An EDA Based on Local Markov Property and Gibbs Sampling

Siddhartha Shakya
Intelligent Systems Research Centre, BT Group
Chief Technology Office
Adastral Park, Ipswich
IP5 3RE, UK
sid.shakya@bt.com

Roberto Santana
Intelligent System Group
Department of Computer Science and Artificial Intelligence
University of the Basque Country
roberto.santana@ehu.es

ABSTRACT

The key ideas behind most of the recently proposed Markov networks based EDAs were to factorise the joint probability distribution in terms of the cliques in the undirected graph. As such, they made use of the global Markov property of the Markov network. Here we presents a Markov Network based EDA that exploits Gibbs sampling to sample from the Local Markov property, the Markovianity, and does not directly model the joint distribution. We call it Markovianity based Optimisation Algorithm. Some initial results on the performance of the proposed algorithm shows that it compares well with other Bayesian network based EDAs.

Categories and Subject Descriptors

I.2.8 [**Artificial Intelligence**]: Problem Solving, Control Methods, and Search
; G.3 [**Probability and statistics**]: Probabilistic algorithms, Stochastic processes

General Terms

Algorithms, Performance, Design

Keywords

Estimation of Distribution Algorithms, Probabilistic graphical models, Markov Networks, Evolutionary Computation

1. INTRODUCTION

An EDA regards a solution $x = \{x_1, x_2, ..., x_n\}$ as a set of values taken by a set $X = \{X_1, X_2, ..., X_n\}$ of random variables. A Markov network [1] on set X is characterised in terms of neighbourhood relationship between variables in X by its *local Markov property* known as *Markovianity* [1][2], which states that the conditional probability of a node X_i given the rest of the variables can be completely defined in

$$p(x_i|x - \{x_i\}) = p(x_i|N_i) \tag{1}$$

A Markov network is also characterised in terms of *cliques*[1] in the undirected graph by its global property, the joint probability distribution, and can be written as

$$p(x) = \frac{1}{Z}\prod_{i=1}^{m}\psi_i(c_i) \tag{2}$$

Where, $\psi_i(c_i)$ is a *potential function* on clique $C_i \in X$, m is the number of cliques in the structure G. Z is the normalising constant known as the *partition function* given by $Z = \sum_{x\in\Omega}\prod_{i=1}^{m}\psi_i(c_i)$. Here, Ω is the set of all possible solutions.

2. MOA WORKFLOW

Most of the EDAs based on Markov network, such as DEUM [5] and MN-EDA [4], use its global property (2) in one form or another. More precisely, they factorise the joint probability distribution in terms of the cliques in the undirected graph and sample it to generate new solutions. In this paper we present Markovianity based Optimisation Algorithm (MOA) (Figure 1) that does not explicitly factorise the joint probability distribution. Instead, it estimates conditional probabilities (1) in terms of neighbouring nodes in the undirected graph and samples from them using Gibbs sampler. MOA incorporates features that have been independently employed in previous implementations of EDAs based on Markov models, but have not been used together. The resulting algorithm is qualitatively different to its predecessors, easier to understand and simpler to implement. It does structural learning of the probabilistic model from

Markovianity based Optimisation Algorithm

1. Generate initial (parent) population P of size M
2. Select set D from P consisting of N solutions, where $N <= M$
3. Estimate structure of a Markov network from D
4. Estimate local Markov conditional probabilities, $p(x_i|N_i)$, for each variable X_i as defined by the undirected structure and sample them to generate new population
5. Replace old population by new one and go to step 2 until termination criteria are met

Figure 1: The workflow of Markovianity based Optimisation Algorithm

[1] Given an undirected graph G, a clique is a fully connected subset of the nodes

GECCO'08, July 12–16, 2008, Atlanta, Georgia, USA.
ACM 978-1-60558-130-9/08/07.

Estimating structure - Step 3 of MOA

1. Create a matrix of mutual information, MI, by estimating cross entropy for each pair of variables in the solution. Cross entropy between two random variables, A and B, is given by

$$CE(A,B) = \sum_{a,b} p(a,b) log \left(\frac{p(a,b)}{p(a) \cdot p(b)} \right)$$

   where sum is over all possible combinations of A and B, and $p(a,b)$ is the joint probability of $A = a$ and $B = b$ computed from D

2. Create an edge between two variables, if the mutual information between them is higher than the given threshold. Here we compute the threshold, TR as $TR = avg(MI) * sig$, where $avg(MI)$ is the average of the elements of the MI matrix and sig is the significance parameter, which for the purpose of this paper is set to 1.5.

3. If the number of neighbours to a variable is higher than the maximum number, MN, allowed, only keep MN neighbours that have the highest mutual information.

Figure 2: The workflow of an undirected structure learning algorithm

the data but it can also take advantage of a priori structural information in a straightforward way. Complex approximations to the joint probability distribution are avoided and the temperature parameter is included to balance the exploration and exploitation of the search space. The use of Gibbs sampling remains as a key component, allowing MOA to deal with interactions represented by cycles.

For this work, we implement a mutual information based approach, described in Figure 2, for leaning a Markov network structure. By its definition, an undirected structure may contain cycles, therefore, we use Gibbs sampler [5], a class of MCMC method, as the sampling method in MOA. Figure 3 describes a version of Gibbs sampler that has been implemented for the purpose of this work. Note that each execution of the Gibbs sampler creates a single solution. Multiple execution of Gibbs sampler should be done in order to create the population of solutions. Also, we set a linear schedule for the temperature as $T = \frac{1}{g \times CR}$, where g is the current generation of MOA and CR is the *cooling rate* parameter. CR can be varied in order to control the convergence of MOA. We set CR to 0.5 for all the experimental results.

3. RESULTS AND CONCLUSION

Since deceptive functions [3] were shown to be hard problem for many lower order EDAs and GAs, we test MOA on two different deceptive functions, trap5 and 3deceptive, of size up to 360 bits. The results show that MOA scales well on these problem and the performance is comparable to that of other Bayesian network based EDAs. This result is encouraging since the MOA implemented here is very basic and uses simpler structure learning and sampling technique. Implementing more efficient structure learning algorithm with higher order independency tests, and more efficient versions of the Gibbs sampler, is likely to further improve the performance of MOA. These works are under way and interesting results are expected in near future.

Gibbs Sampler - Step 4 of MOA

1. Generate a solution $x = \{x_1, x_2, .., x_n\}$ at random.

2. For r iterations (in this paper we set $r = n \times ln(n) \times IT$, where IT, the *iteration coefficient*, is set to 4), do the following:

   (a) Choose a variable x_i from x at random.

   (b) Using selected set of solutions, D, compute conditional probabilities $p(x_i|N_i)$ for each value of x_i as Gibbs probability,

$$p(x_i|N_i) = \frac{e^{p(x_i,N_i)/T}}{\sum_{x_i'} e^{p(x_i',N_i)/T}}$$

   where sum is over all possible values of x_i. For example, in binary case, where $x_i = \{0,1\}$, probability of $x_i = 1$ given the value of its neighbours N_i is written as

$$p(1|N_i) = \frac{e^{p(1,N_i)/T}}{e^{p(1,N_i)/T} + e^{p(0,N_i)/T}}$$

   Here, T is the *temperature coefficient* that controls the convergence of the Gibbs probability distribution. Increasing T makes the distribution close to being uniform, and decreasing T converges it to an extrema.

   (c) Sample $p(x_i|N_i)$ to get new x_i.

3. Terminate with answer x.

Figure 3: The workflow of implemented Gibbs Sampler algorithm

4. REFERENCES

[1] J. Besag. Spatial interactions and the statistical analysis of lattice systems (with discussions). *Journal of the Royal Statistical Society*, 36:192–236, 1974.

[2] S. Z. Li. *Markov Random Field modeling in computer vision.* Springer-Verlag, 1995.

[3] M. Pelikan. *Bayesian optimization algorithm: From single level to hierarchy.* PhD thesis, University of Illinois at Urbana-Champaign, Urbana, IL, 2002. Also IlliGAL Report No. 2002023.

[4] R. Santana. Estimation of Distribution Algorithms with Kikuchi Approximation. *Evolutionary Computation*, 13:67–98, 2005.

[5] S. Shakya. *DEUM: A Framework for an Estimation of Distribution Algorithm based on Markov Random Fields.* PhD thesis, The Robert Gordon University, Aberdeen, UK, April 2006.