Missouri Estimation of Distribution Algorithms Laboratory

Model Accuracy in the Bayesian Optimization Algorithm

Claudio F. Lima, Fernando G. Lobo, Martin Pelikan, and David E. Goldberg

MEDAL Report No. 2010004

February 2010

Abstract

Evolutionary algorithms (EAs) are particularly suited to solve problems for which there is not much information available. From this standpoint, estimation of distribution algorithms (EDAs), which guide the search by using probabilistic models of the population, have brought a new view to evolutionary computation. While solving a given problem with an EDA, the user has access to a set of models that reveal probabilistic dependencies between variables, an important source of information about the problem. However, as the complexity of the used models increases, the chance of overfitting and consequently reducing model interpretability, increases as well.

This paper investigates the relationship between the probabilistic models learned by the Bayesian optimization algorithm (BOA) and the underlying problem structure. The purpose of the paper is threefold. First, model building in BOA is analyzed to understand how the problem structure is learned. Second, it is shown how the selection operator can lead to model overfitting in Bayesian EDAs. Third, the scoring metric that guides the search for an adequate model structure is modified to take into account the non-uniform distribution of the mating pool generated by tournament selection. Overall, this paper makes a contribution towards understanding and improving model accuracy in BOA, providing more interpretable models to assist efficiency enhancement techniques and human researchers.

Keywords

Estimation of distribution algorithms, Bayesian optimization algorithm, Bayesian networks, model overfitting, selection.

Missouri Estimation of Distribution Algorithms Laboratory (MEDAL)
Department of Mathematics and Computer Science
University of Missouri–St. Louis
One University Blvd., St. Louis, MO 63121
E-mail: medal@cs.umsl.edu
WWW: http://medal.cs.umsl.edu/

Model Accuracy in the Bayesian Optimization Algorithm

Claudio F. Lima[1], **Fernando G. Lobo**[1], **Martin Pelikan**[2], **David E. Goldberg**[3]

[1]Department of Electronics and Computer Science Engineering
University of Algarve, Campus de Gambelas, 8000-117 Faro, Portugal
{clima.research,fernando.lobo}@gmail.com

[2]Missouri Estimation of Distribution Algorithm Laboratory (MEDAL)
Department of Mathematics and Computer Science
University of Missouri at St. Louis, St. Louis MO 63121
pelikan@cs.umsl.edu

[3]Illinois Genetic Algorithms Laboratory (IlliGAL)
Department of Industrial and Enterprise Systems Engineering
University of Illinois at Urbana-Champaign, Urbana IL 61801
deg@illinois.edu

Abstract

Evolutionary algorithms (EAs) are particularly suited to solve problems for which there is not much information available. From this standpoint, estimation of distribution algorithms (EDAs), which guide the search by using probabilistic models of the population, have brought a new view to evolutionary computation. While solving a given problem with an EDA, the user has access to a set of models that reveal probabilistic dependencies between variables, an important source of information about the problem. However, as the complexity of the used models increases, the chance of overfitting and consequently reducing model interpretability, increases as well.

This paper investigates the relationship between the probabilistic models learned by the Bayesian optimization algorithm (BOA) and the underlying problem structure. The purpose of the paper is threefold. First, model building in BOA is analyzed to understand how the problem structure is learned. Second, it is shown how the selection operator can lead to model overfitting in Bayesian EDAs. Third, the scoring metric that guides the search for an adequate model structure is modified to take into account the non-uniform distribution of the mating pool generated by tournament selection. Overall, this paper makes a contribution towards understanding and improving model accuracy in BOA, providing more interpretable models to assist efficiency enhancement techniques and human researchers.

1 Introduction

The last decade has seen the rise and consolidation of a new trend of stochastic optimizers known as estimation of distribution algorithms (EDAs) (Pelikan, Goldberg, & Lobo, 2002; Larrañaga & Lozano, 2002; Pelikan, Sastry, & Cantú-Paz, 2006; Lozano, Larrañaga, Inza, & Bengoetxea, 2006). In essence, EDAs build probabilistic models of promising solutions and sample from the corresponding probability distributions to obtain new solutions. Therefore, these algorithms are typically classified according to the complexity of the probabilistic models they rely on. Simpler

EDAs use a model of simple and fixed structure, and only learn the corresponding parameters. At the other side of the spectrum, there are EDAs with adaptive multivariate models such as Bayesian networks (BNs) (Pearl, 1988), which can model complex multivariate interactions. Examples of Bayesian EDAs include the Bayesian optimization algorithm (BOA) (Pelikan, Goldberg, & Cantú-Paz, 1999; Pelikan, 2005), the estimation of Bayesian networks algorithm (EBNA) (Etxeberria & Larrañaga, 1999), and the learning factorized distribution algorithm (LFDA) (Mühlenbein & Mahning, 1999).

While Bayesian EDAs are able to solve a broad class of nearly decomposable and hierarchical problems in a reliable and scalable manner, their probabilistic models oftentimes do not exactly reflect the problem structure (Lima, Goldberg, Pelikan, Lobo, Sastry, & Hauschild, 2007; Hauschild, Pelikan, Sastry, & Lima, 2009; Echegoyen, Lozano, Santana, & Larrañaga, 2007; Mühlenbein, 2008). Because these models are learned from a sample of limited size (population of individuals), particular features of the specific sample are also encoded, which act as noise when seeking for generalization. This is a well-known problem in machine learning, known as model overfitting. However, in the context of EDAs model overfitting is double-sided. While the goal is to model promising solutions rather than the entire search space, focusing on an excessively narrowed portion of this space might not reveal meaningful information about the underlying problem structure, and even reduce the probability of finding the optimum.

In many situations, the knowledge of the problem structure can be as valuable as a high-quality solution to the problem. This is the case for several model-based efficiency enhancement techniques (Sastry, Pelikan, & Goldberg, 2004; Pelikan & Sastry, 2004; Sastry, Lima, & Goldberg, 2006; Sastry & Goldberg, 2004; Lima, Sastry, Goldberg, & Lobo, 2005; Lima, Pelikan, Sastry, Butz, Goldberg, & Lobo, 2006; Lima, Pelikan, Lobo, & Goldberg, 2009; Lima, 2009; Sastry, Abbass, Goldberg, & Johnson, 2005; Yu & Goldberg, 2004; Yu, Sastry, & Goldberg, 2007; Hauschild, Pelikan, Sastry, & Goldberg, 2008; Hauschild & Pelikan, 2008) developed for EDAs that yield *super-multiplicative speedups* (Goldberg & Sastry, 2010). Another important situation is the *offline interpretation* of the probabilistic models (Yu, Goldberg, Sastry, Lima, & Pelikan, 2009; Yu & Goldberg, 2004) to help develop fixed but structure-based operators for specific instances or classes of problems that have similar structure. In this case the EDA can act as a data miner to gain insight about the problem. The importance of analyzing the resulting probabilistic models in EDAs has also been recently highlighted by others (Santana, Larrañaga, & Lozano, 2005; Wu & Shapiro, 2006; Correa & Shapiro, 2006; Echegoyen, Lozano, Santana, & Larrañaga, 2007; Hauschild, Pelikan, Sastry, & Lima, 2009; Lima, Goldberg, Pelikan, Lobo, Sastry, & Hauschild, 2007; Mühlenbein, 2008).

Bayesian network learning is an active topic of research in machine learning, as the choice of the search procedure can have a great influence on model accuracy. However, the problem of finding the best network has been proven to be NP-complete for most scoring metrics (Chickering, Geiger, & Heckerman, 1994). Therefore, in Bayesian EDAs a simple local search procedure is typically used for a good compromise between search efficiency and model quality (Pelikan, Goldberg, & Cantú-Paz, 1999; Pelikan, 2005; Etxeberria & Larrañaga, 1999; Mühlenbein & Mahning, 1999), given the high computational cost of considering more sophisticated alternatives (Echegoyen, Lozano, Santana, & Larrañaga, 2007). Note that Bayesian networks (or any other probabilistic model for that matter) are used as an auxiliary tool in the optimization process, thus it is a good practice to keep the search complexity as simple as possible. On the other hand, this work focus on the best way to integrate BNs within the evolutionary computation framework to improve the expressiveness of the learned models.

This paper investigates model structural accuracy in the Bayesian optimization algorithm, giving particular emphasis to the relationship between the underlying problem structure and the learned Bayesian network structure. The paper also addresses the selection operator as a source of overfitting in Bayesian EDAs. First, a detailed analysis of model learning in BOA is performed to better understand how the problem structure is learned and when inaccuracies are introduced in the network. Next, the role of selection in BN learning is investigated by looking at selection as the mating pool distribution generator, which turns out to have a great impact on model structural accuracy. Particularly, it is shown that tournament selection generates the mating pool according to a power distribution that leads to model overfitting. However, if the metric that scores networks takes into account the resampling bias induced by tournament selection, the model quality can be highly improved and comparable to that of truncation selection which generates a uniform distribution, more suitable for BN learning. The utility of the proposed scoring metric is verified through experiments on three test problems that represent different facets of probabilistic modeling in Bayesian EDAs. These problems are the $m-k$ trap, the onemax, and the hierarchical trap. The results show that the model structural accuracy and interpretability is significantly improved with the modified scoring metric. In general, this paper contributes to better understand and interpret the probabilistic models in BOA, knowledge that can be used to improve several efficiency enhancement techniques (Sastry, Pelikan, & Goldberg, 2004; Pelikan & Sastry, 2004; Sastry, Lima, & Goldberg, 2006; Sastry & Goldberg, 2004; Lima, Sastry, Goldberg, & Lobo, 2005; Lima, Pelikan, Sastry, Butz, Goldberg, & Lobo, 2006; Lima, Pelikan, Lobo, & Goldberg, 2009; Lima, 2009; Sastry, Abbass, Goldberg, & Johnson, 2005; Yu & Goldberg, 2004; Yu, Sastry, & Goldberg, 2007; Yu, Goldberg, Sastry, Lima, & Pelikan, 2009; Hauschild, Pelikan, Sastry, & Goldberg, 2008; Hauschild & Pelikan, 2008; Goldberg & Sastry, 2010) and assist human researchers (Yu, Goldberg, Sastry, Lima, & Pelikan, 2009; Yu & Goldberg, 2004).

The paper is organized as follows. The next section introduces relevant background to understand the purpose of the paper by describing BOA and previous work related to the topic addressed. Section 3 analyses in detail how the model is learned in BOA, while Section 4 investigates the role of selection in model learning and overfitting. Section 5 models the scoring metric gain when overfitting with tournament selection. In Section 6, an adaptive scoring metric is proposed to avoid overfitting, which is shown to considerably improve model accuracy. The paper ends with major conclusions.

2 Background

2.1 Bayesian Optimization Algorithm

The Bayesian optimization algorithm (BOA) (Pelikan, Goldberg, & Cantú-Paz, 1999; Pelikan, 2005) uses Bayesian networks (BNs) to capture the (in)dependencies between the decision variables of the optimization problem. In BOA, the traditional crossover and mutation operators of evolutionary algorithms are replaced by (1) building a BN which models promising solutions and (2) sampling from the probability distribution encoded by the built BN to generate new solutions. The pseudocode of BOA is detailed in Figure 1.

Bayesian networks (Pearl, 1988) are powerful graphical models that combine probability theory with graph theory to encode probabilistic relationships between variables of interest. A BN is defined by its structure and corresponding parameters. The structure is represented by a directed acyclic graph where the nodes correspond to the variables of the problem and the edges correspond to conditional dependencies. The parameters are represented by the conditional probabilities for each variable given any instance of the variables that this variable depends on. More formally, a

```
Bayesian optimization algorithm (BOA)

(1) Create a random population P of $n$ individuals.
(2) Evaluate population P.
(3) Select P' individuals from P using a selection procedure.
(4) Model the selected individuals P' by learning the most adequate Bayesian network B.
(5) Create a new population O by sampling from the joint probability distribution of B.
(6) Evaluate population O.
(7) Replace all (or some) individuals in population P by those from O.
(8) If stopping criteria are not satisfied, return to step 3.
```

Figure 1: Pseudocode of the Bayesian optimization algorithm.

Bayesian network encodes the following joint probability distribution,

$$p(X) = \prod_{i=1}^{\ell} p(X_i|\Pi_i), \tag{1}$$

where $X = (X_1, X_2, \ldots, X_\ell)$ is a vector with all variables of the problem, Π_i is the set of *parents* of X_i (nodes from which there exists an edge to X_i), and $p(X_i|\Pi_i)$ is the conditional probability of X_i given its parents Π_i.

The parameters of a Bayesian network can be represented by a set of conditional probability tables (CPTs) or local structures. Using local structures such as decision trees allows a more efficient and flexible representation of local conditional distributions, improving the expressiveness of BNs (Chickering, Heckerman, & Meek, 1997; Friedman & Goldszmidt, 1999; Pelikan, 2005). In this work we focus on BNs with decision trees.

The quality of a given network structure is quantified by a scoring metric. We consider two popular metrics for BNs: the Bayesian-Dirichlet metric (BD) (Cooper & Herskovits, 1992; Heckerman, Geiger, & Chickering, 1994) and the Bayesian information criterion (BIC) (Schwarz, 1978).

The BD metric for BNs with decision trees (Chickering, Heckerman, & Meek, 1997) is given by

$$BD(B) = p(B) \prod_{i=1}^{\ell} \prod_{l \in L_i} \frac{\Gamma(m_i'(l))}{\Gamma(m_i(l) + m_i'(l))} \prod_{x_i} \frac{\Gamma(m_i(x_i, l) + m_i'(x_i, l))}{\Gamma(m_i'(x_i, l))}, \tag{2}$$

where $p(B)$ is the prior probability of the network structure B, L_i is the set of leaves in the decision tree T_i (corresponding to X_i), $m_i(l)$ is the number of instances in the population that contain the traversal path in T_i ending in leaf l, $m_i(x_i, l)$ is the number of instances in the population that have $X_i = x_i$ and contain the traversal path in T_i ending in leaf l, $m_i'(l)$ and $m_i'(x_i, l)$ represent prior knowledge about the values of $m_i(l)$ and $m_i(x_i, l)$. Here, we consider the K2 variant of the BD metric, which uses an uninformative prior that assigns $m_i'(x_i, l) = 1$.

To favor simpler networks over more complex ones, the prior probability of each network $p(B)$ can be adjusted according to its complexity, that is given by the description length of the parameters required by the network (Chickering, Heckerman, & Meek, 1997; Friedman & Goldszmidt, 1999). Based on this principle, the following penalty function was proposed for BOA (Pelikan, 2005),

$$p(B) = 2^{-0.5 \log_2(n) \sum_{i=1}^{\ell} |L_i|}, \tag{3}$$

where n is the population size, and $|L_i|$ is the number of leaves in decision tree T_i.

The BIC metric is based on the minimum description length (MDL) principle (Rissanen, 1978) and is given by

$$BIC(B) = \sum_{i=1}^{\ell} \left(\sum_{l \in L_i} \sum_{x_i} \left(m_i(x_i, l) \log_2 \frac{m_i(x_i, l)}{m_i(l)} \right) - |L_i| \frac{\log_2(n)}{2} \right). \tag{4}$$

It has been shown that the behavior of these metrics is asymptotically equivalent; however, the results obtained with each metric can differ for particular domains, particularly in terms of sensitivity to noise. In the context of EDAs, when using CPTs to store the parameters, the BIC metric outperforms the K2 metric, but when using decision trees or graphs, the K2 metric has shown to be more robust (Pelikan, 2005). We will confirm this observation in the remainder of the paper.

To learn the most adequate structure for the BN a greedy algorithm is usually used as a good compromise between search efficiency and model quality. We consider a simple learning algorithm that starts with an empty network and at each step performs the operation that improves the metric the most, until no further improvement is possible. The operator considered is the *split*, which splits a leaf on some variable and creates two new children on the leaf. Each time a split on X_j takes place in tree T_i, an edge from X_j to X_i is added to the network. For more details on learning BNs with local structures the reader is referred elsewhere (Chickering, Heckerman, & Meek, 1997; Friedman & Goldszmidt, 1999; Pelikan, 2005).

The generation of new solutions is done by sampling from the learned Bayesian network using probabilistic logic samping (PLS) (Henrion, 1988). Briefly, PLS consists in (1) computing an ancestral ordering of the nodes (where each node is preceded by its parents) and (2) generating the values for each variable according to the ancestral ordering and the conditional probabilities (Equation 1).

The hierarchical BOA (hBOA) was later introduced by Pelikan and Goldberg (Pelikan & Goldberg, 2001; Pelikan, 2005) and results from the combination of BNs with local structures with a simple yet powerful niching method to maintain diversity in the population, known as restricted tournament replacement (RTR) (Harik, 1995). hBOA is able to solve hierarchical decomposable problems, in which the variable interactions are present at more than a single level.

2.2 Related Work

Although the main feature of BOA and other EDAs is to perform efficient mixing of key substructures or building blocks (BBs), they also provide additional information about the problem being solved. The probabilistic model of the population, that represents (in)dependencies among decision variables, is an important source of information which can be exploited to enhance the performance of EDAs even more, or to assist the user in a better interpretation and understanding of the underlying structure of the problem. Examples of using structural information from the probabilistic model for another purpose beyond mixing include the design of structure-aware crossover operators (Yu, Goldberg, Sastry, Lima, & Pelikan, 2009), fitness estimation (Sastry, Pelikan, & Goldberg, 2004; Pelikan & Sastry, 2004; Sastry, Lima, & Goldberg, 2006), induction of global neighborhoods for mutation operators (Sastry & Goldberg, 2004; Lima, Pelikan, Sastry, Butz, Goldberg, & Lobo, 2006; Lima, 2009), hybridization and adaptive time continuation (Lima, Sastry, Goldberg, & Lobo, 2005; Lima, Pelikan, Sastry, Butz, Goldberg, & Lobo, 2006; Lima, Pelikan, Lobo, & Goldberg, 2009; Lima, 2009), substructural niching (Sastry, Abbass, Goldberg, & Johnson, 2005), offline (Yu & Goldberg, 2004) and online (Yu, Sastry, & Goldberg, 2007) population size adaptation, and the speedup of the model building itself (Hauschild, Pelikan, Sastry, & Goldberg, 2008; Hauschild & Pelikan, 2008). Therefore, it is important to understand under which conditions the structural accuracy of the probabilistic models in BOA and other structure-learning EDAs can be maximized. Recently, some studies have been done in this direction (Santana, Larrañaga, & Lozano, 2005; Wu & Shapiro, 2006; Correa & Shapiro, 2006; Echegoyen, Lozano, Santana, & Larrañaga, 2007; Hauschild, Pelikan, Sastry, & Lima, 2009; Mühlenbein, 2008). In the remainder of this section we take a brief look at these works.

Santana, Larrañaga, and Lozano (Santana, Larrañaga, & Lozano, 2005) analyzed the effect of selection on the arousal of bivariate interactions between single variables for random functions. They showed that for these functions, independence relationships not represented by the function structure are likely to appear in the probabilistic model. The authors also noted that even if the function structure plays an important role in the creation of dependencies, this role is mediated by selection (Santana, Larrañaga, & Lozano, 2005). Additionally, an EDA that only used a subset of the dependencies that exist in the data (malign interactions) was proposed. Some preliminary experiments showed that these approximations of the probabilistic model can in certain cases be applied to EDAs.

Wu and Shapiro (Wu & Shapiro, 2006) investigated the presence of overfitting when learning the probabilistic models in BOA and its consequences in terms of overall performance when solving random 3-SAT problems. CPTs (to encode the conditional probabilities) and the corresponding BIC metric were used. The authors concluded that overfitting does take place and that there is some correlation between this phenomenon and performance. The reduction in overfitting was proposed by using an early stopping criterion (based on cross entropy) for the learning process of BNs, which gave some improvement in performance.

The trade-off between model complexity and performance in BOA was also studied by Correa and Shapiro (Correa & Shapiro, 2006). They looked at the performance achieved by BOA as a function of a parameter that determines the maximum number of incoming edges for each node. This parameter puts a limit on the number of parents for each variable, simplifying the search procedure for a model structure. This parameter was found to have a strong effect on the performance of the algorithm, for which there is a limited set of values where the performance can be maximized. These results were obtained using CPTs and the corresponding K2 metric. We should note that in fact this parameter is crucial if CPTs are used with the K2 metric; however, this is not the case for more sophisticated metrics that efficiently incorporate a complexity term to introduce pressure toward simpler models. This can be done better with the BIC metric for CPTs, or with the K2 metric for the case of decision trees (Pelikan, 2005).

Echegoyen et al. (Echegoyen, Lozano, Santana, & Larrañaga, 2007) applied new developments in exact BN learning into the EDA framework to analyze the consequent gains in optimization. While in terms of convergence time the gain was marginal, the models learned by EBNA were more closely related to the underlying structure of the problem. However, the computational cost of learning exact BNs is only manageable for relatively small problem sizes (experiments were made for a maximum problem size of 20).

Hauschild et al. (Hauschild, Pelikan, Sastry, & Lima, 2009) made an empirical analysis of the probabilistic models built by hierarchical BOA for several test problems. The authors verified that the models learned closely correspond to the problem structure and do not change much over consequent iterations. They have also concluded that creating adequate probabilistic models for the 2D Ising spin glasses problem by hand is not straightforward even with complete knowledge of the problem. While in that work Hauschild et al. used truncation selection, this paper demonstrates that the results from (Hauschild, Pelikan, Sastry, & Lima, 2009) do not carry over to other selection methods that assign several copies of the same individual to the mating pool according to a non-uniform distribution.

Recently, Muhlenbein (Mühlenbein, 2008) investigated the Bayesian networks learned for LFDA and BOA when solving a trap-5 decomposable function. He found that in order to find the optimum about $80-90\%$ of the edges have to be correctly identified. Also, the penalty factor used for the BIC metric was shown to have influence on the network density. Although these results are relevant to better understand Bayesian EDAs, there is a fundamental difference from our study—the maximum number of incoming edges was set according to the problem structure—which reduces dramatically the overfitting phenomenon. In real-world optimization this is not typically the case, therefore we let the algorithm learn by itself the adequate complexity. Another important difference is that both LFDA and BOA used CPTs to encode the model parameters, as opposed to DTs used in this work. Additionally, while LFDA used truncation selection, BOA was paired with tournament selection, resulting in worse model quality when compared to LFDA (Mühlenbein, 2008). The author however did not make any remarks for the reason of such quality difference.

On the contrary, this work demonstrates that the difference in model quality is actually caused by the selection procedure rather than the algorithm as a whole. Furthermore, we show when and why overfitting is related to the selection method and propose a method to counterbalance this feature.

3 Analyzing Model Expressiveness

This section analyzes model learning in BOA when solving a problem of known structure and where that knowledge is crucial to solve it efficientlyga. We start by introducing the experimental setup used along the paper and then proceed to a detailed analysis of the learning process of BNs in BOA.

3.1 Experimental Setup for Measuring Structural Accuracy of Probabilistic Models

We start by clarifying some terms that are relevant to the scope of this paper.

Definition 1 *The* model structural accuracy *(MSA) is defined as the ratio of correct edges over the total number of edges in the Bayesian network.*

Definition 2 *An* edge *is* correct *if it connects two variables that are linked according to the objective function definition.*

Definition 3 Model overfitting *is defined as the inclusion of* incorrect (or unnecessary) edges *to the Bayesian network, which leads to excessive complexity.*

To investigate the MSA in BOA, we focus on solving a problem of known structure, where it is clear which dependencies must be discovered (for successful tractability) and which dependencies are unnecessary (reducing the interpretability of the models). The test problem considered is the