

一种基于递增估计 GMM 的连续优化算法

李 斌^{1,2)} 钟润添^{1,2)} 王先基²⁾ 庄镇泉²⁾

¹⁾(中国科学技术大学自然计算与应用实验室 合肥 233027)

²⁾(中国科学技术大学电子科学与技术系 合肥 233027)

摘 要 目前的分布估计算法(estimation of distribution algorithms)中概率模型的学习或多或少存在着对先验知识的依赖,而这些先验知识往往是不可预知的.针对这一问题,文中提出采用集成学习(ensemble learning)的思想实现 EDAs 中概率模型结构和参数的自动学习,并提出了一种基于递增学习策略的连续域分布估计算法,该算法采用贪心 EM 算法来实现高斯混合模型(GMM)的递增学习,在不需要任何先验知识的情况下,实现模型结构和参数的自动学习.通过一组函数优化实验对该算法的性能进行了考查,并与其它同类算法进行了比较.实验结果表明该方法是有效的,并且,相比其它同类 EDAs,该算法用相对少的迭代,可以得到同样或者更好的结果.

关键词 分布估计算法;连续优化;贪心 EM 算法;递增学习;高斯混合模型

中图法分类号 TP18

A Continuous Optimization Algorithm Based-on Progressive Estimation of Guassian Mixture Model

LI Bin^{1,2)} ZHONG Run-Tian^{1,2)} WANG Xian-Ji²⁾ ZHUANG Zhen-Quan²⁾

¹⁾(Nature Inspired Computation and Applications Laboratory, University of Science and Technology of China, Hefei 230027)

²⁾(Department of Electronic Science and Technology, University of Science and Technology of China, Hefei 230027)

Abstract In Estimation of Distribution Algorithms proposed in published literatures, learning of probabilistic model is dependent more or less on the prior-knowledge of the structure of model, which is unavailable in the process of evolutionary optimization. This paper proposes a new idea, which learns probabilistic model in EDAs by an approach similar to ensemble learning in machine learning, to implement automatic learning of both model parameter and model structure. According to this idea, a new EDAs for continuous optimization based on progressive learning of Gaussian Mixture Model is proposed. A greedy EM algorithm is adopted to estimation GMM in a progressive manner, which has the ability of learning the model structure and parameters automatically without any requirement of prior knowledge. A set of experiments on selected function optimization problems are performed to evaluate, and to compare with other EDAs, the efficiency and performance of the new algorithm. The experimental results confirm the feasibility and effect of the idea, and also show that, with a relative small number of generations, the new algorithm can perform better or as well as compared EDAs.

Keywords estimation of distribution algorithms; continuous optimization; greedy EM; progressive learning; Gaussian mixture model

1 引 言

分布估计算法 (Estimation of Distribution Algorithms, EDAs), 也称为基于概率模型的演化算法, 或学习优化算法^[1-2] 是一类新的进化算法, 它通过采用概率模型, 对每一代搜索得到的优良解进行学习, 进而探索问题的内在结构, 再利用该模型引导算法进行搜索, 以加速最优解的获得. 分布估计算法已被用于求解组合优化^[3-7] 和连续优化^[8-14] 两类主要的优化问题, 结果表明这是一类新的高效优化搜索算法.

概率模型的选择与学习是分布估计算法的核心. 简单概率模型 (如概率矢量模型^[6]) 由于无法有效描述变量之间复杂的联结关系 (Linkage), 使算法仅适用于一部分简单优化问题. 而复杂模型 (如 Gaussian Mixture Model^[15]、Bayesian Networks^[5]) 则可以较好地描述变量间的联结关系, 使算法具有更好的通用性. 但是, 基于有限样本的复杂模型学习本身又是一个相当复杂的问题, 因此, 如何设计更为有效的模型学习方法, 以进一步提高 EDA 的优化性能和计算效率, 即成为该领域的一个研究重点.

在连续优化研究领域, 目前已知的大多数 EDAs 采用高斯模型来建模解空间中“好”的区域. 早期的 EDAs^[8-10] 采用单一高斯模型来建模每个变量, 并假设变量间是相互独立的. 这种单变量概率模型在处理变量间存在复杂依赖关系的问题时表现不好. 高斯混合模型 (Gaussian Mixture Model, GMM) 能有效描述变量间的依赖关系, 因此被后来的连续优化 EDAs 广泛采用^[14-16]. GMM 的学习包括结构学习和参数学习两部分. 在已知 EDAs 中, 这两部分学习是分别进行的, 方法主要有基于聚类的算法和期望最大 (EM) 算法两大类. 采用聚类方法估计高斯混合模型一般需要一定的先验知识, 如聚类的个数^[17] 或不同类之间的最小距离^[15], 这在 EDAs 执行过程中往往无法预知. 文献^[16] 采用了另外一种聚类方法 RPCL (Rival Penalized Competitive Learning)^[19], 可以有效减少对先验知识的依赖. RPCL 先指定一个远大于实际聚类数目的“初始聚类数”, 然后在执行过程中根据一定的准则自动消除不存在的类, 直至达到预定的精度. RPCL 算法还是需要实际聚类数目的上界有一个预估, 以确定“初始聚类数”. 文献^[16] 中, 为了满足模型估计的需

要, 维持了一个较大的群体规模 (2000). 事实上, 所有采用聚类方法估计 GMM 的 EDAs 都把概率模型学习分成两步来进行: 先对所有筛选出的较优个体进行聚类, 然后再根据聚类结果估计各高斯混合分量和混合系数. 采用 EM 算法学习 GMM 存在以下两点不足: EM 算法需要事先知道 GMM 中高斯分量的个数, 而这在 EDAs 执行过程中是无法预知的; 另外 EM 算法对参数的初始值比较敏感, 容易陷入似然函数的局部最优. 是否存在一种方法, 能够不依赖任何先验知识, 在完成参数学习的同时实现模型结构的自动学习?

一直以来, 在机器学习有监督学习领域同样存在着模型结构学习的问题, 即在有限训练样本的情况下, 如何通过自动学习得到分类器的结构 (如神经网络的隐层节点数), 以获得满意的分类精度和泛化能力. 近年来发展出一类新的高效分类器构造方法——集成学习 (Ensemble Learning). 其基本思想是: 以某种方式将若干个分类效果不一定很好的“弱”分类器集成在一起, 实现一个具有很高分类精度和泛化能力的“强”分类器. 大量研究与应用表明, 这种多分类器集成 (Multiple Classifiers Ensemble) 的方法能够有效克服传统单分类器结构难确定的问题, 同时达到很高的分类精度和泛化能力. 在 EDAs 领域, 也存在着对模型描述精度和泛化能力两方面的要求. 文献^[20-21] 的研究结果表明, 采用集成学习的方法, 同样能够有效提高概率模型的描述精度和泛化能力. 此外, 本文引入集成学习的主要目的是实现模型结构与参数的同时学习, 进一步降低算法对先验知识的依赖, 以提高 EDA 的自适应能力.

贪心 EM 算法^[22] 是实现 GMM 综合学习的一种有效方法, 它以递增的方式线性组合多个“弱”模型 (单高斯分量), 进而得到一个具有很高似然度的“强”模型 (高斯混合模型), 使得最后所得到的混合模型可以有效地逼近任意复杂的概率分布. 本文将贪心 EM 算法引入 EDAs, 提出了一个基于递增估计 GMM 的 EDA, 与此前 EDAs 不同之处在于, 该算法可以在估计概率分布的过程中自动地学习模型结构和参数, 而不需要聚类和先验知识. 另外, 由于其贪心的本性, 使得整个 EDA 的时间复杂度也相对较低. 通过一组典型的测试函数对算法性能进行了考察, 并与其它几种同类算法进行了比较. 实验结果表明本文算法可以较少的迭代次数搜索到同样甚至更好的解.

2 递增学习 GMM 的贪心 EM 算法

在计算智能领域,有限混合概率模型被广泛用于描述存在不同统计分量的样本群体,一般表示如下:

$$f_k(x) = \sum_{j=1}^k \omega_j \phi(x; \theta_j) \quad (1)$$

其中, $\phi(x; \theta_j)$ 是第 j 个分量的数学描述, θ_j 是它的参数, ω_j 是第 j 个分量的混合系数, k 是混合模型中分量的个数。

对于高斯混合模型,第 j 个分量 $\phi(x; \theta_j)$ 是一个 d 维高斯概率密度函数:

$$\begin{aligned} \phi(x; \theta_j) &= \phi(x; \mu_j, \sigma_j) \\ &= (2\pi)^{-\frac{d}{2}} |\sigma_j|^{-\frac{1}{2}} e^{-\frac{1}{2}(x-\mu_j)^\top \sigma_j^{-1}(x-\mu_j)} \end{aligned} \quad (2)$$

其中, μ_j 和 σ_j 是第 j 个分量的均值和协方差矩阵,合起来用 θ_j 表示。

用 EM 算法估计 GMM 就是估计 $\{\omega_j, \mu_j, \sigma_j\}$ 。首先要设定混合分量的个数 k 、初始参数 $\{\omega_0, \mu_0, \sigma_0\}$ 和一组训练数据,通过不断更新参数 μ, σ 和 ω 来保证似然度单调地增加。文献[23]证明可以通过递增方式逐渐增加混合分量以最大化似然函数,如果新的混合分量是通过一种优化的方式来增加的,那么这种方式就可以学习式(1)所表示的任何混合分布。基于此,文献[22]提出了一种贪心 EM 算法,它从一个高斯分量开始,逐渐增加混合分量,直到满足预定终止判据。具体地,贪心 EM 算法包括两个主要迭代步骤^[22]:首先通过全局搜索找到新增分量的一组初始参数;然后通过局部搜索更新新分量的参数使得似然度最大。

在学习过程中,设某一新分量 $\phi(x; \theta)$ 被加入一个已有 k 个分量的混合密度函数 $f_k(x)$,则新的混合密度如下式所示:

$$f_{k+1}(x) = (1-\alpha)f_k(x) + \alpha\phi(x; \theta) \quad (3)$$

其中, $0 < \alpha < 1$ 。算法的目标是优化权重 α 和参数向量 θ ,以最大化如下对数似然:

$$\begin{aligned} L_{k+1} &= \sum_{i=1}^n \log f_{k+1}(x_i) \\ &= \sum_{i=1}^n \log [(1-\alpha)f_k(x_i) + \alpha\phi(x_i; \theta)] \end{aligned} \quad (4)$$

重复执行上述过程直到满足预定终止判据。对于高斯混合模型,最大似然的学习可以简化为迭代地学习只有两个分量的混合模型 $f_{k+1}(x)$ (如式(3)所示),其中,第一个分量是已有混合模型 $f_k(x)$,第

二个分量是新增分量 $\phi(x; \theta)$,并且第一个分量被设为不变,因此,算法的核心就是优化权重系数 α 和新增分量的参数 $\theta = \{\mu, \sigma\}$ 。文献[22]采用以下两步搜索来优化参数 α, μ 和 σ ,以最大化 L_{k+1} 。

首先,通过全局搜索找到一组新增分量的初始参数 μ, σ 和 α 。为此,在 $\alpha_0 = 0.5$ 处对 L_{k+1} 做二阶泰勒展开,并最大化关于 α 的二次函数,得到似然函数的一个近似:

$$\hat{L}_{k+1} = L_{k+1}(\alpha_0) - \frac{[L'_{k+1}(\alpha_0)]^2}{2L''_{k+1}(\alpha_0)} \quad (5)$$

这里 L'_{k+1} 和 L''_{k+1} 是 L_{k+1} 关于 α 的一次和二次微分。定义:

$$\delta(x, \theta) = \frac{f_k(x) - \phi(x; \theta)}{f_k(x) + \phi(x; \theta)} \quad (6)$$

那么 L_{k+1} 在 $\alpha_0 = 0.5$ 附近的局部最优可求得如下:

$$\hat{L}_{k+1} = \sum_{i=1}^n \log \frac{f_k(x_i) + \phi(x_i; \theta)}{2} + \frac{1}{2} \frac{\left[\sum_{i=1}^n \delta(x_i, \theta) \right]^2}{\sum_{i=1}^n \delta^2(x_i, \theta)} \quad (7)$$

此时 α 取值如下:

$$\hat{\alpha} = \frac{1}{2} - \frac{1}{2} \frac{\sum_{i=1}^n \delta(x_i, \theta)}{\sum_{i=1}^n \delta^2(x_i, \theta)} \quad (8)$$

然后估计新分量的均值 μ 和协方差矩阵 σ 的初值。从式(6)可得 \hat{L}_{k+1} 只依赖于 $\phi(x_i; \mu, \sigma)$,因为协方差矩阵 $\sigma = \sigma_d^2 \mathbf{I}$ (σ_d 是常数)是 μ 到样本 x_i 欧氏距离的一个函数,因此如果限制只为样本 x_j 全局搜索新的 μ ,那么对于所有的 μ ,求 \hat{L}_{k+1} 就需要在算法开始时计算所有样本之间的欧氏距离 $\|x_i - x_j\|$ 。

第二步,通过一种局部搜索来更新新分量的参数,这可以通过 partial-EM 算法^[22]来实现,因为在分配新的分量时,旧的混合模型 $f_k(x)$ 的参数是不变的,partial-EM 算法可以只更新 α 和新增分量的参数 μ 和 σ ,这个过程简单而且快速。

至此,学习高斯混合模型的贪心 EM 算法可以归纳如下。

算法 1. 贪心 EM 算法^[22]。

1. 初始化一个高斯分量, $\mu = E[x]$ 和 $\sigma = Cov(x)$;
2. 搜索所有样本 x_j ,为新分量找到合适的候选点 x_j^* ,设定 μ 为 x_j^* 使得最大化对数似然(式(7));
3. 用估计的 μ, σ 和 $\hat{\alpha}$ 值初始化 partial-EM 算法;
4. 应用 partial-EM 更新 μ, σ 和 $\hat{\alpha}$,直到满足收敛条件: $|L'_k/L_k - 1| < 10^{-6}$;
5. 如果 $L_{k+1} \leq L_k$,则算法终止,否则返回到步 2。

3 基于递增学习 GMM 的连续优化算法

在 EDAs 框架下,通过采用贪心 EM 算法学习高斯混合模型,得到基于递增学习 GMM 的 EDA (Progressive Estimation of Distribution Algorithms, PEDAs),如下所示.

算法 2. PEDAs.

1. 随机产生 N 个个体 $\rightarrow D_l$;
2. For $l=1,2,\dots$ 直到满足终止条件
 - (a) 从 D_{l-1} 选择 $S_c \leq N$ 个个体 $\rightarrow D_{l-1}^S$;
 - (b) 基于 D_{l-1}^S 使用贪心 EM 算法估计 GMM 的概率密

度函数 $\rightarrow p_l(x)$;

(c) 由 $p_l(x)$ 采样得到 N 个新个体 $\rightarrow D_l$;

3. End

由于通过贪心 EM 有效地实现了模型结构和参数的自动学习,而不需要关于混合模型分量个数的先验知识,也不需要聚类过程,这使得 PEDAs 不同于此前采用高斯混合模型的其他 EDAs.

图 1(a), (b), (c), (d) 演示了 PEDAs 在优化 ThreePeaks 函数时搜索区域的收敛过程. 可以看到,随着迭代代数的增加,搜索区域很快从整个解空间逐步收缩到有全局最优解或局部最优解的区域,并最终收缩到只有全局最优解的小区域.

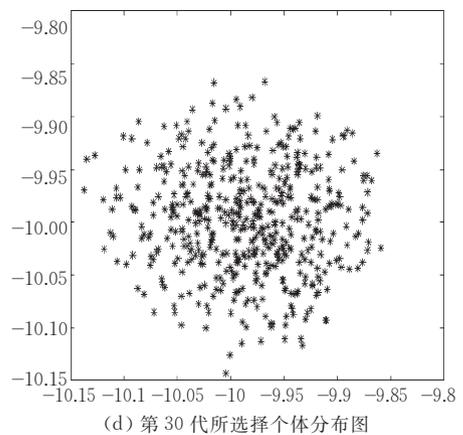
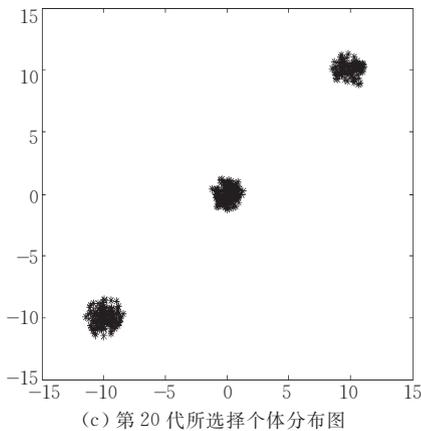
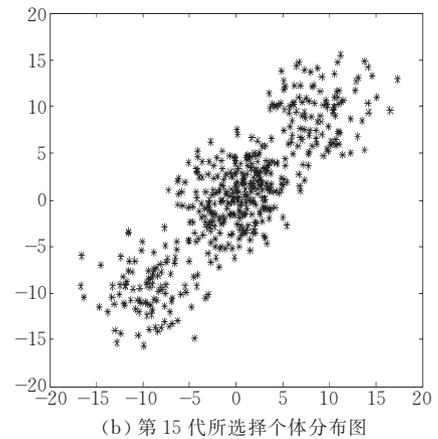
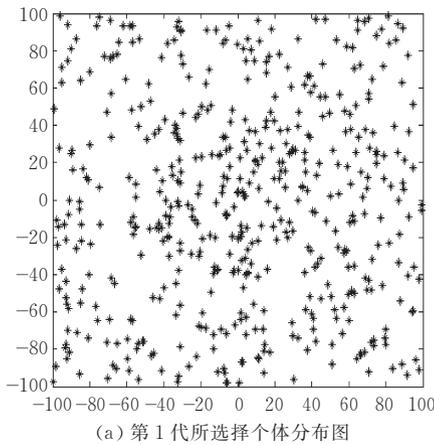


图 1 收敛过程

4 实验结果与分析

为了考察 PEDAs 的性能,并且和文献[16]中算法及此前一些同类算法进行比较,本文选择文献[16]中 5 个测试函数来测试本文算法的性能,算法中的参数设置和文献[16]取的一样.

函数 1: Shpere

$$F(\mathbf{x}) = \sum_{i=1}^n x_i^2 \quad (9)$$

函数 2: SumCan

$$\begin{cases} F(\mathbf{x}) = 1 / (10^{-5} + \sum_{i=1}^n |y_i|) \\ y_1 = x_1 \\ y_i = x_i + y_{i-1}, i \geq 2. \end{cases} \quad (10)$$

函数 3: TwoPeaks

$$F(\mathbf{x}) = \sum_{i=1}^2 \alpha_i N(\mathbf{x}, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \quad (11)$$

这里 $N(\mathbf{x}, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i)$ 是一个多变量正态分布, 均值 $\boldsymbol{\mu}$ 是 n 维向量, 协方差 $\boldsymbol{\Sigma}$ 是一个 $n \times n$ 矩阵, $\alpha_1 = 1000$, $\alpha_2 = 900$, $\boldsymbol{\mu}_1 = (-10, \dots, -10)$, $\boldsymbol{\mu}_2 = (10, \dots, 10)$, $\boldsymbol{\Sigma}_i (i=1, 2)$ 是单位矩阵.

函数 4: ThreePeaks

$$F(\mathbf{x}) = \sum_{i=1}^3 \alpha_i N(\mathbf{x}, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \quad (12)$$

这里除了与 TwoPeaks 一样的设置外, 加上 $\alpha_3 = 500$ 和 $\boldsymbol{\mu}_3 = (0, \dots, 0)$.

函数 5: Shekel

$$F(\mathbf{x}) = \sum_{i=1}^n [(x - a_i)(x - a_i)^T + c_i]^{-1} \quad (13)$$

其中, \mathbf{x} 和 \mathbf{a}_i 为 4 维向量,

i	$a_{ij}, j=1, \dots, 4$	c_i
1	2 2 2 2	0.1
2	4 4 4 4	0.2
3	8 8 8 8	0.2
4	6 6 6 6	0.4
5	3 7 3 7	0.4

实验中, 5 个测试函数的参数设置如表 1.

表 1 测试函数的设置

函数	维数	范围	类型	最优结果
Sphere	30	$[-100, 100]$	Min	0
SumCan	10	$[-0.16, 0.16]$	Max	100000
TwoPeaks	5	$[-100, 100]$	Max	10.1053
ThreePeaks	5	$[-100, 100]$	Max	10.1053
Shekel($n=5$)	4	$[0, 10]$	Max	10.1033

将 PEDA 与 UMDAc^[11], EGNA^[13] 和 CEGDA^[16] 进行比较. 每种算法群体规模和选择个体数的设置如表 2, 对于 CEGDA, $\alpha_c = 0.05$, $\alpha_r = 0.002$.

表 2 实验中算法的参数设置

算法	群体规模	选择个体数
UMDAc	1000	500
EGNA	1000	500
CEGDA	2000	500
PEDA	1000	500

我们对每个测试函数独立运行本文算法 30 次, 其它算法的结果取自文献[16], 实验结果如表 3 所示.

表 3 实验结果

函数	算法	演化代数	最好结果	平均结果	标准偏差
Sphere	UMDAc	200	1.88E-016	3.24E-016	5.59E-017
	EGNA	200	5.86E-010	1.20E-009	3.40E-009
	CEGDA	100	3.38E-008	3.41E-006	8.40E-007
	PEDA	200	1.01E-013	1.31E-013	2.15E-014

(续 表)

函数	算法	一般结果	最好结果	平均结果	标准结果
SumCan	UMDAc	200	698.72	221.771	116.101
	EGNA	200	100000	100000	0
	CEGDA	100	99834.5	99748.1	63.2197
	PEDA	100	99996.827	99995.308	0.765
TwoPeaks	UMDAc	400	10.1053	9.6327	0.1073
	EGNA	400	10.1053	9.8324	0.0828
	CEGDA	200	10.1053	10.0999	5.92E-003
ThreePeaks	UMDAc	400	5.05266	5.05266	8.88E-016
	EGNA	400	5.05266	5.05266	8.88E-016
	CEGDA	200	10.1053	10.1048	7.99E-004
	PEDA	100	10.1053	10.0988	0.01854
Shekel	UMDAc	400	5.1877	4.7331	0.7406
	EGNA	400	8.2036	4.9691	0.7786
	CEGDA	200	10.1033	10.1033	8.8818E-015
	PEDA	100	10.1033	10.0940	0.04227

由表 3 可看出来, 对于单峰函数 Sphere, 4 种算法都可以搜索到很好的解. 因为该函数中变量是相互独立的, 而 UMDAc 采用了单变量模型(假设变量间相互独立), 因此, 更适用于这个函数的求解.

对于函数 SumCan, 虽然也是一个单峰函数问题, 但是由于变量间有较强的依赖关系, 使得 UMDAc 算法的执行效果很差. 在几种 EDA 中, EGNA 由于采用了高斯网络作为概率模型, 能够描述变量间复杂的依赖关系, 因此执行效果最好. 而 PEDA 和 CEGDA 由于采用了高斯混合模型, 对多变量间的结构关系具有一定的描述能力, 因此执行效果也相当好, 求得结果十分接近最优解.

对于函数 TwoPeaks、ThreePeaks、Shekel, 都是多峰函数问题. 从实验结果中可以看出, PEDA 用相对少得多的迭代次数(只用了 100 代), 对几个函数运行的结果要比 UMDAc 和 EGNA 好, 与 CEGDA 的结果几乎一样好. PEDA 对于求解这类问题显示出了更好的性能, 收敛速度更快.

5 结 论

在分布估计算法 EDAs 中, 概率模型学习是非常重要的环节. 对于复杂问题, 需要用复杂的概率模型来描述变量之间的依赖关系. 而复杂模型的学习一般包括两个方面: 模型结构学习和模型参数学习, 在此前的 EDAs 中, 这两个方面通常是分开来进行的, 且对先验知识的依赖较大, 在样本有限的情况下, 使得模型学习的效率较低. 针对这一问题, 本文提出了一种新的、基于递增学习策略的连续域分布估计算法, PEDA, 通过采用贪心 EM 算法来学习 GMM, 实现了模型结构和参数的自动学习, 同时提

高模型的泛化能力,使得 PEDA 比此前的 EDAs 有更好的表现.

参 考 文 献

- [1] Mühlenbein H, Paaß G. From recombination of genes to the estimation of distributions I. Binary parameters//Proceedings of the 5th Parallel Problem Solving from Nature (PPSN V). Amsterdam, The Netherlands, 1998: 178
- [2] Pelikan M, Goldberg D E, Lobo F. A survey of optimization by building and using probabilistic models. IlliGAL Technical Report 99018, 1999
- [3] Mühlenbein H. The equation for response to selection and its use for prediction. *Evolutionary Computation*, 1997, 5(3): 303
- [4] Pelikan M, Mühlenbein H. The bivariate marginal distribution algorithm//Proceedings of the Soft Computing - Engineering Design and Manufacturing. London, 1999: 521
- [5] Pelikan M, Goldberg D E, Cantú-Paz E. BOA: The Bayesian optimization algorithm//Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-99). Orlando FL: Morgan Kaufmann Publishers, 1999, 1: 525
- [6] Harik G R, Lobo F G, Goldberg D E. The compact genetic algorithm. *IEEE Transactions on Evolutionary Computation*, 1999, 3(4): 287
- [7] de Bonet J S, Isbell C L, Viola P. MIMIC: Finding optima by estimating probability densities//Proceedings of the Neural Information Processing Systems. Cambridge, MA: The MIT Press, 1997, 9: 424
- [8] Sebag M, Ducoulombier A. Extending population-based incremental learning to continuous search spaces//Proceedings of the 5th Parallel Problem Solving from Nature (PPSN V). Amsterdam, The Netherlands, 1998: 418
- [9] Rudlof S, Köppen M. Stochastic hill climbing by vectors of normal distributions//Proceedings of the 1st Online Workshop on Soft Computing (WSC1). Nagoya, Japan, 1996
- [10] Servet I, Trave-Massuyes L, Stern D. Telephone network traffic overloading diagnosis and evolutionary computation techniques//Proceedings of the 3rd European Conference on Artificial Evolution (AE'97). Nimes, France, 1997: 137
- [11] Larrañaga P, Etxeberria R, Lozano J A, Peña J M. Optimization in continuous domains by learning and simulation of Gaussian networks//Proceedings of the 2000 Genetic and Evolutionary Computation Conference. Las Vegas, USA, 2000: 201
- [12] Mühlenbein H, Mahnig T, Rodriguez O. Schemata, distributions and graphical models in evolutionary optimization. *Journal of Heuristics*, 1999, 5(2): 215
- [13] Larrañaga P, Etxeberria R, Lozano J A, Peña J M. Optimization by learning and simulation of Bayesian and Gaussian networks. Department of Computer Science and Artificial Intelligence, University Basque Country, Spain; Technical Report EHUKZAA- IK-4/99, 1999
- [14] Bosman P A N, Thierens D. Expanding from discrete to continuous estimation of distribution algorithms: The IDEA//Proceedings of the Parallel Problem Solving from Nature (PPSN VI). Paris, 2000: 767
- [15] Bosman P A N, Thierens D. Advancing continuous IDEA's with mixture distributions and factorization selection metrics//Proceedings of the Optimization by Building and Using Probabilistic Models (OBUPM) Workshop at the Genetic and Evolutionary Computation Conference (GECCO-2001). San Francisco, CA, USA, 2001: 208
- [16] Lu Q, Yao X. Clustering and learning Gaussian distribution for continuous optimization. *IEEE Transactions on Systems, Man and Cybernetics-PART C: Applications and Reviews*, 2005, 35(2): 159
- [17] Pelikan M, Goldberg D E. Genetic algorithms, clustering, and the breaking of symmetry//Proceedings of the Parallel Problem Solving from Nature (PPSN VI), Paris, France, 2000: 385-394
- [18] Gallagher M, Fream M, Downs T. Real-valued evolutionary optimization using a flexible probability density estimator//Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-99), Orlando FL, 1999: 840
- [19] Xu L. Rival penalized competitive learning, finite mixture, and multisets clustering//Proceedings of the International Joint Conference on Neural Networks, Anchorage, Alaska, USA, 1998, II: 2525
- [20] Rosset S, Segal E. Boosting density estimation//Proceedings of the Proceedings of the 16th International Conference on Neural Information Processing Systems (NIPS). Vancouver, British Columbia, Canada, 2002: 641
- [21] Meek C, Thiesson B, Heckerman D. Staged mixture modeling and boosting//Proceedings of the 18th Conference on Uncertainty in Artificial Intelligence. Alberta, Canada, 2002: 335
- [22] Nikos V, Aristidis L. A greedy EM algorithm for Gaussian mixture learning. *Neural Processing Letters*, 2002, 15(1): 77
- [23] Li J Q, Barron A R. Mixture density estimation//Proceedings of the Neural Information Processing Systems 12. Cambridge, MA: The MIT Press, 2000: 279



LI Bin, born in 1970, Ph. D., associate professor. His research interests include computational intelligence and machine learning.

ZHONG Run-Tian, born in 1973, Master candidate. His research interests focus on evolutionary computation.

WANG Xian-Ji, born in 1980, Ph. D. candidate. His research interests focus on machine learning.

ZHUANG Zhen-Quan, born in 1938, professor, Ph. D. supervisor. His research interests include intelligent information processing.

The work of this paper belongs to a new branch of Evolutionary Computation called Estimation of Distribution Algorithms (EDAs), which introduces statistic learning explicitly into the process of evolutionary optimization. Various EDAs has been applied to optimization problems of both continuous and discrete domain, and has been proven to be a new promising evolutionary optimization method. One of the key problems of EDAs is the probability model learning. For complex optimization problems, complex probability models, i. e., Gaussian Mixture Model and Bayesian Networks, are necessary to describe the dependences between variables. But the learning of these complex models are usually nontrivial tasks. Particularly, methods adopted by previous EDAs require some prior knowledge on the structure of models, which, however, is not available in the process of evolutionary optimization. To deal with this problem, this paper introduces the idea of ensemble learning in machine learning into the EDAs to implement the automatic learning of model structure. The experimental results confirm the feasibility and effect of the idea. The experimental results also show that, with a relative small number of generations, the new algorithm can perform better or as well as compared EDAs.

The work of this paper is a part of the project of Natural Science Foundation of China entitled "Genetic Algorithms

based on quantum probability representation", and is also a part of the project of Natural Science Foundation of Anhui Province entitled "Estimation of Distribution Algorithms based on population strategy". The purpose of the two projects is to study the evolutionary algorithms based on various probability models, i. e., quantum probability model, etc.. The project team has got some progress on this purpose. They have proposed a genetic algorithm based on quantum bit probability model and a multi-objective genetic algorithm based on quantum bit probability model, and have applied the algorithms (also some other EDAs) to various application problems, such as 0-1 knapsack problems, function optimization, data mining, hardware/software co-design, etc.. Besides quantum bit probability model, another important quantum probability model is the Gaussian model, which can describe quantum system in more complex states. When developing evolutionary algorithm based on Gaussian probability models, the problem of efficient model learning arisen, the performance of the algorithm is greatly influenced by the model learning method. The work of this paper is motivated to deal with the problem correlated with learning Gaussian mixture model, to make the process more efficient and adaptive.