A Maximum Entropy Approach to Sampling in EDA – The Single Connected Case

Alberto Ochoa[1,2], Robin Höns[1], Marta Soto[2], and Heinz Mühlenbein[1]

[1] Fraunhofer Institute for Autonomous Intelligent Systems, Sankt Augustin, Germany
[2] Institute of Cybernetics, Mathematics and Physics, Havana, Cuba

Abstract. The success of evolutionary algorithms, in particular Factorized Distribution Algorithms (FDA), for many pattern recognition tasks heavily depends on our ability to reduce the number of function evaluations.
This paper introduces a method to reduce the population size overhead. We use low order marginals during the learning step and then compute the maximum entropy joint distributions for the cliques of the graph. The maximum entropy distribution is computed by an Iterative Proportional Fitting embedded in a junction tree message passing scheme to ensure consistency.
We show for the class of single connected FDA that our method outperforms the commonly-used PLS sampling.

1 Introduction

In recent years, evolutionary algorithms (EA) have been successfully applied to a wide range of problems in the field of pattern recognition. The critical issue in this application of EA is to reduce as far as possible the number of fitness function evaluations which depends directly of the population size of the EA.

In this paper we introduce a method which helps in reducing the population size for a particular class of EA: the Estimation Distribution Algorithms (EDA) [14], which can be considered as a substantial improvement of the genetic algorithm paradigm [4].

The tractable subclass of EDA, the so-called Factorized Distribution Algorithms (FDA), learn factorizations of the joint distribution, which are trees, polytrees or general directed acyclic graphs. This information is used to construct a model from which new points are efficiently sampled. FDA algorithms use results of Graphical Models research [13].

A critical parameter both for learning and sampling is the required population size which grows exponentially with the size of the cliques of the graph. This paper introduces a method to reduce the population size overhead. We use low order marginals during the learning step and then compute the maximum entropy joint distributions for the cliques marginals of the graph. The maximum entropy distribution is computed by an Iterative Proportional Fitting embedded in a junction tree message passing scheme to ensure consistency.

A. Sanfeliu and J. Ruiz-Shulcloper (Eds.): CIARP 2003, LNCS 2905, pp. 683–690, 2003.

The outline of the paper is as follows: Section 2 gives a short introduction on Graphical Models and Factorized Distribution Algorithms. Section 3 presents a single connected FDA. In the next section, we introduce our maximum entropy sampling. Then, we present our test bed and discuss the numerical results. Finally, the main conclusions of our research are given.

2 Background

2.1 Bayesian Networks

A Bayesian network is a directed acyclic graph (DAG) containing nodes, representing the variables, and arcs, representing probabilistic dependencies among nodes. In this paper we will consider binary variables, but the results can be extended to the general discrete case.

Let $X = \{X_1, ..., , X_n\}$ denote the set of random variables. For any node X_i and set of parents π_{X_i} the Bayesian network specifies a conditional probability distribution $p(x_i \mid \pi_{x_i})$. We use lower cases to represent the variable values.

In general, Bayesian networks can be multiple connected. In this paper we deal with single connected graphs: these are graphs where no more than one (undirected) path connects every two variables. Examples are chains, trees, forests and polytrees. Whereas in trees each edge is directed away from the root node (so each node has only one parent), in polytrees the direction of edges is not restricted. A polytree generally has many roots (nodes without parents), whereas a tree has only one root.

Polytrees retain many of the computational advantages of trees, but they allow us to describe higher-order interactions than trees, because they allow *head to head* patterns $X \to Z \leftarrow Y$. This type of pattern makes the parents X and Y conditionally dependent given Z, which can not be represented by a tree. A polytree structure can be induced by second-order marginals using a maximum weight spanning tree algorithm, similar to [1].

Given the structure of the probability distribution defined by the Bayesian network, the problem is to find a factorization defining this distribution. This factorization can be determined using a concept called *junction tree.*

2.2 Junction Trees

A junction tree [11,9] is an undirected tree the nodes of which are clusters of variables. The clusters satisfy the *junction property*: For any two clusters V and W and any cluster U on the unique path between V and W in the junction tree $V \cap W \subseteq U$. The edges between the clusters are labeled with the intersection of the adjacent clusters; we call these labels *separating sets* or *separators.*

Junction trees are a very powerful tool for inference in Bayesian networks. For construction of a junction tree, given a general network, we refer to [11,9]. Given a polytree, a junction tree is simple to construct: For each variable that is not a root, create a node containing this variable and all its parents. The separators between the nodes always consist of only one variable.

Algorithm 1 FDA

```
Step 0    Set $t \leftarrow 1$. Generate $N \gg 0$ points randomly.
Step 1    Select $M \leqslant N$ points according to a selection method.
Step 2    Learn a bayesian factorization of the selected set:
              $$p^s(x_1, \cdots, x_n) = \prod_{i=1}^{n} p(x_i \mid x_{i1}, x_{i2}, ..., x_{ir})$$
Step 3    Sample $N$ new points according to the distribution
              $$p(x, t+1) = p^s(x_1, \cdots, x_n)$$
Step 4    Set $t \leftarrow t + 1$. If termination criteria are not met, go to Step 1.
```

2.3 The Factorized Distribution Algorihtms

Generally, in an FDA (see algorithm 1) the estimation (step 2) of the probability factorization of the best individuals is used to sample (step 3) the points of the next generation, there are no mutation nor crossover operators.

The computational cost of an FDA implementation is determined by the number of function evaluations, the memory needed to store, and the time spent to update and sample the probabilistic model. This time is often exponential in the maximum number of variables that interact in the problem, or which is the same, the size of the building blocks. FDA algorithms which use only pairwise dependencies are cheap.

3 PADA2 – FDA Algorithm with Pairwise Independences

The Polytree Approximation Distribution Algorithm (PADA) [17,16] is a specialization of FDA (see algorithm 1) for single connected Bayesian networks. In this paper we use PADA2 [16], which works with second order marginal distributions. PADA2 is inspired by the algorithm proposed by Rebane and Pearl [15]. We shortly review this algorithm.

A polytree with n variables has a maximum of $n-1$ edges, otherwise it would not be single connected. PADA2 chooses the edges that have the largest values for the *mutual information* $H(X) + H(Y) - H(X, Y)$ [2]. The selection of the edges is done by a greedy Maximum Weight Spanning Tree algorithm.

Once we have constructed the skeleton a procedure tries to direct the edges of the skeleton by using the following scheme: if $X - Z - Y \in skeleton$, then whenever $H(X) + H(Y) = H(X, Y)$ we orient the edges to Z. All other edges are directed at random without introducing new head to head connections.

Another distinguishing feature of PADA2 concerns the sampling step 3 (see algorithm 1). To the best of our knowledge, all FDA algorithms introduced so far that are based on Bayesian networks use the same Monte Carlo sampling

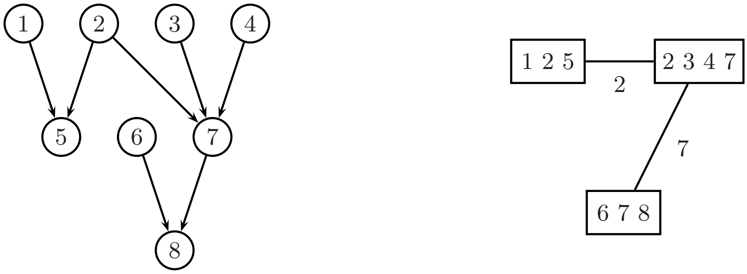

Fig. 1. A polytree learned by PADA2 and its junction tree.

algorithm, namely Probabilistic Logic Sampling (PLS) [6]. It is very simple. Given an ancestral ordering of all the variables (parents before children), the method samples X_i using $p(X_i \mid \pi_{X_i})$. One obvious problem of this method is that the number of points required for estimating the conditional probabilities correctly is exponential in the number of parents.

Both in the process of learning and sampling, FDA that learn general Bayesian networks need a population size which is exponential in the number of parents in order to get reliable estimates of the conditional probabilities. PADA2 is in a different situation: its learning algorithm deals only with second order marginals. Figure 1 shows a polytree learned by PADA2. However, note that the resulting junction tree contains a clique with four variables (7 and its parents $2, 3, 4$) and two cliques with 3 variables, so the PLS sampling requires 4-order and 3-order marginals. Therefore, PADA2 is in a very singular situation when it uses PLS: what is gained during learning is then lost during the sampling step.

In the next section we will present a novel method to overcome this problem. We fix some of the second order marginals used in the learning step, and then compute higher order marginals (like the ones in Fig. 1) as the maximum entropy distributions that obey the given second order marginals. It is important to note that, in contrast to PLS, the computation of these marginals does not need a larger sample size (population size) than the one used for learning.

4 Maximum Entropy

4.1 Entropy and the Maximum Entropy Principle

The *Entropy* [2] of a probability distribution for a random variable X is given by

$$H(X) = -\sum_{x} p(x) \log p(x) \ . \tag{1}$$

If $p(x) = 0$, then $\log p(x)$ is not defined. For this case, we set $0 \log 0 = 0$.

The entropy is a measure of the disorder in the distribution, or of our uncertainty about the outcome of a random experiment.

The *Maximum Entropy Principle* states that, supposing we are looking for a probability distribution fulfilling some given contraints, we should choose among the possible solutions the one with the highest entropy. This is historically founded on Bernoulli's "principle of insufficient reason". It was introduced and advocated by Jaynes [7,8]. For a motivation and discussion of Maximum Entropy, see [5].

4.2 Maximum Entropy Sampling in PADA2

The method is a variation of previous methods for learning probability distributions on junction trees [10,12].

First we construct a junction tree from the polytree, as described in Sect. 2.2. On each node of the junction tree, we maintain a probability distribution of the contained variables (remember, a child and its parents).

Then we find a closed tour through the junction tree that visits each node at least once. On this tour we perform two steps:

1. Calculate the local distribution by *iterative proportional fitting*,
2. *Pass a message* to the next cluster on the tour, in order to ensure consistency between the clusters.

4.3 Iterative Proportional Fitting

The local iteration consists of finding the maximum entropy distribution of a child and its parents, given the second order marginals. This is done by *Iterative Proportional Fitting*. IPF computes iteratively a distribution $q_\tau(\mathbf{x})$ from the given marginals $p_k(\mathbf{x}_k)$, $k = 1, \dots, K$, where $\mathbf{x}_k$ is a subvector of $\mathbf{x}$ and $\tau = 0, 1, 2, \dots$ is the iteration index. Let n be the dimension of $\mathbf{x}$ and d_k be the dimension of $\mathbf{x}_k$. Then, starting from the uniform distribution, the update formula is

$$q_{\tau+1}(\mathbf{x}) = q_\tau(\mathbf{x}) \frac{p_k(\mathbf{x}_k)}{\sum\limits_{y \in \{0,1\}^{n-d_k}} q_\tau(\mathbf{x}_k, \mathbf{y})} \tag{2}$$

with $k = ((\tau - 1) \mod K) + 1$.

For the proof that IPF converges to the maximum entropy solution, see [3] and references therein. Note that the effort is exponential in the clique size.

4.4 Message Passing

A message from a cluster W to a cluster V, separated by $S = V \cap W$, is sent by the following algorithm:

$$q_S^{\text{new}}(\mathbf{x}_S) = \sum_{\mathbf{x}_{W \setminus S}} q_W^{\text{old}}(\mathbf{x}_S, \mathbf{x}_{W \setminus S}) \qquad q_V^{\text{new}}(\mathbf{x}) = q_V^{\text{old}}(\mathbf{x}) \frac{q_S^{\text{new}}(\mathbf{x}|S)}{q_S^{\text{old}}(\mathbf{x}|S)}$$

Here $\mathbf{x}|S$ denotes the vector $\mathbf{x}$, restricted to the variables in S.

4.5 Sampling in the Junction Tree

Using the distributions within the junction tree, the sampling of points works as follows:

Start from any node in the junction tree, sample values for the variables from the local probability distribution. Then, proceed to the neighbors and sample values for the new variables, conditioned on the variables which have already been sampled. When each node has been visited, the sampled individual is complete.

For example, from the structure in Fig. 1, this algorithm samples using the factorization

$$p_{\mathrm{JT}}(\mathbf{x}) = p(x_1, x_2, x_5) p(x_3, x_4, x_7 | x_2) p(x_6, x_8 | x_7) \ ,$$

whereas PLS uses the factorization

$$p_{\mathrm{PLS}}(\mathbf{x}) = p(x_1)p(x_2)p(x_3)p(x_4)p(x_5|x_1,x_2)p(x_6)p(x_7|x_2,x_3,x_4)p(x_8|x_6,x_7)$$

which is not an exact factorization of the underlying distribution.

5 Numerical Results

Now we present the set of additive decomposable functions (ADF) that will be used in our experiments.

1. The *Deceptive Function* of order k, F_k^{dec}, is defined as follows. u denotes the number of 1s in the string. We set $f_k^{\mathrm{dec}}(u) = k$ if $u = k$, and $f_k^{\mathrm{dec}}(u) = k - 1 - u$ otherwise. The function F_k^{dec} is a separable function of subset size k, with $n = k * l$.

$$F_k^{\mathrm{dec}} = \sum_{i=1}^{l} f_k^{\mathrm{dec}}(x_{ki-k+1} + \ldots + x_{ki})$$

2. The next function is also a separable ADF with blocks of length 5. In each block the *FirstPolytree5* function is evaluated. This function has the following property: Its Boltzmann distribution with parameter $\beta \approx 2$ has a polytree structure with edges $x_1 \rightarrow x_3$, $x_2 \rightarrow x_3$, $x_3 \rightarrow x_5$ and $x_4 \rightarrow x_5$. The reader can easily check this by constructing the Boltzmann distribution and then checking marginal dependencies. The definition of the function is given below.

$\mathbf{x}$	$f_5^{\mathrm{Poly}}(\mathbf{x})$	$\mathbf{x}$	$f_5^{\mathrm{Poly}}(\mathbf{x})$	$\mathbf{x}$	$f_5^{\mathrm{Poly}}(\mathbf{x})$	$\mathbf{x}$	$f_5^{\mathrm{Poly}}(\mathbf{x})$
00000	-1.141	01000	-0.753	10000	-3.527	11000	-6.664
00001	1.334	01001	1.723	10001	-1.051	11001	-4.189
00010	-5.353	01010	-4.964	10010	-7.738	11010	-10.876
00011	-1.700	01011	-1.311	10011	-4.085	11011	-7.223
00100	0.063	01100	1.454	10100	1.002	11100	-1.133
00101	-0.815	01101	0.576	10101	0.124	11101	-2.011
00110	-0.952	01110	0.439	10110	-0.013	11110	-2.148
00111	-0.652	01111	0.739	10111	0.286	11111	-1.849

We recall that the basic claim of our research is that our maximum entropy approach to sampling requires a smaller population size than PLS. In this section we will compare these two sampling methods for PADA2.

All the experiments use a fixed truncation selection pressure ($\tau = 0.3$), do not use elitism and are run until a maximum of 20 generations. We perfom 100 runs for each experiment. We use as test functions Deceptive 4 (with 20 variables), Goldberg Deceptive 3 (21 variables) and the FirstPolytree5 (20 variables).

As can be seen in Table 1, the improvement in comparison with conventional PLS is enormous. E. g. for Deceptive 4, our new method finds the optimum in 93 % of the cases for only 800 individuals, whereas PLS even with a population size of 5000 succeeds only in 64 %.

It is also remarkable that the number of generations until success stays the same or even improves. It has also stabilized, as can be seen from the decrease in the standard deviation.

Table 1. Numerical results. D4 - Deceptive 4, D3 - Goldberg Deceptive 3, FP5 - FirstPolytree 5. N - population size, $\%S$ - Success rate, G_c - generation where the optimum is found, MES - maximum entropy sampling, PLS - probabilistic logic sampling.

N			200	600	800	5000
$D4$	PLS	$\%S$	1	12	16	64
		G_c	5 ± 0.0	8.0 ± 3.9	8.3 ± 3.5	9.23 ± 3.5
	MES	$\%S$	21	76	93	100
		G_c	11.14 ± 4.5	8.6 ± 3.2	8.4 ± 2.5	6.1 ± 1.3
$D3$	PLS	$\%S$	0	8	10	92
		G_c	–	9.75 ± 1.5	8.7 ± 3.2	7.21 ± 1.2
	MES	$\%S$	2	69	90	100
		G_c	8.5 ± 0.7	7.4 ± 1.1	7.0 ± 1.2	5.84 ± 0.9
$FP5$	PLS	$\%S$	25	50	54	55
		G_c	10.08 ± 2.08	10.42 ± 2.59	10.59 ± 2.34	10.8 ± 1.5
	MES	$\%S$	59	100	100	100
		G_c	5.14 ± 1.07	3.93 ± 0.7	3.66 ± 0.59	2.92 ± 0.44

6 Summary and Conclusions

The paper introduces a new method for sampling individuals in EDA. Here we restrict ourselves to single connected Bayesian networks (polytrees). In a forthcoming paper, we will discuss the multiple connected case.

The polytree induces canonically a junction tree. Its nodes contain the higher-order marginal distributions that are needed in the sampling phase. These are computed from the given second order marginals using the maximum entropy principle. The conventional "Probabilistic Logic Sampling" is replaced by sampling inside the junction tree.

We explore the method by applying it on three benchmark problems. The improvement in comparison with the previous method turns out to be tremendous. We conclude that using this sampling, we can greatly reduce the population size. This results in a big saving of function evaluations which is critical for any pattern recognition application of evolutionary computation.

References

1. C. K. Chow and C. N. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Trans. on Information Theory*, IT14(3):462–467, 1968.
2. Th. M. Cover and J.A. Thomas. *Elements of Information Theory.* Wiley, New York, 1989.
3. I. Csiszár. I-divergence geometry of probability distributions and minimization problems. *Annals of Probability*, 3:146–158, 1975.
4. D. E. Goldberg. *Genetic Algorithms in search, optimization, and machine learning.* Addison-Wesley, Reading, MA, USA, 1989.
5. P. Grünwald. *The Minimum Description Length Principle and Reasoning under Uncertainty.* PhD thesis, University of Amsterdam, 1998.
6. M. Henrion. Propagating uncertainty in Bayesian networks by probabilistic logic sampling. *Uncertainty in Artificial Intelligence*, 2:317–324, 1988.
7. E. T. Jaynes. Information theory and statistical mechanics. *Phys. Rev*, 6:620–643, 1957.
8. E. T. Jaynes. Where do we stand on maximum entropy? In R. D. Levine and M. Tribus, editors, *The Maximum Entropy Formalism.* MIT Press, Camb., 1978.
9. F. V. Jensen and F. Jensen. Optimal junction trees. In *Proc. of the 10th Conference on Uncertainty in Artificial Intelligence*, pages 360–366, Seattle, Wash., 1994.
10. R. Jiroušek and St. Přeučil. On the effective implementation of the iterative proportional fitting procedure. *Comput. Statistics & Data Analysis*, 19:177–189, 1995.
11. S. L. Lauritzen. *Graphical Models.* Oxford:Clarendon Press, 1996.
12. C.-H. Meyer. *Korrektes Schließen bei unvollständiger Information.* PhD thesis, Fernuniversität Hagen, 1998.
13. H. Mühlenbein, Th. Mahnig, and A. Ochoa. Schemata, distributions and graphical models in evolutionary optimization. *Journal of Heuristics*, 5(2):213–247, 1999.
14. H. Mühlenbein and G. Paaß. From recombination of genes to the estimation of distributions i. binary parameters. In H.-M. Voigt, W. Ebeling, I. Rechenberg, and H.-P. Schwefel, editors, *Lecture Notes in Computer Science 1141: Parallel Problem Solving from Nature – PPSN IV*, pages 178–187, Berlin, 1996. Springer-Verlag.
15. J. Pearl. *Probabilistic reasoning in intelligent systems: Networks of plausible inference.* Morgan Kaufmann, San Mateo, California, 1988.
16. M. Soto. *A Singled Connected Factorized Distribution Algorithm and its cost of evaluation.* PhD thesis, University of Havana, Havana, Cuba, July 2003. (In Spanish, adviser A. Ochoa).
17. M. Soto and A. Ochoa. A Factorized Distribution Algorithm based on polytrees. In *Proceedings of the 2000 Congress on Evolutionary Computation CEC00*, pages 232–237, La Jolla, California, 6–9 July 2000. IEEE Press.