# Continuous Optimization based-on Boosting Gaussian Mixture Model

Bin Li
*Nature Inspired Computation and Applications Laboratory, University of Science and Technology of China, Hefei, China, 230026*
*binli@ustc.edu.cn*

Xian-ji Wang
*Department of Electronic Science and Technology, University of Science and Technology of China, Hefei, China, 230026*
*xjw@mail.ustc.edu.cn*

Run-tian Zhong
*Department of Electronic Science and Technology, University of Science and Technology of China, Hefei, China, 230026*
*rtzhong@mail.ustc.edu.cn*

Zhen-quan Zhuang
*Department of Electronic Science and Technology, University of Science and Technology of China, Hefei, China, 230026*
*zqzhuang@ustc.edu.cn*

## Abstract

*A new Estimation of Distribution Algorithm(EDA) based-on Gaussian Mixture Model (GMM) is proposed, in which boosting, an efficient ensemble learning method, is adopted to estimate GMM. By boosting simple GMM with two components, it has the ability of learning the model structure and parameters automatically without any requirement for prior knowledge. Moreover, since boosting can be viewed as a gradient search for a good fit of some objective in function space, the new EDA is time efficient. A set of experiments is implemented to evaluate the efficiency and performance of the new algorithm. The results show that, with a relatively smaller population and less number of generations, the new algorithm can perform as well as compared EDAs in optimizing multimodal functions.*

## 1. Introduction

Estimation of Distribution Algorithms (EDAs) are population-based search algorithms that use a probabilistic model to estimate the promising area of best solutions in the solution space, then use the estimated model to guide the search of optimal solutions[1]. The EDAs have been proved to be efficient in both combinatorial[2] and continuous optimization[3].

Most EDAs for continuous optimization use the Gaussian probability density function(pdf) to model the promising area of solution space. Early EDAs[3] model each variable with one Gaussian pdf and assume that there is no interaction among variables. These univariate models are not suitable for complex problems with interdependences among variables. Because Gaussian Mixture Model(GMM) can depict the interdependences among variables, it is adopted by many later continuous EDAs[4][5][6]. The learning task of GMM includes two parts: model structure learning and model parameter learning. In previous EDAs, the two parts are implemented separately and executed iteratively as two learning tasks with different evaluation criteria, which is time consuming. Usually clustering techniques are adopted by previous EDAs to estimate the Gaussian Mixture Model. The clustering algorithms usually require prior knowledge, either a predefined cluster number[6] or an minimal distance between different clusters[4]. But this prior knowledge can't be available during the procedure of EDAs. In [5], the authors used another clustering method, Rival Penalized Competitive Learning (RPCL) [7], which could detect the real number of clusters with little prior knowledge. In RPCL, the assigned number of clusters must be bigger than the true number of clusters, the low-bound and upper-bound of the number of clusters must be estimated beforehand, which is infeasible in EDAs. In fact, all the EDAs incorporating clustering techniques divide the probability density function

Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06)
0-7695-2521-0/06 $20.00 © 2006 IEEE

estimation into two steps: cluster the selected samples and estimate the distribution.

In this paper, a new continuous optimization algorithm based on Gaussian mixture model is proposed. Different from previous algorithms, the new algorithm adopts an efficient ensemble learning method, boosting, to estimate GMM[8]. The most important advantage is that it can implement the model structure and parameter learning simultaneously and automatically without any requirement for prior knowledge of data distribution. Our motivation is to construct an EDA that has fewer requirements on preset parameters of algorithm, and thus is more easy to use. By using boosting technique, the proposed EDA needn't do clustering and requires no prior knowledge. In addition, because boosting can be viewed as a gradient descent search for a good fit in function space[9], the new EDA is time efficient. The algorithm is evaluated and compared with other EDAs on a set of selected functions for efficiency and performance. The results show that, with a relatively smaller population, our algorithm can perform very well on all selected functions.

The rest of this paper is organized as follows. Section 2 introduces the continuous EDA based on boosting estimation of GMM. Section 3 presents the experiment results on a set of functions to assess the performance of the proposed EDA. Section 4 gives the conclusion and discussions.

## 2. Continuous Optimization based-on Boosting Estimation of GMM

Boosting is one of the most important developments in ensemble learning methodology, which incrementally builds a linear combination of "weak" learners to generate a "strong" learner. Mason et al. showed that boosting could be viewed as gradient descent on an appropriate cost function in a suitable inner product space[9]. Rosset et al. successfully applied the boosting methodology to the unsupervised learning problem of density estimation and proposed a general boosting density estimation framework[10].

Gaussian Mixture Model (GMM) has been widely used in modeling multi-modal distributions. Usually the EM (Expectation Maximization) algorithm is an effective method to find the optimal parameters of GMM. But EM algorithm still has some limitations:1) It assumes that the number of components is known, which is not the case in the estimation task in EDAs. 2) It is sensitive to the initial parameters, which makes it easily get trapped in local maxima of the likelihood function.

By using simple GMM consists of small number (2 in this paper) of components as the base weak hypothesis, S. Xubo et al. proposed a Boosting-GMM algorithm for density estimation as shown in Figure 1[8]. Step 2(b) is implemented as follows: first, we sort the original samples according to their weights $W_t$, then select some proportion of samples with largest weights to learn GMM.

---

Given dataset $X = \{x_1, x_2, \cdots, x_N\}$

1. Set $G_0(x)$ to uniform on the domain of $x$
2. For t = 1 to T
   (a) Set $W_t(x_i) = 1/G_{t-1}(x_i)$
   (b) Sample the original dataset according to $W_t$ and using EM algorithm to do GMM estimation on the sampled dataset, then output the result $g_t$
   (c) Let $G_t = (1-\alpha_t)G_{t-1} + \alpha_t g_t$
   $\alpha_t = \arg\min_\alpha \sum_i -\log((1-\alpha)G_{t-1}(x_i) + \alpha g_t(x_i))$
3. Output the final model $G_T$

---

**Figure 1. Boosting-GMM algorithm**

In EDA framework, by using the Boosting-GMM algorithm to estimate the probability density function, we get the Boosting-GMM based EDA (BGMMEDA), as shown in Figure 2. At every generation, the structure of a GMM is learned automatically along with the learning of the model parameters, no prior knowledge about the number of components and no clustering procedure are needed, that make BGMMEDA different from the previous EDAs that use Gaussian mixture model.

---

1. Generate $N$ individuals randomly $\rightarrow D_l$
2. For $l = 1, 2, \ldots$ until some termination criteria is reached
   a) Select $S_e \le N$ individuals from $D_{l-1} \rightarrow D_{l-1}^{S_e}$
   b) Using Boosting-GMM to estimate the probability density function based on $D_{l-1}^{S_e} \rightarrow p_l(x)$
   c) Sample $N$ individuals from $p_l(x) \rightarrow D_l$

---

**Figure 2. BGMMEDA**

To illustrate the ability of automatic structure learning of the algorithm, it is used to optimize a 2-peak function(*Func.3* in section 3), and the intermediate states of different generations are

monitored. Figure 3 shows the newly sampled individuals according to the GMM estimated at previous generations. It can be seen that:1) the number of components of GMM changes during the evolution process, 2) the true number of components can be captured correctly by the algorithm.

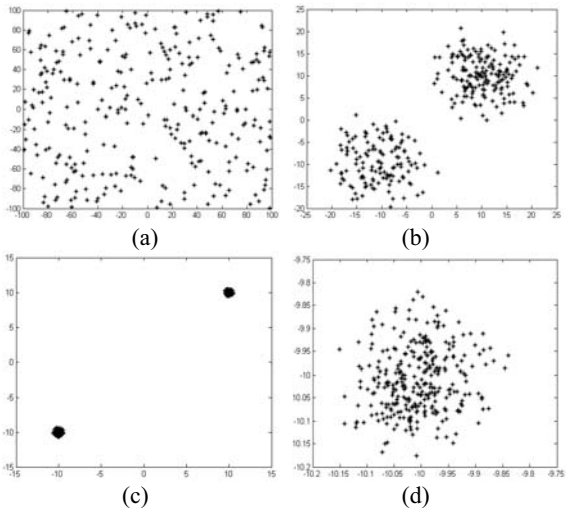


**Figure 3. Distribution of individuals at generation (a)1, (b)35, (c)45, (d)50**

## 3. Experimental Analysis

Five test functions are selected to evaluate the performance of BMMEDA. The settings of the functions are shown in Table 1.

*Func.1*: Sphere $F(\vec{x}) = \sum_{i=1}^{n} x_i^2$

*Func.2*: SumCan

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

*Func.3:* TwoPeaks $F(\vec{x}) = \sum_{i=1}^{2} \alpha_i N(\vec{x}, \mu_i, \Sigma_i)$.

*Func.4:* TreePeaks $F(\vec{x}) = \sum_{i=1}^{3} \alpha_i N(\vec{x}, \mu_i, \Sigma_i)$.

*Func.5*: Shekel (n = 5)

$$F(\vec{x}) = \sum_{i=1}^{n} [(\vec{x} - \vec{a}_i)(\vec{x} - \vec{a}_i)^T + \vec{c}_i]^{-1}$$

**Table 1. Settings for the five functions**

| | Dim | Domain | Type | Optimum |
|---|---|---|---|---|
| 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 | 4 | [0,10] | Max | 10.1053 |

Experimental results are compared with that of three other EDAs: UMDAc, EGNA, and CEGDA. We denote the number of individuals in each generation *Population*, the number of promising solutions selected from individuals in the population *Selection*. The value of *Population* and *Selection* of each algorithm are set as shown in Table 2. For CEGDA, $\alpha_c = 0.05$, $\alpha_r = 0.002$. The simple GMM that we used contains only 2 components.

**Table 2. Settings for the test EDAs**

| Algorithms | Population. | Selection. |
|---|---|---|
| UMDAc | 1000 | 500 |
| EGNA | 1000 | 500 |
| CEGDA | 2000 | 500 |
| BGMMEDA | 600 | 300 |

We run our algorithm 30 times independently on each function. The final results are the averages, which are shown in Table 3-7, it can be seen that with a smaller population and fewer generations, our algorithm can also perform very well compared with the other EDAs. The results of the three other EDAs are from [5].

Function Sphere is easy for all the four EDAs and the results are all good. Since, in this problem, the variables are independent between each other, UMDAc, which assumes single Gaussian distribution and considers no interdependencies, has the best performance.

Function SumCan is also a unimodal function, but the variables are interdependent. Among the four EDAs, UMDAc performs badly because it doesn't take any interdependencies into account. Our algorithm outperforms CEGDA with a smaller population. EGNA performs slightly better than our algorithm, but it takes more generations and uses a larger population. In experiment we find that if we search 150 generations, we can reach the global optimum every time.

TwoPeaks, TreePeaks and Shekel are all multimodal functions. They all have one global optimum with one or several local optima. From Table 5,6,7 we can see that BGMMEDA outperforms UMDAc and EGNA, and perform as well as CEGDA. BGMMEDA can reach the global optimum almost in every run, especially on the TwoPeaks function.

**Table 3. Experimental Results (Sphere)**

| Algorithm | Gen | Best | Mean | Std |
|---|---|---|---|---|
| UMDAc | 200 | 1.88e-16 | 3.24e-16 | 5.59e-17 |
| EGNA | 200 | 5.86e-10 | 1.20e-9 | 3.40e-9 |
| CEGDA | 100 | 3.38e-8 | 3.41e-6 | 8.40e-7 |
| BGMMEDA | 100 | 8.32e-12 | 0.00161 | 0.00564 |

**Table 4. Experimental Results (SumCan)**

| Algorithm | Gen | Best | Mean | Std |
|---|---|---|---|---|
| UMDAc | 200 | 698.72 | 221.771 | 116.101 |
| EGNA | 200 | 100000 | 100000 | 0 |
| CEGDA | 100 | 99834.5 | 99748.1 | 63.2197 |
| BGMMEDA | 100 | 99925 | 99875 | 24.174 |

**Table 5. Experimental Results (TwoPeaks)**

| Algorithm | Gen | Best | Mean | Std |
|---|---|---|---|---|
| UMDAc | 400 | 10.1053 | 9.6327 | 0.1073 |
| EGNA | 400 | 10.1053 | 9.8324 | 0.0828 |
| CEGDA | 200 | 10.1053 | 10.0999 | 5.92e-3 |
| BGMMEDA | 100 | 10.1053 | 10.1053 | 6.37e-5 |

**Table 6. Experimental Results (ThreePeaks)**

| Algorithm | Gen | Best | Mean | Std |
|---|---|---|---|---|
| UMDAc | 400 | 5.05266 | 5.05266 | 8.88e-16 |
| EGNA | 400 | 5.05266 | 5.05266 | 8.88e-16 |
| CEGDA | 200 | 10.1053 | 10.1048 | 7.99e-4 |
| BGMMEDA | 100 | 10.1053 | 10.1030 | 0.01022 |

**Table 7. Experimental Results (Shekel)**

| Algorithm | Gen | Best | Mean | Std |
|---|---|---|---|---|
| UMDAc | 400 | 5.1877 | 4.7331 | 0.7406 |
| EGNA | 400 | 8.2036 | 4.9691 | 0.7786 |
| CEGDA | 200 | 10.1033 | 10.1033 | 8.88e-15 |
| BGMMEDA | 100 | 10.1033 | 10.1031 | 8.91e-4 |

## 4. Conclusion

Learning probabilistic model is the key step in EDAs. For complex problems, complex models that can describe the interdependences among variables are necessary. For these complex models, the learning task includes two aspects: parameter learning and model structure learning. In previous EDAs, the two aspects are usually implemented separately, which makes the model learning an inefficient process. In this paper, a new EDA for continuous optimization is proposed. The novel contribution is that it introduces the boosting technique into the density estimation in EDA. The most important advantage is that it can implement the model structure and parameter learning simultaneously without any requirement for prior knowledge of data distribution.

## 5. Acknowledgement

The work is partially supported by the Natural Science Foundation of China(NSFC) under grand No.60401015, and the Natural Science Foundation of Anhui province under grand No. 050420201.

## References

[1] H. Mühlenbein and G. Paaß. "From recombination of genes to the estimation of distributions I. Binary parameters," *Parallel Problem Solving from Nature* – PPSN V, pages 178-187. Springer, 1998

[2] G.R. Harik, F.G. Lobo, and D.E. Goldberg, "The compact genetic algorithm," in *Proceedings of the International Conference on Evolutionary Computation (ICEC)* 1998, pp. 523-528

[3] M. Sebag and A. Ducoulombier, "Extending population-based incremental learning to continuous search spaces," in *Parallel Problem Solving from Nature—PPSN V*, 1998, pp. 418–427.

[4] P.A.N. Bosman and D. Thierens, "Advancing continuous IDEA's with mixture distributions and factorization selection metrics," in *Proc. Optimization by Building and Using Probabilistic Models (OBUPM) Workshop at the Genetic and Ecolutionary Computation Conf. (GECCO- 2001)*, M. Pelikan and K. Sastry, Eds., San Francisco, CA, 2001, pp. 208–212.

[5] Qiang Lu and Xin Yao. "Clustering and Learning Gaussian Distribution for Continuous Optimization",in IEEE Transactions on Systems,Man and Cybernetics-PART C:Applications and Reviews,VOL.35,NO.2,May 2005.

[6] M. Pelikan and D. E. Goldberg, "Genetic algorithms, clustering, and the breaking of symmetry," in *Proc. Parallel Problem Solving from Nature— PPSN VI*, Paris, France, 2000, pp. 385–394.

[7] L. Xu, "Rival penalized competitive learning, finite mixture, and multisets clustering," in Proc. Int. Joint Conf. Neural Networks, vol.II, pp. 2525-2530, 1997.

[8] Xubo Song, Kun Yang and Misha Pavel, "Density Boosting for Gaussian Mixtures," ICONIP, pp. 508-515, 2004.

[9] L. Mason, J. Baxter, P. Bartlett, "Boosting Algorithm as Gradient Descent in function space," *Advances in Neural Information Processing* Systems12, pp. 512-518, 1999.

[10] S. Rosset and E. Segal, "Boosting density estimation," *Advances in Neural Information Processing* Systems 15, 2002.