化算法深刻的群体智慧背景,同时又有自己许多良好的优化性能,如:能够在较短的时间内求得高质量的解而且具有稳定的收敛特性等。PSO 已广泛应用于函数优化、人工神经网络训练、模糊系统控制等领域,成为目前进化计算研究的一个热点^[3-4]。

传统的 GMM 采用最大似然(Maximum Likelihood, ML)估计对模型参数进行优化,然而 ML 估计是一种局部搜索算法,本质上采用爬山技术来寻找最优解,因此对初值十分敏感,容易陷入局部极小值而得不到最优解。为了克服上述缺点,最近人们提出了各种算法对目标函数进行优化,主要有模糊聚类^[5]、遗传算法^[6-7]等。本文将微粒群算法引入到高斯混合模型的参数估计中,提出了 PSO 算法对 GMM 目标函数

进行优化的新方法。为了增加收敛速度,避免 POS 破坏模型参数,将微粒群算法与 ML 算法相结合,形成混合微粒群算法。实验表明,本文的方法较传统 GMM 参数估计方法能够得到更优的参数,提高了说话人识别系统的识别率。

1 高斯混合模型

高斯混合模型用许多高斯分量的加权和来表示。一个具有 M 个混合度的 D 维 GMM 表示为:

$$p(\mathbf{x} | \lambda) = \sum_{i=1}^M p_i b_i(\mathbf{x}) \quad (1)$$

其中, \mathbf{x} 是一个 D 维的观测矢量; $p_i (i = 1, 2, \dots, M)$ 为混合权值,且 $\sum_{i=1}^M p_i = 1$; $b_i(\mathbf{x})$ 为 D 维高斯函数,即:

$$b_i(\mathbf{x}) = \frac{\exp\{-\frac{1}{2}(\mathbf{x} - \boldsymbol{\mu}_i)' \boldsymbol{\Sigma}_i^{-1} (\mathbf{x} - \boldsymbol{\mu}_i)\}}{(2\pi)^{d/2} |\boldsymbol{\Sigma}_i|^{1/2}} \quad (2)$$

其中, $\boldsymbol{\mu}_i$ 为均值矢量, $\boldsymbol{\Sigma}_i$ 为协方差矩阵。协方差矩阵可以用满矩阵,也可简化为对角矩阵。

整个高斯混合模型便可由各均值矢量、协方差矩阵及混合分量的权值来描述,因此,我们将一个模型 λ 表示为如下三元式:

$$\lambda = \{p_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i, i = 1, 2, \dots, M\}$$

给定一个说话人的训练语音,那么训练说话人的目标就是估计 GMM 的参数。设某说话人的训练语音特征矢量序列为

收稿日期:2007-12-24;修回日期:2008-01-30。

作者简介:许允喜(1978-),男,江苏句容人,讲师,硕士,主要研究方向:智能信息处理、人工智能、说话人识别; 陈方(1987-),女,浙江诸暨人,硕士研究生,主要研究方向:智能信息处理、智能导航。

$X = \{x_t, t = 1, 2, \dots, T\}$, 它对于模型 λ 的似然度为:

$$p(X | \lambda) = \prod_{t=1}^T p(x_t | \lambda) \quad (3)$$

训练的目的就是找到一组参数 λ , 使 $p(X | \lambda)$ 最大, 即:

$$\lambda = \arg \max_{\lambda} p(X | \lambda) \quad (4)$$

利用 ML 估计得到 GMM 参数的重估公式如下:

$$p_i = \frac{1}{T} \sum_{t=1}^T p(i | x_t, \lambda) \quad (5)$$

$$\mu_i = \frac{\sum_{t=1}^T p(i | x_t, \lambda) x_t}{\sum_{t=1}^T p(i | x_t, \lambda)} \quad (6)$$

$$\Sigma_i = \frac{\sum_{t=1}^T p(i | x_t, \lambda) (x_t - \mu_i)^2}{\sum_{t=1}^T p(i | x_t, \lambda)} \quad (7)$$

其中 $p(i | x_t, \lambda)$ 是第 i 个分量的后验概率, 可由式(8)计算得到:

$$p(i | x_t, \lambda) = \frac{p_i b_i(x_t)}{\sum_{k=1}^M p_k b_k(x_t)} \quad (8)$$

ML 估计方法通过式(5)~(8)的不断迭代即可得到最终的高斯混合模型的参数。

最后, 用训练好的 GMM 模型参数对说话人的语音特征进行识别^[1], 不同的训练方法会得到不同的模型参数, 最终影响识别率。GMM 模型参数与结构的联合空间具有不可微、多峰值和欺骗性等特点。从该过程可以看出, 由于使用了梯度下降法, 使得收敛速度不稳定, 有时计算中会出现奇异矩阵导致收敛失败; 同时, ML 算法只能保证收敛到局部极值, 它对每一类的初始值有较强的依赖性, 一旦初始值的估计和真实值偏差较大, ML 算法极易陷入局部最优, 从而影响模型参数估计的质量, 导致说话人识别性能下降。所以, 本文引入微粒群算法对高斯混合模型参数进行优化估计, 以提高参数估计精度。

2 微粒群算法

PSO 算法是一种基于群体搜索的算法, 它建立在模拟鸟群社会的基础上。PSO 将每一个可能产生的解表述为群中的一个微粒, 每个微粒都具有自己的位置向量和速度向量, 以及一个由目标函数决定的适应度。所有微粒在搜索空间中以一定的速度飞行, 通过追随当前搜索到的最优值来寻找全局最优。

假设在一个 D 维目标搜索空间有 m 个微粒, 每个微粒的位置表示一个潜在的解。第 i 个微粒的位置可表示为一个 D 维的向量 $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$ 。第 i 个微粒的“飞翔”速度也是一个 D 维的向量 $V_i = (v_{i1}, v_{i2}, \dots, v_{iD})$, 它经历过的最优位置记为 $P_i = (p_{i1}, p_{i2}, \dots, p_{iD})$, 也称为个体极值 $pbest$ 。整个微粒群迄今为止搜索到的最优位置记为 $P_g = (p_{g1}, p_{g2}, \dots, p_{gD})$, 也称为全局极值 $gbest$ 。PSO 是一种基于迭代的算法。粒子按式(9)、(10) 来调整自己的位置^[2]:

$$V_{i+1} = V_i + c_1 \cdot r_1 \cdot (P_i - X_i) + c_2 \cdot r_2 \cdot (P_g - X_i) \quad (9)$$

$$X_{i+1} = X_i + V_{i+1} \quad (10)$$

其中 V_{i+1} 为更新后的微粒速度; X_{i+1} 为更新后的微粒位置; V_i 为当前微粒速度; X_i 为当前微粒位置; r_1 和 r_2 为范围在 $[0, 1]$ 内变化的随机数; c_1 和 c_2 为加速常数, 表示向 $pbest$ 和 $gbest$ 靠近的速度变化, 取值越大, 则微粒移向 $pbest$ 和 $gbest$ 的加速度

越大。

此外, 微粒的速度 V_i 被一个最大速度 V_{max} 所限制, 以免微粒跑到可行解的范围之外进行搜索。标准 PSO 的算法流程如下:

步骤 1: 初始化一群微粒(群体规模为 m), 包括随机位置和速度;

步骤 2: 评价每个微粒的适应度;

步骤 3: 对每个微粒, 将其适应值与其经历过的最好位置 $pbest$ 作比较, 如果较好, 则将其作为当前的最好位置 $pbest$;

步骤 4: 对每个微粒, 将其适应值与全局所经历的最好位置 $gbest$ 作比较, 如果较好, 则重新设置 $gbest$ 的索引号;

步骤 5: 根据式(9)、(10) 变化微粒的速度和位置;

步骤 6: 如未达到结束条件(通常为足够好的适应值或达到一个预设最大代数), 则返回步骤 2。

文献[8]提出一种惯性权重方法, 将式(9)修改为:

$$V_{i+1} = \omega_i V_i + c_1 \cdot r_1 \cdot (P_i - X_i) + c_2 \cdot r_2 \cdot (P_g - X_i) \quad (11)$$

其中, ω_i 为权重, 在算法运行初期, ω_i 通常较大以保证对整个解空间有较强的搜索能力, 而在运行后期, 其值较小, 保证算法的收敛。

文献[9]中介绍了一种带收敛因子的粒子群优化算法, 其位置和速度更新公式如下:

$$V_{i+1} = \chi [V_i + c_1 \cdot r_1 \cdot (P_i - X_i) + c_2 \cdot r_2 \cdot (P_g - X_i)] \quad (12)$$

其中, $\chi = \frac{2k}{|2 - \varphi - \sqrt{\varphi^2 - 4\varphi}|}$ 为收敛因子, $\varphi = c1 + c2 >$

4。一种典型的设置为 $c1 = c2 = 2.05, k = 0.729$ 。该算法亦称为压缩因子法, 该方法在很多优化问题上都超过了惯性权重方法。所以, 本文采用了带收敛因子的粒子群优化算法。

3 采用混合微粒群算法的 GMM 参数优化

3.1 混合操作

利用 ML 估计作为算法的混合操作, 在每次迭代过程中, 利用式(5)~(8) 对每个粒子重估 2 次, 提高整个群体的适应度, 加快算法的收敛速度。

由于微粒群算法会破坏模型参数的限制条件, 因此在进行参数重估时, 需要对微粒群迭代操作后的混合权值进行归一化, 即 $p_i = p_i / \sum_{i=1}^M p_i$ 。

3.2 粒子群位置编码及初始化

3.2.1 粒子群位置编码

每个粒子的位置为: $\lambda = (\lambda_1, \lambda_2, \dots, \lambda_M)$, 其中 M 为 GMM 中单高斯函数 λ_i 的个数。 λ_i 可以表示为 $\lambda_i = (p_i, \mu_i, \Sigma_i)$, p_i 为单高斯函数 λ_i 的混合权值, $\mu_i = (\mu_{i1}, \mu_{i2}, \dots, \mu_{iD})$ 为均值向量, $\Sigma_i = (\Sigma_{i1}, \Sigma_{i2}, \dots, \Sigma_{iD})$ 为协方差阵的对角线元素组成的向量。

3.2.2 粒子群位置初始化

随机选取 M 个特征矢量作为初始聚类中心, 然后由 K- 均值法来初始化 GMM 模型参数, 再用式(5)~(8) 重估模型参数 3 次, 从而得到粒子群各微粒的初始位置。

由于随机产生的粒子其均值向量 μ_i 的无序性, 两个粒子的均值向量并不互相对应, 如果直接进行微粒群算法, 粒子不能有效利用全局最优位置, 寻优效果很差。为了有效地产生新的有意义的位置, 先对各粒子均值向量 μ_i 进行最近邻匹配。选择最优粒子的位置作为参考粒子, 设为 $\lambda^{gbest} = (\lambda_1^{gbest}, \lambda_2^{gbest}, \dots, \lambda_M^{gbest})$, 其中 $\lambda_i^{gbest} = (p_i^{gbest}, \mu_i^{gbest}, \Sigma_i^{gbest})$ 。对其他粒子

的位置与 λ^{best} 进行匹配, 其余任一粒子设为 $\lambda^{else} = (\lambda_1^{else}, \lambda_2^{else}, \dots, \lambda_M^{else})$, 其中 $\lambda_i^{else} = (p_i^{else}, \mu_i^{else}, \Sigma_i^{else})$ 。对 λ^{best} 中每个 λ_i^{best} , 选择 λ^{else} 中与 μ_i^{best} 距离最近的 μ_i^{else} , 将 λ_i^{else} 与 λ_i^{best} 配对。并且在位置配对过程中, 已经配对的 λ_i^{else} 不再参加后续的配对。这样, λ^{else} 与 λ^{best} 的混合分量最终可以两两配对。再将 λ^{else} 按照 λ_i^{else} 配对的顺序重新排列, 得到新位置。实验表明, 经过位置匹配的粒子群寻优能力更强。

3.3 混合微粒群算法的 GMM 模型优化

适应度函数选择为似然函数式(3), 通过混合微粒群算法寻求使得式(3)最大的模型参数, 具体步骤如下:

- 1) 初始化设置算法各参数, 包括粒子的初始速度、最大位置、最小位置、最大速度、最小速度;
- 2) 按 3.2.2 节方法随机产生初始粒子群;
- 3) 利用式(12)、(10)更新速度以及位置;
- 4) 利用 3.1 节的混合操作对粒子群进行参数重估计算。

表 1 各说话人在传统 ML 算法和混合微粒群算法训练得到的似然函数值

算法	说话人									
	1	2	3	4	5	6	7	8	9	10
传统 ML 算法	13.39	9.54	23.23	19.17	12.82	12.92	26.06	17.18	10.11	15.45
混合微粒群算法	14.31	10.43	24.71	20.10	13.65	13.57	27.14	17.81	11.19	16.32

图 1 给出了用混合微粒群算法对说话人 1 进行模型优化的进化过程(混合度为 16)。从曲线上可以看出混合微粒群算法具有较快的收敛速度。对其余说话人进行实验, 也可以得到同样的进化趋势。

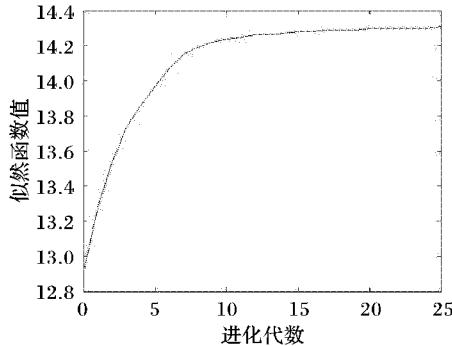


图 1 混合微粒群算法的进化过程曲线

为了比较本文提出的混合微粒群算法及传统的 ML 估计方法对说话人识别的影响, 我们分别使用混合度为 8、16 的 GMM 模型作为说话人的模型, 进行说话人辨认实验。表 2 给出了两种训练方法得到的系统误识率(语音识别长度为 1 s, 每人录制了 100 s 的语音进行实验, 共 1000 个统计样本)。

表 2 两种算法在不同混合度下的误识率比较 %

算法	混合度	
	8	16
ML	9.1	8.3
混合微粒群算法	7.5	6.8

从表 2 可以看出, 在不同混合度下本文提出的混合微粒群算法都能够得到较好的识别效果, 明显优于传统的 ML 估计方法。可见由本文的方法训练得到的 GMM 模型参数更优, 能够更好地描述说话人的特征分布。

5 结语

本文提出了一种新的高斯混合模型优化方法, 利用混合微粒群算法对 GMM 参数进行优化, 较好地克服了传统 ML 算

5) 如果已经满足迭代次数, 则停止迭代, 否则转 3)。

4 实验结果与分析

说话人辨认实验中训练和测试语音数据都在普通实验室环境下用普通声卡进行录制, 采样频率为 11025 Hz, 量化精度为 16 位。参加实验的人数共有 10 人, 训练语音长度为 20 s。采用汉明窗对语音信号进行分帧和加窗, 利用加重系数为 0.95 的一阶高通滤波器对信号进行预加重。提取 12 维的 Mel 倒谱系数作为说话人的特征参数。

PSO 的参数为: 微粒个数为 30, 最大进化代数为 25。ML 算法最大迭代次数为 100。表 1 给出了在 GMM 混合度为 16, 使用 10 个参考说话人时, 两种算法产生的似然函数值。从表 1 可以看出, 使用本文算法得到的似然函数值都大于 ML 算法。这说明混合微粒群算法产生的 GMM 参数要明显优于 ML 算法。

表 1 各说话人在传统 ML 算法和混合微粒群算法训练得到的似然函数值

法局部优化和收敛速度不稳定的缺点。利用本方法进行了与文本无关的说话人辨认实验, 对说话人的训练语音进行 GMM 参数估计, 得到说话人的模型参数, 取不同混合度时本文的方法均能得到更优的模型参数, 获得较好的识别结果, 优于传统 ML 算法, 改善了说话人识别的识别性能。

参考文献:

- [1] REYNOLDS D A, ROSE R C. Robust text-independent speaker identification using Gaussian mixture speaker models [J]. IEEE Transactions Speech Audio Processing, 1995, 3 (1): 72 – 83.
- [2] KENNEDY J, EBERHART R C. Particle swarm optimization [C]// Proceedings IEEE International Conference on Neural Networks. Perth, WA: IEEE Service Center, 1995: 1942 – 1948.
- [3] EBERHART R C, SHI Y. Particle swarm optimization: Developments, applications and resources [C]// Proceedings of IEEE International Conference on Evolutionary Computation. Washington: IEEE Press, 2001: 81 – 86.
- [4] 谢晓峰, 张文俊, 杨之廉. 微粒群算法综述 [J]. 控制与决策, 2003, 18(2): 129 – 134.
- [5] LIN LIN, WANG SHU-XUN. Genetic algorithms and fuzzy approach to Gaussian mixture model for speaker recognition [C]// Proceedings of 2005 IEEE International Conference on Natural Language Processing and Knowledge Engineering (NLP-KE '05). Washington: IEEE Press, 2005: 142 – 146.
- [6] HONG W Y, KWONG S. A genetic classification method for speaker recognition [J]. Engineering Applications of Artificial Intelligence, 2005, 18(1): 13 – 19.
- [7] 林琳, 王树勋. 基于自适应小生境混合遗传算法的说话人识别 [J]. 电子学报, 2007, 35(1): 8 – 12.
- [8] SHI Y, EBERHART R C. A modified particle swarm optimizer [C]// IEEE World Congress on Computational Intelligence. Proceedings of IEEE International Conference on Evolutionary Computation. Washington: IEEE Press, 1998: 69 – 73.
- [9] EBERHART R C, SHI Y. Comparing inertia weights and constriction factors in particle swarm optimization [C]// Proceedings of the IEEE Conference on Evolutionary Computations (ICEC). Washington: IEEE Press, 2000, 1: 84 – 88.