Information Sciences 299 (2015) 262–282

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Evolutionary sampling: A novel way of machine learning within a probabilistic framework

Zhenping Xie [a], Jun Sun [b,*], Vasile Palade [c,*], Shitong Wang [a], Yuan Liu [a]

[a] School of Digital Media, Jiangnan University, Wuxi 214122, China
[b] Key Laboratory of Advanced Process Control for Light Industry (Ministry of Education), Jiangnan University, Wuxi 214122, China
[c] Department of Computer Science, University of Oxford, Parks Road, Oxford OX1 3QD, United Kingdom

ARTICLE INFO

Article history:
Received 3 February 2014
Received in revised form 27 November 2014
Accepted 2 December 2014
Available online 19 December 2014

Keywords:
Evolutionary sampling
Support sample model
Monte Carlo Markov chain
Rejection sampling
Online learning
Particle swarm optimization

ABSTRACT

In many traditional machine learning methods, sampling is only a process of acquiring training data. However, some studies (on sequential Markov chains and particle filters) have demonstrated that sampling can be used for solving some intractable optimization problems in classical learning methods. Along this line of thinking, the relationships between sampling and learning are theoretically exploited in this paper, wherein the key feature of the sampling process is selecting representative samples from original data that can be modeled by a probability distribution. In theory, acquiring reliable samples is not an easy task for an arbitrary probability distribution. Motivated by approaches in evolutionary computation, rejection sampling and function approximation, a novel sampling strategy, called the evolutionary sampling, is proposed in this paper, and a machine learning method, called the evolutionary sampling approach (ESA), is put forward afterwards. Within ESA, a computing model, called the support sample model (SSM), is presented as well and is used to approximate an original density function. Accordingly, a concrete implementation of an evolutionary sampling approach (ESA) is proposed to seek the optimal model parameters of the SSM. Benefiting from the combination of rejection sampling and evolutionary searching, the ESA can theoretically converge to the optimal solution by minimizing the total variation distance, and can do this with high computational efficiency. Moreover, the normalized factor of a density function can be automatically estimated with high precision within the ESA. As a result, the ESA may be suitable for machine learning problems that could be transformed into density function approximation problems within a probabilistic framework. In addition, derived from the rejection sampling strategy, the ESA can also have online learning abilities required by large-scale data stream processing tasks. Theoretical analyses and application studies are carried out in this paper, and the results demonstrate that the ESA, as a novel way of machine learning, has several prominent merits aspired by past researches in machine learning.

* Corresponding authors at: Key laboratory of Advanced Control for Light Industry (Ministry of Education, China), Jiangnan University, No. 1800, Lihu Avenue, Wuxi, Jiangsu 214122, China. Tel.: +86 510 85912136; fax: +86 510 85912136 (J. Sun). Faculty of Engineering and Computing, Coventry University, Priory Street, Coventry CV1 5FB, United Kingdom. (V. Palade)

E-mail addresses: sunjun_wx@hotmail.com (J. Sun), vasile.palade@coventry.ac.uk (V. Palade).

http://dx.doi.org/10.1016/j.ins.2014.12.001
0020-0255/

1. Introduction

In modern information processing, sampling is an important approach for acquiring discrete data from original signals. By using computers, these discrete data can be more efficiently recorded, stored, transmitted, analyzed and visualized than their original forms. Therefore, sampling may be useful in solving those machine learning problems whose solutions could be represented by a group of samples [26,31]. More specifically, if a machine learning problem could be solved within a probabilistic framework, effective sampling strategies are needed to obtain representative samples (patterns). For example, fuzzy rules obtained by learning neuro-fuzzy systems from data can be related to representative samples of the practical problem that is being modeled. Similarly, support vectors in support vector machines (SVM) can also be viewed as pivotal sample vectors of the original dataset, and neural nodes in a neural network can also be related to typical representatives of all available samples. In addition, a group of particles (samples) are used to reliably represent a solution's state configuration in particle filter methods [14,9,40,49].

The above analysis convinced us that, if a solution could be represented by a group of particular samples, the solving (or learning) process of the corresponding problem could also be fulfilled by the sampling process. This new idea lays the foundation of probing novel machine learning methods within a probabilistic framework in this paper. To unify the concept, we introduce a new term, called the "support sample set", to describe the sample set that represents a solution of the problem, such that the problem can be solved by acquiring an optimal support sample set.

Although the support sample set (SSS) should be integrally treated in representing a solution, the procedure of acquiring the optimal support sample set could be implemented by adjusting individual support samples iteratively. This strategy is very similar to evolutionary computation algorithms, especially particle swarm optimization, which evolve the candidate solutions until the optimum is found. We expect that this idea borrowed from evolutionary computation could prove valuable in improving the learning performance when seeking the optimal support sample set. Motivated by the above thinking, we propose a new sampling procedure, called the evolutionary sampling, which is the core of our contributions in this paper. Accordingly, the learning procedure based on evolutionary sampling is called the evolutionary sampling approach (ESA) and is proposed in this paper.

The ESA, which can naturally be considered the combination of sampling approximation (rejection sampling strategies) and evolutionary optimization, will inherit their excellent characteristics. In theory, the ESA is a novel development of rejection sampling and extends its application to the field of machine learning. It also extends the application scope of evolutionary optimization. While ESA is a sampling method with the strategy of evolutionary algorithms incorporated into sampling approximation, estimation of distribution algorithms (EDAs) and EDA-like evolutionary algorithms (such as compact differential evolution, compact particle swarm optimization, disturbed exploitation compact differential evolution and memetic compact differential evolution are modern stochastic optimization methods exploring the space of potential solutions by building and sampling explicit probabilistic models of promising candidate solutions [20,29,33–35]. Although both ESA and EDA-like optimization algorithms appear to be combinations of sampling method and evolutionary algorithm, their purposes are completely different. ESA is a novel sampling method but EDA-like algorithms are optimization techniques.

In brief, the purpose of the evolutionary sampling proposed in this paper is to obtain an optimal approximation of any pointwise computable density function by using finite samples, which is a fundamental problem in statistics and statistical machine learning area. Mainly, our novel ESA algorithm combines the rejection sampling with other strategies in order to address the above goal within the probabilistic framework. Consequently, the ESA can be suitable for almost all the machine learning problems that can be solved within a probabilistic framework. Additionally, some important machine learning problems can be described as (or could be converted into corresponding) density function approximation problem problems, such as the estimation of the density distribution of the characteristic data or the joint distribution between input data and output data. Thus, the evolutionary sampling approach may work well for many machine learning problems. Since the proposed ESA is a general sampling optimization strategy, specific algorithms based on ESA and aiming for different practical applications should be analyzed.

The remainder of the paper is structured as follows. In Section 2, background knowledge and some important discussions are given. Section 3 presents the detailed implementation and the characteristics of the evolutionary sampling approach together with some important theoretical results. In Section 4, the experimental analyses on the ESA are performed. In Section 5, several ESA-based machine learning algorithms are proposed and some theoretical and experimental studies are performed on them. Finally, all important contributions of our studies are concluded in Section 6.

2. Background

2.1. Sampling and problem solving

Although sampling, as an important data processing method, has been widespread used in statistics, it became even more important when the computing science was born. By means of specific sampling methods, Von Neumann and others successfully performed complicated computing and simulations of the motion of physical particles in nuclear physics in the middle of the last century. The Monte Carlo integration [26] was the core of most problems, which computed an estimate of the following integral:

$$I = \int_D g(x)\pi(x)dx,$$

where $\pi(x)$ was a probability density function, $g(x)$ was a function and D was the region of integration. In general, it is very difficult to directly calculate the accurate value of I by analytical methods. However, according to the law of large numbers, Von Neumann et al. proved that the value of I could be estimated using the following formula:

$$I \approx \frac{1}{N}\sum_{i=1}^{N} g(x_i),$$

where $X = \{x_i;\ i = 1,2,\ldots,N\}$ was a sample set generated from $\pi(x)$. Thus, how to obtain a good sample set from $\pi(x)$ becomes the most important problem in the above computing problem. Besides, as a special form of sampling from some standard probability density functions (uniform distribution, normal distribution, etc.), the random number generating approach also plays a fundamental role in many modern computing and simulating problems, such as in evolutionary computation [44,7,16], particle filtering [49], and condensed sampling [4].

2.2. Basic sampling methods

There are three types of practical sampling methods. The first one is represented by the standard pseudo-random number generating methods. Two examples of this type are the multiply-with-carry (MWC) method proposed by George Marsaglia to generate pseudo-random numbers of a uniform distribution on [0, 1], and the normal random number generator to generate pseudo-random numbers of a standard normal distribution [11,36,28]. The second one is the inverse transform sampling method [11], which can perform sampling effectively by using the uniform random number generator on [0, 1] when the corresponding probability density function has an explicit inverse expression. The third one is the rejection sampling method, firstly introduced by von Neumann [50]. Among the three sampling methods, the first two types of sampling strategies have excellent sampling efficiency but poor applicability. Compared to the third method, they need less sampling time to acquire sufficient valid samples, but do not fit for most sampling requirements. The first type of sampling methods is only suitable for a few standard probability density functions. The second type of sampling methods, which have relatively wider applicative scopes than the first type, cannot be widely used though, since most probability density functions have no explicit inverse expressions. In contrast, rejection sampling methods are more general since they can fit almost all sampling demands in theory [26]. Many types of rejection sampling strategies have been proposed, including classical rejection sampling [50], Monte-Carlo Markov Chain [31], and many later variants [22,13,30,5,42]. Although existing rejection sampling strategies have been widely used in many sampling requirements, how to improve their sampling efficiency is still an open problem, where sampling efficiency can be evaluated by the accepted ratio, which is defined as the ratio (probability) of the number of accepted valid samples to that of generated candidates. On the contrary, the candidate samples generated by the other two types of sampling methods are all valid samples so that their accepted ratios equal to 1. That is, they produce many repetitive samples in the sample set using current rejection sampling strategies. In practice, we only need some valid samples, and thus many repetitive samples are not desirable.

Except for some specific sampling methods used in applied statistics (economic statistics, population and social statistics, etc.), the existing sampling methods fall into the aforementioned three types, which are widely used in statistics and machine learning. The first two types, namely the standard pseudo-random number generating methods and the inverse transform methods, are not our main focus in this study and will not be further discussed in this paper. Among all rejection sampling methods, the Metropolis-Hasting (MH) rejection sampling procedure is an important and representative method, which has a better real efficiency than the pioneer method proposed by von Neumann, and can be outlined as follows [26].

Metropolis-hasting rejection sampling	
1	Given current state $x^{(t)}$
2	Draw y from the proposal distribution $T(x^{(t)}, y)$ Draw $U = Uniform[0, 1]$ and update
3	$x^{(t+1)} = \begin{cases} y, & \text{if } U \leqslant r(x^{(t)}, y) \\ x^{(t)} & \text{otherwise} \end{cases}$

where $r(x,y) = \min\left\{1, \frac{\pi(y)T(y,x)}{\pi(x)T(x,y)}\right\}$ as suggested by Metropolis et al. and Hastings [26], and $Uniform[0, 1]$ is the generator for random numbers uniformly distributed on [0, 1].

With the iterative execution of the MH rejection sampling, for any given initial $x^{(0)}$, the probability of generating $x^{(t)}$ approximates $\pi(x)$ and the corresponding Markov chain tends to converge. Thus, k samples $x^{(i)}, i = [n+1, \ldots, n+k]$ can be obtained from $\pi(x)$, where n is the least iteration number needed to reach its stable distribution of $x^{(t)}$.

Apart from the MH sampling, many other rejection sampling strategies in machine learning have also been put forward, such as the weighted re-sampling method [17,2,12], the block sampling method [13], the backward revised sampling method [30], the variational Monte Carlo method [18,6], wherein the weighted re-sampling method is most widely used.

3. The evolutionary sampling learning

Based on the preliminary discussions on the evolutionary sampling strategy in the previous sections, this section provides a complete exploration on the relevant aspects. A novel machine learning method based on evolutionary sampling, named as the evolutionary sampling approach (ESA), is proposed and discussed. The ESA is used to acquire the optimal model parameters of the support sample model related to an expected probability density distribution. Here, the model parameters of the support sample model mainly refer to its support samples. The section begins with the description of the definition of the support sample model (SSM).

3.1. The Support Sample Model (SSM)

As pointed out in the above discussion, some additional explanations imposed on a support sample set can lead to a new computing model, i.e. the support sample model, which can be used to approximate (or substitute) any original probability density distribution. In fact, many current computing models also have their own additional computing rules on their core parts. For example, in many kernel density estimation methods, it is assumed that the probability density of the data can be estimated by computing a weighted combination of a group of Gaussian density functions. The combination operator and the Gaussian density function are essentially the additional explanations on the original data. Similarly, the same situations can be found in neural networks, fuzzy systems, and support vector machines.

To introduce reasonable computing rules on the support sample set, we are to begin our theoretical analysis starting from the support samples themselves. When some sample data are gathered to form support samples under an unknown rule, new relationships among these support samples are also built. It is well-known that the most fundamental strategy for figuring out the relationship between two different objects is to measure their distance. In mathematics, defining a distance measure is also a precondition for the definition of a space. For a set of support samples, when a distance measure on them is imported, their relationships also will be naturally created. Moreover, a probability measure on these support samples can be defined by introducing additional interpretations that reflect the probability of another sample when one sample exists. In present researches, the most classical interpretation is a Gaussian density distribution, as used in kernel density estimation [54,51]. The Gaussian density distribution presumes that the influencing probability of one sample to another sample obeys a Gaussian function in a statistical meaning.

Next, we let X and $\pi_X(\bullet)$ represent the feature data of certain object O_X and the probability distribution of X respectively, where the variable range of X is denoted by D_X (discrete or continuous space). Likewise, we define $\pi_Y(\bullet)$ as the probability distribution on the feature data of the object O_Y, and, then, if there exists the relationship between O_X and O_Y, we may define $\pi_{X\times Y}(\bullet) = \pi_X(\bullet) \oplus \pi_Y(\bullet)$ as their joint probability distribution, where $\oplus$ is a connecting operator. Similarly, the definition $\pi_Z(\bullet) = \pi_{X_1}(\bullet) \oplus \pi_{X_2}(\bullet) \cdots \oplus \pi_{X_m}(\bullet)$ also can be given for the case of connecting multiple objects. Thus, we can describe an object or the relationship to different objects using the probability distribution of its feature data or their feature data. Nonetheless, the exact expressions of these probability distributions are unknown in practice. To tackle this problem, we must introduce computable probability distribution expressions on these objects or their relationships. In the support sample model, we let $Mp_X(\bullet)$ and $Mp_Y(\bullet)$ denote computable expressions of describing O_X and O_Y, respectively, using a group of support samples within the probabilistic framework. Moreover, we consider that $Mp_X(\bullet)$ has the form $Mp_X(x) = Mh(x) \otimes K(x;\ \theta)$, where $\otimes$ is a convolution operator, $Mh(x)$ is the probability distribution of the support samples, and $K(x;\ \theta)$ is an additional interpretation with probabilistic form (like a kernel function). In this paper, a Gaussian kernel function is considered for $K(x;\ \theta)$, thus θ is the Gaussian kernel parameters. Consequently, we can redefine $Mp_X(\bullet)$ in the form $Mp(x) = \frac{1}{N_S}\sum_{x_i \in X_S} K(x, x_i; \theta)$ by means of the Monte Carlo integration formula, where N_S is the cardinal number of the support sample set $\mathbf{X}_S$. Reviewing the definition of $Mp(x)$, we can see that the support samples are the core and $K(x;\ \theta)$ reflects the contribution of any support sample to the total model output. This consideration coincides with the common experience of a human that when recognizing an object is mainly dependent on its representative appearances.

3.2. The evolutionary sampling approach

As introduced in the preceding section, the goal of the evolutionary sampling learning is to acquire an optimal SSM to approximate another probability distribution. Generally, the above problem can be formalized as the following optimization problem within a probabilistic framework.

$$\mathbf{X}_S^* = \arg\min_{\mathbf{X}_S} \{\|Mp(\bullet) - \pi(\bullet)\|_{TV}\}, \tag{1}$$

where $\mathbf{X}_S^*$ is the optimal combination of all possible support samples, $Mp(\bullet)$ and $\pi(\bullet)$ represent the probability distributions related to the SSM and the objective model, respectively, and $\|\bullet\|_{TV}$ denotes the total variation distance. For the convenience of the description, we let $p(\bullet)$ and $G(x, y, \theta)$ substitute $Mp(\bullet)$ and $K(x, y;\ \theta)$, respectively, in the following text. As introduced above, we will introduce a new computing strategy, called the evolutionary sampling approach (ESA), to solve the above problem.

Adopting the core idea of the rejection sampling strategy, a similar sampling procedure is designed for each support sample, thus there are N_S concurrent rejection sampling chains in the ESA. For each sampling chain, a candidate sample is firstly generated and then it is received as the new sample with a certain probability. Generally, we let $A^k(x, y)$ denote the probability of generating a new sample datum y from x at the kth rejection sampling step, and let $\alpha^k(x, y)$ denote the probability of receiving datum y as the new sample datum if previous sample datum is x at the k th rejection sampling step. So, if the current support sample datum is x, then the probability of receiving the sample datum $y(y \neq x)$ as the new support sample is $A^k(x, y) \times \alpha^k(x, y)$ after performing the kth rejection sampling step, which is the same as the traditional rejection sampling strategy. Thus, the concrete expressions of $A^k(x, y)$ and $\alpha^k(x, y)$ should be carefully designed, which will completely determine the final sampling results. According to our explorations, the following formulas are proposed to solve the problem in Eq. (1).

$$A^k(x,y) = \begin{cases} q^k(x,y) \times \alpha_{MH}(x,y) & x \neq y \\ \int q^k(x,t) \times [1 - \alpha_{MH}(x,t)]dt & x = y \end{cases}, \tag{2}$$

$$\alpha^k(x,y) = \max\left\{1 - \frac{\pi(x) \times p^k(y)}{\pi(y) \times p^k(x)}, 0\right\}, \tag{3}$$

where $q^k(x, y)$ is the probability of directly generating a new candidate sample y from x, which has $\int q^k(x,t)dt = 1 \quad q^k(x,t) > 0 \quad \forall x, t, k$, and satisfies the symmetry, that is $q^k(x, y) = q^k(y, x)$. For $\alpha_{MH}(x, y)$, the same form as in the standard Metropolis-Hasting rejection sampling is used, i.e., $\alpha_{MH}(x,y) = \min\left\{1, \frac{\pi(y)}{\pi(x)}\right\}$. So, the equation $\pi(x)A^k(x, y) = \pi(y)A^k(y, x)$ is absolutely satisfied. In practice, $q^k(x, y)$ should have the form that can directly generate the candidate support sample y from x with a probability of $q^k(x, y)$, such as the exponential distribution function and the Gaussian distribution function. Different from traditional rejection sampling strategies, $\alpha^k(x, y)$ is newly designed to ensure that the stable solution of $p^k(x)$ can approximate $\pi(x)$. Wherein, according to the definition of $p(x)$ in the SSM, we have

$$p(x) = \frac{1}{N_S}\sum_{x_i \in \mathbf{X}_S} K(x, x_i; \theta). \tag{4}$$

Next, combining the above discussions, we can put forward the concrete implementation of the ESA as follows.

The Evolutionary Sampling Approach (ESA)

1 Select N_S initial support samples; let the initial support sample set be $\mathbf{X}_S^0 = \left\{x_1^0, x_2^0, \ldots, x_{N_S}^0\right\}$, and let $k = 0$.
2 For each support sample x_i^k in $\mathbf{X}_S^k$, perform step $3 \sim 6$.
3 Generate a candidate support sample datum y_i^C according to the searching probability $q^k(x_i^k, y)$.
4 Get a uniform random number R on [0, 1]; if $R < \min\left\{1, \frac{\pi(y_i^C)}{\pi(x_i^k)}\right\}$ receive y_i^C provisionally, then go to the next step; otherwise reject it and go to step 6.
5 Calculate the value of $\alpha^k(x_i^k, y_i^C)$ according to Eq. (3), generate another uniform random number R' on [0, 1], if $R' < \alpha^k(x_i^k, y_i^C)$ receive y_i^C successfully, otherwise reject it.
6 If y_i^C is received successfully, let $x_i^{k+1} = y_i^C$; otherwise let $x_i^{k+1} = x_i^k$.
7 Update $k = k + 1$ and let new support sample set be $\mathbf{X}_S^k = \left\{x_1^k, x_2^k, \ldots, x_{N_S}^k\right\}$.
8 If the changing difference between two contiguous support sample sets is very small or other termination conditions are satisfied stop the sampling procedure; otherwise go to step 2.
9 Output the final support sample set $\mathbf{X}_S^k$ and the corresponding $p^k(x)$.
10 END

In the above evolutionary sampling approach, the uniform random number can be generated by standard pseudo-random number generating methods, such as the linear congruential generator LCG [36,28]. Except for the uniform random generating methods, the expressions of $K(x, y; \theta)$ and $q^k(x, y)$ should also be precisely defined in the ESA. To answer this problem, some restrictive conditions for $K(x, y; \theta)$ and $q^k(x, y)$ should be firstly considered as follows. (1) All values are non-negative and finite. (2) They are finitely integrable. (3) New candidate sample datum y can be directly generated from x according to $q^k(x, y)$. When the above three restrictions for $K(x, y; \theta)$ and $q^k(x, y)$ are satisfied, exact definitions of them are alterable, such that we can define different forms for different practical applications even for different stages in the same sampling procedure. For example, reasonable $K(x, y; \theta)q^k(x, y)$ could be selected to achieve more exact approximation for certain particular applications, and specific values could be considered to improve the searching performance of the evolutionary sampling procedure in other cases. Of course, it must be pointed out that, in theory, the same optimal results can be equally obtained under different $q^k(x, y)$, but the same $K(x, y; \theta)$. In contrast, different $K(x, y; \theta)$ will produce different approximation results to some extent (more discussions will be given in the later text). Combining the above conclusions and our explorations, the Gaussian kernel function for $K(x, y; \theta)$ and the symmetrical exponential function for $q^k(x, y)$ are proposed as follows.

$$K(x,y;\sigma)=\frac{1}{(2\pi\sigma^2)^{d/2}}\exp\left\{-\frac{\|x-y\|}{2\sigma^2}\right\}, \tag{5}$$

$$q^k(x,y;\beta)\propto e^{-|x-y|/\beta}. \tag{6}$$

In the expression of $K(x, y; \sigma)$, σ is the kernel width, x, y are all d-dimension data, $\|\bullet\|$ is the vector norm and the 2-norm is the default setting in this paper. For $q^k(x, y; \beta)$, the idea mainly came from our previous studies on the quantum-behaved particle swarm optimization QPSO [44,45] and our comparisons, where β is the searching scale factor. Obviously, the scale factor β can be adaptively adjusted in the sampling procedure, which features the idea of evolutionary searching.

In the ESA, it can be found that the sampling procedure is executed in parallel for each support sample, and the final updating probability of support sample set will tend to zero. On the contrary, the ultimate average receiving probability of new candidate samples will converge to a stable nonzero value in traditional rejection sampling methods. Their difference primarily derives from the different solving objectives. That is, the ESA focuses on acquiring optimal support samples to approximate very well the original probability distribution, which demands that the learning process produces a fixed solution. However, since a traditional rejection sampling method generates sample data continuously according to the original probability distribution, its sampling procedure must last forever. Another difference between the ESA and traditional rejection sampling methods can also be clearly found in that the two rejection operators (determined by $A^k(x, y)$ and $\alpha^k(x, y)$, respectively) are performed twice for each candidate sample at every sampling iteration in the ESA, while only once is required in traditional rejection sampling methods. As a result, the real receiving ratio of the new candidate support sample is decided by the product of the two receiving probabilities.

3.3. Theoretical analysis

As described above, the ESA mainly intends to gain an optimal approximation of the original probability distribution. So, the convergence and the actual approximation performance must be analyzed as the two most important performance indexes of the ESA, which will be done in this subsection. Some other properties are also explored, including the computational complexity, how to improve the practical approximation performance, as well as some theoretical discussions on possible applications.

Derived from the concepts of evolutional computation and sampling, the ESA can ensure that the obtained support sample set is probabilistically optimal within limited number of samples and the given SSM. Moreover, the SSM output can steadily and asymptotically converge (in probability) to the original probability distribution by minimizing the total variation distance between the SSM output and the original probability distribution. Although the performing process of the ESA is different from current rejection sampling methods, it is similar to those of usual machine learning methods, such as support regression models [52,3], the coreset model [10], and fuzzy systems [8]. In these approaches, firstly, they all artificially construct a machine model and then use appropriate learning methods to obtain optimal model parameters. Another fact that should be emphasized is that, unlike many traditional machine learning methods that need to solve complicated optimization problems like linear programming and quadratic programming, the ESA is a forward learning process, in a similar way as classical rejection sampling methods. This characteristic makes the ESA have good computing efficiency and low computational complexity. On the other hand, the introduction of evolutionary methods from swarm optimization algorithms will render that the ESA can efficiently achieve the optimal solution.

From the view of evolutionary computation, ESA's learning behavior possess some common features with estimation of distribution algorithms (EDAs), like DE/EDA [46], compact differential evolution [29], compact particle swarm optimization [35], disturbed exploitation compact differential evolution [33], and memetic compact differential evolution [34]. In those EDA-like methods, the optimization procedure is designed to find the local probability of the optimal solution, in which few limited variables are employed to represent the local probability and so those algorithms may only need very low memory usage. Obviously, ESA and EDA-like methods use similar representation strategy to find the probability of the optimal solution. Furthermore, some evolutionary strategies designed in them on those representation variables may be mutually borrowed in theory. Nevertheless, since the two types of algorithms are considered to solve different types of problems as mentioned in the introduction section, they play distinct roles in practical applications.

3.3.1. Convergence analysis on ESA

It is well known that the convergence that will be analyzed here must be ensured for any machine learning algorithms. For ESA, if $p^k(\bullet)$ has a stable solution with the execution of evolutionary sampling, then the algorithm has converged. According to the definition of $p^k(\bullet)$, it completely depends on the corresponding support sample set X_S^k under a given kernel density function $K(\bullet, \bullet; \theta)$. Further, we can consider X_S^k samples from a distribution $h^k(\bullet)$, furthermore, $p^k(\bullet)$ could be viewed as an estimation of $h^k(\bullet)$.

$$p^k(x)=\int h^k(t)\times K(x,t;\theta)dt. \tag{7}$$

For the need of analyzing the convergence of ESA, we firstly introduce the following Theorem 1.

Theorem 1. *Suppose that $\pi(x)$ is a probability density function, $T(x, y)$ is a transition probability function used in the rejection sampling, which satisfies the transition invariance to $\pi(x)$, that is $\pi(x)T(x, y) = \pi(y)T(y, x)$. With any given initial probability density function $p^0(x)$, if $p^{k+1}(x) = \int p^k(t)T(t,x)dt$ holds, then:*

$$\lim_{k\to\infty}\left\|p^k(\bullet) - \pi(\bullet)\right\|_{TV} = 0. \tag{8}$$

Proof. This could be directly derived according to the convergence analysis of the classical rejection sampling method [26].

Because of particular characteristics of ESA, we firstly analyze the case that N_S is inefficiently large and $p^k(\bullet)$ may be viewed as the approximation of $h^k(\bullet)$ in arbitrary high precision. Under this consideration, we can deduce the following Theorem 2. □

Theorem 2. *For the ESA, if $\pi(x)$ is a normalized probabilistic density function, that is $\int \pi(x)dx = 1$, $p^k(\bullet)$ is an approximation of $h^k(\bullet)$ with arbitrary high precision, then with the progress of the evolutionary sampling, the SSM output $p^\infty(\bullet)$ tends to a stable solution $p^*(\bullet)$. Equivalently, $\lim_{k\to\infty} p^k(\bullet) = p^*(\bullet)$, at the same time $p^*(\bullet)$ optimally approximates $\pi(\bullet)$ with the minimal total variation distance between $p^*(\bullet)$ and $\pi(\bullet)$.*

Proof. According to the condition and the iterative procedure of ESA, the following equation exists for $p^{k+1}(\bullet)$, $p^k(\bullet)$ and $\pi(\bullet)$.

$$p^{k+1}(x) = \int p^k(t) \times A^k(t,x) \times \alpha^k(t,x)dt + p^k(x) \times \int A^k(x,t) \times [1 - \alpha^k(x,t)]dt \tag{9}$$

If let

$$p^k(x) = r^k(x)\pi(x) \tag{10}$$

then Eq. (9) can be rewritten as

$$p^{k+1}(x) = r^{k+1}(x)\pi(x) = \int r^k(t)\pi(t) \times A^k(t,x) \times \alpha^k(t,x)dt + r^k(x)\pi(x) \times \int A^k(x,t) \times [1 - \alpha^k(x,t)]dt \tag{11}$$

Moreover:

$$r^{k+1}(x) = \int r^k(t) \times A^k(x,t) \times \alpha^k(t,x)dt + r^k(x) \times \int A^k(x,t) \times [1 - \alpha^k(x,t)]dt \tag{12}$$

Combining Eq. (3) into the above equation, we have

$$\begin{aligned} r^{k+1}(x) = &\int r^k(t) \times A^k(x,t) \times \max\left\{1 - \frac{\pi(t) \times p^k(x)}{\pi(x) \times p^k(t)}, 0\right\}dt + r^k(x) \times \int A^k(x,t) \\ &\times \left[1 - \max\left\{1 - \frac{\pi(x) \times p^k(t)}{\pi(t) \times p^k(x)}, 0\right\}\right]dt \end{aligned} \tag{13}$$

Additionally, $\alpha^k(x,y) = \max\left\{1 - \frac{\pi(x)\times p^k(y)}{\pi(y)\times p^k(x)}, 0\right\} = 1 - \min\left\{\frac{\pi(x)\times p^k(y)}{\pi(y)\times p^k(x)}, 1\right\}$, and substituting it into Eq. (13), we have:

$$\begin{aligned} r^{k+1}(x) &= \int r^k(t)A^k(x,t)dt - \int A^k(x,t)r^k(t)\min\left\{\frac{\pi(t) \times p^k(x)}{\pi(x) \times p^k(t)}, 1\right\}dt + \int A^k(x,t)r^k(x)\min\left\{\frac{\pi(x) \times p^k(t)}{\pi(t) \times p^k(x)}, 1\right\}dt \\ &= \int r^k(t)A^k(x,t)dt \end{aligned} \tag{14}$$

Equivalently,

$$p^{k+1}(x) = \pi(x)\int \frac{p^k(t)}{\pi(t)}A^k(x,t)dt = \int \frac{p^k(t)}{\pi(t)}\pi(t)A^k(t,x)dt = \int p^k(t)A^k(t,x)dt \tag{15}$$

Thus, the Theorem 2 is proved according to Theorem 1. □

In Theorem 2, we consider that $p^k(\bullet)$ is an estimation of $h^k(\bullet)$ in high precision. Practically, it may not be ensured where N_S is finite. Nevertheless, we could think that $p^k(\bullet)$ still will converges to a stable solution that approximates $\pi(\bullet)$ ($\lim_{k\to\infty} p^k(\bullet) \equiv p^*(\bullet) \to \pi(\bullet)$) according to a qualitative analysis.

In Theorem 2, we suppose that $\pi(x)$ is a normalized probability density function admitting $\int \pi(x)dx = 1$. However, the normalization factor of $\pi(x)$ usually do not equal to 1, and is very hard to calculate it in most practical applications. For this case, the following lemma can be deduced.

Lemma 1. *If the normalization factor of $\pi(x)$ is not equivalent to one in the conditions of Theorem 2, then there is $\lim_{k\to\infty} p^k(\bullet) \equiv p^*(\bullet) \to \pi(\bullet)/\lambda_\pi$, where λ_π is the normalization factor of $\pi(\bullet)$, that is $\int \pi(x)dx = \lambda_\pi$.*

Proof. Let $\pi'(x) = \pi(x)/\lambda_\pi$. In terms of the definitions of $A^k(x, y)$ and $\alpha^k(x, y)$, they have the same computational values with respect to $\pi'(x)$ and $\pi(x)$. On the other hand, the final solution of the ESA is only dependent on $A^k(x, y)$ and $\alpha^k(x, y)$, therefore if we replace $\pi(x)$ with $\pi'(x)$ in the ESA, an equivalent solution will be produced. That is, the $\lim_{k\to\infty} p^k(\bullet) \equiv p^*(\bullet)$ still exists according to Theorem 2, and $p^*(\bullet)$ may optimally approximate $\pi'(\bullet)(\pi'(\bullet)/\lambda_\pi)$ in the same way. The lemma is now proven. □

3.3.2. Approximation performance of the ESA

For convenience, we firstly analyze the case when the original probability distribution could be exactly approximated (or equivalently represented) by an SSM with given kernel parameters and N_S. In this case, the practical solution acquired by the ESA might tend to the ideal solution with infinitesimal difference, equivalently $p^k(x) \to \pi(x)$. Accordingly, $\alpha^k(x, y) \to 0$ exists, which is a very noteworthy conclusion and will help us design the excellent termination condition for the ESA. In terms of this conclusion, the updating ratio of the support samples at every iteration could be employed for taking the decision of terminating the sampling process. Obviously, the lower this value the higher the approximation degree.

Nevertheless, the real updating ratio is simultaneously dependent on $A^k(x, y)$ and $\alpha^k(x, y)$ in the ESA. So that, if the updating ratio tends to 0, it cannot consistently reflect $\alpha^k(x, y) \to 0$, but it only indicates that the sampling process tends to converge. As for the evolutionary searching, this discrepancy derives from the difference between the global optimum and the local optimum. In theory, when a searching algorithm tends to converge, it may converge to a local or global optimum solution. In practice, it is not easy to assuredly acquire the global optimum solution even if it is reachable given enough time, because the number of evolutionary iterations must be finite. So, further studies on its practical approximation performance should be performed for the ESA.

According to Eq. (2), which defines $A^k(x, y)$, the updating probability of the first rejection sampling depends on $q^k(x, y;\beta)$ and $\alpha_{MH}(x, y)$, simultaneously. Wherein, the computing expression of $\alpha_{MH}(x, y)$ is invariant during the evolutionary sampling process, and we have $\alpha_{MH}(x, y) > 0$ *for all* x, y. Accordingly, if $A^k(x, y) \to 0$, then we may have $q^k(x, y;\ \beta) \to 0$. To make the algorithm reach a valid solution within a finite number of evolutionary iterations, $q^k(x, y;\ \beta)$ usually tends to 0 by adjusting $\beta \to 0$. In the ESA, β reflects the scalar scope in searching new possible support samples from current support samples. Obviously, a large β will result in better abilities of finding new support samples, but more slower convergence speed, and vice versa. Furthermore, the value of β has no influence on the convergence and the approximation performance of the ESA, as indicated by the theoretical analysis. So, we can adaptively adjust β to achieve different goals in different sampling stages of the ESA. For example, a larger β might be suitable to keep wide searching scope in the starting stage, while a smaller β might be needed to fast converge to the fixed solution in the final stage.

According to the above analysis, in the case that $\pi(\bullet)$ might be exactly approximated by the SSM, the approximation performance is ideal if we could continuously perform the evolutionary sampling endlessly with $\beta > 0$. However, a finite number of sampling iterations must be considered in practice, so we should consider how to obtain better practical approximation performance. That is, how to better decide the sampling termination. Considering the above analysis, the following strategy could be designed. When a local stable solution has been reached, we may increase β to find more possible support samples. Thus, if a sufficient big value of β is used and new support samples still cannot be found, then the sampling learning might be ultimately terminated. This strategy will ensure the ESA can acquire good practical approximation performance.

Along the above discussion, we will examine the practical approximation performance on another case when $\pi(\bullet)$ cannot be exactly expressed by the SSM. For these cases, the following lemma could be concluded directly from Theorem 2.

Lemma 2. *Suppose that $p^*(\bullet)$ is the stable solution of the ESA with given $\pi(\bullet)$, which is a normalized probability density function. Then the following equations exist:*

$$\alpha^*(x_i, y) = 0 \quad \forall x_i \in X_s, \forall y, \pi(y) > \varepsilon \tag{16}$$

$$p^*(\bullet|X_s^*) = \min_{X_s} \|p(\bullet|X_s) - \pi(\bullet)\|_{TV} \tag{17}$$

where ε is extremely small value, where all y with $\pi(y) < \varepsilon$ may considered to be not reached in probability under the final evolutionary searching of ESA. According to Lemma 2, the following theorem could be further deduced.

Theorem 3. *If X_s^* denotes the stable support sample set achieved by the ESA, and $p^*(\bullet)$ is the stable solution, then for any x, we have:*

$$\lambda^* p^*(x) = \pi(x), x \in X_s \tag{18}$$

or

$$\lambda^* p^*(x) \geqslant \pi(x), x \notin X_s, \pi(x) > \varepsilon \tag{19}$$

where λ^* is a constant scalar.

Proof. At first, according to Eq. (16), we have

$$\alpha^*(x, y) = \max\left\{1 - \frac{\pi(x) \times p^*(y)}{\pi(y) \times p^*(x)}, 0\right\} = 0 \ \forall x \in X_s, \forall y, \pi(y) > \varepsilon \tag{20}$$

and equivalently,

$$\frac{p^*(x)}{\pi(x)} \leqslant \frac{p^*(y)}{\pi(y)} \ \forall x \in X_s, \forall y, \pi(y) > \varepsilon \tag{21}$$

According to lemma 2, we have the following equation:

$$\frac{p^*(x)}{\pi(x)} = \frac{p^*(y)}{\pi(y)} \ \forall x, y \in X_s \tag{22}$$

Therefore, we may let $\lambda^* = \frac{p^*(x)}{\pi(x)}$ $x \in X_s$, and Eq. (18) exists. Accordingly, Eq. (19) will be satisfied as well.

In conclusion, Theorem 3 is proven. □

Furthermore, the following lemma also can be deduced.

Lemma 3. *For any $\pi(x)$ with the normalization factor λ_π, and if denote $\lambda_\pi^* = \frac{1}{N_S}\sum_{x_i \in X_S^*} \frac{\pi(x_i)}{p^*(x_i)}$, then λ_π is the lowest bound of λ_π^*, that is $\lambda_\pi^* \geqslant \lambda_\pi$, and the equality $(\lambda_\pi^* = \lambda_\pi)$ is satisfied if and only if $\lambda_\pi^* p^*(x)$ can exactly approximate $\pi(x)$.*

Proof. Combining the definition of λ_π^* and Theorem 3, we have

$$\lambda_\pi^* = \frac{1}{N_S}\sum_{x_i \in X_S^*} \frac{\pi(x_i)}{p^*(x_i)} = \lambda^* \tag{23}$$

Consequently, $\lambda_\pi^* = \lambda^* = \int \lambda^* p^*(x)dx \geqslant \int \pi(x)dx = \lambda_\pi$, where the equation is satisfied if and only if $\lambda_\pi^* p^*(x)$ can consistently approximate π (x).

Hence, the lemma is proven. □

Furthermore, if $\pi(x)$ is not a normalized form, the ESA can use λ_π^* to ideally estimate its normalization factor λ_π, where λ_π^* is defined as:

$$\lambda_\pi^* = \frac{1}{N_S}\sum_{x_i \in X_S^*} \frac{\pi(x_i)}{p^*(x_i)} \tag{24}$$

Moreover, if let

$$\lambda_\pi^k = \frac{1}{N_S}\sum_{x_i \in X_S^k} \frac{\pi(x_i)}{p^k(x_i)} \tag{25}$$

λ_π^k will converge to the stable value when the ESA converges to the optimal stable solution. Hence, the change of λ_π^k between two consecutive k can be used as the decision to terminate the ESA learning process. That is, the ESA can be credibly terminated if λ_π^k is changing less than a tiny scalar even if we augment the searching scope (parameter β) of generating support samples, which is very useful for practical applications.

Thus, the following corollary can be obtained by combining Theorem 3 with Lemma 3.

Corollary 1. *In the ESA, for any $\pi(x)$, when $N_S \to \infty$ and the iteration number tends to infinity, then the optimal solution correlates to the minimal λ_π^*.*

Even if N_S and the iteration number cannot reach infinity, the conclusion of Corollary 1 will help us to theoretically select the optimal kernel function and its parameters.

3.3.3. Computational complexity of the ESA

The computational complexity is a key property for any computing methods in practical applications. Firstly, it is clearly that the space complexity of ESA is $O(N_S)$, which is equivalent to traditional rejection sampling algorithms, but is far less than most of other machine learning algorithms. The time complexity of the ESA comprises two aspects. The first one refers to the time complexity of generating random numbers and performing rejection sampling strategies, and can be expressed as $O(N_S)$ by a direct theoretical analysis, which is also equivalent to traditional rejection sampling methods. The second one refers to the time complexity of computing $p(x)$ for every new generated candidate support sample, which is $O(N_S \log N_S)$ on computing $p(x)$ of N_S support samples, when a fast estimation algorithm [54,53] is used. As a result, the total time complexity of the ESA is $O(LN_S(1 + \log N_S))$, where L is the total number of iterations for performing the evolutionary sampling. Although it is slightly greater than that of the traditional rejection sampling method (i.e., $O(LN_S)$), the time complexity of the ESA is still appealing, because $O(LN_S(1 + \log N_S))/L$ is only dependent on the number of support samples, but not linearly related to N (the number of original data), as required as in almost all traditional machine learning methods.

In summary, when the space and time complexities are considered together, the computing efficiency of the ESA will be remarkably superior to most existing machine learning methods when they are applied to the same problems.

4. Simulation analysis of ESA

In the above section, some theoretical conclusions have been deduced mainly with respect to the theoretical convergence, approximation performance, and computational complexity. Here, an experimental analysis will be performed in order to further assert the theoretical properties of the ESA. At first, the following Hermite polynomial, expressed by π_1, is introduced as the testing density function [24]:

$$\pi_1(x) = (1 + x + 2x^2)e^{-x^2} \tag{26}$$

Obviously, π_1 is point-wise computable but not a normalized form, and $\int \pi_1(x)dx = 2\sqrt{\pi} \approx 3.5449$ by theoretical calculation.

Similar to general machine learning methods, some initial conditions should be predefined in the ESA, including the kernel function and corresponding hyper-parameters, the number of support samples N_S, the maximum number of sampling iterations, initial support samples, and the searching strategy. In this section, some experimental results will be reported with different parameters settings. It should be pointed out that, though more reasonable kernel functions may engender better approximation performance, for an unknown density function, it is difficult to select the optimal kernel function, and it will be ignored in this paper (could be concerned with in practical applications). Thus, according to common experience and the statistical theory, the Gaussian kernel function listed in Eq. (5) will be a reliable selection and is more widely effective when there is no any particular prior knowledge. In addition, fast Gaussian transform [54,53] can be used to compute $p(x)$ effectively.

In theory, the kernel width σ, the number of support samples N_S, the maximum number of iterations $K_{\max}$, and the initial support sample set X_0, all might influence the approximation performance. Accordingly, several groups of different experimental results will be reported, respectively. Because the kernel width plays the crucial role in actual approximation performance, the experimental results with different σ will be firstly delivered, where the parameters setting $N_S = 300$ and $K_{\max} = 500$ are configured. In addition, N_S random data are generated from the standard normal distribution as initial support samples. For $q^k(x, y)$, Eq. (6) is used, where parameter $\beta = \beta_l \frac{1}{N_S}\sum_{x_i \in X_S}\|x_i - \bar{x}\|$ is adopted and β_l is linearly decreased from 20 to 1 with the progress of the evolutionary sampling, which is derived from experimental comparisons and our previous investigations on quantum-based particle swarm optimization algorithms [44,45,15,43], and can perfectly balance the weight between the global optimization and the convergence speed.

Table 1 lists the approximation results obtained by the ESA with different σ, where the optimal results for different performance indexes are bold same as other tables, λ^* refers to the evaluation of the normalization factor of π_1 computed by Eq. (24), J_{err} refers to the approximation error defined as

$$J_{err} = \sqrt{\sum_{i=1}^{N}(p(x_i) - \pi(x_i))^2} \tag{27}$$

where $\pi(x)$ represents the target distribution (here it refers to π_1), and $p(x)$ represents the normalized output, that is $p(x) = \lambda^* p^*(x)$. In addition, 1000 data points uniformly distributed on $[-4, 4]$ are used in calculating J_{err}. Moreover, the average values and the standard deviations of 50 runs are recorded for J_{err} and λ^*.

Fig. 1 gives corresponding visual experimental results, including the approximation curves with $\sigma = 0.5$ and $\sigma = 0.7$, and corresponding changing curves of λ^k.

According to the changing curves of λ^k shown in Fig. 1, it is obvious that sampling learning can accurately converge to the stable solution, which complies with the theoretical results analyzed in the preceding section. By analyzing Table 1 and Fig. 1 together, it can be found that, the sampling learning can obtain actual optimal approximation when reasonable σ is adopted, which contains not only the consistent output value but also the high evaluation precision of the normalization factor. Specifically, when $\sigma = 0.5$, the error between the real value and the evaluation value obtained by the ESA is only 0.22%. Furthermore, the obtained error is only 0.054% when $\sigma = 0.3$. From Table 1 and Fig. 1, another apparent conclusion is that, if λ^* is closed to the real value, the corresponding approximation error J_{err} is also very small, and vice versa, which accords as well with the theoretical conclusions. Here, a phenomenon ought to be explained is that the obtained evaluation value λ^* with $\sigma = 0.3$ is slightly smaller than the exact value, which is seemingly not consistent with Lemma 3. The reason is that the actual λ^* is obtained with finite N_S, while the conclusion in Lemma 3 is obtained with infinite N_S. So, due to numerical computation error with finite N_S, the actual λ^* might be slightly different than the real normalization factor.

Table 1
The approximation performance on π_1 obtained by the ESA with different kernel width σ.

σ	J_{err}	λ^*
0.3	0.2259 ± 0.0513	**3.5433 ± 0.0045**
0.5	**0.1118 ± 0.0293**	3.5529 ± 0.0056
0.7	2.6864 ± 0.0039	4.0072 ± 0.0039
0.9	6.0151 ± 0.0324	4.6367 ± 0.0072
1.1	8.4591 ± 0.0418	5.1083 ± 0.0126

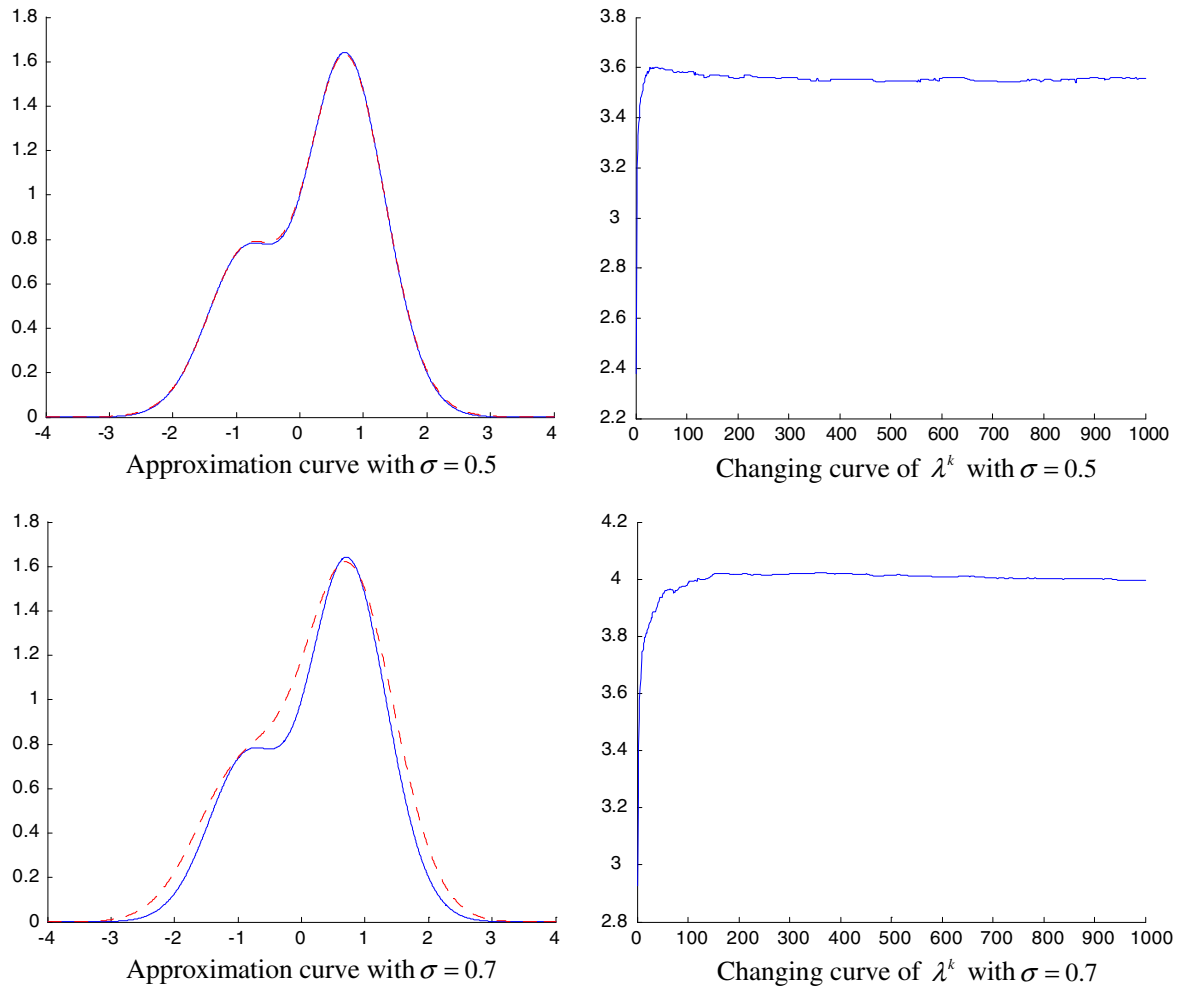

Fig. 1. The approximation results obtained by the ESA with $\sigma = 0.5$ and $\sigma = 0.7$.

A good approximation performance can be obtained when suitable σ is adopted, but bad σ will still result in poor performance. When $\sigma = 0.7, 0.9, 1.1$, the obtained λ^* becomes far from the real value, and the corresponding approximation precisions also decrease, which also accords with the conclusions of Lemma 3 and Corollary 1. So, a simple kernel parameter selection mechanism in terms of λ^* is still required for the ESA in practical applications.

Table 2 reports the experimental results on π_1 obtained by the ESA with $\sigma = 0.5$, $K_{\max} = 500$ and different N_S, where all performance indices are calculated by means of 50 runs.

From Table 2, it is obvious that increasing N_S will remarkably improve the approximation performance, while the average computing time will increase nearly linear. In addition, increasing N_S will not cause any overfitting that must be carefully dealt with in almost all traditional machine learning methods including neural networks, fuzzy systems, support vector machines, etc. Furthermore, Table 3 gives the experimental results on π_1 obtained by the ESA with $\sigma = 0.5$, $N_S = 200$ and different $K_{\max}$. Wherein, it can be found that if N_S is fixed, then increasing $K_{\max}$ will not remarkably improve the approximation performance, while the actual computing time will increase linearly.

Table 2
The different approximation performance on π_1 obtained by the ESA with different N_S.

N_S	J_{err}	λ^*	*Avg_time*
100	0.3033 ± 0.1117	3.5609 ± 0.0141	0.7065
200	0.1585 ± 0.0513	3.5556 ± 0.0084	1.2795
400	0.0971 ± 0.0356	3.5537 ± 0.0041	2.3725
600	0.0732 ± 0.0209	3.5521 ± 0.0030	3.4391
1000	**0.0538 ± 0.0136**	**3.5505 ± 0.0023**	5.5912

Table 3
Different approximation performances on π_1 obtained by the ESA with different $K_{\max}$.

$K_{\max}$	J_{err}	λ^*	*Avg_time*
200	0.2091 ± 0.0616	3.5577 ± 0.0089	0.4947
400	0.1680 ± 0.0490	3.5569 ± 0.0072	1.0113
600	0.1617 ± 0.0508	3.5558 ± 0.0087	1.5234
1000	0.1490 ± 0.0517	3.5542 ± 0.0063	2.5882
2500	0.1471 ± 0.0410	3.5525 ± 0.0080	6.7631

According to the theoretical analysis, increasing N_S or $K_{\max}$ will improve the actual approximation performance. However, the experimental results demonstrate that a larger N_S will be more effective than a larger $K_{\max}$ with limited computing capacity, the reason of which can be explained as follows. Obviously, increasing N_S will engender a larger variable scope for the SSM output and make the ESA obtain better support samples. In contrast, if N_S is fixed, increasing $K_{\max}$ cannot enlarge the variable scope of the SSM output, and better solution will not be included into the valid solution space.

In summary, we suggest that if enough computing capacity is available, we should make N_S as adequate as possible, while $K_{\max}$ can be adaptively adjusted based on the changing status of λ^k. Moreover, according to Lemma 3, whether N_S is adequate can be determined by inspecting the changing of λ^*_π with different parameters. In general, if increasing N_S does not bring a significant decrease of λ^*_π, then N_S has been adequate. In the same way, the optimal kernel function and corresponding parameters could be adaptively selected as well by means of Lemma 3 and Corollary 1.

Although theoretical results show that the optimal σ should relate to the minimal λ^*_π, however, selecting the optimal σ that relates to the right inflection point on the changing curve of λ^*_π may provide more parameters' robustness in practical applications.

Be means of the above experiments, the actual performance of the ESA with different parameter settings have been examined, including different kernel parameters like N_S and $K_{\max}$. The experimental results indicate that the conclusions are consistent with the theoretical analyses. Moreover, the theoretical analyses have also proved that different initial support samples do not influence the ultimate approximation precision, which has also been verified in the above experiments. For example, although we set initial support samples far from the region with high probability, such as the range [15,50], the ESA can still converge to an accurate solution, where only some additional evolutionary searching steps are required. Even so, if the probability values of given initial support samples with respect to the target distribution are extremely small (for example support samples are all around 250 in the above experiments), then reasonable solution will not be achieved due to the limited numerical precision of ordinary computers.

5. Applications

Derived from ideas in rejection sampling and evolutionary computation, a novel way of machine learning, the evolutionary sampling learning has been proposed, which includes the support sampling model and the so-called evolutionary sampling approach. Theoretical and experimental results demonstrated that the proposed ESA can acquire an optimal explicit approximation by minimizing the total variation distance to any density function where only point-wise computability is required. This remarkable characteristic cannot be easily achieved by current machine learning methods according to our knowledge, which creates a novel way to solve machine learning problems within a probabilistic framework.

Obviously, if a certain problem can be transformed into a probability approximation problem within the probabilistic framework, then it might be resolved by evolutionary sampling learning. Not surprisingly, many application problems can be transformed into such approximation problems. Specifically, at least the following problems faced in machine learning and statistics have the above characteristics, including mainly the Monte Carlo Integration problem, classical sampling problems, data modeling problems, etc. Next, we will start to explore how to solve these practical application problems, for which some novel algorithms should be put forward.

5.1. The Monte Carlo integration problem

As introduced in Section 2, the Monte Carlo integration problem can be expressed as the following calculation problem.

$$f_\pi = \int f(x)\pi(x)dx \tag{28}$$

where $\pi(x)$ is a probability density function and is point-wise computable, and $f(x)$ is any computable function. This problem is a basic form of many popular estimation problems in statistics and machine learning. Unfortunately, we usually cannot explicitly compute f_π, and numerical methods should be adopted to deal with it. The following computing formula, primarily put forward by von Neumann, etc., is the most popular way.

$$f_\pi = \int f(x)\pi(x)dx \approx \frac{1}{N}\sum_{i=1}^{N} f(x_i) \tag{29}$$

where $\{x_i\}$ is a sample set generated from $\pi(x)$, which reflects the intrinsic property of $\pi(x)$ from a statistical viewpoint. Obviously, the larger N the higher evaluation accuracy will be, and an adequate N is required in practical applications.

Nonetheless, to gain a good sample set $\{x_i\}$ that can really reflect $\pi(x)$ is a very difficult task. Nowadays, the rejection sampling strategies are the most valid method to obtain reliable samples for arbitrary $\pi(x)$. However, there is a substantial disadvantage for the existing rejection sampling methods as they require a huge sampling chain length to work well. That is, even if given a large value of N, all traditional rejection sampling methods could not effectively gain reasonable samples in theory, and it is difficult to evaluate the rationality of these obtained samples. In contrast, derived from its novel design, the ESA can wonderfully tackle such practical problem.

When the SSM is introduced, problem (28) can be rewritten as

$$f_\pi = \int f(x)\pi(x)dx = \frac{1}{N_S}\sum_{i=1}^{N_S}\int f(x)k(x;x_i)dx \tag{30}$$

In addition, $k(x;\theta)$ is a type of standard kernel density function, and can directly generate valid samples. This property will not only offer a better chance to get the explicit expression of f_π, but also help the algorithm achieve a better evaluation precision of f_π by means of a standard Monte-Carlo integration formula. Consequently, the ESA has noteworthy advantages for Monte-Carlo integration problems compared to traditional rejection sampling methods.

5.2. Classical sampling problem

For a classical sampling problem, we mainly try to gain N samples from any point-wise computable probability density function $\pi(x)$, such as the uniform distribution, and normal distribution. In theory, how to ensure the obtained N samples can exactly reflect the original distribution is still an open problem in most cases (except for a few standard probability distributions). As introduced before, there are three types of methods that can be used, including pseudo-random number generating methods [11,36,28], inverse transform sampling methods [11] and rejection sampling methods [50]. The first two types of methods have higher performance but poor applicability. In contrast, the rejection sampling has wide applicability but dissatisfactory performance. That is, it cannot be assured that the obtained finite samples really reflect the original distribution, and many duplicate samples will exist. It is very difficult to perfectly tackle such problem in theory due to the intrinsic disadvantages of current methods.

Nevertheless, the evolutionary sampling learning presented in this paper provides a new possibility to solve well the above problem, of which the concrete implementation can be considered as follows. For any point-wise computable density distribution $\pi(x)$, we firstly obtain an approximation $p(x)$ by means of ESA, and then auxiliary sampling procedures based on $p(x)$ are borrowed to get samples. In fact, if $p(x)$ is a good approximation of $\pi(x)$ with extremely high precision, and random data samples can be directly generated from $p(x)$, then the Metropolized independence sampler MIS [26] can be used to obtain perfect samples from $\pi(x)$. Obviously, if a Gaussian function is used as the kernel function, we can directly generate data samples from $p(x)$. In summary, the following data sampling method based on the evolutionary sampling approach is proposed below.

A data sampling algorithm using the evolutionary sampling approach

1 Given $\pi(x)$ and desired sample number N, construct an initial empty sample set $X = \{\}$, and let $k = 1$.
2 Obtain an approximation $p(x)$ to $\pi(x)$ using the ESA.
3 Generate a starting sample x^1 from $p(x)$.
4 Generate a candidate sample y^c from $p(y)$.
5 Generate a random number R from a uniform distribution on [0, 1] and, if $R \leqslant \alpha(x^k, y^c) = \min\left\{1, \frac{p(x^k)\pi(y^c)}{p(y^c)\pi(x^k)}\right\}$ is satisfied, then y^c is successfully accepted, and let $x^{k+1} = y^c$; otherwise reject it and let $x^{k+1} = x^k$.
6 Set $k = k + 1$; if $k \geqslant k_T$, then set $\mathbf{X} = \mathbf{X} \cup \{x^k\}$.
7 If cardinal number $|\mathbf{X}| < N$, then go to step 4.
8 Output N desired samples from $\mathbf{X}$.
9 END

In the above algorithm, k_T is the required number of iterations for which the corresponding Monte Carlo Markov Chain (MCMC) has been executed to approach its invariant distribution $\pi(x)$, which comes from the original rejection sampling strategy also required in MIS (Metropolized independence sampler). Though selecting a reasonable k_T is a hard task in traditional rejection sampling methods, however, if $x^1 \sim \pi(x)$ is satisfied, then $k_T = 1$ might be considered. On the other hand, because the probability of generating x^1 ($p(x)$) is approximately equivalent to $\pi(x)$, with a high precision, so parameter $k_T = 1$ also can be configured as the default setting in our new algorithm, which will bring more practicality.

Next, two following test distribution functions π_2 and π_3 are employed to analyze the actual performance of the data sampling algorithm using the ESA (DSAuESA).

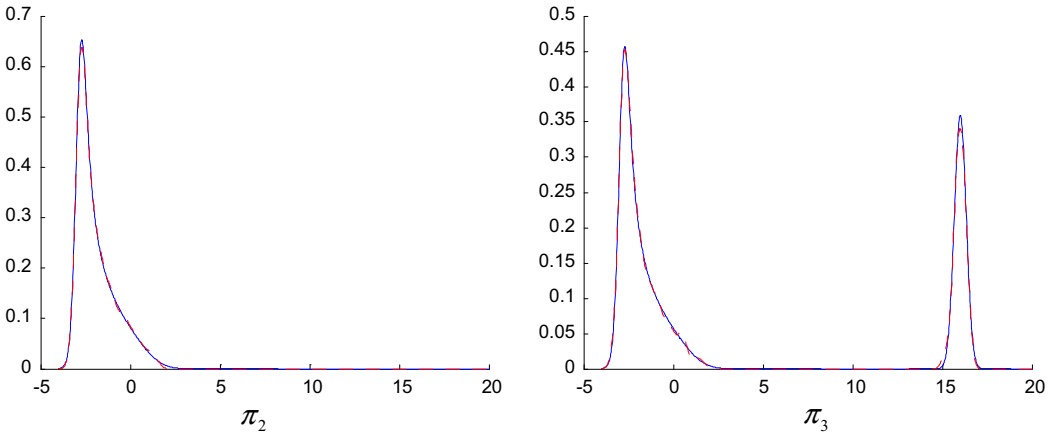

Fig. 2. The output approximation results on π_2 and π_3 obtained by the ESA with $\sigma = 0.2$.

$$\pi_2(x) = \frac{1}{8}\frac{1}{\sqrt{2\pi}}\sum_{i=0}^{7}\left(\frac{1}{\sqrt{h_i}}e^{-(x+3-3h_i)^2/2/h_i}\right), \quad h_i = \left(\frac{2}{3}\right)^i \tag{31}$$

$$\pi_3(x) = 0.7\pi_2(x) + 0.3\frac{1}{3\sqrt{2\pi}}e^{-(x-16)^2/2/9} \tag{32}$$

Obviously, we have $\int \pi_2(x)dx = 1$ and $\int \pi_3(x)dx = 1$. Besides, similar to π_1, π_2 is a unimodal function, while π_3 is a bimodal function. Here, experimental configurations are considered as follows. A Gaussian kernel function is used, $Ns = 300$, $K_{\max} = 300$, and the test data used in evaluating the error between $p(x)$ and $\pi(x)$ are generated from *Uniform* $[-4, 20]$. Fig. 2 gives two typical approximation results of $p(x)$ to $\pi(x)$, where the Gaussian kernel width is $\sigma = 0.2$.

Fig. 2 shows that the ESA can achieve good approximation results on π_2 and π_3, where original distribution curves are displayed with blue solid lines, and approximation outputs are drawn with red dotted lines. Furthermore, Table 4 lists the approximation performance obtained by the ESA on π_3 with different σ. From Table 4, it can be found that, with the increase of σ, λ^* firstly increases and then decreases, where the range 0.1–0.2 of σ relates to the smaller approximation errors, and the errors between λ^* and real values are all less than 1%.

Fig. 3 shows the sample data histograms obtained by our new algorithm on π_2 and π_3, where the number of sample data is 1000 and $\sigma = 0.2$. Clearly, the obtained histograms are consistent with the original distribution. Moreover, the average accepting ratios of candidate samples at step 5 in our algorithm are 98.6% and 97.5% for π_2 and π_3, respectively, in terms of 50 times of running results. The high average accepting ratio reflects that there are only few repetitive samples in the obtained sample set, which is very good for many practical applications.

In fact, because the SSM output can approximate not only the desired distribution $\pi(x)$ with high precision, but also $p(x)$ can be directly sampled, the good performance of the DSAuESA is predictable and provable in theory. Obviously, on top, the experimental results accord with the theoretical predictions and demonstrate the effectiveness of DSAuESA. In addition, we should point out that, for traditional sample acquisition methods, it is a trouble to deal with bimodal or multimodal distributions. Especially, in traditional reject samplers, constructing suitable proposal distributions is an open problem too. For example, when classical MIS and the Gaussian proposal distribution are used to get samples for the objective distribution π_3, our experimental studies indicate that, even if the optimal kernel width is chosen by hand, the sampling efficiency is still extremely low.

Table 4
The approximation performance obtained by the ESA on the test distribution π_3 with different kernel width σ.

σ	J_{err}	λ^*	*Avg_time*
0.05	0.6947 ± 0.1897	1.1808 ± 0.0807	0.5331
0.1	0.1723 ± 0.0231	0.9941 ± 0.0046	0.5809
0.2	**0.1022 ± 0.0192**	**0.9985 ± 0.0024**	0.6653
0.3	0.2603 ± 0.0221	1.0420 ± 0.0028	0.7388
0.4	0.5785 ± 0.0182	1.1319 ± 0.0022	0.7903
0.5	0.9213 ± 0.0145	1.2450 ± 0.0035	0.8275

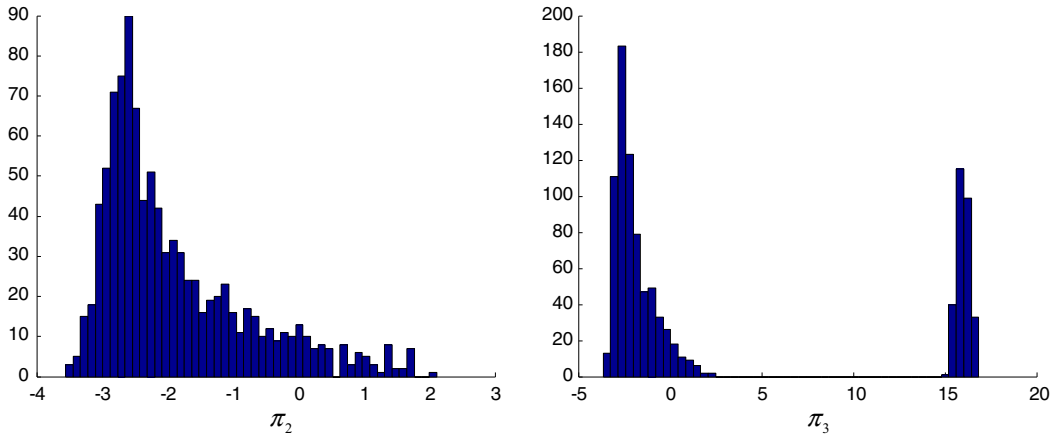

Fig. 3. Sample data histograms on π_2 and π_3 obtained by our new algorithm, where the number of samples is 1000 and $\sigma = 0.2$.

In summary, because the good proposal distribution can be taken, DSAuESA gains better (absolutely not inferior) performance than current data sampling methods. In fact, DSAuESA can have even more advantages for the cases of multimodal distributions or those cases that only small quantities of samples are desired.

5.3. Data modeling

Data analysis is a basic task in machine learning, including data density estimation, data clustering analysis, and pattern analysis, in which the focus is how to model the data. Many data modeling methods have been presented, including mainly the kernel density method [25,51], the Gaussian mixture model [1,38], the (hierarchical) Dirichlet process model [32,47,48], the Gaussian process model 39], the manifold model [41,55] and so on.

Within a probabilistic framework, for any one group of observed data X_O (O refers to "Observed" or "Objective"), there must exist a probability distribution π_O that reflects the possibility of the occurrence of each data x_i. In theory, π_O represents the intrinsic characteristics of X_O, and contains almost all information about X_O. So, reconstructing π_O is a fundamental strategy to modeling the original data, from which many other data processing algorithms might be implemented like clustering and classification.

Nonetheless, as the number of samples in X_O is finite, X_O cannot represent anything else than themselves in theory, such that it is an impossible task to know the real π_O from X_O. That is, only in terms of X_O, we could never know what is the actual thing behind X_O, and so we can only obtain the estimation π_E (or π_M in machine intelligence) of π_O by means of introducing empirical knowledge (or artificial explanations). Accordingly, in order to obtain the approximation π_M of π_O, it is necessary to impose some artificial interpretations on the observed dataset X_O.

In practice, empirical knowledge (or prior knowledge) could be introduced to achieve a good estimation π_M of π_O by using machine learning methods. The prior knowledge includes the type of model representation, the model parameters and other auxiliary conditions. For example, in kernel density estimation methods, prior knowledge refers to the type of kernel function and kernel parameters. If these artificial interpretations are removed, then the obtained results will have no actual meanings.

Among all data density estimation methods, the Parzen window method and the kernel density estimation method, as the two classical representatives, are widespread used [37,19,23], but they have difficulties in selecting a suitable kernel width [21,27,51] and in dealing with large-scale data [10,8]. It should be clarified that, although many modified kernel density estimation methods declared they can adaptively select the kernel width, they also implicitly introduce other artificial restrictions [51] according to our theoretical analysis.

As known, to obtain the approximation estimation π_M of π_O from X_O, two problems should be firstly considered. The first one is how to explicitly express π_M, and the other one is how to measure the approximation difference. Among all density estimation methods, the kernel density estimation method, the (hierarchical) Dirichlet process model and the Gaussian process model all primarily concern the first problem. Conversely, all parameter selection strategies aim to deal with the second problem. In this paper, a novel density estimation method, named as the density estimation using evolutionary sampling learning (ESLDE), is proposed here. Wherein, the support sample model is employed to express π_M, and the evolutionary sampling learning is used to train the support sample model and construct an approach of selecting the optimal kernel width.

To evaluate the approximation precision of π_M to π_O, a measuring standard must be predefined, which may not be directly formulized from π_O, in the sense that π_O cannot be explicitly expressed based on X_O. To tackle this problem, we could employ

some characteristics of π_O as the measure index, because that X_O will actually inherit some intrinsic characteristics of π_O within the statistic sense. In other words, the fact that X_O is a representative of π_O indicates that X_O will have the same characteristics as π_O, which makes them tightly connect with each other. Therefore, it will be good to obtain the optimal π_M by minimizing the difference, on these measure indices, between X_O and π_M.

In general, two types of characteristics may be concerned for any probability density π_O. The first type is its intrinsic features related to its moments (for example the 1^{st} order moment), and the second type refers to all other apparent properties. Obviously, if π_M has more similar characteristics to X_O, then π_M will approximate π_O with higher precision. In this study, an apparent characteristic is considered as the measure index, of which the concrete forms with respect to π_O and X_O are defined as follows.

$$f(x,\theta_1;\pi_O)=\int \pi_O(t)g(x,t;\theta_1)dt \tag{33}$$

$$f(x,\theta_1;X_O)=\frac{1}{N_O}\sum_{i=1}^{N_O} g(x,x_i;\theta_1) \tag{34}$$

where N_O is the number of samples in the dataset X_O, $g(x, x_i; \theta_1)$ is also a kernel function (in general, a Gaussian kernel function may be considered).

Thus, if π_M is represented by the SSM, then we have:

$$\pi_M(x,\theta_2)=\frac{1}{N_S}\sum_{i=1}^{N_S} g(x,x_i;\theta_2) \tag{35}$$

Furthermore,

$$f(x,\theta_1;\pi_M)=\int \pi_M(t,\theta_2)g(x,t;\theta_1)dt=\frac{1}{N_S}\sum_{i=1}^{N_S}\int g(t,x_i;\theta_2)g(x,t;\theta_1)dt \tag{36}$$

Again, if $g(x, x_i; \theta_1)$ is a Gaussian kernel function, then the following equation is satisfied.

$$\int g(t,x_i;\theta_2)g(x,t;\theta_1)dt=g(x,x_i;\theta_3) \tag{37}$$

where $\theta_3^2=\theta_1^2+\theta_2^2$. Moreover, $f(x, \theta_1; \pi_M)$ can be written as

$$f(x,\theta_1;\pi_M)=\int \pi_M(t,\theta_2)g(x,t;\theta_1)dt=\frac{1}{N_S}\sum_{i=1}^{N_S} g(x,x_i;\theta_3) \tag{38}$$

In summary, the following learning objective can be concluded in the ESLDE.

$$f(x,\theta_1;\pi_M)\to f(x,\theta_1;\pi_O)\approx f(x,\theta_1;X_O) \tag{39}$$

For the problem in Eq. (39), $f(x,\theta_1;X_O)$ is clearly point-wise computable with given θ_1. So, the above learning objective can be solved by means of ESA, where the optimal solution depends on the numerical approximation degree between $f(x, \theta_1; X_O)$ and $f(x, \theta_1; \pi_O)$ regardless of the parameters of the SSM.

Nonetheless, if the ESA is directly used to solve the above problem, the required computational complexity will be relatively large. Although the space complexity only requires $O(N_O + N_S)$, the time complexity will reach $O(LN_S(1 + N_O \log N_S))$. To avoid this high computational burden, a variant of the ESA is proposed to better tackle the problem (39). In the new algorithm, only one sample is processed at every iteration, so that the learning procedure can be accomplished in an online manner (as opposed to batch learning). The detailed procedure is described as follows.

Table 5
The estimation results on $\pi_3(x)$ obtained by the ESLDE and SKDE, respectively, where $\alpha = 20$.

θ_1	J_{err}^{ESLDE}	J_{err}^{SKDE}	λ^*	*Avg_time*
0.1	0.3613 ± 0.0859	0.2229 ± 0.0633	1.0095 ± 0.0016	0.2546
0.12	0.3737 ± 0.1456	**0.1834 ± 0.0267**	1.0096 ± 0.0007	0.2854
0.14	0.3353 ± 0.1501	0.2018 ± 0.0273	1.0097 ± 0.0004	0.3595
0.16	0.1844 ± 0.0374	0.2377 ± 0.0299	**1.0106 ± 0.0021**	0.6444
0.18	**0.1723 ± 0.0403**	0.2861 ± 0.0249	1.0157 ± 0.0050	1.0222
0.20	0.2203 ± 0.0324	0.3373 ± 0.0205	1.0304 ± 0.0041	1.0516
0.22	0.3577 ± 0.0322	0.3994 ± 0.0241	1.0528 ± 0.0055	1.0528

Density estimation using evolutionary sampling learning – ESLDE

1 Given dataset X_O, configure kernel parameters θ_1, θ_2, and $\theta_3 = \sqrt{\theta_1^2 + \theta_2^2}$.
2 Configure the value of N_S, generate N_S sample data to construct initial support sample set X_S of SSM by randomly selecting from original dataset X_O, and let $k = 1$.
3 Randomly select a candidate support sample y^c from X_O with the same probability value.
4 For each x_i in X_S, update $f(x_i, \theta_1; X_O)$ using the online estimation strategy.
5 Let $j = \max_i\{p(x_i) - f(x_i, \theta_1; X_O)\}$, and use greedy strategy to choose the jth support sample as candidate updating sample at this iteration.
6 Get a random number $R \sim Uniform[0, 1]$; if $R \leqslant \alpha(x_j, y^c) = \max\left\{1 - \frac{f(x_j, \theta_1; X_O)}{p(x_j)}, 0\right\}$, then receive the new candidate support sample successfully, otherwise reject it.
7 If new candidate support sample is successfully received, then update the jth support sample $x_j = y^c$ and $p(x_i)x_i = 1, 2, \ldots, N_S$, and then let $f(x_i, \theta_1; X_O) = p(x_i)$.
8 If $\lambda_{X_O}^k < \varepsilon$ or $k \geqslant k_{\max}$, then go to the next step; otherwise let $k = k + 1$ and go to step 4.
9 Output the density estimation $\pi_M(x, \theta_2)$ of $\pi_O(x)$ according to the obtained support samples and formula (35).
10 END

For step 4 in the above algorithm, the following strategy is designed to update $f(x_i, \theta_1; X_O)$. Firstly, we compute $f(x_i, \theta_1; y^c)$ for all i, then the new values of $f(x_i, \theta_1; X_O)$ for all i are recomputed by the weighted average between previous $f(x_i, \theta_1; X_O)$ and $f(x_i, \theta_1; y^c)$ (weighted ratio might be considered as $0.3 \times k$:1). In the ESLDE, ε is a threshold used to terminate the learning procedure. According to theoretical results, there is $\lambda_{X_O}^k \geqslant 1$, and a smaller value will produce better results. Therefore, with respect to ε, a value with a slightly larger than 1 may be configured. From our preliminary studies, $\varepsilon = 1.01$ is a good value for practical applications.

Next, the effectiveness of the ESLDE is explained. First, for $f(x_i, \theta_1; X_O)$, with the progress of evolutionary sampling, more and more (even repeated) original sample data are used to estimate $f(x_i, \theta_1; X_O)$, and the final estimation result will converge to the exact result calculated by formula (34). Thus, to prove that the learning procedure of ESLDE is effective, we only need to prove that the output of the obtained SSM, $p(x_i) = f(x, \theta_1; \pi_M)$, will gradually approximate $f(x_i, \theta_1; X_O)$. Compared with the standard ESA, the following suppositions will hold in the ESLDE.

(a) $A^k(t, x) = \pi(x)$ and $\pi(t)A^k(t, x) = \pi(x)A^k(x, t)$ holds.
(b) for every chosen updating support sample z, there is $\frac{\pi(z)}{p(z)} = 1$.
(c) the candidate updating support sample (a certain support sample in the current SSM) is greedily selected based on $j = \max_i\{p(x_i) - \pi(x_i)\}$.

Based on the above three preconditions, the sampling procedure of the ESLDE is the same as the standard ESA, so the ESLDE still holds similar convergence and approximation performance; that is, $p(x_i) = f(x, \theta_1; \pi_M)$ will gradually approximate $f(x_i, \theta_1; X_O)$.

In the ESLDE, the selection of Kernel width θ_1 and θ_2 should be pre-considered for practical applications. Obviously, we should select optimal θ_1 to make $f(x, \theta_1; X_O) \to f(x, \theta_1; \pi_O)$ with more precision, and optimal θ_2 to make $\pi_M(x, \theta_2)$ completely represent $\pi_O(x)$ when $N_S \to \infty$. In practice, $\theta_2 = \theta_1$ might be considered, in a similar way to the density estimation algorithm proposed by Klemela [23]. Further studies indicate that θ_2 may be slightly larger than θ_1, and $\theta_2 = 1.618\theta_1$ is a better configuration from our experimental comparisons. Thus, only one parameter should be set. Similar to the ESA, we can conclude that lower λ^* will relate to better θ_2. In addition, when the robustness of the actual computation with finite support samples is considered, taking the right inflection point in the changing curve of θ_2 as the optimal θ_2^* to λ^* is still a good selection.

Next, $\pi_3(x)$ is also employed as the test probability density function to examine the estimation performance and the parameter influence of the ESLDE. In the following experimental results, we set $k_{\max} = 10{,}000$, $N_S = 300$, and let the number of original data $N = \alpha N_S$, where α is a ratio factor.

Table 5 reports a group of experimental results on $\pi_3(x)$, where $\alpha = 20$ is fixed and θ_1 is variable. Besides, the standard kernel density estimation (SKDE) method is also used for comparison, in which the Gaussian kernel function is used as well.

In Table 5, J_{err}^{ESLDE} denotes the estimation error between the result obtained by the ESLDE and the original probability density function $\pi_3(x)$; likewise J_{err}^{SKDE} denotes the estimation error between $\pi_3(x)$ and the result obtained by the standard kernel density estimation method, where the error calculation is the same as definition (27), and the kernel width used in the SKDE equals to θ_1. *Avg_time* records the average computational time of the ESLDE in 50 runs. Table 5 illustrates that different θ_1 will result in different λ^*, especially the changing degree of λ^* between $\theta_1 = 0.18$ and $\theta_1 = 0.2$ increases rapidly. According to the above discussion, $\theta_1 = 0.16$ or $\theta_1 = 0.18$ may be considered as the optimal kernel width. Consequently, the estimation error J_{err}^{ESLDE} is also very small. Compared to SKDE, the ESLDE obtains better estimation results than the SKDE with their respective optimal settings for the kernel width. Besides, it is significant that the optimal kernel width in the ESLDE can be

Table 6
Estimation results on $\pi_3(x)$ obtained by the ESLDE and SKDE, respectively, with different α, and their respective kernel widths.

α	J_{err}^{SLDE}	J_{err}^{SKDE}	λ^*	*Avg_time*
3	0.2977 ± 0.0523	0.3300 ± 0.0610	1.0211 ± 0.0052	1.0612
5	0.2739 ± 0.0559	0.2725 ± 0.0546	1.0162 ± 0.0066	0.8886
10	0.2292 ± 0.0417	0.2130 ± 0.0373	1.0116 ± 0.0032	0.8285
20	0.1823 ± 0.0465	0.1889 ± 0.0334	1.0096 ± 0.0005	0.6147
30	**0.1822** ± **0.0713**	**0.1812** ± **0.0311**	**1.0099** ± **0.0002**	0.7485

adaptively selected through a theoretically trustworthy manner, which is different from traditional kernel density estimation methods.

To further investigate the estimation performance of the ESLDE with different α, Table 6 lists a group of experimental results obtained by the ESLDE and the SKDE with different α, where $\theta_1 = 0.16$ for ESLDE (by adaptive setting) and $\theta = 0.12$ (by hand) for SKDE.

From Table 6, it can be found that the larger number of original data will result in higher estimation precision for both the ESLDE and the SKDE. Although almost equivalent results can be obtained for two estimation methods with their respective kernel width settings, how to select the optimal kernel width is very difficult in the SKDE, as no theoretical foundation can be used. In contrast, the kernel width can be adaptively selected by reliable rules in the ESLDE.

In addition, if $\theta_1 = 0.18$ is adopted in the ESLDE, the estimation error will reach $J_{err}^{ESLDE} = 0.1450 \pm 0.0304$ when $\alpha = 30$, which is greatly superior to the minimal J_{err}^{SKDE}. On the whole, the actual estimation performance of the ESLDE is superior to that of the standard kernel estimation method.

Next, the computational complexity of the ESLDE will be analyzed. Clearly, its space complexity is $O(N_S + 1)$, independent of the number of original data N, but the space complexity of the standard kernel density estimation method is $O(N)$. In general, we have $N \geqslant N_S$, so the space complexity of the ESLDE is very low. Because the pre-learning process is required for the ESLDE, its time complexity contains two parts: one that concerns the learning process, and the other one the computing output. From simple analysis, the former is $O(L \times N_S)$, and the later is $O(N_S)$, where L is the total number of iterations executed in the whole sampling learning procedure. Comparatively speaking, the time complexity of the SKDE is $O(N)$, where no learning process is required. The above considerations demonstrate that, with some additional expense in learning time, the ESLDE achieves several novel and significant algorithm performance improvements, such as higher practical efficiency and lower space complexity, strongly concerned in modern massive-data processing methods [8]. Furthermore, the ESLDE can accomplish the leaning process in an online manner, when the original data can be continuously dealt with one by one (in sequence), which is very useful for online large-scale data processing tasks.

In addition, it can be proven that, the approximation objective of the ESLDE is equivalent to a weighted kernel density estimation method [51] with a particular kernel width. However, due to the different learning approach, the ESLDE possesses more flexibility and better performance in selecting the kernel width and in dealing with online data, by means of low computational complexity and excellent solving procedure (i.e., the ESLDE does not require solving a linear or quadratic programming problem).

In summary, the ESLDE is very suitable for online data analysis, which is the most significant merit of the ESLDE compared to all existing density estimation methods. More than this, the sampling learning may be exploited more and it has good potential in data analysis and processing by combining current clustering and classification strategies within a probabilistic framework. Such studies will be carried out in our future works.

5.4. Other discussions

In the above subsections, three types of application problems that can be solved by evolutionary sampling learning have been studied, including the Monte Carlo integration problem, traditional random number generating problems, data modeling problems. On the other hand, the theoretical studies show that the evolutionary sampling learning might also be valid for other many machine learning problems, such as the evolutionary computation, particle filters, and probabilistic neural networks. Next, some qualitative analysis will be discussed.

Evolutionary computation, as an important method in computational intelligence, has been successfully applied to many problems. In our previous studies, Sun et al. proposed a novel particle swarm optimization algorithm named as the quantum behaved particle swarm optimization (QPSO) [44,45,15,43], in which a good optimization performance has been demonstrated by many experimental results on practical applications. In evolutionary sampling learning, a similar searching strategy is used to find the most possible candidate support sample. In theory, a better searching strategy of generating candidate support samples will produce a better global converge performance. As for the evolutionary optimization, its main objective is to rapidly find the optimal solution. When the evolutionary sampling strategy is used to search the optimal solution, the SSM may be employed to record last optimal solutions that have been found. Thus, this new designed evolutionary optimization algorithm based on the evolutionary sampling strategy can be viewed as the combination of the QPSO and the simulated annealing optimization. Our experimental results display that, although the new algorithm based on the evolutionary

sampling strategy cannot gain more prominent advantage than the QPSO on some classical test searching functions, however, deserved from the integrated merits of the QPSO and simulated annealing, the new algorithm might be more suitable than the QPSO and traditional simulated annealing method in some particular optimization problems.

Particle filter [14,9,40,49] is a very effective tool for solving (or simulating) sophisticated system identification problems, like the object tracking problem in machine vision. In particle filter, a solution (described by the probability distribution of different possible positions) is represented by a group of particles, and these particles evolve with time to instantaneously reflect the new system state in real time. In general, resampling is used as a tool to adjust the number of different particles with different positions. Wherein, the core of particle filter is to rapidly obtain a new particle set that can better represent the current problem solution, that is, the probability distribution of all possible positions. However, resampling methods can only select new particles from those anterior particles, but not from all possible positions, which is also the meaning of resampling. Thus, it is difficult to obtain an optimal particle set related to the real problem solution. In contrast, if evolutionary sampling strategy is used to evolve particles, then all possible positions are reachable for new particles, which will help achieve the optimal particles continuously.

For other possible applications, it has been analyzed that, if a problem can be represented or transformed into a density function approximation problem within a probabilistic framework, then this problem may be solved by means of the evolutionary sampling strategy. Wherein, many novel properties owned by the evolutionary sampling strategy will also be inherited by the new designed methods. We deem that the above novel proposals and conclusions will help us come up with many original and effective machine learning methods and will drive us to carry out more explorations on evolutionary sampling learning.

6. Conclusions

Motivated by ideas in evolutionary computation, rejection sampling and function approximation, a novel machine learning strategy, the evolutionary sampling learning was put forward in this paper, which can obtain an approximation expression to any point-wise computable probability density function. Based on the above strategy, a new machine model – the support sample model (SSM) – and a novel solving method – the evolutionary sampling approach (ESA) – were proposed, respectively. Our theoretical and experimental studies have demonstrated that the evolutionary sampling leaning can be used to solve many practical application problems which can be expressed as density function approximation problems within a probabilistic framework. ***This brilliant property expands the application scope of machine learning, has visible theoretical and practical significance and draws forth a new thought on the way of performing machine learning.*** More specifically, compared with existing machine learning methods, the evolutionary sampling learning owns the following important characteristics:

1. Due to the introduction of the concept of support samples, a novel method was developed to express sampling problems. Thus, the evolutionary sampling learning not only inherits the good convergence and the learning ability of rejection sampling, but also gains more widespread applicability.
2. Benefiting from the combined merits of evolutionary computation and rejection sampling, the evolutionary sampling learning can stably converge to the optimal solution without any other limitations like specific initial conditions or different learning parameters. Clearly, for many machine learning methods, different initial conditions or different learning parameters may influence not only the learning efficiency but the accuracy too.
3. Derived from its intrinsic merits, the evolutionary sampling learning can effectively determine when its evolutionary learning should be terminated, in terms of the evaluated normalized factor λ^k, which is an impossible task in the traditional rejection sampling framework.
4. The evolutionary sampling learning can also be viewed as a new development of evolutionary computation with application to machine learning. Traditionally, evolutionary computation is only used to solve optimization problems raised by machine learning methods. On the contrary, in evolutionary sampling learning, learning problems are directly described as probabilistic problems. Moreover, the particular evolutionary sampling approach can be used to obtain the optimal objective solution, where the evolutionary mechanism is deeply coupled into the problem objective. More specifically, the evolutionary sampling learning is directly guided by the local difference between the actual output and the desired output related to every position (x), but not by the total difference between the actual output and the desired value that is empirically considered in existing machine learning methods.
5. The space complexity of evolutionary sampling learning is very low, and it only depends on the number of the support sampling models and it does not totally relate to the number of learning data. Moreover, the evolutionary sampling learning can also accomplish an online learning task. The above two properties make ESL more practical for the online real-time processing of large-scale data, whose ability would be strongly aspired by many current researches.

In summary, we proposed a novel way of realizing machine learning within a probabilistic framework, called the evolutionary sampling strategy, where some very worthy merits, concluded as above, were demonstrated by theoretical analysis and experimental results. In addition, by developing more theoretical extensions, some troubled machine learning problems

could be solved as well by the means of evolutionary sampling strategy, which represent our motivation to continue to further explore this field in our near future, with further applications and theoretical studies on the ESA.

Acknowledgements

This work is supported by Jiangsu Province Natural Science Foundation of China (Project Number: BK20130161 and BK2010143), by National Natural Science Foundation of China (Project Numbers 61170119, 61105128, 61272210, 61373055), by the Program for New Century Excellent Talents in University (Project Number: NCET-11-0660), and by the RS-NSFC International Exchange Programme (Project Number: 61311130141).

References

[1] J. Bilmes, A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian Mixture and Hidden Markov models, Int. Comput. Sci. Inst. 4 (1998) 1–13.
[2] P. Brasnett, L. Mihaylova, et al., Sequential Monte Carlo tracking by fusing multiple cues in video sequences, Image Vis. Comput. 25 (8) (2007) 1217–1227.
[3] F. Camci, R.B. Chinnam, General support vector representation machine for one-class classification of non-stationary classes, Pattern Recogn. 41 (10) (2008) 3021–3034.
[4] E.J. Candes, M.B. Wakln, An introduction to compressive sampling, IEEE Signal Process. Mag. (2008) 21–30.
[5] O. Cappe, S.J. Godsill, et al., An overview of existing methods and recent advances in sequential Monte Carlo, Proc. IEEE 95 (5) (2007) 899–924.
[6] F.P. Casey, J.J. Waterfall, et al., Variational method for estimating the rate of convergence of Markov-chain Monte Carlo algorithms, Phys. Rev. E 78 (4) (2008) (046704. 1–12).
[7] J.-F. Chang, S.-c. Chu, et al., A parallel particle swarm optimization algorithm with communication strategies, J. Inf. Sci. Eng. 21 (4) (2005) 809–818.
[8] F.-L. Chung, Z. Deng, et al., From minimum enclosing ball to fast fuzzy inference system training on large datasets, IEEE Trans. Fuzzy Syst. 17 (1) (2009) 173–184.
[9] J. Czyz, B. Ristic, et al., A particle filter for joint detection and tracking of color objects, Image Vis. Comput. 25 (8) (2007) 1271–1281.
[10] Z. Deng, F.-L. Chung, et al., FRSDE: fast reduced set density estimator using minimal enclosing ball approximation, Pattern Recogn. 41 (4) (2008) 1363–1372.
[11] L. Devroye, Non-Uniform Random Variate Generation, Springer-Verlag, New York, 1986.
[12] R. Douc, E. Moulines, Limit theorems for weighted samples with applications to sequential Monte Carlo methods, Ann. Stat. 36 (5) (2008) 2344–2376.
[13] A. Doucet, M. Briers, et al., Efficient block sampling strategies for sequential Monte Carlo methods, J. Comput. Graphical Statist. 15 (3) (2006) 693–711.
[14] G. Fan, V. Venkataraman, et al., A comparative study of boosted and adaptive particle filters for affine-invariant target detection and tracking, CVPRW 06 (2006) 138–139.
[15] W. Fang, J. Sun, et al., Convergence analysis of quantum-behaved particle swarm optimization algorithm and study on its control parameter, Acta Phys. Sinica 59 (6) (2010) 3686–3694.
[16] H.-M. Feng, J.H. Horng, et al., Bacterial foraging particle swarm optimization algorithm based fuzzy-VQ compression systems, J. Inf. Hiding Multimedia Signal Process. 3 (3) (2012) 2073–4212.
[17] J.F.G.d. Freitas, M. Niranjan, et al., Sequential Monte Carlo methods to train neural network models, Neural Comput. 12 (4) (2000) 955–993.
[18] N.d. Freitas, P. Hojen-Sorensen, et al., Variational MCMC, in: Proceedings of the 17th Conference in Uncertainty in Artificial Intelligence, Morgan Kaufmann Publishers Inc., 2001, pp. 120–127.
[19] M. Girolami, C. He, Probability density estimation from optimally condensed data samples, IEEE Trans. Pattern Anal. Mach. Intell. 25 (10) (2003) 1253–1264.
[20] M. Hauschild, M. Pelikan, An introduction and survey of estimation of distribution algorithms, Swarm Evol. Comput. 1 (3) (2011) 111–128.
[21] M. Jones, J. Marron, et al., A brief survey of bandwidth selection for density estimation, J. Am. Stat. Assoc. 91 (433) (1996).
[22] M. Klaas, D. Lang, et al., Fast maximum a posteriori inference in Monte Carlo state spaces, Artif. Intell. Statist. (2005).
[23] J. Klemela, Density estimation with stagewise optimization of the empirical risk, Mach. Learn. 67 (3) (2007) 169–195.
[24] C.-S. Leung, J.P.-F. Sum, A fault-tolerant regularizer for RBF Networks, IEEE Trans. Neural Networks 19 (3) (2008) 493–507.
[25] F. Liang, K. Mao, et al., Nonparametric Bayesian Kernel Models, ISDS Discussion Paper, Duke University, 2006.
[26] J.S. Liu, Monte Carlo Strategies in Scientific Computing, Springer, 2001.
[27] C.R. Loader, Bandwidth selection: classical or plug-in?, The Annals of Statistics 27 (2) (1999) 415–438
[28] G. Marsaglia, Random number generators, J. Modern Appl. Stat. Methods 2 (1) (2003) 2–13.
[29] E. Mininno, F. Neri, et al., Compact differential evolution, IEEE Trans. Evol. Comput. 15 (1) (2011) 32–54.
[30] P.D. Moral, A. Doucet, et al., Sequential Monte Carlo samplers, J. R. Stat. Soc.: Ser. B (Stat. Methodol.) 68 (3) (2006) 411–436.
[31] L. Murray, Advances in Markov chain Monte Carlo methods, University of Cambridge, London, UK. 2007, phd.:176.
[32] R. Neal, Markov chain sampling methods for Dirichlet process mixture models, J. Comput. Graphical Statist. 9 (2) (2000) 249–265.
[33] F. Neri, G. Iacca, et al., Disturbed exploitation compact differential evolution for limited memory optimization problems, Inf. Sci. 181 (2011) 2469–2487.
[34] F. Neri, E. Mininno, Memetic Compact differential evolution for cartesian robot control, IEEE Comput. Intell. Mag. (2010) 54–65.
[35] F. Neri, E. Mininno, et al., Compact particle swarm optimization, Inf. Sci. 239 (2013) 96–121.
[36] S.K. Park, K.W. Miller, Random number generators: good ones are hard to find, Commun. ACM 31 (10) (1988) 1192–1201.
[37] E. Parzen, On estimation of a probability density function and mode, Ann. Math. Stat. 33 (3) (1962) 1065–1076.
[38] K. Pyun, J. Lim, et al., Image segmentation using hidden Markov Gauss mixture models, IEEE Trans. Image Process. 16 (7) (2007) 1902–1911.
[39] C.E. Rasmussen, C.K.I. Williams, Gaussian Processes in Machine Learning, MIT Press, Cambridge, Massachusetts, 2006.
[40] S. Särkkä, A. Vehtari, et al., Rao-Blackwellized particle filter for multiple target tracking, Inform. Fusion 8 (1) (2007) 2–15.
[41] H.S. Seung, D.D. Lee, The manifold ways of perception, Science 290 (5500) (2000) 2268–2269.
[42] S.A. Sisson, Y. Fan, et al., Sequential Monte Carlo without likelihoods, Proc. Nat. Acad. Sci. 104 (2007) 1760–1765.
[43] J. Sun, W. Fang, et al., QoS multicast routing using a quantum-behaved particle swarm optimization algorithm, Eng. Appl. Artif. Intell. 24 (1) (2011) 123–131.
[44] J. Sun, B. Feng, et al., Particle swarm optimization with particles having quantum behavior, in: Congress on Evolutionary Computation, 2004, CEC2004, vol. 1, 2004, pp. 325–331.
[45] J. Sun, W. Xu, et al., A global search strategy of quantum-behaved particle swarm optimization, in: 2004 IEEE Conf. on Cybernetics and Intelligent Systems, 2004, pp. 111–116.
[46] J. Sun, Q. Zhang, et al., DE/EDA: a new evolutionary algorithm for global optimization, Inf. Sci. 169 (2005) 249–262.
[47] Y.W. Teh, M.I. Jordan, et al., Hierarchical Dirichlet processes, Adv. Neural Inf. Proc. Syst. 17 (2004).
[48] Y.W. Teh, M.I. Jordan, et al., Hierarchical Dirichlet processes, J. Am. Stat. Assoc. 101 (476) (2006) 1566–1581.

[49] N. Vaswani, Particle filtering for large-dimensional state spaces with multimodal observation likelihooks, IEEE Trans. Signal Process. 56 (10) (2008) 4583–4597.
[50] J. von Neumann, Various techniques used in connection with random digits, Nat. Sureau Stand. Appl. Math. Ser. 12 (1951) 36–38.
[51] B. Wang, X. Wang, Bandwidth Selection for Weighted Kernel Density Estimation, 2007, arXiv/0709.1616.
[52] S. Wang, J. Zhu, et al., Theoretically optimal parameter choices for support vector regression machines with noisy input, Soft. Comput. 2005 (9) (2004) 732–741.
[53] C. Yang, R. Duraiswami, et al., Efficient kernel machines using the improved fast Gauss transform, Adv. Neural Inf. Process. Syst. 17 (2005) 1561–1568.
[54] C. Yang, R. Duraiswami, et al., Improved fast gauss transform and efficient kernel density estimation, in: Proceedings. Ninth IEEE International Conference on Computer Vision, 2003, 2003, pp. 664–671.
[55] Q. Zhang, R. Souvenir, et al., On manifold structure of cardiac MRI data: application to segmentation, in: Proceedings of the 2006 IEEE computer society conference on computer vision and pattern recognition, vol. 1, 2006, pp. 1092–1098.