

COLO notes

Marc Toussaint

April 13, 2005

1 05-04-13

Benchmarking

GAs or EDAs are often compared against repeated hill climbing — not random search. The Algorithm is:

Algorithm 1.1 (Repeated Hill Climbing).

1. Pick a random sequence
2. For every symbol, check if a flip leads to an improvement. This is done in random order of the symbols.
3. As soon as an improvement is found, keep it and repeat with step 2.
4. If no improvement is found, memorize the sequence and repeat with step 1.

Models of the empirical distribution

The general kind of algorithm we were talking about today:

- keep the set of all so far evaluated sequences in memory
- Assign probabilities (like selection probabilities) to each of these sequences, depending on their evaluation
- This gives you an empirical distribution
- Learn a model of the empirical distribution
- sample from that model a new sequence

Many kinds of models can be fitted to the data. What Börn (on the second level) did roughly corresponds to what I'd call “identically factorized”:

$$P(x) = \prod_i p(x_i), \quad (1)$$

$x \in \Sigma^*$ is a sequence from the alphabet Σ , x_i is the i th symbol, where $p(x_i)$ is a distribution over Σ independent of i .

(Actually, a full model of the empirical distribution would also have to include a model of the length distribution $P(n)$. But let's neglect that.)

On his third level, he distinguished different $p(x_i)$ for different parts of the sequence. Distinguishing different $p(x_i)$ for every i leads to PBIL:

Other models used in EDAs are (cut-and-paste from a paper of mine...)

Factorial Estimates the marginals $P(x_i)$ for each variable separately (cf. PBIL [1]); the distribution model is

$$P(x) = \prod_{i=1}^n P(x_i) . \quad (2)$$

Tree First calculates the mutual information between each pair of variables; then generates a maximum spanning tree with random root, maximizing the total mutual information associated to the edges (cf. COMIT [2]). The distribution model is

$$P(x) = P(x_0) \prod_{i=2}^n P(x_i | x_{\pi(i)}) , \quad (3)$$

where $\pi(i) \in \{1, \dots, i-1\}$ is the parent of i and the indexing of variables is topologically sorted w.r.t. the tree.

3rd order graphical model First calculates the mutual information between each *triplet* of variables ($I(i, j, k) = H(i) + H(j) + H(k) - H(i, j, k)$). Then generates a graph, where each node (except the first two) has exactly two parents. This graph is generated following the same scheme as for the maximum spanning tree: At every step, find a variable x_k that has not yet been added to the graph and *two* nodes x_i and x_j that have already been added, which maximize $I(i, j, k) - I(i, j)$; then add x_k to the graph with edges from x_i and x_j . The distribution model is

$$P(x) = P(x_0) P(x_1 | x_0) \prod_{i=3}^n P(x_i | x_{\pi_1(i)}, x_{\pi_2(i)}) , \quad (4)$$

where $\pi_1(i) \neq \pi_2(i) \in \{1, \dots, i-1\}$ are the two parents of i (node indexing is topologically sorted w.r.t. the graph). The root x_0 is chosen randomly and the second node x_1 as for the tree building.

2 05-02-09: EDAs, ANOVA

2.1 Notation

X	search space
$p \in \Lambda^X$	a probability distribution or <i>population</i> over the search space: it may also represent a finite (multi)set (=population) over the search space when it is composed as a normalized finite sum of δ 's; or also weighted population ...
$\mathcal{S}^\lambda : \Lambda^X \rightarrow \Lambda^X$	sampling operator: maps any distribution to the a population of λ search points (think of stochastic universal sampling)

2.2 EDAs

General notation: q is usually a *search distribution* (or a population search points), whereas p is the *selection distribution* (or a population of selected)

An EDA simply works as follows:

Algorithm 2.1.

1. Initialize the search distribution $q^{(0)}$ over X , e.g., as uniform

2. Sample λ points from $q^{(t)}$:

$$\tilde{q}^{(t)} \sim \mathcal{S}^\lambda q^{(t)} .$$

[Now, $\tilde{q}^{(t)}$ is a population $\tilde{q}^{(t)}(x) = \frac{1}{\lambda} \sum_{i=1}^{\lambda} \delta(x_i, x)$.]

3. Evaluate all points: this means to change the weighting of the search points like

$$p^{(t)}(x) = \frac{1}{Z} \exp[-f(x)] \tilde{q}^{(t)}(x)$$

(Note the Bayesian perspective here!)

4. Construct a new search distribution: Let Q be the space of feasible search distributions (determined by the probabilistic model you choose), then choose the new search distribution as

$$q^{(t+1)} = \operatorname{argmin}_{q \in Q} D(p^{(t)}, q) .$$

Here, D is some estimation objective (or “metric”) measuring how well q models $p^{(t)}$

5. Repeat from 2.

Please note that, according to this definition, an EDA is in principle determined by the choice of Q (i.e., the probabilistic model used for estimation) and the choice of the estimation objective D . Practically, an EDA is also characterized by the method by which it tries to minimize $D(p^{(t)}, q)$ since this is non-trivial for complex Q . Table 1 displays some EDAs I know of, characterized by the choice of Q , D , and method for minimization.

2.3 Anova \sim PBIL

Let $f : X \rightarrow \mathbb{R}$ be the objective function to be minimized. Let’s call it energy. It’s Boltzmann distribution then is

$$F \propto \exp(-f) .$$

Let $X = X_1 \times \dots \times X_n$ and each $X_i = \{A, B, \dots\} = \mathcal{A}$. The Anova model then reads

$$\begin{aligned} f(x) &= \beta_0 + \sum_{i;a \in \mathcal{A}} \beta_{ia} \delta(x_i, a) + \sum_{i,j;a,b \in \mathcal{A}} \beta_{ijab} \delta(x_i, a) \delta(x_j, b) + \dots \\ &= \beta_0 + \sum_i \beta_{ix_i} + \sum_{ij} \beta_{ijx_i x_j} + \dots \end{aligned}$$

Here $\delta(x_i, a) \in \{0, 1\}$ is one iff the statement $x_i = a$ is true.

Consider the Boltzmann distribution:

$$\begin{aligned} F(x) &\propto \exp\left\{-\sum_i \beta_{ix_i} - \sum_{ij} \beta_{ijx_i x_j}\right\} \\ &= \prod_i \exp\{-\beta_{ix_i}\} \cdot \prod_{ij} \exp\{-\beta_{ijx_i x_j}\} \\ &= \prod_i p(X_i = x_i) \cdot \prod_{ij} p(X_i = x_i, X_j = x_j) , \end{aligned}$$

where we identified

$$\begin{aligned} p(X_i = x_i) &= \exp\{-\beta_{ix_i}\} \\ p(X_i = x_i, X_j = x_j) &= \exp\{-\beta_{ijx_i x_j}\} . \end{aligned}$$

Neglecting the second order terms, F factorizes.

	class Q of probabilistic models	estimation objective D for minimization	method of minimization
PBIL [1] Population-based Incremental Learning	factored $\prod_i P(x_i)$	trivial (sets equal)	counting
UMDA [5] Univariate Marginal Distribution Algorithm	factored $\prod_i P(x_i)$	trivial	counting
MIMIC [3] Mutual Information Maximization for Input Clustering	Markov $\prod_i P(x_i x_{i-1})$	trivial	counting
COMIT [2] Combining Optimizer with Mutual Information Trees	trees		
BOA [8] Bayesian Optimization Algorithm	Graphical model	Bayesian Metric	hill-climbing on graph structure, counting for local CPTs
hBOA [7] hierarchical Bayesian Optimization Algorithm	..with decision trees at each node		
FDA [6] Factorized Distribution Algorithm	weird		
Compression EDAs [9]	factored on compressed rep.	Kullback-Leibler divergence	grammar-kind heuristic compression

Table 1: EDAs: All of these are for discrete search spaces which are decomposed as $X = X_1 \times \dots \times X_n$. See [4] for EDAs in continuous domains. Blank means I was too lazy to look it up...

2.4 General remark: Modeling the objective function \sim EDA

Generally, modeling the energy f is equivalent to modeling the distribution F . To goal of such modeling is, of course, to sample the good solutions from this model. In the case of EDAs, the models used typically make it easy to sample good solutions from them (DAGs). This is not so easy for the second order Anova model: finding minima of this function is itself an optimization problem. I think, when dropping some terms in the second order Anova model, it corresponds to a feasible graphical model, like a tree, from which one can sample easily.

In my view, the only principle difference between an EDA and the traditional Anova approach is that an EDA iterates the procedure of “modeling–sampling–modeling–sampling–...” several times, relying on some heuristic that the models become better in each iteration (also because they may specialize locally on regions of good solutions), whereas the Anova model does this only once: “modeling–sampling–basta” (sampling here actually means optimizing w.r.t. Anova model).

References

- [1] S. Baluja. Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning. Technical Report CMU-CS-94-163, Comp. Sci. Dep., Carnegie Mellon U., 1994.
- [2] S. Baluja and S. Davies. Using optimal dependency-trees for combinatorial optimization: Learning the structure of the search space. In *Proc. of Fourteenth Int. Conf. on Machine Learning (ICML 1997)*, pages 30–38, 1997.
- [3] J. S. de Bonet, C. L. Isbell, Jr., and P. Viola. MIMIC: Finding optima by estimating probability densities. In M. C. Mozer, M. I. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems*, volume 9, page 424. The MIT Press, 1997.
- [4] S. Kern, S. D. Müller, N. Nansen, D. Büche, J. Ocenasek, and P. Koumoutsakos. Learning probability distributions in continuous Evolutionary Algorithms – A comparative approach. *Natural Computing*, 3:77–112, 2003.
- [5] H. Mühlenbein. Equation for response to selection and its use for prediction. *Evolutionary Computation*, 5:303–346, 1998.
- [6] H. Mühlenbein, T. Mahnig, and A. O. Rodriguez. Schemata, distributions and graphical models in evolutionary optimization. *J. of Heuristics*, 5:215–247, 1999.
- [7] M. Pelikan and D. E. Goldberg. Hierarchical BOA solves Ising spin glasses and MAXSAT. In *Genetic and Evolutionary Computation Conference 2003 (GECCO-2003)*, pages pp. 1271–1282. Springer, 2003.
- [8] M. Pelikan, D. E. Goldberg, and E. Cantú-Paz. Linkage problem, distribution estimation, and Bayesian networks. *Evolutionary Computation*, 9:311–340, 2000.
- [9] M. Toussaint. Compact genetic codes as a search strategy of evolutionary processes. In *Foundations of Genetic Algorithms 8 (FOGA VIII)*, LNCS. Springer, 2005.