An analysis of the performance of the Mixture of Trees Factorized Distribution Algorithm when priors and adaptive learning are used

Roberto Santana[1]

Institute of Cybernetics, Mathematics, and Physics (ICIMAF),
Calle 15, entre C y D, Vedado, C-Habana, Cuba
{Rsantana}@cidet.icmf.inf.cu

Abstract. This paper analyzes the behavior of the Mixture of Trees Factorized Distribution Algorithm (MT-FDA) when priors are incorporated. It is shown that the addition of priors provokes a mutation like effect during the search. Adaptive priors that relate the rate of mutation to the quality of the search are also introduced. Additionally, the learning step of the MT-FDA is changed to avoid the overfitting of data. The results of the experiments show that our proposals improve the trade off between exploration and exploitation displayed by the MT-FDA.

1 Introduction

Population Based Search Methods that use Selection (PBSMS) are non deterministic heuristic search strategies, commonly used as optimization methods. Their main characteristics are: They use a population of points instead of a single point to conduct the search. In every iteration (usually called generation) a subset of points is selected, and by applying some operators a new population is created. In this way the algorithm iterates (evolves) until one stop condition is satisfied.

Genetic Algorithms [2] are members of a subclass of PBSMS where recombination and mutation operators are applied to the selected set of individuals to obtain the new population. Another class of PBSMS comprises those algorithms characterized by the use of probabilistic modeling of the information contained in the selected set, instead of the genetic operators. These algorithms are the focus of this paper.

In this paper an Estimation Distribution Algorithm is any evolutionary algorithm that uses the estimation of probability distributions to improve the search. Although no all the proposals fit well in the EDAs scheme this conceptual framework has been taken before [3] as a model to study the algorithms under analysis. A special subclass of EDAs will group the algorithms that use factorizations of the probability distribution. This subclass of algorithms will be called Factorized Distribution Algorithms (FDAs)[1].

[1] In the literature the term FDA is usually given to a Factorized Distribution Algorithm that uses the same factorization along the evolution

In this paper we concentrate on the Mixture of Trees FDA (MT-FDA) that was introduced in [12]. The algorithm uses as its probabilistic model a mixture of trees. Our main contribution in [12] was to show that a FDA that uses as its probabilistic model a mixture of trees can perform better than other simpler FDAs and be competitive, and some times superior to FDAs that use more complex probabilistic models. In the present study we show how the results of the MT-FDA can be improved by adding a mutation like effect using priors, and modifying the algorithm used for learning the probabilistic model.

Along the paper we will concerned with the maximization of a function $f : B^n \rightarrow R$, and $X \in B^n$ is a set of random binary variables. We will use x_i ($x_i \in \{0,1\}$) to denote a value of X_i, the i-th component of X. $P = \{x^1, \ldots x^N\}$ will denote a set of vectors where N is the number of binary vectors in the set.

The paper's outline is as follows: In the next section we will present the mixture of trees as a probabilistic model and the MT-FDA. In section 3 we discuss the relationship that exits in EDAs between the goals of exploration and exploitation and probabilistic modeling. Section 4 describes how priors can be inserted in the MT-FDA to obtain a mutation like effect. Adaptive priors are defined for the first time in EDAs. Section 5 presents a modification to the learning algorithm based on halting the learning procedure to avoid overfitting. Experimental results on the application of the proposed modifications are shown in section 6. The conclusions of the paper and future research trends are presented in section 7.

2 Trees and Mixture of Trees Models

Modeling by finite mixture of distributions [1] concerns modeling a statistical distribution by a mixture (or weighted sum) of other distributions. The mixture of trees was introduced as a probabilistic model in [5] where its usefulness as a density estimator algorithm and a classification tool was demonstrated on a set of problems. Mixture of trees, and in general mixture distributions have a number of distinctive attributes that make them particularly appealing for their use in the framework of FDAs. Maybe the most important is the possibility of representing, condensed in just one model, different patterns of interactions among the variables of the problem. In Bayesian Networks (BN) the change in one variable's value can determine changes only in the parameters of other variables, no in their structural relation (edges or arcs in the graphical representation). In mixture of finite distributions the structure of dependencies among a set of variables can change depending on the values of the choice variable they depend on.

Now the probabilistic models are formally introduced. We will utilize the same notation used in [5]. A probability distribution T that is conformal with a tree is defined as:

$$T(x) = \prod_{v \in V} T_{v|pa(v)}(x_v|x_{pa(v)}) \; . \tag{1}$$

The distribution T itself will be called a tree when no confusion is possible. The graph (V, E) represents the structure of the distribution T. A mixture of trees is defined to be a distribution of the form:

$$Q(x) = \sum_{k=1}^{m} \lambda_k T^k(x) \ . \tag{2}$$

with $\lambda_k \geq 0$, $k = 1, \ldots, m$, $\sum_{k=1}^{m} \lambda_k = 1$.

The tree distributions are the mixture components, and the λ_k are called mixture coefficients. A mixture of trees can be viewed as containing an unobserved choice variable z, which takes values $k \in \{1, \ldots, m\}$ with probability λ_k. Conditioned on the value of z the distribution of the visible variables V is a tree. The m trees may have different structures and different parameters.

Algorithm 1: **MT-FDA**

1 Set $t \Leftarrow 0$. Generate $N \gg 0$ points randomly.
2 **do** {
3 Select a set S of $k \leq N$ points according to a selection method.
4 Calculate a mixture of trees Q that approximates the distribution of points in S.
5 Generate new points sampling from Q.
6 $t \Leftarrow t + 1$
7 } **until** Termination criteria are met.

Above, the pseudo-code of the MT-FDA is presented. The first population of the algorithm is generated randomly. From the current population, a subset of points is selected. A mixture of trees Q that fits the selected set is found using the Iterated Estimation Maximization (IEM) algorithm [5]. New points are generated by sampling from Q. Different replacing strategies can be used to combine points from the current population and new generated points in the next population. When proportional or Boltzmann selection are used the step 3 of algorithm 1 is not needed and in the step 4 of the algorithm the model is calculated using the probabilities of selection associated to each point in the population by the selection method. This can be done because graphical models can be learned too from a joint probability distribution.

3 Exploration and Exploitation in EDAs

When generating a new population the two traditional goals of an efficient search, exploitation and exploration of the search space, have to be accomplished. In the case of FDAs the achievement of these goals will depend on the power of expression of the probabilistic model, the way it has been learned and the method used to sample points from it. In principle, a good probabilistic model has to

be able to exploit the good points that have been already identified, in our case represented by the set of selected points and possibly some elite points, but also able to explore new areas of the search space where eventually are located better points.

In the model learning phase of Bayesian FDAs [11, 3] the quality of the model is usually evaluated using different measures of the likehood of the data of the selected points. It is clear that a probabilistic model with a maximal likehood of the data can be useless to explore new areas of the search space. While an exact learning of the probabilistic model from the data can benefit exploitation, if learning is pursued beyond certain limits the phenomenon of overfitting arises. An overfitted model can less likely generate, during the sampling step, points that belong to unexplored areas of the search space. Different alternatives have been proposed to cope with this problem in probabilistic modeling. In the context of FDAs we identify the following possible solutions to avoid overfitting:

- To allow the learning algorithm to learn a very exact (possibly overfitted) model and modify this model after.
- To stop the learning process when a predefined quality in the approximation of the data has been achieved.
- To do the search of the probabilistic model in a constrained space of (simple) models.

The first idea has been implemented for the Univariate Marginal Distribution Algorithm (UMDA) [10], the Factorized Distribution algorithm with a fixed model (FDA^*) [9] and Learning FDA (LFDA) [8] by means of considering Bayesian priors during the learning of the model [4]. In the implementation, the learned model's parameters are changed after the structure of the model has been learned. The effect of this type of priors is similar to the effect of mutation in Genetic Algorithms. The authors show that, also for FDAs, mutation increases in many cases the performance of the algorithm. This can be explained by the important role played by mutation in avoiding premature convergence and allowing the exploration of new regions during the search. In our first implementation of the MT-FDA we did not consider the use of priors. One of the improvements incorporated to the algorithm in this paper is the use of priors.

Although the use of priors can in general improve the results of the search, it has been shown in [4] that it can decrease the quality of the results for certain functions. This fact makes convenient to investigate the second alternative mentioned before, i.e. to stop the learning process when a predefined quality in the approximation of the data has been achieved. In section 5 we analyze this issue.

4 Use of Priors

In [4] it is explained how to introduce mutation into FDAs and how to choose the mutation rate based on a theoretically derived result. In the paper it is shown that mutation increases in many cases the performance of the algorithms

and decreases the dependence on the correct choice of the population size. The authors state that:

Let τ denote the parameter for truncation selection, I_τ is the strength of selection that depends on τ [7], and M is the size of selected set. When r is the prior for a single binary variable, the prior r' for a factor $p(x_1, \ldots, x_k)$, and the prior r^* for a factor $p(x_k|x_1, ..., x_{k-1})$ should be

$$r' = r^* = 2^{-(k-1)} \cdot r \; . \tag{3}$$

Using $r'_i = 2^{-k_i - 1} r$ with $r = \frac{I_\tau M}{n}$ is a reasonable choice for the Bayesian prior for truncation selection [4].

We use the proposed prior in our experiments adapting it to the case of the MT-FDA where the marginals used can have only order one or two. Additionally, we have introduced an adaptive prior that can be different for each tree. This prior increases the probability of the appearance of events with zero probability proportionally to the approximation of the data, and inversely proportionally to the coefficient of the tree (i.e. its weight in the mixture approximation).

Let us introduce some new notation.

We define D^c as the set formed by exactly one copy of all the different vectors in $D = \{x_1, x_2, \ldots, x_M\}$. This means $x \in D^c \Longrightarrow x \in D$ and $x_i, x_j \in D^c \Longrightarrow x_i \neq x_j$. We denote as $\tilde{P}^k(D^c)$ to the sum of the probabilities assigned by the tree T^k to all the points in D^c. Respectively $\tilde{P}(D^c)$ is the sum of the probabilities assigned by the mixture to all the points in D^c.

$$\tilde{P}^k(x_i) = \lambda_k T^k(x_i) \; . \tag{4}$$

$$\tilde{P}(D^c) = \sum_{x_i \in D^c} \sum_{k=1}^{m} \tilde{P}^k(x_i) \; . \tag{5}$$

$\tilde{P}$ is not a probability in D^c (e.g. $\tilde{P}(D^c) \neq 1$). Only when all the points of the search space are in D^c, $\tilde{P}(D^c) = 1$. The prior r^k for the k tree is defined as follows.

$$r^k = \frac{\tilde{P}(D^c) M}{\lambda^k n} \tag{6}$$

Finally, we substitute r by r^k in 3. This choice of the priors is related to the adaptive mutation schedules used in Genetic Algorithms.

5 Adaptive Learning

We address now the problem of how precise has to be the approximation given by the mixture of trees in order to avoid overfitting. The IEM algorithm, used in [5] to learn the mixtures, improves the likehood of the data in each iteration by modifying the structure and parameters of the mixture. In the experiments done in [13] we have confirmed that stopping the learning process just when no

improvement is detected in the likelihood can provoke a premature convergence of the MT-FDA because the learned model overfits the data.

Now we present a solution to the problem of determining the extent of learning. As a measure for assessing the quality of the learned model in every step of the mixture of trees learning algorithm we will use $\tilde{P}$, previously defined in equation 5. $\tilde{P}$ gives an idea of which is the probability given by the model to points that are not in the data set D, $\tilde{P}(x \in X, x \notin D) = 1 - \tilde{P}(D^c)$. So, in principle we can run the learning algorithm while the probability given by the model to points that are in D does not exceed a given parameter μ. μ defines a measure of overfitting, or in terms of a population based search method, a measure of exploitation

When the model is a bad approximation of the data, or the data points are a very small sample of the state space $\tilde{P}(D^c) \approx 0$. When the model completely overfits the data points $\tilde{P}^c = 1$. To summarize, we present the modification done to the IEM learning algorithm: The calculation of μ is incorporated and the condition $\tilde{P}(D) \geq \mu$ is added to the termination criteria.

6 Experiments

In our experiments we will evaluate the impact of all the introduced proposals in the performance of the MT-FDA. We will focus on the analysis of the algorithm's behavior when the complexity of the model is constrained using μ as a parameter. Additionally we study the influence of the number of trees and the use of priors. The test functions used were: $f_{deceptive3}$, $f_{deceptive4}$, F_{IsoP}, $f_{3deceptive}$, and $Isotorus$. Deceptive functions have served to study the performance of GAs in the presence of deception. The F_{IsoP} and $Isotorus$ functions respectively allow the study of problems with a chainlike and grid based structure. The interested reader is referred to [3] for an account of the performance of other FDAs for these functions. All the functions are defined below and in every case $u = \sum_{i=1}^{n} x_i$.

Function f_{dec}^3:

$$f_{dec}^3 = \begin{cases} 0.9 \; for \; u = 0 \\ 0.8 \; for \; u = 1 \\ 0.0 \; for \; u = 2 \\ 1.0 \; for \; u = 3 \end{cases} . \tag{7}$$

Function $f_{3deceptive}$:

$$f_{3deceptive}(x) = \sum_{i=1}^{i=\frac{n}{3}} f_{dec}^3(x_{3i-2}, x_{3i-1}, x_{3i}) \; . \tag{8}$$

Function general deceptive of order k, $fdecK$:

$$fdecK(x) = \begin{cases} k-1 & for\ u=0 \\ k-2 & u=1 \\ & \cdots \\ k-i-1 & u=i \\ & \cdots \\ k \cdot n & for\ u=k \end{cases} . \tag{9}$$

Function $f_{deceptiveK}$:

$$f_{deceptivek}(x) = \sum_{i=1}^{i=\frac{n}{k}} fdecK(x_{ki-k+1}, \ldots, x_{ki}) . \tag{10}$$

Function $IsoPeak$:

x	00	01	10	11
$IsoC_1$	m	0	0	$m-1$
$IsoC_2$	0	0	0	m

(11)

$$F_{IsoP}(x) = \sum_{i=1}^{m-1} IsoC_1(x_{2i-1}, x_{x2i}) + IsoC_2(x_{2m-1}, x_{x2m}) .$$

with $n = m + 1$.
Function $Isotorus$:

u	0	1	2	3	4	5
$IsoT_1$	m	0	0	0	0	$m-1$
$IsoT_2$	0	0	0	0	0	m^2

(12)

$$F_{IsoTorus}(x) = IsoT_1(x_{1-m+n}, x_{1-m+n}, x_1, x_2, x_{1+m}) + \sum_{i=2}^{n} IsoT_2(x_{up}, x_{left}, x_i, x_{right}, x_{down}) .$$

where x_{up}, etc., are defined as the appropriate neighbors, wrapping around.

In the experiments the number of variables is fixed to 36, as well as the selection algorithm (Truncation selection with truncation parameter 0.15).

6.1 Numerical Results

Table 1 presents the results of the MT-FDA with 6 trees when the parameter μ is changed and two different types of priors are used: the recommended and adaptive priors. The table shows the number of trials (of 100) where the optimum was found and the average number of evaluations. For the deceptive functions it is evident the influence of parameter μ in the behavior of the algorithm. For these functions a low value of μ gives better results than when a higher one is used. For the Iso functions the same behavior can not be clearly appreciated, particularly for function $F_{IsoTorus}$.

For deceptive functions the priors increase the number of times the optimum is found, although in the case of the adaptive prior there is an interaction with the parameter μ, when μ is high the successful rate diminishes. The increment in the number of successful trials reached by using priors is achieved at the cost of an increment in the number of function evaluations. In the case of the Iso functions there is not a significative difference when priors are added. Even more, in the case of function $IsoPeak$ results deteriorate.

In order to create an algorithm of practical use we would like to reduce as much as possible the number of parameters or to find rules of thumb that can be used to assign their values. We hypothesize that a good choice for parameter μ is around 0.2. Notice that the case when $\mu = 1$ is equivalent to run the IEM until no further improvement in the likehood is achieved (as we did in previous implementations). In the first generations, when the data is very diverse a good approximation is very hard to reach. The same happens when a high prior is used. In these cases the learning algorithm will stop when no further improvement in the likehood is achieved, or the improvement is under a given threshold (we use 0.005 as the threshold value).

We evaluate the validity of our proposal in the next experiment. For the 5 functions considered before and different values of the number of trees we run the MT-FDA and determine the value μ in $[0.2, 0.4, 0.6, 0.8, 0, 1.0]$ for which best results are achieved. When the same successful rate is obtained for more than one value of μ we choose the smallest μ . Table 2 presents the results that confirm that our choice is correct even if there exist exceptions. In most of the cases the 0.2 value was the best or among the best assignments for μ. This experiment also allows to evaluate the influence of the number of trees in the performance of the MT-FDA for the different functions.

When priors are not used the algorithm is very sensitive to an increment in the number of trees. If priors are incorporated the succesful rate is the same, or can even increase when the number of trees is augmented, but at the expense of a higher number or function evaluations. The exception is function F_{IsoP}, for this function results remarkably deteriorate with the increment in the number of trees. F_{IsoP} is an example of a function for which a simple model gives better results. Also in the case of $f_{deceptive3}$ it can be appreciated that a simple model can find the optimum with a lesser number of evaluations. In the case of the other three functions results improve when mixtures are used.

7 Conclusions and Future Research

In this paper we have introduced a mutation like effect in the MT-FDA by using priors. We have proposed a way for doing adaptive mutation by relating the mutation rate of the FDA to the quality of the approximation and the strength of the components of the mixture model. This is an example of the number of interesting possibilities that arise in the integration of the theory of Graphical Models and Evolutionary Computation. The results of the experiments show that the changes introduced to the MT-FDA improve its performance.

Table 1. Results of the MT-FDA with 6 components in the optimization of different functions, $n = 36$, $T = 0.15$

$Function$	$Popsize$	$NoPrior$		$BestPrior$		$Adap.Prior$	
	μ	$succ.$	$eval.$	$succ.$	$eval.$	$succ.$	$eval.$
$f_{deceptive3}$	0.2	80	4807	100	5271	100	5579
	0.4	90	4847	99	5310	100	5963
	0.6	82	4979	99	5296	100	5886
	0.8	67	4967	100	5683	99	7831
	1.0	41	4928	98	5643	65	7034
$f_{deceptive4}$	0.2	22	5497	63	7135	87	11257
	0.4	17	5839	62	8129	84	12516
	0.6	6	6059	61	8377	75	14782
	0.8	3	6059	55	8821	51	17983
	1.0	2	3845	40	9490	3	6525
$f_{3deceptive}$	0.2	24	10657	62	14454	98	17677
	0.4	17	11283	63	14780	99	16964
	0.6	6	10490	62	15389	90	21490
	0.8	1	7993	47	16112	65	28418
	1.0	1	17983	45	16694	9	27529
F_{IsoP}	0.2	28	5888	15	6861	30	6894
	0.4	16	5808	17	7640	23	7949
	0.6	21	5852	15	6661	17	6760
	0.8	24	5828	19	6994	16	6744
	1.0	21	5614	8	7243	13	5226
$F_{IsoTorus}$	0.2	93	5436	100	6134	97	5706
	0.4	95	5511	95	6268	95	5585
	0.6	93	5404	99	6348	94	5538
	0.8	92	5441	100	6314	87	5627
	1.0	89	5456	98	6015	90	5507

Table 2. Results of the MT-FDA with different number of components in the optimization of different functions $n = 36$, $T = 0.15$ when the number of trees is increased

$Function$	$N.Trees$	$NoPrior$			$BestPrior$			$Adap.Prior$		
		$Best\mu$	$succ.$	$eval.$	$Best\mu$	$succ.$	$eval.$	$Best\mu$	$succ.$	$eval.$
$f_{deceptive3}$	2	0.4	100	3859	0.2	100	4440	0.2	100	3851
	4	0.2	98	4309	0.2	100	4810	0.2	100	4705
	12	0.2	66	5222	0.2	100	5858	0.2	100	6753
$f_{deceptive4}$	2	0.4	47	4760	0.8	71	6164	0.2	67	5916
	4	0.2	34	5490	0.2	70	7400	0.4	72	8185
	12	0.2	15	6059	0.2	75	8035	0.2	88	12916
$F_{IsoTorus}$	2	0.2	94	5549	1.0	98	6515	0.2	92	5580
	4	0.4	97	5490	0.6	100	6345	0.4	98	5710
	12	0.2	93	5608	0.4	99	6358	0.2	99	6056
F_{IsoP}	2	0.2	77	5100	0.2	41	5654	1.0	70	5053
	4	0.2	45	5573	0.4	32	6089	0.6	38	5653
	12	0.2	23	6386	0.2	21	8183	0.2	18	8992
$f_{deceptive3}$	2	0.2	37	9370	0.2	66	12034	1.0	63	11053
	4	0.2	23	10338	0.2	66	13835	0.2	93	15942
	12	0.2	11	11172	0.2	79	13911	0.2	95	20244

As trends of future research we identify the creation of learning algorithms for the mixtures of trees that allow to determine the number of trees incrementally during the learning phase. Such types of algorithms would be able to adjust the number of components of the mixture to the characteristics of the data. We plan to evaluate the convenience of using other types of priors, like structural priors that allow to change the structure of the trees, and not only the parameters, and "smoothing with the marginal". This method is thought to allow the appearance of events with zero probability. The probability distributions obtained from the data are smoothed with some more general probability distribution by interpolating both distributions. This technique can be seen as a Dirichlet prior derived from the pairwise marginal distributions for the data set [6].

References

1. B. Everitt and D. Hand. *Mixture Models: Inference and Applications to Clustering.* Chapman and Hall, London, 1981.
2. J. H. Holland. *Adaptation in natural and artificial systems.* University of Michigan Press, Ann Arbor, MI, 1975.
3. P. Larrañaga and J. A. Lozano. *Estimation Distribution Algorithms. A new tool for Evolutionary Optimization.* Kluwer Academic Publishers, Boston/Dordrecht/London, 2001.
4. T. Mahnig and H. Mühlenbein. Optimal mutation rate using Bayesian priors for Estimation of Distribution Algorithms. In K. Steinhöfel, editor, *Proceedings of the*

First Symposium on Stochastic Algorithms: Foundations and Applications, SAGA-2001, volume 2264 of *Lecture Notes in Computer Science*, pages 33–48. Springer, 2001.

5. M. Meila. *Learning Mixtures of Trees.* PhD thesis, Massachusetts Institute of Technology, 1999.
6. M. Meila and M. Jordan. Learning with mixtures of trees. *Journal of Machine Learning Research*, 1:1–48, 2000.
7. H. Mühlenbein. The equation for response to selection and its use for prediction. *Evolutionary Computation*, 5(3):303–346, 1997.
8. H. Mühlenbein and T. Mahnig. Evolutionary synthesis of Bayesian networks for optimization. *Advances in Evolutionary Synthesis of Neural Systems, MIT Press*, pages 429–455, 2001.
9. H. Mühlenbein, T. Mahnig, and A. Ochoa. Schemata, distributions and graphical models in evolutionary optimization. *Journal of Heuristics*, 5(2):213–247, 1999.
10. H. Mühlenbein and G. Paaß. From recombination of genes to the estimation of distributions I. Binary parameters. In A. Eiben, T. Bäck, M. Shoenauer, and H. Schwefel, editors, *Parallel Problem Solving from Nature - PPSN IV*, pages 178–187, Berlin, 1996. Springer Verlag.
11. M. Pelikan, D. E. Goldberg, and E. Cantú-Paz. BOA: The Bayesian Optimization Algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-99*, volume I, pages 525–532, Orlando, FL, 1999. Morgan Kaufmann Publishers, San Francisco, CA.
12. R. Santana, A. Ochoa, and M. R. Soto. The Mixture of Trees Factorized Distribution Algorithm. In L. Spector, E. Goodman, A. Wu, W. Langdon, H. Voigt, M. Gen, S. Sen, M. Dorigo, S. Pezeshk, M. Garzon, and E. Burke, editors, *Proceedings of the Genetic and Evolutionary Computation Conference GECCO-2001*, pages 543–550, San Francisco, CA, 2001. Morgan Kaufmann Publishers.
13. R. Santana, A. Ochoa, and M. R. Soto. The Mixture of Trees Factorized Distribution Algorithm. Technical Report ICIMAF 2000-129, Institute of Cybernetics, Mathematics and Physics, Havana, Cuba, January 2001.