Bayesian Evolutionary Algorithms for Learning and Optimization

Byoung-Tak Zhang

Artificial Intelligence Lab (SCAI)
School of Computer Science and Engineering
Seoul National University
Seoul 151-742, Korea
E-mail: btzhang@scai.snu.ac.kr

Abstract

Bayesian evolutionary algorithms (BEAs) are a probabilistic model of evolutionary computation that is based on Bayesian inference. Though originally developed as a method for evolutionary *learning*, BEAs can be shown to be applicable to evolutionary *optimization* as well. We investigate the relationship of BEAs to other distribution estimation algorithms (i.e. EDAs) to develop a unified probabilistic framework of evolutionary computation for solving both optimization and learning problems. The generality and distinguishing features of the Bayesian framework are discussed and compared to existing probabilistic evolutionary models.

1 INTRODUCTION

Recently, a number of evolutionary algorithms have been proposed that explicitly model the population of good solutions and use the constructed model to guide further search. These include, among others, PBIL, UMDA, MIMIC, BMDA, FDA, and BOA (see [Zhang and Shin, 2000] and references therein) and are generally known as the estimation of distribution algorithms or EDAs [Mühlenbein et al., 1999]. They use global information contained in the population, instead of using local information through crossover or mutation of individuals. From the population, statistics of the hidden structure are derived and used when generating new individuals. All these methods are designed for optimization.

In contrast, Bayesian evolutionary algorithms (BEAs) are a probabilistic model of evolutionary computation [Zhang, 1999] that were originally presented in the context of evolutionary learning of models from given data, i.e. function approximation. In BEAs, evolutionary computation is formulated as a probabilistic process of finding an individual with the maximum a posteriori probability (MAP). Explicit modeling of fitness distributions in terms of probabilities and the generational transition by means of Bayes formula are two distinguishing features of BEAs from other evolutionary algorithms.

In this paper, we extend the Bayesian evolutionary framework presented in [Zhang, 1999] to encompass probabilistic evolutionary algorithms for function optimization as well as those for function approximation. In Section 2, we formally define the problems of function optimization and function approximation (learning) and examine their relationship. Section 3 presents the unified Bayesian framework of evolutionary computation for learning and optimization. Section 4 briefly discusses the experimental results and concludes with some remarks on the implications of the Bayesian evolutionary approach to learning and optimization.

2 FUNCTION OPTIMIZATION AND APPROXIMATION

2.1 FUNCTION OPTIMIZATION

The goal of function optimization is to find $\mathbf{x}^*$ that minimizes an objective function f:

$$\mathbf{x}^* = \operatorname*{argmin}_{\mathbf{x} \in X} f(\mathbf{x}), \tag{1}$$

where X is the input space of f. Here, the objective function f is given as an equation (or the equivalent) and the fitness of any search points $\mathbf{x}$ can be evaluated directly from this function. Evolutionary algorithms have been extensively used for function optimization. Starting from the initial population of search points, an evolutionary algorithm applies mutation and recombination operators to generate a new population.

This process is repeated until the optimum point $\mathbf{x}^*$ is found or the maximum allowed generation is reached.

2.2 FUNCTION APPROXIMATION

Learning or function approximation involves constructing models f_θ of a target function f given a set D of input-output pairs $(\mathbf{x}_c, f(\mathbf{x}_c))$, i.e., $D = \{(\mathbf{x}_c, f(\mathbf{x}_c)) \mid c = 1, ..., N\}$. Here $f(\mathbf{x}_c)$ is the observed output of the target function given the input $\mathbf{x}_c$. The objective is to find the model θ^*, i.e., the functional structure and associated parameters, whose output $f_{\theta^*}(\mathbf{x})$ best predicts the output $f(\mathbf{x})$ of the target function given an arbitrary input $\mathbf{x}$. Using the squared error criterion as the objective function, this can be formulated as a minimization problem:

$$\theta^* = \underset{\theta \in \Theta}{\operatorname{argmin}} \left\{ \sum_{\mathbf{x}_c \in D} ||f_\theta(\mathbf{x}_c) - f(\mathbf{x}_c)||^2 \right\}, \qquad (2)$$

where Θ is the space of all possible models in consideration and D is the data set available. Evolutionary computation has been used from its inception for automatic induction of models for a given system or process. For example, L. Fogel used simulated evolution to induce finite state machines that predict a sequence of symbols from an environment. Recent development includes genetic programming where Lisp-like symbolic programs are evolved from training data.

2.3 OPTIMIZATION VS. LEARNING

There are several differences between function approximation (learning) and function optimization. First, the objective function f in optimization is given as an "explicit" functional form and the fitness of any search points $\mathbf{x}$ can be evaluated directly from this function (there are some exceptions, but most of evolutionary optimization is formulated in this way). In contrast, the objective function in learning is given "implicitly" as a finite set D of function values, and thus the fitness of search points (in this case, models θ) is evaluated on this finite set of fitness cases. Learning also differs from optimization in the way how the optimization result is used. The result θ^* of learning in (2) is usually used later to solve multiple problem-instances (e.g., to predict outputs $f_{\theta^*}(\mathbf{x})$ given different inputs $\mathbf{x}$) while the solution $\mathbf{x}^*$ of an optimization problem (1) is itself the final solution for the problem instance at hand.

Despite the differences in their goals and assumptions, function approximation is formally in close connection with 'regular' function optimization. To see the relationship, we define the objective function F of function approximation (2) in terms of D as

$$F(\theta) = \sum_{\mathbf{x}_c \in D} ||f_\theta(\mathbf{x}_c) - f(\mathbf{x}_c)||^2. \qquad (3)$$

Then, the function approximation problem can be viewed as a function optimization problem:

$$\theta^* = \underset{\theta \in \Theta}{\operatorname{argmin}}\ F(\theta), \qquad (4)$$

where Θ is the input space of the function F (see Equation (1) for comparison).

On the other hand, function optimization can be facilitated by using function approximation as a subroutine. That is, instead of attempting to directly optimize the objective function $f(\mathbf{x})$, we can build a model $f_\theta(\mathbf{x})$ of the objective function using the search points observed so far or their subset $X^t = \{\mathbf{x}_c \mid c = 1, ..., N\}$. The new problem is then formulated as:

$$\mathbf{x}^* = \underset{\mathbf{x} \in X}{\operatorname{argmin}}\ f_\theta(\mathbf{x}), \qquad (5)$$

where f_θ is the approximated function of the original objective function f. This is reasonable since many optimization problems have underlying structure in their search space. Using this structure can help the search for the optimal solution. The distribution estimation algorithms (EDAs) described in Section 1 are examples of this approach.

3 A BAYESIAN EVOLUTIONARY FRAMEWORK FOR LEARNING AND OPTIMIZATION

In this section, we present a general probabilistic framework for evolutionary computation that handles function optimization, function approximation, and their combinations in a uniform way.

In the Bayesian approach to evolutionary computation [Zhang, 1999] the fitness of the individuals is formulated as a posterior probability. More formally, let θ denote the parameter vector for the model, let $\pi(\theta)$ be the prior probability distribution for the models and $p(D|\theta)$ the likelihood of the model for the data $D = \{(\mathbf{x}_c, y_c) \mid c = 1, ..., N\}$. Then, using Bayes formula the posterior probability $\pi(\theta|D)$ of model θ is given as

$$\pi(\theta|D) \;=\; \frac{p(D|\theta)\pi(\theta)}{p(D)}, \qquad (6)$$

where $p(D)$ is a normalizing constant.

Starting from a prior probability distribution $\pi_0(\theta)$, evolution is considered as an iterative process of revising the posterior distribution of individuals $\pi_t(\theta|D)$ by combining the prior $\pi_t(\theta)$ with the likelihood $p(D|\theta)$. In each generation, Bayes theorem (6) is used to estimate the posterior fitness of individuals from their prior fitness values. The posterior distribution $\pi_t(\theta|D)$ is then used to generate its offspring.

The aim of Bayesian evolutionary computation is twofold, depending on the problem to address. One is to choose a model θ_{MAP} that maximizes the posterior probability (MAP):

$$\theta_{MAP} = \underset{\theta \in \Theta}{\operatorname{argmax}}\ \pi(\theta|D). \tag{7}$$

The MAP model is then used to predict the output values y for given input values $\mathbf{x}$:

$$y = f(\mathbf{x}; \theta_{MAP}). \tag{8}$$

This is the approach we take for function approximation or learning. An example of this approach is given in [Zhang and Cho, 2000]. Alternatively, the samples from the posterior distribution $\pi(\theta|D)$ can be used to compute the posterior *predictive* distribution $p(\mathbf{x}|D)$ of inputs $\mathbf{x}$ as follows:

$$p(\mathbf{x}|D) \quad = \quad \int p(\mathbf{x}|\theta)\pi(\theta|D)d\theta. \tag{9}$$

This is the approach we take to generate promising new points for function optimization [Zhang and Shin, 2000].

The distribution estimation algorithms for optimization mentioned in the foregoing sections can be regarded as special cases of the Bayesian evolutionary approach. That is, a single function $p(\mathbf{x}|\hat{\theta})$ of maximum likelihood estimate $\hat{\theta}$ is used to approximate the objective function $f(\mathbf{x})$ rather than using $p(\mathbf{x}|\theta)$'s for all possible θ's to take into account their distribution $\pi(\theta|D)$. Note that this is a reasonable approximation since if the prior is relatively flat and the peak of the likelihood function is relatively sharp, then the integral in (9) will be dominated by the region around the maximum likelihood estimate $\hat{\theta}$, and the posterior predictive distribution can be given approximately by

$$p(\mathbf{x}|D) \approx p(\mathbf{x}|\hat{\theta}) \int \pi(\theta|D)d\theta = p(\mathbf{x}|\hat{\theta}), \tag{10}$$

where we used $\int \pi(\theta|D)d\theta = 1$, and $\hat{\theta}$ is the parameter vector that maximizes $p(\mathbf{x}|\theta)$. In [Zhang and Shin, 2000], we describe a method that uses a probabilistic graphical model known as Helmholtz machines to approximate the posterior predictive distribution function $p(\mathbf{x}|D)$ by the maximum likelihood $p(\mathbf{x}|\hat{\theta})$.

4 CONCLUDING REMARKS

We presented a unified probabilistic framework for solving optimization and learning problems using evolutionary computation. In this framework, evolutionary computation is formulated as a probabilistic process of Bayesian inference, which leads to a class of Bayesian evolutionary algorithms or BEAs. The method of Bayesian evolutionary optimization is very closely related to the distribution estimation algorithms. Similar to the usual EDAs, a BEA builds a probabilistic model of discrete samples. However, BEAs are more general in the sense that the distribution estimated is the posterior distribution which encompasses likelihood functions (as in most EDAs) as a special case. It should also be noted that BEAs are different from BOA [Pelikan et al., 1999] though both use the term 'Bayesian'; BOA uses Bayesian networks as a method for estimating population distributions in a regular evolutionary process, while in BEA the evolutionary process itself is Bayesian, regardless of using Bayesian networks or other methods for distribution estimation.

Acknowledgments

This research was supported by KOSEF Grant 981-0920-350-2, KISTEP Grant BR-2-1-G-06, and the BK21-IT Program.

References

Mühlenbein, H., Mahnig, T. and Ochoa, A. (1999). Schemata, distributions and graphical models in evolutionary optimization, *J. Heuristics*, **5**:215-247.

Pelikan, M., Goldberg, D.E., and Cantú-Paz, E. (1999). BOA: The Bayesian optimization algorithm, In *Proc. 1999 Genetic and Evolutionary Computation Conference* (GECCO-99), Morgan Kaufmann, pp. 525-532.

Zhang, B.-T. (1999). A Bayesian framework for evolutionary computation. In *Proc. 1999 Congress on Evolutionary Computation* (CEC99), IEEE Press, pp. 722-727.

Zhang, B.-T. and Cho, D.-Y. (2000). Evolving neural trees for time series prediction using Bayesian evolutionary algorithms, In *Proc. First IEEE Workshop on Combinations of Evolutionary Computation and Neural Networks* (ECNN-2000), San Antonio, TX, 2000 (to appear).

Zhang, B.-T. and Shin, S.-Y. (2000). Bayesian evolutionary optimization using Helmholtz machines, submitted to PPSN-2000, Paris, France.