Convergence Properties of Bayesian Evolutionary Algorithms with Population Size Greater Than 1

Si-Eun Lee
Artificial Intelligence Lab (SCAI)
School of Computer Sci. and Eng.
Seoul National University
Seoul 151-742, Korea
selee@scai.snu.ac.kr

Byoung-Tak Zhang
Artificial Intelligence Lab (SCAI)
School of Computer Sci. and Eng.
Seoul National University
Seoul 151-742, Korea
btzhang@cse.snu.ac.kr

Arnaud Doucet
Signal Processing Group
Engineering Department
University of Cambridge
Trumpington Str.Cambridge CB2 1PZ, UK
ad2@eng.cam.ac.uk

Abstract- A Bayesian evolutionary algorithm is a probabilistic model of evolutionary computation for learning and optimization. It explicitly estimates the posterior distribution of the individuals and then samples offspring from the distribution. In the previous paper, using the asymptotic results from Markov chain Monte Carlo and annealing techniques, the asymptotic convergence of Bayesian evolutionary algorithms was shown for the case of population size 1. This paper presents convergence properties of Bayesian evolutionary algorithms with population size greater than 1. The basic idea is that BEAs can be reduced to Bayesian particle filters. The Bayesian particle filter approximates the posterior distribution of individuals at each generation. As the individuals evolve, the approximated posterior distribution also evolves. Then using the convergence properties of particle filters under some mild conditions, it is shown that as the number of individuals increases, a BEA converges to the posterior distribution.

1. Introduction

Bayesian evolutionary algorithms (BEAs) are probabilistic models of evolutionary computation and are based on the Bayesian inductive principle. Starting from a population of individuals drawn from the prior distribution, a Bayesian evolutionary algorithm iteratively generates a new population by estimating the posterior fitness distribution of parent individuals and then sampling offspring individuals from the distribution via variation and selection operators. BEAs are distinguished from conventional evolutionary algorithms [3] in that the individuals are generated from a probability distribution as in Markov chain Monte Carlo (MCMC) methods [7, 11]. BEAs have more exploratory nature than conventional evolutionary algorithms because they generate offspring by sampling from a probability distribution. BEAs have the advantage of efficiency in that search is population-based while maintaining the theoretical validity of MCMC methods.

In the previous work [13], we present convergence properties of Bayesian evolutionary algorithms with population size 1, using the asymptotic results from Markov chain Monte Carlo. Annealing techniques can be used to show asymptotic convergence of non-homogeneous Markov chain generated by BEAs.

In this paper, we study the convergence properties of Bayesian evolutionary algorithms. We are especially interested in the convergence properties of BEAs with population size greater than 1.

We convert BEAs to Bayesian particle filters that approximate posterior distribution of individuals. BEAs estimate posterior distribution by combining prior and likelihood. Then we sample individuals from the posterior distribution and modify them to generate offspring for next generation. Bayesian particle filters generate offspring by multiplyng or discarding individual with respect to fitness. The posterior distribution evolves over time in both BEAs and Bayesian particle filters. Using the convergence properties of particle filters, we show that the Bayesian particle filter converges to the posterior distribution.

The paper is organized as follows. In Section 2, we review canonical Bayesian evolutionary algorithm and convergence properties of that with population size 1. Section 3 presents generic particle filters and convergence properties of them. Section 4 presents the description of the Bayesian particle filter. Section 5 gives convergence properties of the Bayesian particle filter. Section 6 gives conclusion remarks.

2. Bayesian Evolutionary Computation

In the Bayesian probabilistic model of evolutionary computation [12, 14], the fitness of the individuals is represented as a posterior probability. Let $\pi(\theta)$ be the prior distribution for individual $\theta \in \Theta$ where Θ is the

0-7803-6657-3/01/$10.00 ©2001 IEEE

search space. Let $f(D \mid \theta)$ be the likelihood of θ for data D. Bayesian theorem is used to estimate posterior fitness $\pi(\theta \mid D)$ from a population of individuals at each generation such that

$$\pi(\theta \mid D) = \frac{f(D \mid \theta)\pi(\theta)}{\int f(D \mid \theta)\pi(\theta)d\theta} \approx \frac{f(D \mid \theta)\pi(\theta)}{\sum_{\theta' \in \Theta} f(D \mid \theta')\pi(\theta')}. \quad (1)$$

Evolution is considered as an iterative process of revising the posterior distribution of individuals $\pi(\theta \mid D)$ by combining the prior $\pi(\theta)$ with the likelihood $f(D \mid \theta)$. The posterior distribution is then used to generate its offspring.

The canonical Bayesian evolutionary algorithm can be summarized as follows.

Algorithm 2.1 (Canonical BEA)

1. **(Initialize)** Generate $\Theta^0 = \{\theta_1^0, \ldots, \theta_M^0\}$ from $\pi_0(\theta)$. Initialize temperature T_0. Set generation count $t \leftarrow 0$.
2. **(D-step)** Generate (observe) D. Compute likelihoods $f(D \mid \theta_i')$.
3. **(P-step)** Estimate posterior distribution $\pi_t(\theta_i' \mid D)$.
4. **(V-step)** Generate L variations $\Theta' = \{\theta_1', \ldots, \theta_L'\}$ by sampling from $\pi_t(\theta)$.
5. **(S-step)** Select M individuals from Θ' into $\Theta' = \{\theta_1', \ldots, \theta_M'\}$ by $f(D \mid \theta_i')$.
6. **(R-step)** Revise prior distribution $\pi_t(\theta)$. Update temperature T_t.
7. **(loop)** Set $t \leftarrow t + 1$ and go to Step 2.

The algorithm consists of five steps: D (data), P (posterior), V (variation), S (selection), and R (revision). The steps of R, D and P involve computation of prior, likelihood and posterior probabilities, respectively. The V and S steps realize the sampling from the posterior distribution at time t.

More specifically, we assume the exponential family for the likelihood function and prior distribution (e.g., Gaussian distributions),

$$f(D \mid \theta_i') = \frac{\exp\{-E(D \mid \theta_i')/T_t\}}{\sum_{\theta_j' \in \Theta'} \exp\{-E(D \mid \theta_j')/T_t\}}, \quad (2)$$

$$\pi_t(\theta_i') = \frac{\exp\{-C(\theta_i')/T_t\}}{\sum_{\theta_j' \in \Theta'} \exp\{-C(\theta_j')/T_t\}}, \quad (3)$$

where $E(D \mid \theta_j')$ and $C(\theta_j')$ are arbitrary component measures for evaluating raw fitness of individuals and T_t is the temperature parameter for controlling the randomness of the stochastic process. The fitness of individuals is written as

$$\pi_t(\theta_i' \mid D) = \frac{\exp\{-F(\theta_i' \mid D)/T_t\}}{\sum_{\theta_j' \in \Theta'} \exp\{-F(\theta_j' \mid D)/T_t\}}, \quad (4)$$

where $F(\theta_j' \mid D) = E(D \mid \theta_j') + C(\theta_j')$.

A theoretical analysis of full-fledged BEAs seems practically impossible since Markov chains generated by them are non-homogeneous. Annealing techniques can be used to show asymptotic convergence of non-homogeneous Markov chains. If we make simplifying assumptions such as restricting population size to one and considering state spaces of fixed dimensions, BEA can be reduced to a Markov chain Monte Carlo method for which geometric convergence results are well-known. The convergence results for Metropolis-Hastings algorithm apply in Metropolis-Hastings version of the BEA that has a single chain where $T_t = T$ and $M = 1$. In case of fixed data set D with temperature scheduling, annealed version of the BEA has a convergence property equivalent to that of simulated annealing. Annealed BEA is in fact a Metropolis algorithm, which simulates the density proprotional to $\exp\{-F(\theta_i' \mid D)/T_t\}$. Incremental BEA has D-step which cares for incremental growth of data sets $D = D'$. In incremental BEA, no temperature scheduling is applied and growth of data size only relates revision step. Incremental data growth plays the role of a cooling schedule in simulated annealing, so we arrive at convergence results of incremental BEA.

3. Particle Filters

3.1 Generic Particle Filters

Particle filters are powerful sampling-based inference and learning algorithms. They have been introduced to handle the state estimation problem [1] and appeared in many fields under such names as "sequential Monte Carlo", "condensation" and "survival of the fittest" [8, 9, 10].

In sampling-based methods one represents the probability distribution $\pi(\theta \mid D)$ by a set of N random samples $\Theta = (\theta_1, \cdots, \theta_N)$ drawn from it. We can do this because of the essential duality between the samples and the probability distribution from which they are generated. From the samples we can always approximately reconstruct the probability distribution.

Particle filters give a complete representation of the posterior distribution of the states, so any statistical estimates can be easily computed. They can treat any type of probability distribution, nonlinear and non-stationary, and are not restricted to Gaussian distribution. We can approximate the posterior distribution using particle filters.

Particle filters are generally two-step Markov processes [5]. Standard assumption that the process is Markov can be removed [4]. In the first step at time t, a particle filter $\Theta^t = (\theta_1^t, \cdots, \theta_N^t)$ is composed of N particles and each particle θ_i^t is assigned to the weight w_i^t. Proportional to the assigned weight, each particle generates offspring by copying itself and the particles with small weights are deleted. Then we obtain intermediate particle filter $\overline{\Theta}^t$. It is interesting to see analogy to the steps in evolutionary algorithms. It is possible for many particles to generate no offspring, whereas others generate a large number of offspring. In this case, there is a severe reduction in the diversity of samples.

Because of degenerating of the variety of samples, at the second step each particle evolves independently according to system dynamics whose proposal distribution is the posterior distribution. We can get new particle filter Θ^{t+1} and rearrange the weights of particles to make random samples.

The particles that approximate the posterior distribution evolve over time by adapting to take account of information contained in D. The particles evolve over time, so the posterior distribution approximated by the particles evolves. Let δ_a be Dirac delta function at $a \in \Theta$. Posterior distribution $\pi(\theta \mid D)$ can be approximated by the following empirical estimate,

$$\hat{\pi}(\theta \mid D) = \frac{1}{N}\sum_{i=1}^{N} \delta_{\theta_i}. \qquad (5)$$

Expectation for arbitrary function g

$$E(g(\theta)) = \int g(\theta)\pi(\theta \mid D)d\theta \qquad (6)$$

is approximated by

$$\overline{E(g(\theta))} = \frac{1}{N}\sum_{i=1}^{N} g(\theta_i). \qquad (7)$$

But it is often impossible to sample directly from the posterior distribution. We can circumvent this difficulty by sampling from a known, easy-to-sample, importance distribution $q(\theta)$. Thus particle filters rely on importance sampling and require design of proposal distribution.

The idea of importance sampling is to draw independent samples from a simpler proposal distribution q, then apply importance sampling corrections i.e., weight these samples by importance weight $w(\theta)=\pi(\theta)/q(\theta)$ to obtain a fair representation of π. Importance sampling is an effective estimation technique when q approximates π over most of the domain. Except in a few special cases, the posterior distribution cannot be obtained analytically. Particle filters utilize random particles (samples) based representation of the posterior distribution.

A generic particle filter algorithm is summarized as Importance Sampling-step, Selection-step and MCMC-step [4]. The algorithm proceeds as follows.

Algorithm 3.1 (Generic Particle Filter)

1. **(Initialize)** Sample $\theta^0 \sim \pi_0(\theta)$. Set $t \leftarrow 1$.
2. **(Importance Sampling-step)** Sample $\tilde{\theta}^t \sim q_t(\theta^{t-1}, D, \hat{\pi}_{t-1})$ and evaluate importance weights $w(\tilde{\theta}^t)$.
3. **(Selection-step)** Multiply/discard particles $\{\tilde{\theta}^t\}$ with respect to high/low importance weights to obtain N particles $\{\bar{\theta}^t\}$.
4. **(MCMC-step)** Sample $\theta^t \sim K(\{\bar{\theta}^t\}, d\theta^t)$ where K is a Markov kernel of invariant distribution $\pi_t(\theta)$.
5. **(loop)** Set $t \leftarrow t+1$ and go to Step 2.

At time t in Importance Sampling-step, q_t depends on the observed data D and the current approximated posterior distribution. Each particle θ_i^t is assigned to importance weight w_i^t. In Selection-step, particles are resampled to obtain an unweighted empirical distribution of the weighted measure. A selection procedure associates to each particle θ_i^t a number of offspring m_i^t such that $m_i^t = Nw_i^t$ and $\sum_{i=1}^{N} m_i^t = N$, so allows particles to give birth to some particles at the expense of particles with samll weights which are deleted. If the distribution of the importance weights is highly skewed then the particles which have high importance weights are selected many times and thus many particles will be identical. In MCMC-step, we can use any of

the standard MCMC methods to overcome this depletion of samples. If we apply a Markov chain transition kernel of invariant distribution $\pi_t(\theta)$, particles can be moved to more interesting areas [2].

3.2 Convergence of Generic Particle Filters

Theoretical convergence of particle filters is studied actively. Here we present some results from [4, 5]. Let $B(\Re^n)$ be the space of Borel measurable functions g at $\Re^n$. We denote

$$\|g\| = \sup_{x \in \Re^n} |g(x)|. \quad (8)$$

Let us consider the following assumptions. Importance distribution is chosen so that the corresponding importance weights are bounded above . We also assume that selection (resampling) scheme does not introduce too strong discrepancy like this, there exists a constant c so that

$$E\left[\left|\sum_{i=1}^{N} (m_i^t - Nw_i^t) g(\theta_i^t)\right|^2\right] \le c\, N \|g\|^2. \quad (9)$$

Then at each step of the particle filter algorithm, the approximation produced admits a mean square error of order $1/N$. The following theorem is straightforward consequence of Theorem 1 in [4].

Theorem 3.1

If the importance weights $\{w^t\}$ are upper bounded for any (θ^{t-1}, D) and offspring are generated proportional to importance weights, then for all $t \ge 0$, there exists constant c_t independent of N such that for any $g \in B(\Re^n)$

$$E\left[\left(\frac{1}{N}\sum_{i=1}^{N} g(\theta_i^t) - \int g(\theta)\pi_t(d\theta \mid D)\right)^2\right] \le c_t \frac{\|g\|^2}{N}.$$

If we assume more restrictive conditions about importance distribution and MCMC kernel, then we get weak convergence of empirical estimate $\hat{\pi}_t(\theta \mid D)$ toward posterior distribution $\pi_t(\theta \mid D)$. Theorem 3.2 is from Theorem 2 in [4].

Theorem 3.2

If the importance distribution and MCMC kernel are Feller Kernels, the importance weights are bounded continuous and offspring are generated proportional to importance weights, then for all $t \ge 0$,

$$\lim_{N \to \infty} \hat{\pi}_t(\theta \mid D) = \pi_t(\theta \mid D).$$

4. The Bayesian Particle Filter

Particle filters and BEAs have Bayesian properties i.e, incorporate observations of data into a prior updating routine and posterior distribution evolves over time through the accumulation of data arises. Canonical BEA observes data and computes likelihood. Then, updates posterior distribution and samples offspring from it. A generic particle filter starts by predicting particles in Importance Sampling-step and then update particles by observation of data.

We show that canonical BEA can be converted to a generic particle filter. We name particle filter version of the BEA as the Bayesian particle filter.

The algorithm of the Bayesian particle filter is as follows. We don't consider temperature scheduling and incremental data growth in canonical BEA. We assume fixed data set D which does not arrive sequentially and $L = M = N$ in canonical BEA for simplicity.

Algorithm 4.1 (Bayesian Particle Filter)

1. **(Initialize)** Generate random particles $\Theta^0 = \{\theta_1^0, \ldots, \theta_N^0\}$ from $\pi_0(\theta)$. Set weight of each particle $1/N$. Set generation count $t \leftarrow 0$.
2. **(D-step)** Observe data D and compute likelihoods $f(D \mid \theta_i^t)$.
3. **(P-step)** Estimate posterior distribution $\pi_t(\theta_i^t \mid D)$.
4. **(V-step)** Generate N variations $\Theta' = \{\theta_1', \ldots, \theta_N'\}$ by multiplying θ_i^t a number of offspring m_i^t where $m_i^t = Nw_i^t$, $\sum_{i=1}^{N} m_i^t = N$ and
$$w_i^t = \frac{f(\theta_i^t \mid D)}{\sum_{\theta_j' \in \Theta'} f(\theta_j^t \mid D)}.$$
5. **(S-step)** Each particle evolves independently according to Markov transition.
6. **(R-step)** Revise prior distribution. Rearrange weight of each particle $1/N$ to make random samples.
7. **(loop)** Set $t \leftarrow t + 1$ and go to Step 2.

We need no importance sampling distribution because canonical BEA samples from the posterior distribution directly. After observation of data in D-step, the posterior distribution is updated. In P-step, we estimate the posterior distribution. In general case, we cannot estimate the

posterior explicitly so we approximate the posterior distribution by empirical estimates of particles.

Selection-step of a generic particle filter that generates offspring proportional to weight is converted to V-step in the Bayesian particle filter. In V-step of canonical BEA, offspring are sampled from the posterior distribution via variation operators. In V-step of the Bayesian particle filter, the weight of each particle is determined proportional to its likelihood. Then each particle generates offspring proportional to the weight by multiplying itself with respect to high weight or discarding with respect to low weight. In S-step, each particle evolves independently according to Markov transition to prevent lack of variety of samples. Since we assume $L = M = N$, we select all evolved particles.

In R-step, we revise prior distribution by replacing it with current posterior distribution for next generation.

5. Convergence Properties of the Bayesian Particle Filter

We first see asymptotic convergence of the Bayesian particle filter using convergence properties of particle filters if simple sufficient conditions are given. Then, we show that canonical BEA that can be converted to the Bayesian particle filter converges to true posterior distribution under more restrictive conditions.

Our Bayesian particle filter doesn't require Importance Sampling-step. It consists of Selection-step and MCMC-step of generic particle filter algorithm. So we need to show that after each step of the Bayesian particle filter, approximation produced admits a mean square error of order $1/N$ if suitable conditions are imposed.

Theorem 5.1

For all $t \geq 0$, average mean square error of the Bayesian particle filter converges weakly to 0 as the number of particles increases.

(Proof)

We can use induction on generation count t. For $t = 0$, we assume we can sample N random particles exactly from $\pi_0(\theta)$, so for any $g \in B(\Theta)$, we get

$$E\left[\left(\frac{1}{N}\sum_{i=1}^{N} g(\theta_i^0) - \int g(\theta)\pi_0(d\theta \mid D)\right)^2\right] \leq \frac{\|g\|^2}{N}$$

Assume after R-step at generation $t-1$, there exists a constant c_{t-1} independent of N, we get

$$E\left[\left(\frac{1}{N}\sum_{i=1}^{N} g(\theta_i^{t-1}) - \int g(\theta)\pi_{t-1}(d\theta \mid D)\right)^2\right] \leq c_{t-1}\frac{\|g\|^2}{N}.$$

At the next generation t, P-step estimates posterior of each particle that depends on observed data in D-step. In V-step, we have $\{\theta_i'\}_{i=1}^N$ by generating a number of offspring m_i' for each particle θ_i' such that we get for some constant c_t' independent of N,

$$E\left[\left(\sum_{i=1}^{N}\left(m_i' g(\theta_i') - N w_i' g(\theta_i')\right)\right)^2\right] \leq c_t' N \|g\|^2.$$

In S-step, we apply to each particle $\{\theta_i'\}_{i=1}^N$ a Markov transition kernel whose invariant distribution is the posterior distribution, then the new particles $\{\theta_i^t\}_{i=1}^N$ are still distributed according to the posterior distribution of interest. Since importance weights $\{w_i'\}$ are upper bounded, by Theorem 3.1, we have for some constant c_t independent of N,

$$E\left[\left(\frac{1}{N}\sum_{i=1}^{N} g(\theta_i^t) - \int g(\theta)\pi_t(d\theta \mid D)\right)^2\right] \leq c_t\frac{\|g\|^2}{N}.$$

□

Theorem 5.2

For all $t \geq 0$, Bayesian evolutionary algorithms converge weakly to their true posterior distributions as the number of indiviuals increases.

(Proof)

Bayesian evolutionary algorithms can be converted to Bayesian particle filters as we mentioned in Sec 4. We simply check that BEAs have suitable sufficient conditions to ensure convergence. In P-step of BEAs, we sample from the posterior. Especially canonical BEA assumes it is Gaussian distribution. It is straightforward to check that this distribution is continuous and has Feller property. Also, importance weight functions (i.e, normalized version of the likelihood) are bounded above and continuous. In canonical BEAs, MCMC kernel can be omitted. So by Theorem 3.2, BEAs converge weakly to their true posterior distributions as the number of individuals increases. □

6. Conclusions

We proposed the Bayesian particle filter and presented convergence properties of Bayesian evolutionary algorithms not restricted to population size 1. We convert BEAs to Bayesian particle filters that approximate posterior distributions of individuals. Using the convergence properties of the particle filters, we show the BEAs converge to posterior distributions as the number of individuals increases. Future work includes convergence behaviors of BEAs under more general settings.

Acknowledgments

This work was supported in part by the Korea Ministry of Science and Technology through KISTEP under grant and by the Brain Korea 21 Project in 2000.

Bibliography

[1] Akashi, H. and Kumamoto, H. (1977), Random sampling approach to state estimation in switching environments, *Automatica* 13:429-434.

[2] Andrieu, C., de Freitas, J. F. G. and Doucet, A. (1999), Sequential MCMC for Bayesian model selection, *IEEE Higher Order Statistics Workshop*, Ceararea, Israel, pp. 130-134.

[3] Bäck, T. (1996), *Evolutionary algorithms in Theory and Practice*, Oxford University Press.

[4] Crisan, D. and Docet, A. (2000), Convergence of generalized particle filters, *Technical report CUED /FINFE-NG/TR 381*, Cambridge University Engineering Department.

[5] Crisan, D., Del Moral, P. and Lyons, T. (1999), Discrete filtering using branching and interacting particle systems, *Markov processes and Related Fields* 5(3): 293-318

[6] de Freitas, J. F. G. (1999), *Bayesian Methods for Neural Network*, PhD thesis, Department of Engineering, Cambridge University, Cambridge, UK.

[7] Gilks, W.R., Richardson, S., and Spiegelhalter, D. J. (1996), *Markov Chain Monte Carlo In Practice,* Chapman & Hall.

[8] Isard, M. and Blake, A. (1998), Condensation-conditional density propagation for visual tracking, *Int. J. Computer Vision*, 29, 1: 5-28.

[9] Kanazawa, K., Koller, D. and Russel, S. (1995), Stochastic simulation algorithms for dynamic probabilistic networks, *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, Morgan Kaufmann, pp. 346-351.

[10] Liu, J. S. and Chen, R. (1998), Sequential Monte Carlo methods for dynamic systems, *Journal of the American Statistical Association* 93: 1032-1044

[11] Roberts, C. P. and Casella, G. (1999), *Monte Carlo Statistical Methods*, Springer, New York.

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

[13] B. –T. Zhang, Gerhard Paass, Heinz Mühlenbein (2000), Convergence properties of incremental Bayesian evolutionary algorithms with single Markov chains. *In Proc. 2000 Congress on Evolutionary Computation (CEC2000),* IEEE Press, pp. 938-945.

[14] B. –T. Zhang (2000), Bayesian methods for efficient genetic programming. *Genetic Programming and Evolvable Machines*, vol. 1, no. 3. pp. 217-242.