Optimization by ℓ_1-constrained a Markov Fitness Modelling

Gabriele Valentini[1], Luigi Malagò[2], and Matteo Matteucci[2]

[1] IRIDIA, CoDE, Université Libre de Bruxelles,
gabriele.valentini@ulb.ac.be
[2] Department of Electronics and Information, Politecnico di Milano
{malago,matteucci}@elet.polimi.it

Abstract. When the function to be optimized is characterized by a limited and unknown number of interactions among variables, a context that applies to many real world scenario, it is possible to design optimization algorithms based on such information. Estimation of Distribution Algorithms learn a set of interactions from a sample of points and encode them in a probabilistic model. The latter is then used to sample new instances. In this paper, we propose a novel approach to estimate the Markov Fitness Model used in DEUM. We combine model selection and model fitting by solving an ℓ_1-constrained linear regression problem. Since candidate interactions grow exponentially in the size of the problem, we first reduce this set with a preliminary coarse selection criteria based on Mutual Information. Then, we employ ℓ_1-regularization to further enforce sparsity in the model, estimating its parameters at the same time. Our proposal is analysed against the 3D Ising Spin Glass function, a problem known to be NP-hard, and it outperforms other popular black-box meta-heuristics.

Keywords: Estimation of Distribution Algorithms, Markov Fitness Model, DEUM, ℓ_1-constrained Linear Regression, Least Angle Regression

1 Introduction

Black-box optimization consists of a set of meta-heuristics used to search for the optimum of a function when no information about its structure is available. Such approach to optimization can be used to define general purpose algorithms that do not depend on the function to be optimized, and it becomes the only possible approach when the mathematical formulation of the function is unknown. In particular model based meta-heuristics [25] introduce a statistical model to represent correlations among variables and to guide the search for the optimum.

Most of the model based meta-heuristics, cf. [25], use a probabilistic description of the given problem to drive their search towards solutions with the best value. The most general model is the joint probability distribution p which characterizes the correlations among all the variables involved in the objective function f. In the discrete setting, the probability simplex is able to capture

any possible order of interactions among the variables; however, its dimension equals the cardinality of the search space, and the estimation of its parameters is unfeasible.

In practice, most of the problems we are interested in, even NP-hard problems, are characterized by a limited set of correlations and can be characterized by a sparse pattern of interactions. It follows that model based search in a black-box context, to be really effective, must face the problem of selecting a lower dimensional model, which is computationally tractable, and would be able to capture all, or at least most, of relevant correlations. The family of the model and the way in which it is chosen define the particular class of meta-heuristics.

Once the model has been selected, model based algorithms implement different techniques to search for the optimal distribution in the model, for instance by applying estimation and sampling techniques, as in Estimation of Distribution Algorithm (EDA) [10], or by following the gradient of the expected value of f as in CMA-ES [8] and SNGD [11]. Within EDAs, a distinctive feature of the algorithms that belong to the Distribution Estimation Using Markov Networks (DEUM) [18] framework is the direct use of a probabilistic model of the objective function, which is sampled to search for a global minimum.

Selecting a model and estimating the parameters correspond, respectively, to a model selection and a model fitting problem, and in the general case are computationally expensive to address. On the other hand being able to recover the correct set of interactions, or at least a model that capture most of them, allows to work with tractable models with good properties, i.e., from any point the gradient of the expected value of the original function points in the direction of the global optimum, so that different optimization algorithms are less prone to end up with local minima, [12].

In DEUM, the joint probability distribution is represented using the formalism of Markov Networks (MNs) [22], also known as Markov Random Fields (MRFs), an example of undirected Probabilistic Graphical Models (PGMs). The structure of the MN, i.e., the set of conditional independences, can be either fixed a priori [18, 19], in which case we refer to fixed structure DEUM algorithms, or learned from scratch using model selection criteria such as Mutual Information [17] or χ^2-independence test [4]. Once the structure is identified, the parameters of the model are estimated from a subset of points with least square method, and then the model is sampled to look for a global optimum.

A common hypothesis when learning a model in EDAs is to limit the search to pairwise interactions. This reduces the number of possible interactions to $\binom{n}{2}$. Other additional hypothesis [4, 17] limit the maximum size of the neighbourhood of each variable to force a sparse pattern of interactions. A different approach to model selection in DEUM has been proposed in [13] where ℓ_1-regularized logistic regression is employed to recover the neighbourhood of each variable, cf. [16]. This choice allows to shrink the conditional probability distribution of a variable given its neighbourhood through a regularization parameter. The approach showed promising results both in terms of model selection and optimization performance. However, the computational effort was still very expensive.

The aim of this paper is to provide a novel method to estimate the statistical model used in DEUM by introducing a sparse model selection approach when estimating the Markov Fitness Model, thus dealing with model selection and model fitting at the same time. To obtain this result, we formalize the estimation problem as an ℓ_1-constrained linear regression problem, also known as the Lasso [20]. In this formulation, the penalizing ℓ_1-constraint addresses model selection, while the least square error minimization allows to estimate the coefficients of the model. Since candidate interactions grows exponentially in the problem size in the general case and quadratically if we restrict to pairwise interactions, we firstly use a preliminary coarse selection criteria based on Mutual Information to reduce the size of this set, similarly to the approach in [17], but with no constraint on the size of the neighbourhood.

The remaining of the paper is organized as follows. In Section 2 we describe the Markov Fitness Model underlying the DEUM framework. In Section 3 we introduce our approach based on ℓ_1-constrained linear regression to estimate the set of interactions and associated parameters of the model. In Section 4 we present the Sparsified DEUM algorithm (sDEUM), while in Section 5 we provide an empirical analysis of its performance using the well-known 3D Ising Spin Glass function [2] as a benchmark. The paper ends in Section 6 with conclusions and future directions of research.

2 Objective Function Modelling by Markov Networks

EDAs and more in general most model-based meta-heuristics make use of a statistical model, i.e., a set of probability distributions, to represent the interactions among the variables of an optimization problem. Usually the model is estimated from a subset of points, selected from a larger sample according to the value of f. The same applies for the algorithms in the DEUM framework, with the difference that the statistical model is employed to learn a model of f, rather than to estimate the correlations among its variables.

We consider the unconstrained optimization problem of minimizing a real-valued function f defined over a vector of n binary variables $X = (X_1, \ldots, X_n)$ with values in $\Omega = \{-1, +1\}^n$. Since the domain is finite, and $x^2 = 1$, any f can be written as a square-free polynomial

$$f(x) = \sum_{\alpha \in I} c_\alpha x^\alpha. \tag{1}$$

Here, we employ a notation based on the multi-index $\alpha = (\alpha_1, \ldots, \alpha_n) \in I \subset \{0, 1\}^n$, with $x^\alpha = \prod_{i=1}^n x_i^{\alpha_i}$. The associated real coefficients $c_\alpha \in \mathbb{R} \setminus \{0\}$ are indexed by α. Each monomial represents an interaction among a set of variables in f. We say that the set of interactions in function f is sparse if $\#(I) \ll 2^n$, where $\#(I)$ represents the cardinality of I. Many well known functions belong to this class, and even if the number of interactions is limited the optimization of such functions can be an NP-hard problem. For instance, the energy function of an Ising Spin glass problem [2] defined over a 3D toroidal lattice has $\#(I) = 3n$

interactions, where $l = \sqrt[3]{n}$ is the size of the grid. In the maximum cut [23] problem the cardinality of I corresponds to the number of edges in the graph, and in general $\#(I) \leq \binom{n}{2}$.

In the DEUM framework, probabilities of points in the search space Ω are assigned under the hypothesis that the probability of x should be proportional to the value of f, i.e.,

$$p(x) \equiv \frac{f(x)}{Z}, \qquad \text{with } Z = \sum_{x \in \Omega} f(x). \tag{2}$$

In particular, DEUM uses the Gibbs distribution as a statistical model, which is an example in the exponential family of distributions that can be equivalently represented with the formalism of MNs. The Gibbs distribution is used to learn a model of the objective function, by means of the Markov Fitness Model [3].

2.1 Markov Networks and Gibbs Distribution

Most EDAs make use of PGMs to represent the statistical model they use. In particular, the algorithms in the DEUM framework employ undirected graphical models called MNs. One of the advantages of a PGM is that the graph represents the conditional independence structure of the random variables, and provides a way to factorize the joint probability distribution associated to the graph.

Given a vector $X = (X_1, \ldots, X_n)$ of random variables, a MN is defined by a pair $(\mathcal{G}, \Phi)$, where $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ is an undirected graph and Φ is a set of local energy functions φ associated to the cliques. Each random variable X_i in X corresponds to a vertex $v_i \in \mathcal{V}$, while the edges $e_{ij} \in \mathcal{E}$ define the topology of the graph. We denote with $\mathcal{N}_i$ the *neighbourhood* of a variable X_i, i.e., the set of vertices v_j such that $e_{ij} \in \mathcal{E}$. A set X_C of fully connected vertexes of $\mathcal{G}$ is called *clique*. A clique is *maximal* if it is not contained in the set of vertices of any other clique.

The topology of the MN determines a set of conditional independence statements according to the absence of edges in the graph. As stated in the Hammersley-Clifford theorem [7], a positive probability distribution satisfies all the Markov properties with respect to the graph $\mathcal{G}$ if and only if it factorizes according to the graph itself. This implies that the joint probability distribution of X can be expressed as the product of a set of non-negative functions φ_C, called *potential functions*, defined over the clique $C \in \mathcal{C}$, i.e.,

$$p(x) = \frac{1}{Z} \prod_{C \in \mathcal{C}} \varphi_C(x_C) \tag{3}$$

where Z is a normalization constant that ensures the probabilities sum to 1. Without loss of generality, by absorbing cliques in maximal cliques, in the rest of the paper we restrict the factorization to the product of potential functions defined over the maximal cliques of $\mathcal{G}$.

Moreover, the Hammersley-Clifford theorem implies the equivalence of the probability distribution p in (3) associated to $\mathcal{G}$ and the *Gibbs* (or *Boltzmann*)

distribution of the form

$$p(x) = \frac{1}{Z} \mathrm{e}^{-U(x)/T}, \qquad \text{with } Z = \sum_{x \in \Omega} \mathrm{e}^{-U(x)/T}. \tag{4}$$

In statistical physics, the normalizing constant Z is called *partition function*, $T > 0$ is the *temperature* of the distribution, and $U(x)$ the *energy function*. The temperature parameter controls the sharpness of the distribution. Indeed, for $T \to \infty$, Equation (4) tends to the uniform distribution over Ω, while for $T \to 0$ the probability mass concentrates over the global minima of the energy function. The energy function of the Gibbs distribution is defined as the sum of local functions u_C associated to φ_C defined over the maximal cliques of $\mathcal{G}$, i.e.,

$$U(x) = \sum_{C \in \mathcal{C}} u_C(x_C). \tag{5}$$

In EDAs, the search space Ω is explored by sampling from a density in a statistical model. However, sampling from (4) is non trivial due to the presence of the partition function Z, whose computation requires a summation over the entire space Ω, and thus is unfeasible since it is exponential in n. Nevertheless, the Gibbs distribution can be sampled using a Gibbs sampler, and exploiting the local Markov property, so that the conditional probability of X_i only depends on its neighbourhood $\mathcal{N}_i$. Moreover, due to the $\{\pm 1\}$ encoding, we have

$$p_i(x_i|\mathcal{N}_i) = \frac{p(x)}{\sum_{x_i \in \{\pm 1\}} p(x)} = \frac{\mathrm{e}^{-U(x)/T}}{\sum_{x_i \in \{\pm 1\}} \mathrm{e}^{-U(x)/T}} = \frac{1}{1 + \mathrm{e}^{x_i \Delta_i U(x)/T}}, \tag{6}$$

where $\Delta_i U(x)$ is the difference between $U(x)$ and $U(\tilde{x}^i)$, where $\tilde{x}^i$ equals x except for the sign of x_i that has been changed. Since all terms in $U(x)$ and $U(\tilde{x}^i)$ agree except those containing x_i, and thus $\Delta_i U(x)$ only depends on $\mathcal{N}_i$, its computation can be further simplified.

2.2 The Markov Fitness Model

In the DEUM framework, probabilities are assigned to points in Ω proportionally to the value of f, and a model is chosen in the family of Gibbs distributions. By setting $T = 1$, in order to simplify the formulas, and combining Equations (2), (4) and (5), we have

$$p(x) \equiv \frac{f(x)}{\sum_{\Omega} f(x)} = \frac{e^{-\sum_{C \in \mathcal{C}} u_C(x_C)}}{\sum_{\Omega} e^{-\sum_{C \in \mathcal{C}} u_C(x_C)}},$$

that in particular is implied by setting

$$-\ln(f(x)) = \sum_{C \in \mathcal{C}} u_C(x_C), \tag{7}$$

i.e., when $U(x)$ is supposed to be a good model for f. This relationship between the energy function of the Gibbs distribution and f is called Markov Fitness Model (MFM) [3]. Notice that Equation (7) defines a probabilistic model of f.

Every u_C is defined over a subset of the variables in x according to the nodes in the maximal clique. Thus u_C admits a polynomial expansion as for f in Equation (1), and

$$-\ln(f(x)) = \sum_{C \in \mathcal{C}} \sum_{\alpha \in I_C} \theta_{\alpha,C} x^{\alpha}, \tag{8}$$

where the set of interactions identified by I_C depends on the variables in the maximal clique. Every $\theta_{\alpha,C} \in \mathbb{R}$ is a parameter associated to the expansion of u_C. By grouping similar terms and introducing a set M for all the monomials that appear in Equation (8), the expression can be simplified to

$$-\ln(f(x)) = \sum_{\alpha \in M} \theta_{\alpha} x^{\alpha}. \tag{9}$$

The statistical model used in the MFM in (9) can be written as an m-dimensional exponential family, with $m = \#(M)$,

$$p(x;\theta) = \exp\left\{ \sum_{\alpha \in M} \theta_{\alpha} x^{\alpha} - \psi(\theta) \right\}, \tag{10}$$

where $\psi(\theta) = \ln Z(\theta)$ is the normalizing factor and x^{α} are the sufficient statistics.

In order to reduce the number of parameters of the statistical model, further assumptions can be made in the choice of the monomials that appear in u_C in Equation (8). For instance, in the Ising DEUM algorithm [19], where the $\mathcal{G}$ is a toroidal 2D lattice and all maximal cliques have size 2, every $u_{ij}(x_i, x_j)$ takes the form of $\theta_{ij} x_i x_j$, so that all linear terms are not included among the sufficient statistics of the exponential model since they are not required to capture such class of Ising Spin Glass functions.

3 Sparse Learning of the Markov Fitness Model

To make the estimation of the MFM computationally feasible, we need to consider a reduced set of monomials as support statistics in (10) by imposing sparsity on the interactions pattern of the variables. This can be done a priori by making proper assumptions on the model, for instance limiting the neighbourhood size of each variable or the total number of interactions in the graph. On the other hand sparsity can be obtained by employing machine learning techniques such as ℓ_1-regularization in the estimation of the model. For instance, Ravikumar et al. [16] address sparse model selection by solving a set of n ℓ_1-constrained logistic regression problems. Other approaches, such as [9], solve the problem of sparse structure learning by evaluating pseudo-likelihoods. In the literature of discrete EDAs, some related methods have been applied in L1BOA [24] and DEUM$_{\ell_1}$ [13].

3.1 Problem Statement and Theoretical Approach

Let consider the MFM in Equation (8), where the set of monomials identified by indices in M defines a set of interactions among the variables in f. In the DEUM framework the coefficients θ are estimated by solving a linear system of equations. More in general the estimation of θ can be seen as a linear regression problem where, given a sample of observations, $-ln(f(x))$ corresponds to the response variable, and x^α to the covariates. By introducing a shrinkage regression technique in estimating the value of the parameters we can perform model selection by zeroing a subset of coefficients, and thus obtaining a sparse model. As a consequence, by applying a shrinkage technique in estimation, we can perform model selection at the same time of model fitting.

In particular we learn the MFM by solving an ℓ_1-constrained linear regression problem, also known as the Lasso [20]. The solution of the Lasso gives a sparse estimation of θ, hence, it selects a set of sufficient statistics for the statistical model of f in (10). The ℓ_1-constrained linear regression problem can be formalized as the minimization problem

$$\min_{\theta \in \mathbb{R}^m} \left\{ \frac{1}{2} || - \ln(f(x)) - \sum_{\alpha \in M} \theta_\alpha x^\alpha ||_2^2 + \lambda ||\theta||_1 \right\}, \tag{11}$$

where the first term represents the residual sum of squares, and the second term is an ℓ_1-constraint weighted by a control parameter λ, called *regularization parameter*. The value of the regularization parameter strongly affects the sparsity pattern of the vector of coefficients. Indeed, for $\lambda \to \infty$ all coefficients will vanish, while, for $\lambda \to 0$ the solution of the Lasso corresponds to the usual least square estimation of the MFM, which in general is not sparse.

To correctly dimension the value of the regularization parameter λ we refer to the asymptotic results presented in [5]. In particular dimensioning λ as

$$\lambda = K \sqrt{\frac{\log(m)}{N}}, \tag{12}$$

where K is a constant, m is the number of monomials in the exponential family, and N is the size of the sample used for the regression, guarantees that the correct correlations can be identified as $N \to \infty$. The same result has been applied in [16], where the authors show how N may depend on the topology of the graph. Such result is obtained under the hypothesis that the sample is i.i.d. from to an unknown probability distribution. Usually such hypothesis cannot be satisfied in black-box optimization, since f is unknown. In order to deal with this issue, we propose to perform ℓ_1-constrained linear regression over a subset of samples selected from a randomly generated initial sample according to the value of f. This procedure can only approximate an i.i.d. sample, but from our experiments it was sufficient to correctly reconstruct the topology of the MN.

A solution of the minimization problem defined in Equation (11) gives an estimation of the MFM that approximates a statistical model of f. However, the number of potential covariates in the regression problem grows exponentially

with n, making the minimization problem computationally unfeasible. Indeed its complexity is bounded by $\mathcal{O}(m^3)$. Even under the hypothesis of limiting the maximum order of interactions to the second, we have $m = \binom{n}{2}$ and the problem does not scale very well. For this reason, we propose to apply a rough selection procedure to reduce the set of covariates in the MFM before solving the Lasso.

3.2 Taking Care of Dimensionality: Candidate Edges Reduction

In order to reduce the complexity of the ℓ_1-constrained linear regression problem we only consider pairwise interactions among variables, so that the MFM in Equation (8) can be represented as a complete pairwise graph $\mathcal{G}(\mathcal{V}, \mathcal{E})$, such that $(i, j) \in \mathcal{E}$ for every $j > i$. However, the number of terms to consider still grows quadratically with n. In order to further reduce the number of edges before solving the Lasso, we select first a subset with a computationally lighter but yet less accurate method based on a measure of correlation among the variables.

Similarly to [17], we evaluate Mutual Information (MI) for each pair of random variables in the original function. MI is a metric that measures the mutual dependence between random variables. Given a pair of discrete random variables X_i and X_j, their Mutual Information $\mathcal{I}$ is defined as

$$\mathcal{I}(X_i, X_j) = \sum_{x_i, x_j \in \{\pm 1\}} p_{ij}(x_i, x_j) \log \left(\frac{p_{ij}(x_i, x_j)}{p_i(x_i) p_j(x_j)} \right), \tag{13}$$

where p_i and p_j are the marginal probabilities, and p_{ij} is their joint probability. If the MI between X_i and X_j is higher than a given threshold, then we include the associated monomial during the solution of the Lasso; otherwise, we remove the edge (i, j) from the graph.

The overall procedure can be summarized as follow. Given a sample we compute the Mutual Information matrix A, which is symmetric and has dimension $n \times n$. Then, we proceed by removing from the initial complete graph every edge (i, j) whose Mutual Information a_{ij} is lower than the threshold $b \cdot \overline{a}$, where b is a weight coefficient and $\overline{a}$ is the average Mutual Information of variables in X.

Such procedure allows us to reduce the candidate set of interactions in the regression problem. The optimal value of b such that only real interactions are recovered, strongly depends from both the original function f and the sample. In principle, correctly dimensioning b represents a hard task to address. Higher values of b cut most of the edges, while less restrictive choices give a dense network, that in turn, is more like to contain all the relevant interactions of the problem together with many undesired ones. We choose non-restrictive values for b since the purpose of this preliminary selection is to reduce the number of edges rather then selecting a good model for X. Shakya et al. [17] reduce furthermore the density of the network by making hypothesis on the maximum neighbourhood size of the nodes. We do not apply such step here, since we would like model selection to be as much independent as possible on prior knowledge about f, and leave to the Lasso the task of identifying the correct interactions.

Algorithm 1: sDEUM$(P, b, s_{mi}, s_{\ell_1})$

Let $\mathcal{E}$ be the set of edges of a fully connected pairwise MN
Randomly generate initial sample $\mathcal{P}$ of size P
Evaluate f for each point in $\mathcal{P}$
Select a subset $\mathcal{P}_{mi}$ from $\mathcal{P}$ of size $s_{mi}P$
Compute the MI matrix A and the average MI $\overline{a}$ given $\mathcal{P}_{mi}$
Select a subset of edges $\mathcal{E}_{mi}$ from $\mathcal{E}$ according to A and the threshold $b\overline{a}$
Let $m = \#(\mathcal{E}_{mi})$
Select a subset $\mathcal{P}_{\ell_1}$ from $\mathcal{P}$ of size $N = s_{\ell_1}P$
Estimate a distribution p in the MFM, by solving the Lasso with covariates associated to $\mathcal{E}_{mi}$ and observations in $\mathcal{P}_{\ell_1}$, with $\lambda = K\sqrt{\frac{\log m}{N}}$, as in Eq. (11)
Sample p with the Gibbs sampler by evaluating conditional probabilities in Eq. (6)

4 Shrinkage DEUM Optimization Algorithm

In this section, in the light of the machine learning techniques described in the first part of the paper, we present the *Shrinkage Distribution Estimation Using Markov Networks* algorithm (sDEUM). The sDEUM algorithm consists of a black-box meta-heuristics able to learn from scratch a sparse probabilistic model of the function to be minimized. The use of ℓ_1-penalized linear regression allows sDEUM to shrinkage the size of the θ parameters of the MFM. Due to the ℓ_1 constraint and according to the size of the λ parameter, some coefficients are fixed to zero with high probability, so that an implicit model selection is performed while solving the regression problem. The model is then sampled to generate new points, possibly with optimal values for f.

Algorithm 1 summarizes the procedure implemented in sDEUM. The meta-heuristic is characterized by some parameters: the population size P, the MI coefficient b, the percentages of selection s_{mi} and s_{ℓ_1}, and the constant K. Two different subsets are selected from the same initial random sample: $\mathcal{P}_{mi}$ for the computation of the MI matrix and $\mathcal{P}_{\ell_1}$ for solving the Lasso, respectively. This choice allows a better sizing of observations for the two different estimation tasks. In both cases a truncation selection operator is employed, other policies are possible, but they are not investigated here.

Once the MFM is estimated, next step in the DEUM framework consists of sampling the distribution to search for the optimum of f. In sDEUM, as in [19, 13], we use a Gibbs sampler, i.e., a Monte Carlo Markov Chain sampling method. The Gibbs sampler allows to generate instances with minimum values for the energy U of the Gibbs distribution by cooling the temperature during sampling. Refer to [19] for a presentation of the sampling schema employed in DEUM.

When the estimated model is good enough, repeatedly sampling the model with an adequate cooling schema yields with high probability the global optima of f. As a consequence, as in most of the DEUM framework algorithms, model learning in sDEUM is performed only once, and the learned model is repeatedly sampled using the Gibbs sampler (single generation approach).

5 Empirical Performance Analysis

In this section we present the results of an empirical analysis of the performance of sDEUM. We set up a series of experiments in order to evaluate the ability of the algorithm to reconstruct the correct set of interactions among the variables in the model and to find the global minimum of the function. In all the experiments, we evaluated the performance using the 3D Ising spin glass problem [2] as a benchmark, whose interaction structure is known and can be used to determine a set of model selection statistics, such as precision, recall and F1 score. First we analyse the behaviour of sDEUM when its parameters are changed, in order to find the best configuration, then we compare its performance in solving the energy minimization problem with those of DEUMce [17], Simulated Annealing [1] and hBOA [15]. DEUMce and Simulated Annealing have been tuned to achieve best performance, while results of hBOA are taken from [14]. Since the difficulty of the 3D Ising spin glass problem may depend from the particular instance, we averaged the results over 10 different instances, and for each of them we run 30 independent executions of every algorithm. In order to simplify the experimental comparison and evaluation, sDEUM, DEUMce and Simulated Annealing were implemented within the Evoptool toolkit [21]. The source code of the algorithms and the Ising spin glass instances can be found on the Evoptool homepage[3].

5.1 Experimental Setting and 3D Ising Spin Glass Problem

In statistical physics, the Ising spin glass problem is an energy minimization problem in the space of binary configurations of a set of spins $\sigma = (\sigma_1, \ldots, \sigma_n)$, where each spin can be either up, $\sigma_i = +1$, or down, $\sigma_i = -1$. The optimal solutions, i.e., the ground states of the spin glass, are those configurations that minimize the energy function

$$E(\sigma) = -\sum_{i \in L} h_i \sigma_i - \sum_{i<j \in L} J_{ij} \sigma_i \sigma_j, \tag{14}$$

where L is a toroidal lattice of n sites, while h_i and J_{ij} are coupling constants respectively of a single spin σ_i and a pair of spins (σ_i, σ_j). The difficulty of the problem is strongly related to the dimensionality of the lattice. Indeed, even if with particular choices of h and J the problem in 1D and 2D can be solved in polynomial time, it becomes NP-hard for all kind of coupling constants, as soon as it reaches the third dimension, and in particular when the edge degree of each vertex equals 6, see [2].

In our experiments we use spin glasses defined over a 3D grid with periodic boundaries [2]. The contribution to the energy given by singleton spins is not taken into account, therefore $h_i = 0$ for every spin. The instances of the problem are randomly generated with couplings J_{ij} that takes values in $\{\pm 1\}$ with equal probability. Instances of the problem and their optimal solutions are generated using the spin glass ground states server by the group of Prof. Michael Jünger[4].

[3] Available at `http://airwiki.ws.dei.polimi.it/index.php/Evoptool`

[4] Available at `http://www.informatik.uni-koeln.de/ls_juenger/research/sgs/`

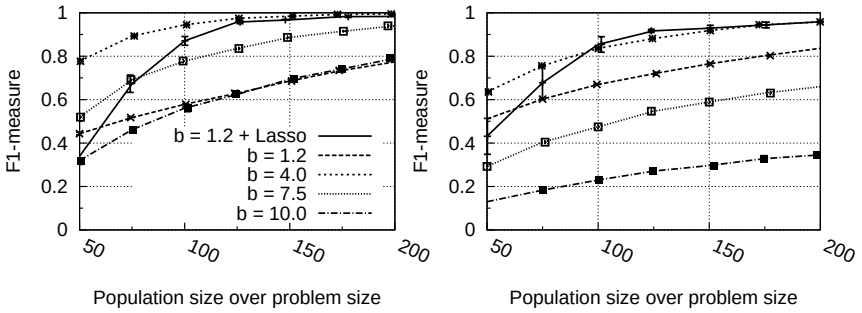

Fig. 1. F1 measure of model selection based on MI vs preliminary selection based on MI followed by ℓ_1-constrained linear regression for the Ising spin glass problem for $n = 64$, (left) 2D lattice; (right) 3D lattice.

The sDEUM algorithm has been run for different values of its parameters: the sample size P, the threshold coefficient of MI b, and the percentages of selection s_{mi} and s_{l1}. After preliminary tests, the constant K in (12) has been fixed to the value of $K = 16$. In particular, to solve the ℓ_1-constrained linear regression problem, we employed the R package `lars` available on CRAN, implementing the Least Angle Regression (LARS) [6] algorithm.

The performance of sDEUM is compared with those of DEUMce, Simulated Annealing (SA) and the Hierarchical Bayesian Optimization Algorithm (hBOA). DEUMce is a DEUM algorithm with model learning capability based on the evaluation of the Mutual Information plus a structure refinement mechanism that bounds the maximum edge degree of each node. SA is a meta-heuristic characterized by the number P of starting points, by the initial temperature T and the cooling rate c of the Metropolis sampler. The hBOA algorithm is an optimization meta-heuristic belonging to the family of EDAs based on Bayesian Networks (BNs). At each generation, hBOA employs a niching mechanism to select individuals in the population. The sample of individuals is used to learn a BN, which in turns is sampled to produce the set of solutions. For further details on the implementations of DEUMce, SA and hBOA refer to [17], [1] and [15], respectively.

The performance of the algorithm is evaluated according to a set of statistics that concerns the F1 measure, the probability of success and the average number of evaluations of f required to find the first ground state at each execution. In particular, the F1 measure is defined as the harmonic mean of precision and recall statistics, while the probability of success is computed as the rate of successful executions, i.e., the percentage of runs in which at least an optimal solution is sampled.

5.2 Impact of Learning Parameters

In order to successfully minimize a given objective function f, it is essential to recover a good statistical model for the variables in the problem, i.e., to learn

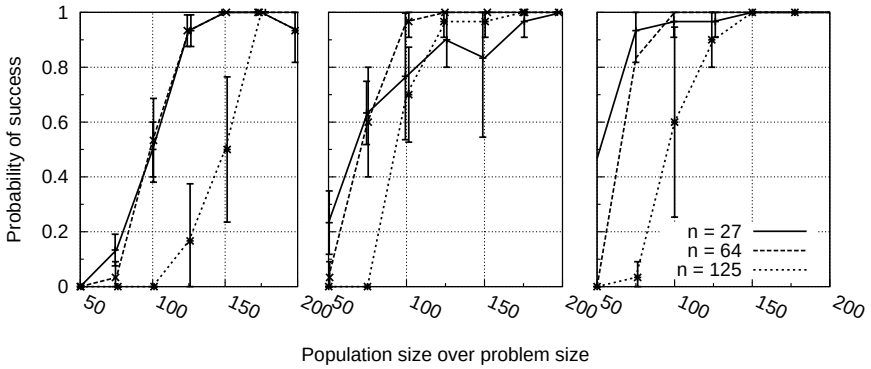

Fig. 2. Probability of success over normalized size of initial population (P/n). Benchmark: 3D Ising Spin Glass function, $n \in \{27, 64, 125\}$. sDEUM parameters: $s_{mi} = 0.3$; (left) $b = 1.5, s_{\ell_1} = 0.1$; (center) $b = 1.5, s_{\ell_1} = 0.3$; (right) $b = 1.1, s_{\ell_1} = 0.3$.

most of the interactions present in f and to correctly estimate the value of their parameters.

The threshold coefficient b of the preliminary selection based on MI, as well as the λ parameter of the Lasso, determine the sparsity level of the recovered structure. However, to correctly dimension b a preliminary tuning phase which depends on the problem is usually required, while, in contrast, the λ parameter can be chosen according to Equation (12) to ensure good theoretical performance.

In Fig. 1 we compared the model selection performance of our approach with those of the model selection based on MI, when solving the Spin Glass function with 2D and 3D structure and 64 variables. As we can see, in case of model selection based only on MI the results vary greatly according to the value of b. In contrast, when MI is followed by the ℓ_1-constrained regression, the choice of value for b results less problem dependent. Indeed, thanks to the ℓ_1-constraint, the value of b required to recover a good model can be chosen in a broader range of less selective values.

In Fig. 2 we can see the probability of success plotted against the size of the initial population P for problem size $n \in \{27, 64, 125\}$, and for different values of the threshold coefficient $b \in \{1.1, 1.5\}$, that determines how dense the MFM is after preliminary model selection based on MI. When $n = 27$ or $n = 64$, a less restrictive value of b provides better performance, see Fig. 2 (right); while when the size of the problem increases, $n = 125$, a higher value of the coefficient b results in earlier convergence, see Fig. 2 (center).

These results suggest that the value of b should increase with n. A possible explanation is given by the fact that the number of interactions grows linearly as $3n$ for a 3D lattice, while the number of total interactions is quadratic, for this reason a more restrictive choice of b helps to reduce the number of candidate interaction before the Lasso is solved.

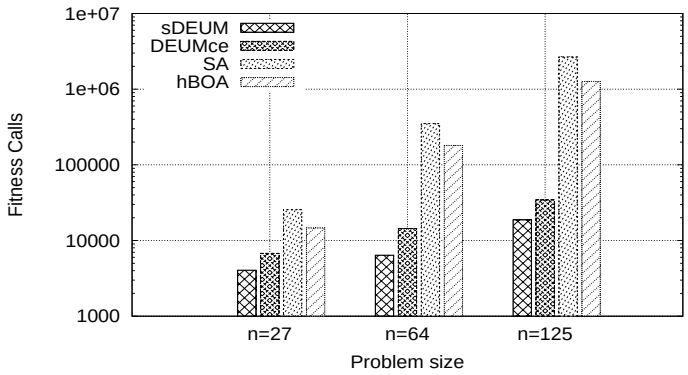

Fig. 3. Average number of evaluations of f (log scale) over problem size required to find first optimal solution with probability 1. Benchmark: 3D Ising Spin Glass function, $n \in \{27, 64, 125\}$. Algorithms: sDEUM, DEUMce, SA, hBOA.

In a black-box scenario an i.i.d. sample is not available to solve the lasso. Instead we choose a subset of the sample based on the value of the function f, and we compared the performance of the algorithm for different values of s_{ℓ_1}. In Fig. 2(left) and Fig. 2(center) we show the results for s_{ℓ_1} equal to 0.1 and 0.3, respectively. It is possible to notice that even if selection helps identify a good sample with respect to the random observations generated when the algorithms starts, decreasing that percentage too much results in lower performances. This result suggests that if the output of a selection is a sample not informative enough, then we have preliminary convergence and a larger population is necessary.

5.3 Analysis of Optimization Performance

In this section we compare the performance of sDEUM to find the ground states of the 3D Ising Spin Glass function with those of DEUMce, Simulated Annealing and hBOA. We analyse the results in terms of average amount of evaluations of the objective function required to find the optimum with a probability of success equals to 1 for each algorithm on 10 instances of the problem. The parameters of sDEUM, DEUMce and Simulated Annealing have been chosen after a preliminary tuning phase on this set of experiments. This was not possible for the hBOA algorithm, and results provided[5] here are taken from [14].

The trend highlighted in Fig. 3 suggests that sDEUM algorithm requires a lower number of evaluations of f with respect to other meta-heuristics on this benchmark. Indeed, the overall number of evaluations for sDEUM appears

[5] The performance of hBOA in [14] are evaluated over a set of instances of the 3D Ising Spin Glass function different from our set but with the same setting: 3D toroidal lattice, $h_i = 0$, $J_{ij} \in \{\pm 1\}$.

to grow polynomially as $\mathcal{O}(n^{2.04})$, while the same metric grows as $\mathcal{O}(n^{2.16})$, $\mathcal{O}(n^{3.06})$ and $\mathcal{O}(n^{2.91})$, for DEUMce, SA and hBOA [14], respectively.

The lower requirements in terms of fitness evaluations of both sDEUM and DEUMce with respect to SA and hBOA are due to the single iteration approach characteristic of the DEUM algorithms. Indeed, most of the evaluations in DEUM, concern the initial sample, before selection is applied. Moreover, the shrinkage method used in sDEUM compared with the approach of DEUMce based on MI and structure refinement is able to recover a good model with a smaller sample of observations and thus further reduce the number of evaluations of the objective function.

6 Conclusions

In this paper we presented a novel approach to the estimation of the MFM based on ℓ_1-regularized linear regression. Our proposal allows to perform both model selection and model fitting at the cost of solving a single regularized linear regression problem. The advantage of this approach is due to theoretical results on the dimensioning of λ, that in contrast to the threshold parameter of Mutual Information, permits to be more robust and less problem dependent.

In the context of the DEUM framework, we developed a novel algorithm called sDEUM that estimates the MFM using an approach based on shrinkage regression. In order to make Lasso more efficient, sDEUM uses a preliminary σ coarse selection based on Mutual Information in order to find a candidate set of interactions for the MFM. This candidate set is then used to solve the regularized regression problem by means of Least Angle Regression (LARS). We showed that sDEUM is able to learn a probabilistic description of the objective function and to successfully use it to address optimization. We remark lower requirements in terms of number of evaluations of f to reach optimality with respect to other popular algorithms in the EDA framework. In particular, solving the Lasso defined on the MFM allows to reduce the necessary number of observations with respect to performing ℓ_1-regularized logistic regression on the conditional probability distribution of each variable, as previously done in DEUM$_{\ell_1}$ [13].

Acknowledgements. This work was supported by the META-X project, an Action de Recherche Concertée funded by the Scientific Research Directorate of the French Community of Belgium, and by the PRIN 2009 project ROAMFREE (Robust Odometry Applying Multisensor Fusion to Reduce Estimation Errors) funded by the Italian Ministry of University and Research.

References

1. E. Aarts and J. Korst. *Simulated annealing and Boltzmann machines: a stochastic approach to combinatorial optimization and neural computing.* John Wiley & Sons, Inc., New York, NY, USA, 1989.

2. F. Barahona. On the computational complexity of Ising spin glass models. *Journal of Physics A: Mathematical and General*, 15(10):3241–3253, 1982.
3. D. Brown, A. Garmendia-Doval, and J. McCall. Markov random field modelling of royal road genetic algorithms. In *Artificial Evolution*, pages 35–56. Springer, 2002.
4. A. Brownlee, J. McCall, S. Shakya, and Q. Zhang. Structure learning and optimisation in a Markov Network based Estimation of Distribution Algorithm. *Exploitation of Linkage Learning in Evolutionary Algorithms*, pages 45–69, 2010.
5. F. Bunea, A. Tsybakov, and M. Wegkamp. Sparsity oracle inequalities for the lasso. *Electronic Journal of Statistics*, 1:169, 2007.
6. B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *The Annals of statistics*, 32(2):407–499, 2004.
7. J. Hammersley and P. Clifford. Markov fields on finite graphs and lattices. 1971.
8. N. Hansen, S. Müller, and P. Koumoutsakos. Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES). *Evolutionary Computation*, 11(1):1–18, 2003.
9. H. Höfling and R. Tibshirani. Estimation of sparse binary pairwise markov networks using pseudo-likelihoods. *JMLR*, 10:883–906, 2009.
10. P. Larrañaga and J. A. Lozano, editors. *Estimation of Distribution Algoritms. A New Tool for evolutionary Computation*. Springer, 2001.
11. L. Malagò, M. Matteucci, and G. Pistone. Stochastic natural gradient descent by estimation of empirical covariances. *Proceedings of IEEE CEC*, 2011.
12. L. Malagò, M. Matteucci, and G. Pistone. Towards the geometry of Estimation of Distribution Algorithms based on the exponential family. In *Proceedings of XI Foundation of Genetic Algorithms (FOGA)*, January 2011.
13. L. Malagò, M. Matteucci, and G. Valentini. Introducing ℓ_1-regularized logistic regression in Markov Networks based EDAs. *Proceedings of IEEE CEC*, 2011.
14. M. Pelikan, D. Goldberg, J. Ocenasek, and S. Trebst. Robust and scalable black-box optimization, hierarchy, and ising spin glasses. Technical report, IlliGAL, 2003.
15. M. Pelikan and D. E. Goldberg. A hierarchy machine: Learning to optimize from nature and humans. *Complexity*, 8(5):36–45, 2003.
16. P. Ravikumar, M. J. Wainwright, and J. D. Lafferty. High-dimensional Ising model selection using ℓ_1-regularized logistic regression. *The Annals of Statistics*, 2010.
17. S. Shakya, A. Brownlee, J. McCall, F. Fournier, and G. Owusu. A fully multivariate DEUM algorithm. *Proceedings of IEEE CEC*, 2009.
18. S. Shakya and J. McCall. Optimization by Estimation of Distribution with DEUM framework based on Markov random fields. *IJAC*, 4(3):262–272, 2007.
19. S. Shakya, J. McCall, and D. Brown. Solving the Ising spin glass problem using a bivariate EDA based on Markov random fields. *Proceedings of IEEE CEC*, 2006.
20. R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 267–288, 1996.
21. G. Valentini, L. Malagò, and M. Matteucci. Evoptool: an extensible toolkit for evolutionary optimization algorithms comparison. *Proceedings of IEEE WCCI*, 2010.
22. G. Winkler. *Image Analysis, Random Fields and Dynamic Monte Carlo Methods: A Mathematical Introduction.* Springer, second edition, 2003.
23. L. A. Wolsey. *Integer Programming*. Wiley-Interscience, 1998.
24. J. Yang, H. Xu, Y. Cai, and P. Jia. Effective structure learning for EDA via L1-regularized bayesian networks. *Proceedings of GECCO*, 2010.
25. M. Zlochin, M. Birattari, N. Meuleau, and M. Dorigo. Model-based search for combinatorial optimization: A critical survey. *Ann Oper Res*, 131(1-4):375–395, 2004.