Implicit Model Selection based on Variable Transformations in Estimation of Distribution

Emanuele Corsano, Davide Cucci, Luigi Malagò, and Matteo Matteucci

Department of Electronics and Information, Politecnico di Milano
Via Ponzio, 34/5, 20133 Milano, Italy
emanuele.corsano@mail.polimi.it {cucci,malago,matteucci}@elet.polimi.it

Abstract. In this paper we address the problem of model selection in Estimation of Distribution Algorithms from a novel perspective. We perform an implicit model selection by transforming the variables and choosing a low dimensional model in the new variable space. We apply such paradigm in EDAs and we introduce a novel algorithm called I-FCA, which makes use of the independence model in the transformed space, yet being able to recover higher order interactions among the original variables. We evaluated the performance of the algorithm on well known benchmarks functions in a black-box context and compared with other popular EDAs.

Keywords: Estimation of Distribution Algorithms, Transformation of Variables, Implicit Model Selection, Minimization of Mutual Information.

1 Introduction

Estimation of Distribution Algorithms (EDAs) belong to the class of metaheuristics for optimization where the search is guided by a statistical model able to capture the interactions among the variables in the problem. The choice of the model is crucial, indeed much of the literature in the EDAs community is focused on applying machine learning techniques for model selection, able to identify the correct interactions among the variables from a sample of observations. Some examples are the algorithms which learn the structure of a Bayesian Network, as in the Bayesian Optimization Algorithms (BOA) [4], clustering algorithms for the variables that appear to be correlated, extended Compact Genetic Algorithm (eCGA) [2] or model selection for Markov Random Field, as in DEUM [5]. Although very powerful, these techniques have their main drawback in the computational complexity of the model selection and sampling phases [1].

In this paper we propose a novel approach to the problem of model selection based on the idea of applying a transformation of variables and then employing fixed, low dimensional model in the new transformed space. This corresponds to implicitly identify a different statistical model in the original space which depends on the particular transformation applied. Obviously we moved much of the computational complexity from model selection to the choice of a good

transformation of variables; on the other side it becomes easier to select models able to capture higher order interactions among the variables. Instead of limiting the search up to a given order of interactions, due to the family of transformations we introduced we are able to identify non hierarchical models that can be efficiently employed in an EDAs.

This paper is organized as follows: we first introduce how transformation of variables can be employed in EDAs, then we present I-FCA, a novel algorithm which employs this technique. Finally we compare the performances with other popular EDAs.

2 Variable Transformations in EDAs: Function Composition Algorithms

In this section we apply the idea of choosing a transformation of variables and then considering low-dimensional statistical models in the transformed space to introduce a novel family of EDAs called Function Composition Algorithms (FCAs). In the following we address the maximization of $f(x) : \Omega^n \to \mathbb{R}$, $\Omega = \{\pm 1\}$.

We introduce a new vector of variables $y = (y_1, \ldots, y_n)$ in Ω and a one-to-one map $h : \Omega \to \Omega$ such that $y = h(x)$. We can thus express f as the composition of a function $g(y) : \Omega \to \mathbb{R}$ with h, i.e., $f = g \circ h$ and $g = f \circ h^{-1}$. Since h defines a permutation of the points in Ω, follows that $\max g = \max f$.

Recall the basic iteration of an EDA:

$$\mathcal{P}^t \xrightarrow{\text{selection}} \mathcal{P}_s^t \xrightarrow{\text{estimation}} p(x;\theta^t) \in \mathcal{M} \xrightarrow{\text{sampling}} \mathcal{P}^{t+1}$$

At each iteration, EDAs start with a population $\mathcal{P}^t$, chose a subset of individuals according to a selection policy and use this sample to estimate the parameters of a distribution $p(x;\theta)$ belonging to a model $\mathcal{M}$. For instance this can be done by means of statistical techniques such as max-likelihood estimation. A new population $\mathcal{P}^{t+1}$ is finally generated sampling individuals from $p(x;\theta)$. In the estimation phase, some algorithms, such as UMDA [3], employ a fixed model while more powerful EDAs, such as BOA [4], DEUM [5] perform a model selection step using machine learning techniques in order to chose a good model able to express the interactions among variables in the selected population $\mathcal{P}_s$.

We introduce the following variation of an EDA, where estimation and sampling are preceded and followed by two transformation steps: first a one-to-one map $y = h(x)$ is applied to each individual in the selected sample, obtaining $\tilde{\mathcal{P}}_s$, then the new sample $\tilde{\mathcal{P}}^{t+1}$ is mapped back in the original space with h^{-1}:

$$\mathcal{P}_s^t \xrightarrow{y=h(x)} \tilde{\mathcal{P}}_s \xrightarrow{\text{estimation}} q(y;\xi^t) \in \mathcal{N} \xrightarrow{\text{sampling}} \tilde{\mathcal{P}}^{t+1} \xrightarrow{x=h^{-1}(y)} \mathcal{P}^{t+1}$$

Here $\mathcal{N}$ identifies a model for the transformed variables y which corresponds to a model $\mathcal{M}$ for x which depends on the particular map h applied. Both models are characterized by the same dimension of the parameter space.

In the following we give the details of Independence-FCA (I-FCA), a novel EDA which fixes $\mathcal{N}$ to be the independence model for Y and performs an implicit

model selection step among a wide family of n-variate models by means of the choice of the one-to-one map h. At each iteration a map h is chosen among a subset of all the possible one-to-one maps by means of a greedy strategy which maximizes the likelihood of $q(y;\xi^t)$ with respect to $\tilde{\mathcal{P}}_s$. The resulting low-dimensional model $\mathcal{M}$ for X achieves a better approximation of the sample $\mathcal{P}_s$ with respect to the independence model for X.

The subset of the class of the one-to-one maps employed by I-FCA, indexed by j, $k \in \{1,\ldots,n\}$, with $j \neq k$, is defined such that

$$h_i^{(j,k)} : \begin{cases} y_i = x_i x_k & \text{if } i = j \\ y_i = x_i & \text{otherwise.} \end{cases}$$

Obviously we have $n(n-1)$ different $h^{(j,k)}$ transformations. It is easy to see that they are one-to-one and that $h^{-1} = h$, since $x_i^2 = 1$ and $\Omega = \{\pm 1\}$. Next we extend the class of transformations we consider by allowing elements h to be the composition of a finite number $\overline{m}$ of maps of the form $h^{(j,k)}$:

$$h = h^{(j_1,k_1)} \circ \ldots \circ h^{(j_m,k_m)} \circ \ldots \circ h^{(j_{\overline{m}},k_{\overline{m}})}.$$

Since the inverse of each transformation in the sequence of compositions is the element itself, it is easy to see that h^{-1} is the compositions of all the $h^{(j_m,k_m)}$ in the inverse order.

In I-FCA we propose a strategy for the choice of map h based on the maximization of the likelihood of the transformed selected sample $\tilde{\mathcal{P}}_s$ with respect to the estimated distribution $q(y,\hat{\xi}) \in \mathcal{N}$, where $\mathcal{N}$ is the independence model for Y. This is equivalent to minimize the Kullback-Leibler divergence between the empirical distribution representing the selected population and its projection on the independence model, which gives a measure of the loss of information which occurs when $\tilde{\mathcal{P}}_s$ is approximated with $q(y,\xi)$. In order to make the search for h feasible, we chose a greedy approach: we initialize h to be the identity map $y = x$, then we iteratively examine all the $n(n-1)$ maps $h^{(j,k)}$ and compose the h map obtained at the previous step with the map $h^{(j,k)}$ which better improves the likelihood of $(h \circ h^{(j,k)})(\mathcal{P}_s)$ with respect to the independence model. The iteration stops when no improvement in the likelihood is achievable composing further maps of the form $h^{(j,k)}$ or when the maximum number $\overline{m}$ of transformations in h has been reached. See Algorithm 1.

Since the chosen encoding for h is redundant, the procedure IsAllowed() is needed to avoid the evaluation of maps which lead to configurations already appeared in previous stages. The worst case time complexity of the search strategy for h is $\mathcal{O}(n^2\overline{m}N)$, where N is the population size, even though it is possible to take advantage of the likelihood decomposition to cut most of the complexity which comes from iterations over the selected population.

3 Experimental Results

In this section we present the results of a preliminary performance evaluation for the novel I-FCA algorithm on a set of well known benchmarks functions:

Algorithm 1 I-FCA - Choice of the map h

$m \leftarrow 0$;
$max\mathcal{L} \leftarrow \mathcal{L}_{ind}$; ▷ The likelihood of $\mathcal{P}_s$ w.r.t the independence model
repeat
  $h[m] \leftarrow$ NULL; ▷ The m-th element of the composition sequence
  for all $j, k \in \{1, \ldots, n\}, j \neq k$ **do**
    if IsAllowed($h^{(j,k)}$) **then**
      $\tilde{\mathcal{P}}_s \leftarrow h^{(j,k)}(\mathcal{P}_s)$;
      $\hat{\theta} \leftarrow$ MaxLikelihoodEstimation($\tilde{\mathcal{P}}_s$);
      $\mathcal{L} \leftarrow$ Likelihood($\tilde{\mathcal{P}}_s, q(y;\, \hat{\theta})$);
      if $\mathcal{L} > max\mathcal{L}$ **then**
        $h[m] \leftarrow h^{(j,k)}$;
      end if
    end if
  end for
  $\mathcal{P}_s \leftarrow h[m](\mathcal{P}_s)$;
  $m \leftarrow m + 1$;
until $m \geq \overline{m} \vee h[m-1] =$ NULL;
return h;

Alternated Bits, 2D Ising spin glass and Trap3. The first two functions are quadratic while Trap3 includes hierarchical interactions up to order three.

After a preliminary tuning the I-FCA parameters were chosen as follows. As the selection policy we perform truncation selection and keep the S highest fitness individuals, where S is a function of the problem size n but it is independent with respect to N. We found that $S = 5n$ is a good choice for all the benchmark functions considered. Moreover we set $\overline{m} = n$. This result is also supported by an analysis on the set of models $\mathcal{M}$ obtainable mapping the independence model for Y into a model for X by means of h. The success ratio as a function of the population size N is shown in Figure 1, for different problem sizes n. We next compare the performances of I-FCA with three well known EDAs: UMDA [3] , hBOA and DEUMce [5]. UMDA employs the independence model and it is identical to I-FCA once h is fixed to be the identity map $y = x$. hBOA make use of densities which factorize according to the structure of a Bayesian Network which is learned at every iteration from the selected sample. DEUMce employs a Cross Entropy criterion to learn the structure of a Markov random field where interactions up to order two are considered.

Straightforward implementations of these algorithms have been implemented in Evoptool [6] and all code is available at[1]. We run experiments with different parameter settings and population sizes and we computed the normalized value of f, the success ratio, the number of fitness evaluations, time and algorithm iterations when the best found individual appeared in the population, averaged over multiple runs. The results for the parameter settings which gave highest success ratio and least number of fitness function evaluations are presented in

[1] http://airlab.elet.polimi.it/index.php/Evoptool

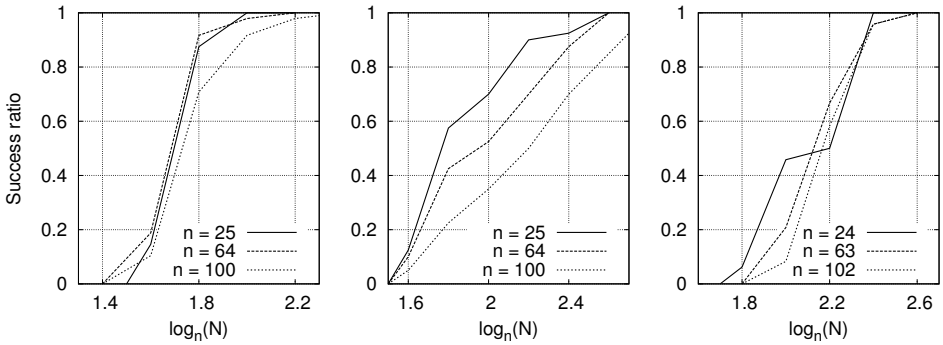

Fig. 1. Probability of success for three problems as a function of the population size for different problem sizes n. (a) Alternated Bits, $n \in \{25, 64, 100\}$, 50 runs (b) 2D Ising spin glass, $n \in \{25, 64, 100\}$, 5 instances, 16×5 runs (c) Trap3, $n \in \{24, 63, 102\}$, 50 runs

Table 1. It is possible to see that I-FCA outperforms UMDA: this proves the viability of the variables transformation approach. Moreover, the performances of I-FCA are comparable with hBOA, although the latter is able to achieve reliable convergence to the global optimum with smaller populations. Part of this comes from the more advanced selection scheme employed. DEUMce fails to find any good solution on Trap3, because of the third interaction present in the function, which instead are correctly handled by I-FCA and hBOA.

4 Conclusions

In this work we have introduced a novel EDA called I-FCA and we have tested out algorithm on three well known benchmark function. I-FCA has a low number of parameters for which we were able to give problem independent settings. Although a wider set of benchmark functions has to be analyzed, our preliminary experiments have shown that I-FCA can challange algorithms which learn expressive models, such as hBOA, employing only a low dimensional model, once a proper variable transformation has been learnt.

References

1. C. Echegoyen, Q. Zhang, A. Mendiburu, R. Santana, and J. Lozano. On the limits of effectiveness in estimation of distribution algorithms. In *Evolutionary Computation (CEC), 2011 IEEE Congress on*, pages 1573 –1580, june 2011.
2. G. Harik. Linkage learning via probabilistic modeling in the eCGA, 1999. Harik, G. R. (1999). Linkage Learning via Probabilistic Modeling in the ECGA (IlliGAL Report No. 99010). University of Illinois at Urbana-Champaign.

	I-FCA	UMDA	hBOA	DEUMce	I-FCA	UMDA	hBOA	DEUMce	I-FCA	UMDA	hBOA	DEUMce
	Alternated Bits											
n	25				64				100			
N	$n^{2.0}$	$n^{2.2}$	$n^{2.0}$	$n^{2.2}$	$n^{2.2}$	$n^{1.4}$	$n^{2.2}$	$n^{2.2}$	$n^{2.2}$	$n^{1.4}$	$n^{2.2}$	$n^{2.0}$
f_{best}[%]	100.0	94.4	100.0	97.7	100.0	88.8	100.0	100.0	100.0	85.4	100.0	100.0
success [%]	100.0	14.6	100.0	83.3	100.0	0.0	100.0	100.0	97.9	0.0	100.0	100.0
$f_{evals} \times 10^3$	2.43	13.14	3.82	1.12	51.08	7.42	119.0	9.42	179.3	15.69	444.9	10.01
iteration	3.60	10.62	4.71	–	5.19	21.56	11.25	–	6.98	24.50	16.35	–
t [s]	0.26	0.22	0.48	0.98	8.77	0.18	79.19	1.72	23.07	0.50	1649.7	8.04
	Ising spin glass 2D											
n	25				64				100			
N	$n^{2.4}$	$n^{2.4}$	$n^{2.2}$	$n^{2.4}$	$n^{2.4}$	$n^{1.8}$	$n^{2.2}$	$n^{2.4}$	$n^{2.4}$	$n^{1.6}$	$n^{2.2}$	$n^{2.4}$
f_{best}[%]	99.5	97.3	100.0	94.9	99.7	92.8	99.7	100.0	99.6	85.2	99.9	100.0
success [%]	92.5	55.0	100.0	50.0	87.5	2.5	87.5	100.0	70.0	0.0	95.0	100.0
$f_{evals} \times 10^3$	5.55	26.78	6.89	2.03	130.4	43.16	155.5	21.62	488.0	39.27	498.4	63.10
iteration	2.23	11.40	4.42	–	5.80	23.82	15.15	–	7.53	24.32	18.48	–
t [s]	0.17	0.24	0.73	2.88	10.67	0.91	95.26	2.54	76.83	1.18	1707.6	37.53
	Trap3											
n	24				63				102			
N	$n^{2.4}$	$n^{2.6}$	$n^{2.4}$	$n^{1.8}$	$n^{2.6}$	$n^{1.8}$	$n^{2.2}$	$n^{1.8}$	$n^{2.4}$	$n^{1.8}$	$n^{2.0}$	$n^{1.8}$
f_{best}[%]	100.0	95.5	100.0	90.9	100.0	90.3	100.0	82.3	100.0	90.2	100.0	77.9
success [%]	100.0	4.2	100.0	0.0	100.0	0.0	100.0	0.0	95.8	0.0	100.0	0.0
$f_{evals} \times 10^3$	4.81	6.78	13.68	0.32	159.8	23.65	150.8	0.91	348.3	82.15	287.5	2.46
iteration	2.12	1.29	5.35	–	3.15	13.21	15.29	–	5.19	19.48	26.29	–
t [s]	0.19	0.10	1.18	4.26	8.69	0.62	97.63	2.05	20.43	2.31	992.7	15.74

Table 1. Statistics of the best solutions averaged over 48 runs for Alternated Bits and Trap3 and 8 runs $\times$ 5 instances for Sping glass. UMDA: truncation selection 50%, DEUMce: truncation selection 30%, Cross Entropy min significance 2.0. CPU: AMD Opteron$^{\text{TM}}$ 6176, 2.3 GHz

3. H. Mühlenbein and T. Mahnig. Mathematical analysis of evolutionary algorithms. In *Essays and Surveys in Metaheuristics, Operations Research/Computer Science Interface Series*, pages 525–556. Kluwer Academic Publisher, 2002.
4. M. Pelikan and D. Goldberg. Hierarchical bayesian optimization algorithm. In M. Pelikan, K. Sastry, and E. CantPaz, editors, *Scalable Optimization via Probabilistic Modeling*, volume 33 of *Studies in Computational Intelligence*, pages 63–90. Springer Berlin / Heidelberg, 2006.
5. S. Shakya, A. Brownlee, J. McCall, F. Fournier, and G. Owusu. A fully multivariate DEUM algorithm. In *IEEE Congress on Evolutionary Computation,*, 2009.
6. G. Valentini, L. Malagò, and M. Matteucci. Evoptool: an extensible toolkit for evolutionary optimization algorithms comparison. In *Proceedings of IEEE World Congress on Computational Intelligence*, pages 2475–2482, July 2010.