Applied Soft Computing 13 (2013) 2412–2432

Contents lists available at SciVerse ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

Regularized continuous estimation of distribution algorithms

Hossein Karshenas [a,*], Roberto Santana [b], Concha Bielza [a], Pedro Larrañaga [a]

[a] Computational Intelligence Group, Facultad de Informática, Universidad Politécnica de Madrid, Campus de Montegancedo, 28660 Boadilla del Monte, Madrid, Spain
[b] Intelligent System Group, Department of Computer Science and Artificial Intelligence, University of the Basque Country, Paseo Manuel de Lardizabal 1, 20080 San Sebastian-Donostia, Spain

ARTICLE INFO

Article history:
Received 19 April 2012
Received in revised form
26 November 2012
Accepted 28 November 2012
Available online 11 December 2012

Keywords:
Estimation of distribution algorithm
Regularized model estimation
Continuous optimization
High-dimensionality

ABSTRACT

Regularization is a well-known technique in statistics for model estimation which is used to improve the generalization ability of the estimated model. Some of the regularization methods can also be used for variable selection that is especially useful in high-dimensional problems. This paper studies the use of regularized model learning in estimation of distribution algorithms (EDAs) for continuous optimization based on Gaussian distributions. We introduce two approaches to the regularized model estimation and analyze their effect on the accuracy and computational complexity of model learning in EDAs. We then apply the proposed algorithms to a number of continuous optimization functions and compare their results with other Gaussian distribution-based EDAs. The results show that the optimization performance of the proposed RegEDAs is less affected by the increase in the problem size than other EDAs, and they are able to obtain significantly better optimization values for many of the functions in high-dimensional settings.

1. Introduction

Estimation of distribution algorithms (EDAs) [1–6] are a class of evolutionary algorithms based on estimating a *probability distribution model* for the space of possible candidate solutions to the given problem. This probabilistic model, which is learnt from a set of candidate solutions selected according to their quality, is used to generate new candidate solutions in the search space.

Assuming that the method used for generating new solutions is more likely to sample regions of the search space that have a higher probability, the ultimate goal of the model learning step in EDAs is to estimate probabilistic models that assign higher probabilities to a close neighborhood of optimal problem solutions (specified by the corresponding fitness function). It is almost impossible to estimate such a model directly at one go, especially for high-dimensional problems with a complex structure and high dimensionality.

Iterative model learning and factorized estimation of the probability distribution are two main techniques employed to facilitate model learning in EDAs. If the model is estimated across several generations, the algorithm can visit more regions of the search space and gradually improve its estimation as, due to the limitation of computational resources, algorithms have to work with a finite population of solutions. Techniques like univariate or bivariate factorization, or more generally, multivariate Bayesian network learning, which imposes a factorization over problem variables, are able to estimate the joint probability distribution as the product of simpler factors.

The use of probabilistic modeling in EDAs allows these algorithms to better exploit the information obtained up to the current stage of the search, in order to speed up convergence. Many of the probabilistic models employed in EDAs can also approximate the relationships or linkages between variables which is necessary for finding the optimal solutions to many problems. The successful application of EDAs to many real-world problems in different domains like: machine learning [7,8], bioinformatics [9–11], scheduling [12–14], industrial design and management [15,16], protein folding [17,18], software testing [19] and composite materials [20] have proved their usefulness in practice.

Despite promising performance for solving many real-world problems, there are still shortcomings in the behavior of EDAs that have made them the topic of active research. Several studies have tried to analyze the behavior of EDAs [21–26]. However, their results are mainly based on impractical assumptions or are limited to only specific problems. In continuous domains, especially, which is the scope of this paper, there are many difficulties with model estimation that prevent EDAs from exhibiting the expected behavior.

The ability of the chosen probabilistic model to fit the solutions of a given problem, which is referred to as model capacity [27], can greatly affect model estimation. Thanks to their analytical properties, Gaussian distributions have been the probabilistic model of

* Corresponding author. Tel.: +34 913363675; fax: +34 913524819.
E-mail addresses: hkarshenas@fi.upm.es, hosseinkarshenas@gmail.com (H. Karshenas), roberto.santana@ehu.es (R. Santana), mcbielza@fi.upm.es (C. Bielza), pedro.larranaga@fi.upm.es (P. Larrañaga).

1568-4946/$ – see front matter © 2012 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.asoc.2012.11.049

choice in most continuous EDAs [28–30,2,31]. However, a robust estimation of Gaussian distribution relies on acquiring adequate statistics that are often not available from the population of continuous EDAs. This will usually cause EDAs to fall into premature convergence (or rather stalemate). To overcome this shortcoming, techniques like variance scaling [32–34] or eigenvalue resetting [35,36] have been proposed in the literature.

Regularization techniques [37–40] are widely used in statistics and machine learning to obtain a more robust estimation of probabilistic models with lower prediction error. Regularized model estimation attempts to decrease the general prediction error of the estimated model by reducing the high variance caused for the prediction of new and unseen samples at the cost of introducing a little bias into the model [41,42]. The large-scale application of these techniques for model estimation, especially in high-dimensional problems where the number of samples is small compared with the number of variables, has proved useful.

Model estimation in EDAs has some characteristics that motivate the use of regularization techniques. Lack of adequate statistics can cause the estimated model to become highly biased to specific regions of the search space. This reduces its generalization ability which is an important factor when sampling the model. The use of regularization can reduce the generalization error of the estimated model in EDAs. Another important issue is the EDA scalability with regard to problem size. Estimating the probability distribution model of huge search spaces requires large population sizes. Since the model estimation and sampling parts of EDAs are very time-consuming, algorithm performance will decline steeply if population sizes are large, not to mention the memory constraints regarding large datasets. Being able to estimate a model of comparable quality using much smaller populations is a major requirement in these algorithms.

Very recently, regularization has been used in EDAs for discrete optimization. Yang et al. [43] used regularized regression in the context of a Bayesian optimization algorithm [44] to obtain a reduced set of candidate parents for each variable before searching for the correct Bayesian network structure. Luigi et al. [45] proposed the use of regularized logistic regression to learn the structure of the Markov network in the DEUM framework [46]. In a different context, Karshenas et al. [47] studied some of the methods for integrating regularization techniques into the model estimation of continuous EDAs.

This paper analyzes some of the methods to regularized model learning in EDAs and shows how they can be applied to continuous optimization in high-dimensional settings. The rest of the paper is organized as follows. Section 2 reviews some of the background material about continuous EDAs and regularization techniques, used in other sections. Section 3 discusses the incorporation of different regularization techniques into EDAs and studies their effect on model estimation using synthetic data. The results of applying the proposed algorithms on different well-known optimization functions are presented in Section 4. Finally, the conclusions and future perspectives are given in Section 5.

2. Background

2.1. Multivariate Gaussian distribution

A joint *multivariate Gaussian distribution* (MGD) for n random variables $X_1, \ldots, X_n$ is determined with two overall parameters: $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\mu}$ is an n-dimensional vector of mean values for each variable, and $\boldsymbol{\Sigma}$ is a $n \times n$ symmetric and positive semidefinite covariance matrix. The total number of individual parameters (free parameters) that have to be estimated in order to determine an MGD is $(n^2 + 3n)/2$, i.e. of $O(n^2)$ complexity.

Positive definite matrices are interesting since they are full-ranked and non-singular, implying that their inverse exists. The inverse of a positive definite covariance matrix, which is called the precision or concentration matrix, represents partial covariances between variables and any zero entry in this matrix implies that the corresponding two variables are conditionally independent given all other variables. Therefore, the zero pattern of the precision matrix directly induces the graphical structure of a Markov network. The positive definiteness of the covariance matrix also allows for a unique triangular decomposition, known as Bartlett or Cholesky decomposition [48], that can be used to generate samples from the corresponding MGD. These types of sampling algorithms have also been extended to work for MGDs with positive semidefinite matrices.

In many application domains, the covariance matrix of MGD is obtained with maximum likelihood (ML) estimation using a dataset of N samples, which is denoted with $\boldsymbol{S}$ (assuming row-wise vectors)

$$\boldsymbol{S} = \frac{1}{N-1}\sum_{i=1}^{N}(\boldsymbol{x}_i - \overline{\boldsymbol{x}})^{\mathrm{T}}(\boldsymbol{x}_i - \overline{\boldsymbol{x}}), \tag{1}$$

where $\overline{\boldsymbol{x}}$ is the ML estimation of $\boldsymbol{\mu}$. However, the covariance matrices obtained with ML estimation (Eq. (1)) usually result in a poor generalization of MGD [42,49]. For many applications, the covariance matrix should be positive definite or at least the partial correlations between the variables should be known. Therefore, several techniques for improving the estimation of the covariance matrix or its inverse have been proposed, some of which will be discussed in the following sections.

2.2. Estimation of distribution algorithms

Algorithm 1 shows the basic steps taken by an EDA for optimization. The algorithm starts from an initial population (step 1), which is usually generated randomly, though other techniques are applicable. In each generation, after selecting a subset of solutions according to their fitness values, a probability distribution model $\hat{\rho}_g(\boldsymbol{x})$ is learnt from the selected solutions to encode the general characteristics of these solutions (step 6). A set of new candidate solutions to the optimization problem is then generated using a sampling algorithm, which is incorporated into the EDA population (steps 7 and 9). This procedure is repeated until one of the stopping criteria (e.g. maximum number of generations, optimal solution(s), population convergence) is met (step 4).

Algorithm 1. The basic steps of an estimation of distribution algorithm

Estimation of Distribution Algorithm
Inputs:
A representation of solutions
An objective function f
1 $P_0 \leftarrow$ Generate initial population according to the given representation
2 $F_0 \leftarrow$ Evaluate individuals of P_0 using f
3 $g \leftarrow 1$
4 **while** termination criteria are not met **do**
5 $S_g \leftarrow$ Select a subset of P_{g-1} according to F_{g-1} using a selection mechanism
6 $\hat{\rho}_g(\boldsymbol{x}) \leftarrow$ Estimate the probability of solutions in S_g
7 $Q_g \leftarrow$ Sample $\hat{\rho}_g(\boldsymbol{x})$ according to the given representation
8 $H_g \leftarrow$ Evaluate individuals of Q_g using f
9 $P_g \leftarrow$ Incorporate Q_g into P_{g-1} according to F_{g-1} and H_g
10 $F_g \leftarrow$ Update F_{g-1} according to the solutions in P_g
11 $g \leftarrow g+1$
12 **end while**
Output: The best solution(s) in P_{g-1}