Chapter 10
Continuous Estimation of Distribution Algorithms Based on Factorized Gaussian Markov Networks

Hossein Karshenas, Roberto Santana, Concha Bielza, and Pedro Larrañaga

Abstract. Because of their intrinsic properties, the majority of the estimation of distribution algorithms proposed for continuous optimization problems are based on the Gaussian distribution assumption for the variables. This paper looks over the relation between the general multivariate Gaussian distribution and the popular undirected graphical model of Markov networks and discusses how they can be employed in estimation of distribution algorithms for continuous optimization. A number of learning and sampling techniques for these models, including the promising regularized model learning, are also reviewed and their application for function optimization in the context of estimation of distribution algorithms is studied.

10.1 Introduction

Approaches to continuous optimization with estimation of distribution algorithms (EDAs) [24], can be divided into two general categories: (i) Discretization of problem domain and then application of discrete EDAs [49]; (ii) Direct application of EDAs based on continuous probabilistic models [3, 5, 13].

In the latter approach, Gaussian distributions have been the probabilistic model of choice in most of the research in this area, considering either non-overlapping factorized distributions (e.g., Gaussian UMDA) or distributions defined by dependencies encoded in graphical models (e.g., Gaussian networks). The research on the use of undirected graphical models (Markov networks) in EDAs however, has been mainly focused on discrete domain optimization [39, 43–45]. In this paper,

Hossein Karshenas · Concha Bielza · Pedro Larrañaga
Computational Intelligence Group, Faculty of Informatics, Technical University of Madrid
e-mail: {hkarshenas,mcbielza,pedro.larranaga}@fi.upm.es

Roberto Santana
Intelligent Systems Group, Faculty of Informatics,
University of the Basque Country (UPV/EHU), San-Sebastian, Spain
e-mail: roberto.santana@ehu.es

S. Shakya and R. Santana (Eds.): Markov Networks in Evolutionary Computation, ALO 14, pp. 157–173.
springerlink.com

continuous EDAs based on Gaussian distribution are analyzed from the Gaussian Markov random field [46] perspective, a probabilistic graphical model successfully applied for handling uncertainty in many practical domains. It is shown that the analysis of undirected graphical models, as it is done in discrete Markov network-based EDAs, can be also extended to continuous domains. An important role in this analysis is played by the precision matrix which connects the dependencies between the variables to the Gaussian distributions. We focus on marginal product factorizations as a particular case of undirected graphical models. Next section will discuss the theoretical issues related to Gaussian distributions and their relation to Gaussian Markov random fields. Section 10.3 describes the main steps of the introduced algorithm. Special attention is given to different variants of incorporating regularization into model learning and the description of affinity propagation, the clustering technique used by the proposed EDA for finding the factorization of the distribution. Section 10.4 discusses previous works related to our proposal. The experiments that illustrate the behavior of the introduced approach using two different types of functions are presented in Section 10.5. Finally, Section 10.6 concludes the paper.

10.2 Multivariate Gaussian Distribution

Multivariate Gaussian distribution (MGD) is the most frequently used probability distributions for continuous optimization problems in estimation of distribution algorithms [2, 4, 28]. A multivariate Gaussian distribution $\mathcal{N}(\mu, \Sigma)$ over a vector of n random variables $X = (X_1, \ldots, X_n)$ is defined with two parameters: μ is the n-dimensional vector of mean values for each variable, and Σ is a positive-definite and symmetric $n \times n$ covariance matrix.

A square covariance matrix Σ is positive definite if $x\Sigma x^T > 0$, $\forall x \in \mathbb{R}^n \setminus \{0\}$. Positive definite matrices are guaranteed to be full-ranked and non-singular. When the '$>$' relation is replaced with '$\geq$', the square matrix becomes positive semi-definite which can be singular. Although in general MGDs can have positive semi-definite covariance matrices, but since only positive definite covariance matrices are invertible, this type of MGDs are only considered in this paper.

Geometrically, MGDs specify a set of parallel ellipsoidal contours around the mean vector. The mean vector determines the bias of each variable's values from origin and the variances, i.e., entries along the diagonal of the covariance matrix, are responsible for specifying the spread of values. The covariances (i.e., the off-diagonal entries in the covariance matrix) determine the shape of the ellipsoids. The mean vector and the covariance matrix are computed as the first two moments of the Gaussian distribution: (considering row-wise vectors)

$$\mu = E(X)$$
$$\Sigma = E((X-\mu)^T(X-\mu)) = E(X^T X) - \mu^T \mu$$

The typical representation of an MGD, sometimes referred to as the *moment* form, is given by

$$p_{\mathcal{N}(\mu,\Sigma)}(x) = \frac{1}{\sqrt{(2\pi)^n|\Sigma|}} \exp\left(-\frac{1}{2}(x-\mu)\Sigma^{-1}(x-\mu)^T\right) \tag{10.1}$$

This equation can be transformed to the *information* form representation of MGD [25], also known as the canonical or natural form

$$p_{\mathcal{N}^{-1}(h,\Theta)}(x) = \frac{\exp\left(-\frac{1}{2}h\Theta^{-1}h^T\right)}{\sqrt{(2\pi)^n|\Theta^{-1}|}} \exp\left(-\frac{1}{2}x\Theta x^T + xh^T\right) \tag{10.2}$$

where $h = \mu\Sigma^{-1}$ is called the potential vector and $\Theta = \Sigma^{-1}$ is the inverse covariance matrix, known as the *precision*, concentration or information matrix. For a valid MGD represented in the information form, the precision matrix should be positive definite.

10.2.1 Markov Networks

Similar to other probabilistic graphical models, a Markov network $\mathcal{M}(\mathcal{G}, \Phi)$ is composed of two components: (i) Graphical structure $\mathcal{G}$ and (ii) Parameters Φ. The nodes in the structure represent the variables, and the undirected edges correspond to probabilistic interactions between the neighboring nodes.

The parameters of a Markov network represent the affinities between related variables and the compatibility of their values. They are represented with factors $\phi_k \in \Phi$ (non-negative functions) that are defined over network cliques (complete subgraphs) $C_k \subseteq X$. The normalized product of these factors (according to factors multiplication rule) define the so called *Gibbs* distribution, factorized over the Markov network, which gives the joint probability distribution encoded in the network

$$p_{\Phi}(x) = \frac{1}{Z} \prod_{\phi_k \in \Phi, C_k \in \mathcal{C}} \phi_k(x_{C_k}) \tag{10.3}$$

where Z is the normalization term usually called the *partition* function, and $\mathcal{C}$ is a subset of network cliques that covers all variables in X.

10.2.2 Gaussian Markov Random Fields

When the interactions between the variables are symmetrical and there is no specific direction for the influence of the variables over each other, an undirected graph is more appropriate for representing the correlations. Markov networks are a type of probabilistic graphical models fitted to this need.

A widely used class of Markov networks is the pairwise Markov networks [15], where all of the factors are defined over either single or pairs of variables. More specifically, a pairwise Markov network has two types of factors:

i) Node factors $\phi_i(X_i)$ defined over every single variable X_i,
ii) Edge factors $\phi_{ij}(X_i, X_j)$ defined over the ending variables X_i and X_j of every edge.

MGDs and pairwise Markov networks are closely related. The precision matrix of an MGD represents partial covariances between pairs of variables variables. A zero value in any entry θ_{ij} of this matrix implies that the corresponding two variables are conditionally independent given all other variables and vice versa:

$$\theta_{ij} = 0 \quad \Longleftrightarrow \quad (X_i \perp X_j \mid X \setminus \{X_i, X_j\}) \in \mathscr{I}_p$$

where $\mathscr{I}_p$ represents the set of conditional independencies satisfied by MGD. This type of conditional independence is exactly the pairwise Markov property [29] encoded by Markov networks, and therefore the zero pattern of the precision matrix directly induces a Markov network, which is known as *Gaussian Markov Random Field* (GMRF) [38].

The structure of this Markov network is obtained by introducing an edge between every two nodes whose corresponding variables are partially correlated (a non-zero entry in the precision matrix). The parameters of the model are node and edge factors obtained, for example, by decomposing the variable part of the exponential factor of the MGD in Equation (10.2) to terms consisting of single and pairs of variables

$$\begin{aligned}\exp\Big(-\frac{1}{2}X\Theta X^T + Xh^T\Big) = \Big[&\exp(-\frac{1}{2}\theta_{11}X_1^2 + h_1X_1)\cdots \\ &\exp(-\frac{1}{2}\theta_{nn}X_n^2 + h_nX_n)\Big] \cdot \Big[\exp(-\theta_{12}X_1X_2)\cdots \\ &\exp(-\theta_{1n}X_1X_n)\cdots\exp(-\theta_{n-1,n}X_{n-1}X_n)\Big]\end{aligned}$$

The joint Gibbs distribution factorizing over this Markov network (Equation (10.3)) can then be obtained by computing the partition function

$$Z = \int_{-\infty}^{\infty} \exp\left(-\frac{1}{2}X\Theta X^T + Xh^T\right) dX = \frac{\sqrt{(2\pi)^n|\Theta^{-1}|}}{\exp\left(-\frac{1}{2}h\Theta^{-1}h^T\right)}.$$

While transforming an MGD to its corresponding GMRF is straightforward, thanks to the information representation in Equation (10.2), it should be noted that it is not possible to obtain a valid MGD from every pairwise Markov network with log-quadratic factors. This is because not every pairwise Markov network with log-quadratic factors will result in a positive definite precision matrix [25].

10.2.3 Marginal Product Factorizations

Factorization of the joint probability distribution can represent both marginal and conditional independence relationships between the variables. If the factorization is only based on the marginal independence relationships between the sets of variables

then it is called a marginal product factorization. Marginal product factorizations can be represented with directed and undirected probabilistic graphical models. In particular, when a GMRF represents a marginal product model (MPM), its cliques do not overlap and therefore it is possible to independently estimate the probability distribution for the variables in each clique.

In this paper, we consider EDAs using MPM representation of the probability distribution with GMRFs. This is appropriate when variables in the problem can be divided into a number of disjoint groups. Many of the real-world problems actually consist of several smaller components that are either independent or weakly connected. This type of GMRFs also allows to evaluate the learning algorithms we introduced in this chapter. In the discrete case, two well known examples of EDAs that use MPMs are the univariate marginal distribution algorithm (UMDA) [34] and the extended compact genetic algorithm (ECGA) [16]. In Section 10.4 we also review some of the continuous EDAs based on marginal product factorizations.

10.3 GMRF-Based EDA with Marginal Factorization

We have seen that GMRFs can represent independence relationships between continuous variables. The abstraction of the problem regularities captured by these undirected graphical models can be useful for continuous optimization in EDAs. A necessary step in incorporating undirected models in optimization is designing algorithms to feasibly learn models from data, from memory and time constraints points of view.

In this section we propose an approach for learning a subclass of GMRFs that represent MPMs. There are several alternatives for learning these undirected continuous models within EDAs, from which some are:

i) Estimation of MGD's covariance or precision matrix.
ii) Learning the structure of GMRF and its factors.
iii) Hybrid approaches.

The first approach includes maximum likelihood estimation as well as other covariance matrix selection techniques discussed in the literature [37, 50, 51]. The methods for inducing an undirected (in)dependence structure between variables are the typical choice in the second approach. Usually these methods use techniques like statistical hypothesis tests or mutual information (entropy) between variables to decide about their (in)dependence. Based on these (in)dependencies, the local neighborhood of each variable is obtained which can then be combined to obtain a full structure and compute the factors for the related variables. The third class of methods combines the computation of the covariance or precision matrix with the identification of the neighborhood structure of the variables. The approach introduced in this chapter belongs to the third class of methods.

10.3.1 A Hybrid Approach to Model Learning of GMRF

The main idea of our algorithm is to initially identify the putative neighborhood of each variable X_i by learning a regularized regression model [19, 48]. In this model, the dependence of X_i on each of the variables in $X \setminus X_i$ is represented with a weight. In the second step, variables are clustered into disjoint factors according to the strength of their dependence weights. Finally, an MGD is estimated for each factor using one of the methods in the first approach. The main steps of the proposed method are presented in Algorithm 10.1. Each of these steps are explained in detail in the following sections.

---

1. Set $t \Leftarrow 0$. Generate M points randomly.
2. **for** each variable
3. Compute its linear dependence weights on all other variables
4. Cluster the variables into disjoint cliques using the weight matrix
5. **for** each clique
6. Estimate an MGD for the variables in the clique

Fig. 10.1 Hybrid GMRF-EDA learning algorithm

10.3.2 Model Learning Using Regularization

Regularized model learning is one of the promising methods proposed in statistics for estimating a more accurate and sparser model. In regularization, the value of model parameters are penalized to shrink them toward zero. This is done by adding a regularization term of the form $\lambda J(\beta)$ to model estimation, where λ is the regularization parameter and $J(\cdot)$ is a function of model parameters (β_i). Some of the regularization techniques have the interesting property of setting some of the model parameters exactly equal to zero, thus implicitly performing variable selection [48].

The hybrid GMRF learning method proposed in this chapter, uses regularized linear regression models to find the dependencies between the variables. The parameters of the model learnt for each variable X_i, are the weights specifying the influence of other variables on the values of X_i. Thanks to regularized estimation, some of these weights will be exactly zero, suggesting that the two variables are independent, given the rest of variables. The regularization techniques considered in this study are:

- LASSO (lasso) or ℓ_1 regularization [48]: This technique penalizes the absolute values of the model parameters (regression coefficients), and can act as a variable selection operator. It is one of the most frequently used regularization techniques and different variants have been proposed for it in the literature.

- Elastic net (elnet) [52]: The penalization term in this technique is a linear combination of the previous ℓ_1 regularization term with an ℓ_2 regularization term [20], causing a grouping effect where correlated variables will be in or out of the model together.
- Least angle regression (lars) [9]: In this method the variables are added to the model (setting their coefficients to non-zero) one at a time based on their correlation with the error of estimation, and in exactly N steps, where N is the size of the learning data. This algorithm is able to compute the whole solution path (all possible coefficients configurations) of a regression model for different λ values very efficiently.

Regularization can also be applied for obtaining a regularized estimation of the covariance matrix or its inverse. In these methods, sparser estimation of covariance or precision matrices are obtained by applying regularization on their entries, causing marginal or conditional independencies between variables. To evaluate the proposed hybrid model learning algorithm, it is also compared with the following two regularization methods used for obtaining a regularized estimation of the covariance matrix:

- Shrinkage estimation (Shrink) [42]: In this method the sample covariance matrix, obtained from maximum likelihood estimation, is shrunk towards a target covariance matrix with smaller number of free parameters (e.g., a diagonal matrix). This method uses analytical relations to compute the proper value of the regularization parameter and can result in a statistically more efficient estimation of the covariance matrix.
- Graphical LASSO (Glasso) algorithm [12]: This method is based on obtaining a regularized maximum likelihood estimation of the MGD. An ℓ_1 regularization is applied on the entries of the precision matrix, forcing sparser Markov network structures. Here, the regularization parameter, determining the strength of penalization is left as an open parameter.

10.3.3 Clustering by Affinity Propagation

Clustering methods are used to group objects into different sets or clusters, in such a way that each cluster comprises similar objects. Clusters can then be associated to labels that are used to describe the data and identify their general characteristics. Among the the best known clustering algorithms are k-means [17] and k-center clustering [1].

Affinity propagation (AP) [11] is a recently introduced clustering method which takes as input a set of measures of similarity between pairs of data points and outputs a set of clusters of those points. For each cluster, a typical or representative member, which is called exemplar, is identified. We use affinity propagation as an efficient way to find the Markov network neighborhood of the variables from the linear weights output by regularized regression methods.

AP takes as input a matrix of similarity measures between each pair of points and a set of preferences which are a measure of how likely each point is to be chosen as

exemplar. The algorithm works by exchanging messages between the points until a stop condition, which reflects an agreement between all the points with respect to the current assignment of the exemplars, is satisfied. These messages can be seen as the way the points share local information in the gradual determination of the exemplars. For more details on AP, see [11]. In the context of GMRF-EDA, each variable will correspond to a point and the similarity between two points X_i and X_j will be the absolute value of weight w_{ij} obtained from the regularized regression model of variable X_i. Since in general $w_{ij} \neq w_{ji}$, the similarity matrix is not symmetric. AP also takes advantage of the sparsity of the similarity matrix obtained from regularized estimation, when such distribution of similarity values is available. The message-passing procedure may be terminated after a fixed number of iterations, when changes in the messages fall below a threshold, or after the local decisions stay constant for some number of iterations.

10.3.4 Estimating an MGD for the Factors

In the final step of the proposed hybrid model learning algorithm, an MGD of the variables in each factor is estimated. We expect that the factorized distribution obtained by this model estimation will give a more accurate estimation of the target MGD (of all variables) in comparison to learning a single multivariate distribution for all of the problem variables.

10.3.5 Sampling Undirected Graphical Models for Continuous Problems

Sampling is one of the most problematic steps in EDAs based on undirected graphical models. The problem is mainly related to the loops existing in the model structure that do not allow a straightforward application of simple sampling techniques like the probabilistic logic sampling (PLS) method [18]. The application of Markov Chain Monte Carlo (MCMC) methods like Gibbs sampling [14] also has a high computational complexity.

Obtaining decomposable approximations of the Markov networks to allow the application of PLS algorithm [39], merging cliques of the Markov network to capture as many dependencies as possible before applying exact or approximate sampling algorithms [21], and computation of the most probable configurations based on belief propagation [33] are among other options for sampling undirected graphical models that have been used in discrete EDAs and could be applied to continuous problems. Here, we independently sample the MGDs for each of the factors.

10.4 Related Work

A number of works have proposed the use of MPMs for continuous problems. In [10] and [30] two different algorithms are proposed that learn variants of ECGA

for real-valued problems. Both algorithms are based on discretizing the continuous variables previous to the construction of the MPM. Lanzi et al. [26] propose an ECGA that instead of mapping real values into discrete symbols, models each cluster of variables using a multivariate probability distribution and guides the partitioning process using an MDL metric for continuous distributions. To learn the clusters of variables, an adaptive clustering algorithm, namely the BEND random leader algorithm, is used.

Recently, Dong et al. [8] have proposed to compute the correlation matrix as an initial step to do a coarse learning such as identifying weakly dependent variables. These variables are independently modeled using a univariate Gaussian distribution while the other variables are randomly projected into subspaces (clusters) that are modeled using a multivariate Gaussian distribution.

Although the introduction of regularization techniques in EDAs is very recent, different variants of its application have been tried. In particular, the approach in which the local neighborhood of each variable is modeled with a regularized regression method and then a specific combination strategy is applied to aggregate these models [32] has been applied in different contexts. It has been incorporated into EDAs for optimization based on undirected graphical models both for discrete [40] and continuous domains [23]. In [31], the task of selecting the proper structure of the Markov network is addressed by using l1-regularized logistic regression techniques. In [35], the class of shrinkage estimation of distribution algorithms has been introduced. The results presented there show that shrinkage regularization can dramatically reduce the critical population size needed by EMNA in the optimization of continuous functions.

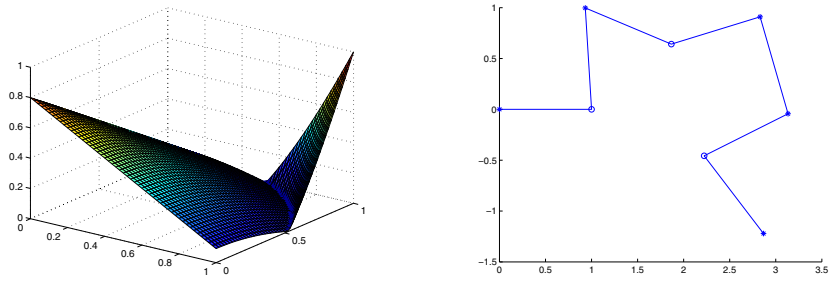
AP has been previously applied in evolutionary computation. It has been used to learn MPMs in the learning phase of EDAs that learn discrete MPMs [41]. In a different context, it has been applied as a niching procedure for EDAs based on Markov chain models [6].

10.5 Experiments

The main objective of our experiments is to study the proposed GMRF-EDAs in learning a factorized distribution model and compare their behavior to that of EDAs that use regularization methods for learning a single Gaussian distribution. The results are also compared with those of UMDAc [27], the EDA that assumes total independence between the variables. We also analyze the accuracy of the learning methods in recovering an accurate structure of the problem.

10.5.1 Benchmark Functions

To evaluate the performance of the algorithms, we use two classes of functions that represent completely different domains of difficulty in terms of optimization: An additive deceptive function and a simplified protein folding model.



a) b)

Fig. 10.2 a) 2-dimensional continuous trap function. b) One possible configuration of the Fibonacci sequence $S_5 = BABABBAB$.

The 2D-deceptive function [36] is composed of the aggregation of 2-dimensional trap functions which have a local optimum with a large basin of attraction and an isolated global optimum (Fig. 10.2a)), and should be maximized.

$$f_{2D-deceptive}(\mathbf{x}) = \sum_{i=1}^{n/2} f_{2D-trap}(x_{2i-1}, x_{2i})$$

where

$$f_{2D-trap}(x,y) = \begin{cases} 0.8 - \sqrt{\frac{x^2+y^2}{2}} & \text{if } \sqrt{\frac{x^2+y^2}{2}} \leq 0.8 \\ -4 + 5\sqrt{\frac{x^2+y^2}{2}} & \text{otherwise} \end{cases}.$$

Off-lattice models are simplified protein models that, in opposition to the HP simplified model [7], do not follow a given lattice topology. Instead, the 2D or 3D coordinate in the real axis define the positions of the protein residues. Among the off-lattice models with known lowest energy states is the *AB* model [47], where *A* stands for hydrophobic and *B* for polar residues. The distances between consecutive residues along the chain are held fixed to $b = 1$, while non-consecutive residues interact through a modified Lennard-Jones potential. There is an additional energy contribution from each angle θ_i between successive bonds. The energy function for a chain of n residues, that is to be minimized, is shown in equation (10.4).

$$E = \sum_{i=2}^{n-1} E_1(\theta_i) + \sum_{i=1}^{n-2} \sum_{j=i+2}^{n} E_2(r_{ij}, \zeta_i, \zeta_j), \tag{10.4}$$

where

$$E_1(\theta_i) = \frac{1}{4}(1 - cos\theta_i) \tag{10.5}$$

and

$$E_2(r_{ij}, \zeta_i, \zeta_j) = 4(r_{ij}^{-12} - C(\zeta_i, \zeta_j) r_{ij}^{-6}) \tag{10.6}$$

Here, r_{ij} is the distance between residues i and j (with $i < j$). Each ζ_i is either A or B, and $C(\zeta_i, \zeta_j)$ is $+1$, $+\frac{1}{2}$, and $-\frac{1}{2}$ respectively, for AA, BB, and AB pairs, giving strong attraction between AA pairs, weak attraction between BB pairs, and weak repulsion between A and B [22].

We only consider Fibonacci sequences defined recursively by

$$S_0 = A, \quad S_1 = B, \quad S_{i+1} = S_{i-1} * S_i \tag{10.7}$$

where $*$ is the concatenation operator. Fig. 10.2b) shows a possible configuration for sequence $S_5 = BABABBAB$.

A 2D off-lattice solution of the AB model can be represented as a set of $n-2$ angles. Angles are represented as continuous values in $[0, 2\pi]$. The first two residues can be fixed at positions $(0,0)$ and $(1,0)$. The other $n-2$ residues are located from the angles with respect to the previous bond. We look for the set of angles that defines the optimal off-lattice configuration. As instances, we consider Fibbonacci sequences for numbers $(5,6,7,8)$. The respective sizes of these sequences, in the same order, are $n \in (13,21,34,55)$.

10.5.2 EDA Parameters

All of the EDA variants used truncation selection with $\tau = 0.5$. The population size was $5n$ and the maximum number of allowed generations was 2000. We conduct 30 experiments for small instances ($n \leq 30$) and 15 experiments for larger instances.

10.5.3 Influence of the Regularization Parameter

In the first experiment, we investigated the accuracy of the structural learning algorithm as a function of the regularization parameter (λ). Analyzing the influence of the regularization parameter is very important to understand the way in which the introduction of regularization affects the behavior of the algorithms. We did this analysis for the 2D-deceptive function for which the problem structure is known.

Fig. 10.3 represents the typical precision matrices obtained at the end of an EDA run using regularized covariance matrix estimation, with two different configuration of this parameter, on a 10 variable 2D-deceptive function with cross-related variables (i.e. first with last, second with penultimate, etc.).

Fig. 10.3a) uses a constant penalization value throughout the whole EDA run while the other (Fig. 10.3b)) starts with a small value and dynamically increases it along the evolution. As it can be seen, the employed regularization technique allows the algorithm to obtain a very good estimation of the precision matrix and its corresponding GMRF structure.

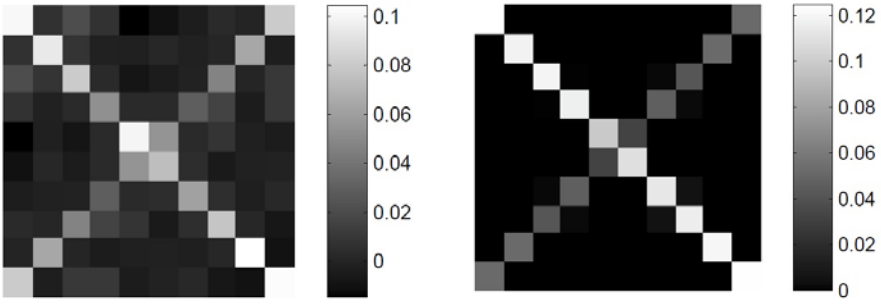


Fig. 10.3 Precision matrices learned by two different learning methods. a) $\lambda = 0.001$. b) Dynamic λ.

10.5.4 Optimization Performance

In the second step we compare the results of different GMRF-EDAs for the problems selected. Fig. 10.4 shows the average best value of the 2D-deceptive function ($n = 30$) in each generation. The average has been computed from the set of all experiments conducted. It can be appreciated that UMDAc, which considers a fixed total factorization of the distribution, starts outperforming other algorithms in the first generations but as the evolution continues, the initial lead of this algorithm is lost. On the other hand, EDAs based on regularized model learning using the graphical LASSO and shrinkage estimation methods, which consider no explicit factorization of the distribution, have a poor behavior and are not able to reach the best solutions achieved by UMDAc. The GMRF-EDAs using the proposed hybrid model learning algorithm, are able to constantly improve the average best value and obtain better results and the end of optimization.

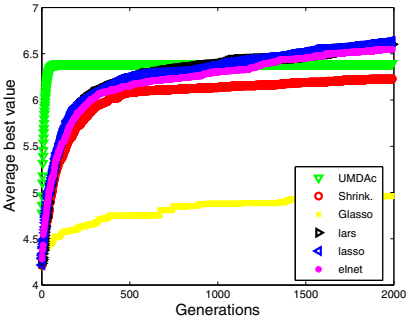


Fig. 10.4 Results of different EDAs for the $f_{2D-deceptive}$ function.

We also investigate the behavior of the algorithms for the off-lattice protein folding model. It is worth to mention that this is a very hard problem where the structural interactions between the variables are not clearly defined. There seems to be a clear dependence between adjacent variables in the AB sequence. However, the way in which other interactions between the AB residues are translated to dependencies in the model is not clear.

Fig. 10.5 shows the average best value of the off-lattice protein folding models corresponding to Fibonacci sequences 6 and 7 in each generation. For the first problem shown in (Fig. 10.5a)), it can be seen that UMDAc is outperformed by all other EDAs from the initial stages of the search. In this instance, the best results are achieved when estimating a single MGD with graphical LASSO, although the MPM learnt with elastic net regularization technique closely follows.

On the second instance of off-lattice protein problem (Fig. 10.5b)), the performance of UMDAc gets very close to EDAs using graphical LASSO and shrinkage estimation methods, and outperforming GMRF-EDAs that try to decompose the problem by learning dependencies between variables. There seems to be an important variability between the characteristics of the different off-lattice protein instances. In some situations such as the Fibonacci sequence number 6, capturing dependencies between the variables of the problem contributes to improve the quality of the obtained solutions. However, there are cases where explicitly learning the dependencies does not improve the results of the simpler univariate models.

Another interesting issue is to observe the disparate behavior exhibited by some EDAs like the one that uses graphical LASSO. It behaves very different in comparison to all other algorithms for 2D-deceptive function, achieving the worst results. Nevertheless, it reaches the best results for the off-lattice protein models. More research on this type of regularized model learning is needed to discern which mechanisms explain the difference between the performance of the algorithm for these two classes of functions.

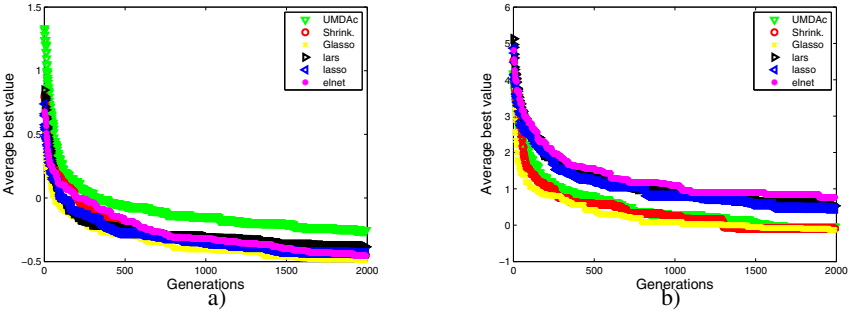


Fig. 10.5 Results of different EDAs for two different instances of the Off-lattice AB models. a) Sequence s_6. b) Sequence s_7.

10.6 Conclusions

This paper discussed how the multivariate Gaussian distribution, used by many continuous estimation of distribution algorithms, has a close relationship with a special type of Markov networks, namely the Gaussian Markov random field. It was shown how the (in)dependencies encoded in the precision matrix of the Gaussian distribution corresponds to the edges in the structure of the induced Markov network, and how factors of the network can be driven from the joint probability density function.

Based on this correspondence, some of the topics in learning and sampling these probabilistic graphical models, when employed in EDAs, were discussed. Specifically, some of the methods that can be used to approach the learning and sampling of Gaussian Markov random fields were presented. Regularized model learning was pointed out as a promising method for this purpose, especially in high dimensional problems.

We have introduced a GMRF-EDA that combines regularized regression with affinity propagation to learn regularized marginal product models. This is a different approach to learn more accurate MPMs for continuous problems. We have presented preliminary results on two different functions but more extensive experimentation is needed to determine how the characteristics of the regularization method influence the outcome of the EDAs.

It is worth mentioning that EDAs that learn undirected graphical models do not necessarily need to stick to Gaussian distributions and can employ other types of multivariate probability distributions once they have obtained the structure. The proposed hybrid model learning algorithm can be straitforwardly adopted for this purpose. The main goal of these approximations should be to exploit the information extracted from the set of promising solutions, allowing the EDA to explore promising areas of the search space.

Acknowledgements. This work has been partially supported by TIN2010-20900-C04-04, TIN2010-14931, Consolider Ingenio 2010-CSD2007-00018, Cajal Blue Brain projects (Spanish Ministry of Science and Innovation), the Saiotek and Research Groups 2007-2012 (IT-242-07) programs (Basque Government).

References

1. Agarwal, P., Procopiuc, C.: Exact and approximation algorithms for clustering. Algorithmica 33(2), 201–226 (2002)
2. Ahn, C.W., Ramakrishna, R.S., Goldberg, D.E.: Real-Coded Bayesian Optimization Algorithm: Bringing the Strength of BOA into the Continuous World. In: Deb, K., et al. (eds.) GECCO 2004. LNCS, vol. 3102, pp. 840–851. Springer, Heidelberg (2004)
3. Bengoetxea, E., Miquélez, T., Larrañaga, P., Lozano, J.A.: Experimental results in function optimization with EDAs in continuous domain. In: Larrañaga, P., Lozano, J.A. (eds.) Estimation of Distribution Algorithms. A New Tool for Evolutionary Computation, pp. 177–190. Kluwer Academic Publishers, Boston (2002)

4. Bosman, P.A.N., Thierens, D.: Expanding from Discrete to Continuous Estimation of Distribution Algorithms: The IDEA. In: Deb, K., Rudolph, G., Lutton, E., Merelo, J.J., Schoenauer, M., Schwefel, H.-P., Yao, X. (eds.) PPSN 2000. LNCS, vol. 1917, pp. 767–776. Springer, Heidelberg (2000)
5. Bosman, P.A.N., Thierens, D.: Numerical optimization with real-valued estimation-of-distribution algorithms. In: Pelikan, M., Sastry, K., Cantú-Paz, E. (eds.) Scalable Optimization via Probabilistic Modeling: From Algorithms to Applications. SCI, pp. 91–120. Springer (2006)
6. Chen, B., Hu, J.: A novel clustering based niching EDA for protein folding. In: Proceedings of the World Congress on Nature & Biologically Inspired Computing, NaBIC 2009, pp. 748–753. IEEE (2010)
7. Dill, K.A.: Theory for the folding and stability of globular proteins. Biochemistry 24(6), 1501–1509 (1985)
8. Dong, W., Chen, T., Tino, P., Yao, X.: Scaling up estimation of distribution algorithms for continuous optimization. CoRR, abs/1111.2221 (2011)
9. Efron, B., Hastie, T., Johnstone, I., Tibshirani, R.: Least angle regression. The Annals of Statistics 32(2), 407–499 (2004)
10. Fossati, L., Lanzi, P., Sastry, K., Goldberg, D., Gomez, O.: A simple real-coded extended compact genetic algorithm. In: Proceedings of the IEEE Congress on Evolutionary Computation, CEC 2007, pp. 342–348. IEEE (2007)
11. Frey, B.J., Dueck, D.: Clustering by passing messages between data points. Science 315, 972–976 (2007)
12. Friedman, J., Hastie, T., Tibshirani, R.: Sparse inverse covariance estimation with the graphical LASSO. Biostatistics 9(3), 432–441 (2008)
13. Gallagher, M., Frean, M., Downs, T.: Real-valued evolutionary optimization using a flexible probability density estimator. In: Proceedings of the Genetic and Evolutionary Computation Conference, GECCO 1999, Orlando, FL, vol. I, pp. 840–846. Morgan Kaufmann Publishers, San Francisco (1999)
14. Geman, S., Geman, D.: Stochastic relaxation, Gibbs distributions, and Bayesian restoration of images. IEEE Transactions on Pattern Analysis and Machine Intelligence 6(6), 721–741 (1984)
15. Hammersley, J.M., Clifford, P.: Markov fields of finite graphs and lattice. Technical report, University of California-Berkeley (1968)
16. Harik, G.: Linkage learning via probabilistic modeling in the ECGA. IlliGAL Report 99010, University of Illinois at Urbana-Champaign, Illinois Genetic Algorithms Laboratory, Urbana, IL (1999)
17. Hartigan, J., Wong, M.: Algorithm AS 136: A K-means clustering algorithm. Applied Statistics 28(1), 100–108 (1979)
18. Henrion, M.: Propagating uncertainty in Bayesian networks by probabilistic logic sampling. In: Lemmer, J.F., Kanal, L.N. (eds.) Proceedings of the Second Annual Conference on Uncertainty in Artificial Intelligence, pp. 149–164. Elsevier (1988)
19. Hesterberg, T., Choi, N., Meier, L., Fraley, C.: Least angle and L1 penalized regression: A review. Statistics Surveys 2, 61–93 (2008)
20. Hoerl, A.E., Kennard, R.W.: Ridge regression: Applications to nonorthogonal problems. Technometrics 12(1), 69–82 (1970)
21. Höns, R.: Estimation of Distribution Algorithms and Minimum Relative Entropy. PhD thesis, University of Bonn, Bonn, Germany (2006)
22. Hsu, H.-P., Mehra, V., Grassberger, P.: Structure optimization in an off-lattice protein model. Physical Review E 68(2), 4 Pages, article number 037703 (2003)

23. Karshenas, H., Santana, R., Bielza, C., Larrañaga, P.: Regularized model learning in estimation of distribution algorithms for continuous optimization problems. Technical Report UPM-FI/DIA/2011-1, Department of Artificial Intelligence, Faculty of Informatics, Technical University of Madrid (January 2011)
24. Kern, S., Müller, S.D., Hansen, N., Büche, D., Ocenasek, J., Koumoutsakos, P.: Learning probability distributions in continuous evolutionary algorithms– A comparative review. Natural Computing 3(1), 77–112 (2004)
25. Koller, D., Friedman, N.: Probabilistic Graphical Models: Principles and Techniques. Adaptive Computation and Machine Learning. The MIT Press (August 2009)
26. Lanzi, P., Nichetti, L., Sastry, K., Voltini, D., Goldberg, D.: Real-coded extended compact genetic algorithm based on mixtures of models. Linkage in Evolutionary Computation, 335–358 (2008)
27. Larrañaga, P., Etxeberria, R., Lozano, J., Peña, J.: Optimization in continuous domains by learning and simulation of Gaussian networks. In: Wu, A. (ed.) Conference on Genetic and Evolutionary Computation (GECCO 2000) Workshop Program, pp. 201–204. Morgan Kaufmann (2000)
28. Larrañaga, P., Lozano, J. (eds.): Estimation of Distribution Algorithms: A New Tool for Evolutionary Computation. Kluwer Academic Publishers, Norwell (2001)
29. Lauritzen, S.L.: Graphical Models. Oxford Statistical Science Series, vol. 17. Clarendon Press, Oxford (1996)
30. Li, M., Goldberg, D., Sastry, K., Yu, T.: Real-coded ECGA for solving decomposable real-valued optimization problems. Linkage in Evolutionary Computation, 61–86 (2008)
31. Malagó, L., Matteo, M., Gabriele, V.: Introducing l1-regularized logistic regression in Markov networks based EDAs. In: Proceedings of the 2011 Congress on Evolutionary Computation, CEC 2011, pp. 1581–1588. IEEE (2011)
32. Meinshausen, N., Bühlmann, P.: High-dimensional graphs and variable selection with the LASSO. Annals of Statistics 34(3), 1436–1462 (2006)
33. Mendiburu, A., Santana, R., Lozano, J.A.: Introducing belief propagation in estimation of distribution algorithms: A parallel framework. Technical Report EHU-KAT-IK-11/07, Department of Computer Science and Artificial Intelligence, University of the Basque Country (October 2007)
34. Mühlenbein, H., Paaß, G.: From Recombination of Genes to the Estimation of Distributions I. Binary Parameters. In: Ebeling, W., Rechenberg, I., Voigt, H.-M., Schwefel, H.-P. (eds.) PPSN 1996. LNCS, vol. 1141, pp. 178–187. Springer, Heidelberg (1996)
35. Ochoa, A.: Opportunities for Expensive Optimization with Estimation of Distribution Algorithms. In: Tenne, Y., Goh, C.-K. (eds.) Computational Intel. in Expensive Opti. Prob. ALO, vol. 2, pp. 193–218. Springer, Heidelberg (2010)
36. Pelikan, M., Goldberg, D.E., Tsutsui, S.: Getting the best of both worlds: Discrete and continuous genetic and evolutionary algorithms in concert. Information Sciences 156(3-4), 147–171 (2003)
37. Ravikumar, P., Raskutti, G., Wainwright, M., Yu, B.: Model selection in Gaussian graphical models: High-dimensional consistency of l1-regularized MLE. Advances in Neural Information Processing Systems (NIPS) 21 (2008)
38. Rue, H., Held, L.: Gaussian Markov Random Fields: Theory and Applications. Monographs on Statistics and Applied Probability, vol. 104. Chapman & Hall, London (2005)
39. Santana, R.: A Markov Network Based Factorized Distribution Algorithm for Optimization. In: Lavrač, N., Gamberger, D., Todorovski, L., Blockeel, H. (eds.) ECML 2003. LNCS (LNAI), vol. 2837, pp. 337–348. Springer, Heidelberg (2003)

40. Santana, R., Karshenas, H., Bielza, C., Larrañaga, P.: Regularized k-order Markov models in EDAs. In: Proceedings of the 2011 Genetic and Evolutionary Computation Conference, GECCO 2011, Dublin, Ireland, pp. 593–600 (2011)
41. Santana, R., Larrañaga, P., Lozano, J.A.: Learning factorizations in estimation of distribution algorithms using affinity propagation. Evolutionary Computation 18(4), 515–546 (2010)
42. Schäfer, J., Strimmer, K.: A shrinkage approach to large-scale covariance matrix estimation and implications for functional genomics. Statistical Applications in Genetics and Molecular Biology 4(1), 32 (2005)
43. Shakya, S.: Markov random field modelling of genetic algorithms. Technical report, The Robert Gordon University, Aberdeen, UK (2004)
44. Shakya, S., McCall, J.: Optimization by estimation of distribution with DEUM framework based on Markov random fields. International Journal of Automation and Computing 4(3), 262–272 (2007)
45. Shakya, S., Santana, R.: A Markovianity based optimisation algorithm. Genetic Programming and Evolvable Machines (2011) (in press)
46. Speed, T., Kiiveri, H.: Gaussian Markov distributions over finite graphs. The Annals of Statistics 14(1), 138–150 (1986)
47. Stillinger, F., Head-Gordon, T., Hirshfeld, C.: Toy model for protein folding. Physical Review E 48, 1469–1477 (1993)
48. Tibshirani, R.: Regression shrinkage and selection via the lasso. Journal of the Royal Statistical Society. Series B (Methodological) 58(1), 267–288 (1996)
49. Tsutsui, S., Pelikan, M., Goldberg, D.E.: Evolutionary algorithm using marginal histogram in continuous domain. In: Optimization by Building and Using Probabilistic Models, OBUPM 2001, San Francisco, California, USA, pp. 230–233 (July 2001)
50. Witten, D.M., Tibshirani, R.: Covariance-regularized regression and classification for high dimensional problems. Journal of the Royal Statistical Society. Series B. Methodological 71(3), 615–636 (2009)
51. Yuan, M., Lin, Y.: Model selection and estimation in the Gaussian graphical model. Biometrika 94(1), 19–35 (2007)
52. Zou, H., Hastie, T.: Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 67(2), 301–320 (2005)