Available online at www.sciencedirect.com
ScienceDirect
Information Sciences 177 (2007) 3110–3128
INFORMATION SCIENCES
AN INTERNATIONAL JOURNAL
www.elsevier.com/locate/ins

Adaptive estimated maximum-entropy distribution model

Ling Tan, David Taniar *

Clayton School of Information Technology, Monash University, Clayton, Vic. 3800, Australia

Received 7 August 2005; received in revised form 6 January 2007; accepted 21 January 2007

Abstract

The *Estimation of Distribution Algorithm* (EDA) model is an optimization procedure through learning and sampling a conditional probabilistic function. The use of conditional density function permits multivariate dependency modelling, which is not captured in a population-based representation, like the classical Genetic Algorithms. The Gaussian model is a simple and widely used model for density estimation. However, an assumption of normality is not realistic for many real-life problems. Alternatively, the maximum-entropy model can be used, which makes no assumption of a normal distribution. One disadvantage of the maximum-entropy model is the learning cost of its parameters. This paper proposes an *Adaptive Estimated Maximum-Entropy Distribution* (Adaptive MEED) model, which aims to reduce learning complexity of building a model. Adaptive MEED exploits the fact that samples have a low average fitness in the early stage, but they gradually converge to an optima towards the end of the search. Hence, it is not necessary to inference the model with a full account of observed constraints in the early stage of the search. The proposed model attempts to estimate the density function with a dynamic set of samples and active constraints. In addition, the proposed model includes a global sampling function to address the issue of a missing mutation operator. The ergodic convergence properties of the proposed model are discussed with the Markov Chain analysis. The preliminary experimental evaluation shows that the proposed model performs well against genetic algorithms on several clustering problems.
© 2007 Elsevier Inc. All rights reserved.

Keywords: Estimation of Distribution Algorithms (EDA); Genetic Algorithms (GA); Adaptive method; Global convergence; Clustering; Data mining

1. Introduction

Classical *Genetic Algorithms* (GAs) are well known global optimization techniques. GAs encode candidate solutions in string-based data structures, and they search for globally optimal solutions with the help of three important operators: the selection operator, which chooses good candidate solutions from the randomly generated population; the crossover, which recombines these good solutions; and the mutation operator, which probabilistically adds new diversity to the recombined solutions.

* Corresponding author. Tel.: +61 3 99059693; fax: +61 3 99055159.
E-mail address: David.Taniar@infotech.monash.edu.au (D. Taniar).

0020-0255/$ - see front matter © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.ins.2007.01.029

The success of the GA's mechanism is illustrated in Goldberg's building-block theory [17]. The theory explains that the ability to recombine partial solutions in individual data structures helps GA to locate optimal solutions. In order for such a mechanism to work efficiently, as noticed by Mühlenbein and Paaß [35] and Inza et al. [24], one critical condition is that partial solutions are explicitly encoded with their whereabouts in the data structures. When the condition does not hold, recombination and mutation often break those partial solutions of long chain or wide gap, which subsequently leads to slow convergence or sub-optimal solutions.

To circumvent the problems of implicit coding in GAs, Mühlenbein and Paaß [35] proposed the *Estimation of Distribution Algorithm* (EDA). EDA does not make any assumption about the location of partial solutions. The partial solutions are explicitly coded and kept by means of the probabilistic distribution function. For example, a pairwise conditional probabilistic function represents the second-order partial solutions. In general, EDA may use a multivariate conditional density function to capture arbitrary interactions of variables (or partial solutions) in the solution landscape.

In essence, EDA needs to estimate the density function from the underlying population. The problem of density estimation has been well studied in statistics. There are basically two approaches: (i) *parametric density estimation*, and (ii) *non-parametric density estimation*. The first approach assumes that the form of distribution is known, and the problem is simplified by learning the parameters from one of the known density functions. The second approach does not make any assumption about the structure of the function but, in general, it has higher computational requirements.

This paper adopts the parametric approach in order to reduce learning complexity. The major problem of this approach is to decide which form of distributions should be used. One popular and widely used distribution is the *Gaussian* distribution. However, the assumption of normality is not realistic for many real-life problems. Alternatively, the *Maximum-Entropy* model could be used which makes no assumption about a normal distribution.

This paper applies the maximum-entropy model as a statistical learning method to estimate the density function in EDAs. The maximum-entropy-based EDA has an advantage in dealing with uncertainty. With incomplete information presented in the system, a maximum-entropy distribution is the least biased distribution subject to given constraints [27]. These constraints can be frequency counts from data, or other information known a priori. Prior information from data is not necessary. This feature uniquely distinguishes the maximum-entropy method from the maximum likelihood method, where the latter models only on the available data.

To estimate the maximum-entropy density function, a set of *Lagrangian* parameters needs to be solved with an iterative procedure. In general, a maximum-entropy learning procedure exhibits intractable computation when the number of constraints and the dimension of training data increase. Two major approaches are used to alleviate this problem. One method is to improve optimization function [13,19], and the second method is to use approximation methods to reduce the number of constraints [18,48].

This paper extends the existing approximation techniques and proposes an *Adaptive Maximum-Entropy Estimated Distribution* approach (Adaptive MEED) to effectively reduce the constraint dimension in model learning. The potential dimension reduction in problem size is much larger than that of existing one-dimensional approaches. The proposed adaptive approach selects constraints based on the relative fitness of an n-ordered parameter, and the number of constraints is dynamically determined by the population fitness. In this way, non-essential constraints are discarded when constructing a maximum-entropy model.

In addition, the proposed model includes a global sampling function to address the issue of a missing mutation operator in the EDAs. By removing the mutation operator, the conventional EDA algorithms such as [8,32,35] do not have a global randomization function. With only sampling and selection operators, EDA is a local search method by definition, and its search space is confined to the space of the original probabilistic function. Therefore, EDA algorithms may not converge to global optima in theory. The proposed Adaptive MEED model rectifies this problem by including a global sampling function. This function operates on the entire search space and is independent of the current population. In contrast, the classic mutation operator operates on the current population and generates new individuals by flipping a small number of bits in the parent strings. Due to the low probability of mutation, the new samples are highly dependent on the parent samples. For example, in a traveling salesman problem [22], the global sampling function would generate traveling routes from all possible routes with equal probabilities; whereas the mutation operator would generate routes resembling their parent routes with a high proportion.

This paper discusses the ergodic convergence properties of the proposed adaptive model via Markov chain analysis. It proves that the Adaptive MEED model can be modelled with an ergodic Markov chain. Based on this theoretical foundation, the proposed model can be applied to classification and clustering problems in data mining. Finally, preliminary experiments were conducted to demonstrate the potentials of the proposed model in several clustering problems.

The rest of this paper is organized as follows. Section 2 describes the related work. Section 3 presents a detailed description of the estimated maximum-entropy distribution model. Section 4 presents the proposed adaptive estimated maximum-entropy distribution model. Section 5 provides a Markov chain analysis of the proposed model. Section 6 illustrates the applications of the proposed model in data mining. Section 7 presents the empirical performance study of the proposed model to clustering problems. Finally, Section 8 presents the conclusion.

2. Related work

Many approaches have been proposed in the framework of EDAs. They are similar in the way that a probability distribution is estimated from the selected individuals. According to the complexity of variable dependency, they can be categorized into three groups: (i) *univariate dependency* [6,21,32,35]; (ii) *bivariate dependency* [8,12,32]; and (iii) *multivariate dependency* [8,20,30–32,34,40]. The EDA methods can further be divided into discrete variable techniques and continuous variable techniques.

In univariate EDAs, one variable is independent of all others, so no interaction is modelled among the variables. In this case, the joint estimated density function is simplified as the product of the independent univariate distribution functions. Let the variables be denoted as $X = (x_1, \ldots, x_n)$, and the selected population at the $(k-1)$th generation be D^S_{k-1}. The estimated density function without dependency at the kth generation can be written as $\hat{p}_k(X) = \prod_i p(x_i|D^S_{k-1})$.

In the domain of discrete variable, Mühlenbein and Paaß [35] proposed *Univariate Marginal Distribution Algorithm* (UMDA). Each univariate marginal distribution is estimated from marginal frequency, i.e. $p(x_i) = n_i/N$, where n_i is the number of selected individuals having value of x_i, and N is the total number of selected individuals. UMDA is proven to be theoretically equivalent to the classic GA with uniform crossover and proportional selection. In addition, Baluja [6] proposed the *Population-Based Incremental Learning* (PBIL). PBIL estimates the current probability function by taking into account the previous probability function. For this purpose, a linear function is employed to update the marginal probability distribution, and each univariate marginal distribution is estimated in the following way: $p_k(x_i) = \alpha \cdot p_{k-1}(x_i) + (1-\alpha) \cdot n_i/N$, where $\alpha = [0, 1]$. In the case of $\alpha = 0$, PBIL is equivalent to UMDA. Similarly, *Compact Genetic Algorithm* (CGA) proposed by Harik et al. [21] also considers the previous probability function in estimating the current univariate function. CGA uses a Bernoulli distribution to initialize the population. An evaluation scheme is used to compare the marginal values of two individuals, and the best values are used to update the marginal probability function. The univariate marginal distribution function of CGA has the following form: $p_k(x_i) = p_{k-1}(x_i) \pm 1/k$, where k is a constant. The positive sign is used when the corresponding bit position is "1", and the negative sign is used when the bit is "0". One of the main differences between CGA and PBIL is that CGA requires less memory than PBIL, because only two individuals are sampled at any generation in CGA.

In the domain of continuous variable, Larrañaga et al. [32] proposed *UMDA Continuous* (UMDAc). UMDAc models the univariate function with the Gaussian distribution and uses the maximum likelihood to estimate the parameters. In the case of Gaussian distribution, UMDAc has the following form of the univariate density function:

$$p_k(x_i; \mu_i, \sigma_i^2) = \frac{1}{\sigma_i\sqrt{2\pi}} \exp\left[-\frac{(x_i - \mu_i)^2}{2\sigma_i^2}\right]$$

where μ_i and σ_i^2 refer to the mean and the variance respectively.

All univariate EDAs share one problem that they do not consider conditional dependency among random parameters. In the bivariate EDAs, pairwise variable interaction is modelled with the conditional probability

function, in which a variable is conditionally dependent on exactly one other variable. The estimated density function with pairwise dependency at the kth generation can be written as $\hat{p}_k(X) = \prod_{i,j} p_k(x_i|x_j)$.

In the domain of discrete variable, Baluja and Davies [5] proposed *Combining Optimizer with Mutual Information Tree* (COMMIT). COMMIT estimates the probability function with tree-structured Bayesian networks. The networks are constructed by iteratively adding the largest branch, which is determined by minimizing the mutual information criteria. In COMMIT, each bivariate probability function has the form: $p_k(x_i|x_j) = p_k(x_i, x_j)/p_k(x_j)$, where $(x_i, x_i) = \arg\min_{i,j} I(X_i, X_j) = \arg\min_{i,j}(H(X_i) - H(X_i|X_j))$. $I(X_i, X_j)$ is the mutual information measure, $H(X_i)$ is the uncertainty of x_i, and $H(X_i \mid X_j)$ is the uncertainty of x_i given X_j. COMMIT adds new search diversities by applying the hill-climbing procedure and the PBIL-like linear function to the sampled individuals. In addition, De Bonet et al. [12] proposed the *Mutual Information Maximization for Input Clustering* (MIMIC), which makes use of an information-theoretic greedy function to estimate the probability function. Specifically, MIMIC applies the Kullback–Leibler divergence to search for an order list of bivariate probability functions, and each bivariate marginal function has the form: $p_k(x_i \mid x_{i+1})$, where $x_i = \arg\min_{j \neq i+1,\ldots,n} H(X_j|X_{i+1})$ for $i \in [1, n-1]$ and $x_n = \arg\min_j H(X_j)$.

In the domain of continuous variable, Larrañaga et al. [32] proposed *MIMIC Continuous* (MIMICc). MIMICc models the pairwise marginal density function with the bivariate Gaussian distribution. MIMICc searches for an order list of bivariate density function by minimizing the conditional entropy

$$H(X_i|X_j) = \frac{1}{2}\left[(1 + \log 2\pi) + \log\left(\frac{\sigma_i^2\sigma_j^2 - \sigma_{ij}^2}{\sigma_j^2}\right)\right]$$

where σ_{ij}^2 is the covariance.

In the multivariate EDAs, higher-order variable interaction is modelled, in which a variable is conditionally dependent on a set of variables. The estimated density function with multiple dependencies at the kth generation can be written as $\hat{p}_k(X) = \prod_i p_k(x_i|Y_i)$, where $Y_i = \{y_{i1}, \ldots, y_{ij}\}$. The estimation of the conditional probability is the most challenging task for all multivariate density estimation evolutionary algorithms.

In the domain of discrete variable, Mühlenbein and Mahnig [34] proposed *Factorized Distribution Algorithm* (FDA). FDA requires additively decomposed functions for the factorization of probability distribution. FDA may require expert knowledge to determine the appropriate decomposition. Harik [20] proposed *Extended Compact Genetic Algorithm* (ECGA). ECGA represents the joint distribution function with the product of clustered univariate distributions. The clustering of variables helps to reduce the structure complexity, and the clustering is achieved by means of the greedy search method. ECGA uses the metric of minimum description length to evaluate the probability density structure. In addition, Pelikan et al. [40] proposed *Bayesian Optimization Algorithm* (BOA). BOA models the joint distribution function with the Bayesian networks. It uses the search method with the Bayesian Dirichlet metric to learn the network structure. A constraint on the maximum number of parents is used to control the network complexity. Larrañaga et al. [30] proposed *Estimation of Bayesian Networks Algorithm* (EBNA), which also uses the Bayesian networks to estimate the multivariate distribution function. EBNA uses the *Bayesian Information Criteria* (BIC) metric to evaluate the network structure.

In the domain of continuous variable, Larrañaga et al. [31] proposed *Estimation of Multivariate Normal Algorithm* (EMNA). EMNA estimates the parameters of multivariate Gaussian distributions with the maximum likelihood method. The multivariate density function is updated in two approaches. The first approach compares the sampled individual with the worst individual in the population and it replaces the worst individual if the sampled individual has a better fitness. The second approach directly adds the sampled individual into the population, without removing any existing individuals. In addition, Larrañaga et al. [32] proposed *Estimation of Gaussian Networks Algorithm* (EGNA). EGNA approximates the density function with the Gaussian networks. The networks can be determined by several metrics including the Bayesian score, the penalized maximum likelihood score, and the edge exclusion tests. Furthermore, Bosman and Thierens [8] proposed a framework of *Iterative Density Evolutionary Algorithms* (IDEA). IDEA considers both parametric (i.e. normal distribution) and non-parametric density models (i.e. histogram distribution and kernel methods). IDEA represents the marginal conditional Gaussian density function as a one-dimensional normalized Gaussian $p_k(x_i|Y_i) = g(\bar{\mu}_i, \bar{\sigma}_i)$. For the interested readers, excellent literature reviews on EDAs can be found in [5,29].

Theoretically, the *Maximum-Entropy* method (MaxEnt) can be used to inference the probability distribution with arbitrary dependency. Multivariate dependency is modelled with conditional maximum-entropy distribution functions. The form of conditional maximum-entropy distribution functions is provided in the next section. In practice, maximum-entropy estimated distribution model of bivariate dependency has been applied to feature selection problem by Yu and Scheunders [49]. More recently, Wright et al. [47] applied the 2nd-order maximum-entropy-based EDA to a set of test problems including deception-trap functions and NK landscapes.

The computational issue of maximum-entropy-based EDA is twofold. Firstly, MaxEnt method demands excessive computation in learning. Secondly, probability estimation is iterative in the EDA algorithm. It means that cost for maximum-entropy learning has to be repeated till the convergence of EDA. Two major approaches have been proposed to alleviate this problem.

The first approach is to improve the optimization function of the maximum-entropy model. Traditionally, an iterative method called *Generalized Iterative Scaling* (GIS) proposed by Darroch and Ratcliff [11] has been used as a popular method for parameter optimization. In addition, Della Pietra el al [13] proposed *Improved Iterative Scaling* algorithm (IIS). IIS does not require the sum of input variables to be a constant over all the training examples. In addition, Goodman [19] proposed *Sequential Conditional Generalized Iterative Scaling* (SCGIS) method. SCGIS learns the model parameters sequentially rather than simultaneously, which leads to an order of magnitude faster than GIS. Recently, Malouf [33] compared a number of algorithms in estimating the maximum-entropy model parameter and found that the general-purpose, nonlinear optimization methods like the conjugate gradient and the quasi-Newton method outperform GIS and IIS in some test problems.

The second approach is to reduce the dimensionality of the maximum-entropy model. Yan and Miller [48] proposed an approximation method to restrict the support of joint probability distribution to a subset of parameter space. Specifically, constraints without support from data sets are not considered in model training. In addition, Goodman [18] proposed the clustering approach to reduce the size of non-zero support constraints. These dimension reduction methods are static in nature. And the reduction is applied only to constrain the size of the model.

The two approaches described above are limited to the scope of maximum-entropy modelling, and only the first-order dimension reduction is considered (i.e. the number of constraints). This paper proposes an adaptive approach in model learning. The proposed dynamic approach is more effective in dimension reduction. It is guaranteed to obtain a subset dimension no larger than that of the static dimension reduction method. In addition, the proposed method considers dimension reduction from the perspective of the population-based MEED model. It dynamically adjusts selected population size based on the population fitness. Unlike the existing approaches, the proposed method reduces problem size in both constraints and data size.

3. Estimated maximum-entropy distribution method

This section introduces an *Estimation of the Distribution Algorithm* (EDA) based on *Maximum-Entropy Distribution* (MEED). Firstly we discuss the motivation of the MEED model. Next, we present the procedure of the MEED model. Finally, we discuss the computational complexity of this model.

3.1. Motivations

The MEED model combines the benefits of maximum-entropy probability modelling and the search power of EDA. The MEED model makes use of the MaxEnt model to obtain a conditional probability distribution. Furthermore, it generates the population of the next generation by sampling from the current model. The entire MEED process generates a chain of models from the initial random distribution to the true distribution of optimal solution space. Along the chain, each model is a step closer to the true probability distribution. The advantages of the proposed MEED model compared with the classic genetic algorithm are twofold.

Firstly, it removes the limitation of structure representation in GAs. GA relies on a string-structured representation of parameter space. Its effectiveness in solving a problem largely depends on how well the string-structure is encoding the problem. GA does not have a mechanism to take into account the information about the causal relationship among variables. This causes ad hoc linkage of different subsets of parameter space. In

addition, the expressiveness of string-structures is limited in GAs and it is not straightforward in representing hierarchical or networked problems.

Secondly, MEED does not suffer from the GA's bias towards short partial solutions. The uniform crossover in GA favors short partial solutions. The solutions with long-distance or wide gap are very likely to be disrupted by the crossover operator. The probabilistic framework in MEED does not have these limitations. It has been shown empirically that EDAs outperform GAs in several optimization problems including graph matching [7], linear optimization problems [39], and deception-trap functions and NK landscapes [47].

However, it must be noted that the conventional EDAs do not include a global random function. The conventional EDAs consist of three important components including the sampling operator, the selection operator, and the density estimation. The sampling operator generates a set of samples from a given density function. The samples are constrained by the parameter space of the distribution function. The sampling size controls only the range that it searches within the density function. By applying an appropriate selection scheme, the selection operator reduces the samples to a subset. So the selection operator is also limited to the same parameter space of the given density function. Again, the density estimation operation does not jump out of the original parameter space because the inference is based on the selected samples. Therefore, EDAs without a global random function are highly dependent on the initialization.

To reduce the dependence on the initial samples, many EDAs introduce new search diversities when updating the density function. The following approaches have been proposed: (i) using the linear function to incorporate individuals from the previous generation, as in PBIL; (ii) performing a local search for the selected individuals, as in COMMIT [5]; and (iii) incrementally adding all newly sampled individuals, as in EMNA [31]. However, all these methods can be regarded as the variations of the local search method. EDAs without a global randomization function may have several problems. For example, it is sensitive to initialization, and it is likely to be trapped by local optima. The proposed MEED model rectifies this problem by including a global randomization function. The proof of the global convergence for the proposed method is also provided.

3.2. Maximum-entropy distribution

The maximum-entropy distribution model provides a clear and logical way to estimate the probability distribution [28,48] for EDAs. With uncertainty or incomplete information present in the system, there are often infinite probability distributions satisfying given constraints. The maximum-entropy model chooses the least biased distribution, which maximises uncertainty (or entropy) in the distribution subject to given constraints. It is well known that many standard probability distributions (e.g. Gaussian and Gamma distributions) maximize the information entropy when satisfying the given moment constraints [28]. For example, the Gaussian distribution maximizes the entropy among all distributions subject to its mean and variance; and the exponential distribution maximizes the entropy subject to its given mean. Since the maximum-entropy distribution does not assume any particular structure of the density function, maximum-entropy distribution does not make any assumption about normality of the underlying data.

In information theory, entropy is a measure of uncertainty. It defines the average amount of information contained in any random variable. The entropy provides an aggregation measure over all individual data. In particular, it can be used to calculate the information-theoretic properties of an entire population from individual data. This ability to make aggregated predictions based on individual data is the main advantage of entropy over other measures. In addition, entropy provides a theoretical lower bound of the amount of channel capacity required to reliably represent the entire information source.

Let a random variable with n components be denoted as $X = \{x_1, \ldots, x_n\}$, and its corresponding probability be denoted as $P = \{p_1, \ldots, p_n\}$. The entropy of X is defined as

$$H = -\sum p_i \lg p_i \tag{1}$$

According to Jaynes [27] and Kapur and Kesavan [28], MaxEnt method suggests maximising entropy H subject to the given constraints:

$$\sum_{i=1}^{n} p_i = 1 \tag{2}$$

$$\sum_{i=1}^{n} p_i g_r(x_i) = a_r \tag{3}$$

where $r \in \{1, 2, \ldots, m\}$ and $p_i \geqslant 0$. In Eq. (3), a_r refers to algebraic moments of X, for example, mean and variance; and $g_r(X)$ is a generalized function of x. Maximizing H with the constraints (2) and (3) is a constrained optimization problem. The *Lagrangian* converts the problem into the non-constrained as follows:

$$L \equiv -\sum_{i=1}^{n} p_i \ln p_i - (\lambda_0 - 1)\left(\sum_{i=1}^{n} p_i - 1\right) - \sum_{r=1}^{m} \lambda_r \left(\sum_{i=1}^{n} p_i g_{ri} - a_r\right) \tag{4}$$

where λ_r is *Lagrangian* multiplier and $|\ \lambda_r\ |$ corresponding to the $m + 1$ constraints in (2) and (3). Differentiate with respect to P, and we obtain:

$$\frac{\partial L}{\partial p_i} = 0 \Rightarrow \ln p_i + \lambda_0 + \sum_{r=1}^{m} \lambda_r g_{ri} = 0 \tag{5}$$

$$p_i = \exp\left(-\lambda_0 - \sum_{r=1}^{m} \lambda_r g_{ri}\right) \tag{6}$$

If the maximum-entropy distribution P exists subject to the constraints, we have the marginal probability distribution:

$$p_i = \frac{\exp\left(-\sum_{r=1}^{m} \lambda_r g_{ri}\right)}{\sum_{i=1}^{n} \exp\left(-\sum_{r=1}^{m} \lambda_r g_{ri}\right)} \tag{7}$$

For multivariate dependency, the conditional maximum-entropy distribution is written as

$$p_i(J) = p(i|j_0, j_1, \ldots, j_{k-1}) = \frac{\exp\left(-\sum_r \lambda_r g_{ri}\right) \cdot \prod_j \exp\left(-\sum_r \lambda_r g_{rj}\right)}{Z_i \cdot \prod_j Z_j} \tag{8}$$

where a random variable x_i is conditionally dependent on k other variables $x_{j0}, x_{j1}, \ldots, x_{jk-1}$, and $Z_j = \sum_j \exp\left(-\sum_r \lambda_r g_{rj}\right)$. The joint probability distribution is written as

$$P = \prod_{i \in n} p(i|J_i) \tag{9}$$

With the support of a training data set, e.g. a set of selected individuals in a generation, the parameters λ_r can be estimated by minimizing the *Kullback–Leibler* divergence between the empirical distribution Q and the model P, and the *Kullback–Leibler* divergence is defined as

$$D(Q, P) = \sum q_i \lg \frac{q_i}{p_i} \tag{10}$$

Or equivalently, it maximizes the log-likelihood of the empirical distribution Q, which is defined as

$$L(Q) = \sum q_i \lg p_i \tag{11}$$

Substitute Eq. (7) into Eq. (11), and the log-likelihood is:

$$L(Q) = -\sum_{i=1}^{n} q_i \sum_{r=1}^{m} \lambda_r g_{ri} - \sum_{i=1}^{n} q_i \lg \sum_{i=1}^{n} \exp\left(-\sum_{r=1}^{m} \lambda_r g_{ri}\right) \tag{12}$$

Differentiating with respect to the parameter λ_r, we have:

$$\frac{\partial L(Q)}{\partial \lambda_r} = -\sum_{i=1}^{n} q_i \sum_{r=1}^{m} g_{ri} - \frac{\sum_{i=1}^{n} q_i \sum_{i=1}^{n} \exp\left(-\sum_{r=1}^{m} g_{ri}\right)}{\sum_{i=1}^{n} \exp\left(-\sum_{r=1}^{m} g_{ri}\right)} \tag{13}$$

Setting Eq. (13) to zero yields the globally optimal conditions of the log-likelihood with respect to parameter λ_r. However, it does not give a closed form solution.

A general-purpose optimization technique such as GIS or conjugate gradient is often used to find an optimum parameter λ_r. For instance, GIS is used in maximum-entropy-based language modelling [19]. In the parameter learning, it first calculates the numerator of RHS of Eq. (7) and normalization factor Z for the entire data set and all outputs. Next the probability distribution can be obtained, and the expected model values can be updated. To evaluate the current set of *Lagrangian* multipliers $\bar{\lambda}_{r,t}$, the expected parameter values are compared with the observed values. The difference denoted as δ is then updated into $\bar{\lambda}_{r,t+1}$. The above process iterates until the difference δ is smaller than a predefined threshold.

3.3. Procedure of the MEED model

The MEED model is the process of estimating the maximum-entropy function with a set of the population-based constraints. The process reaches its equilibrium when the fitness improvement does not exceed a threshold.

In the initialization, the population is randomly sampled from a uniform distribution. The size of population N is determined by the sufficiency of sampling. From the generated population, selection is performed to shortlist promising individuals S. Many methods exist to select individuals, such as tournament selection or roulette wheel selection [17]. The selected population is denoted as D^S_{k-1} where $k-1$ is the index of iteration.

It is important to determine the size S. If it is too small, it may not include all promising search points. If it is too large, it increases the learning overhead. An ad hoc MEED model may select S with a fixed size. After selection, it applies a global randomization operation to the selected individuals.

Next, it estimates the probability distribution from the selected population. This is done by updating *Lagrangian* multipliers $\bar{\lambda}_{r,i}$. The convergent model is the estimated conditional probability distribution of the selected individuals, i.e. $\widehat{P}_k(X|D^S_{k-1}) = \prod_{i\in n} p_k(i|J_i, D^S_{k-1})$. If the change of population fitness from the model is greater than the threshold, the procedure continues. The MEED procedure is depicted in Fig. 1.

3.4. Complexity analysis of MEED

This analysis focuses on the complexity of parameter learning in Step 2.2 of Fig. 1. The discussion assumes that GIS is used for parameter estimation, and in general the time complexity holds for other iterative methods. Choosing other methods like conjugate gradient may result in a better update size.

It is known that GIS has the time complexity of $\mathbf{O}(M \cdot N \cdot T)$ for an iteration, where M is the number of constraints, N is the feature dimension, and T is the number of training instances. In order to avoid repetitive updates of the same change in other parameters, GIS introduces the slowing factor f_{GIS}, which is set to the maximum number of active constraints [19]. The update step size of GIS is inversely proportional to f_{GIS}. Thus, MEED model has the time complexity of $\mathbf{O}\left(\frac{M\cdot N\cdot T}{f_{\mathrm{GIS}}\cdot f_{\mathrm{EDA}}}\right)$, where f_{EDA} is the convergence factor.

As the size N is constant, one needs to reduce the dimension of M or T in order to reduce overall time complexity. The number of constraints M has an exponential form $|x|^n \cdot \binom{N}{n}$ where $|x|$ is value dimension, n is

Step 1: $D_0 \leftarrow$ Generate initial population with size N.
Step 2: Repeat for k = 1,2, … until a stop criterion is met.
  Step 2.1: $D^S_{k-1} \leftarrow$ Select S <= N individuals from D_{k-1} according to a method. And apply global randomization operation.
  Step 2.2: $\bar{\lambda}_{r,i} \leftarrow$ Learning model parameter vector $\bar{\lambda}_r$ for $\hat{p}_i$ from D^S_{k-1} according to an optimization method.
  Step 2.3: $\hat{P}_k(X \mid D^S_{k-1}) \leftarrow$ Estimate the conditional probability distribution of selected individuals.
  Step 2.4: $D_n \leftarrow$ Sample N individuals from $\hat{P}_k(X \mid D^S_{k-1})$.

Fig. 1. The MEED procedure.

the order of constraints, and $n < N$. For example, M is equal to $\sum_{i\in N}|x_i|$ when the constraint is univariate (i.e. $n = 1$), and it increases to $\sum_{i\in I, j\in J}|x_i| \cdot |x_j|$ for $I = \{1, \ldots, N-1\}$ and $J = \{i+1, \ldots, N\}$, when the constraint is bivariate (i.e. $n = 2$). In general, the constraint size M follows a power law distribution as n increases.

From the above discussion, it is clear that the time complexity of estimating maximum-entropy distribution is dominated by a large number of low-order constraints. One needs to reduce the constraint size when a multivariate EDA model is trained. In the next section, an adaptive method is proposed to reduce the constraint dimensionality.

4. The proposed adaptive MEED model

The proposed model exploits our observation that the population fitness is relatively low in the early stage of the search process. It is expected that a relatively large divergence exists in the average population fitness between the early stage and the convergence stage of the search. So it is not necessary to estimate the probability function with a full account of constraints in the early search.

In order to determine the constraints for the model estimation, the proposed adaptive approach selects the constraints based on the relative fitness of n-ordered parameter, and the number of constraints is dynamically determined by the population fitness. By removing the individuals below average fitness, the model effectively excludes weak constraints in estimating the density function. Since the weak constraints are part of non-zero constraints, it is guaranteed that the constraint size obtained in the adaptive approach is not larger than the non-zero constraints. One might argue that the discarded constraints should be included as additional information. But the information contains a high level of noise because of their relatively low fitness support. Therefore, the trade-off of these constraints for the computational speed-up seems reasonable.

In addition, the proposed model dynamically adjusts the selection size based on the population fitness. In this way, a good balance is achieved between the reservation of the promising search points and the reduction of the estimation overhead. In summary, the proposed model reduces both constraint dimension and the selection size based on the population fitness. The following section presents the details of the proposed adaptive method.

4.1. Adaptive control functions

The minimum support of constraints is set to the normalized average fitness value of the selected population. Let $\bar{f}$ denote the normalized average fitness of the selected population, and let $f(x_i)$ denote the fitness of the individual having value x_i, satisfying the condition $f(x_i) \geqslant \bar{f}$. The estimated marginal conditional probability on a single variable is written as

$$\hat{p}(i|j) = \sum_{\geqslant \bar{f}} f(x_i, x_j) \Big/ \sum_{\geqslant \bar{f}} f(x_j) \tag{14}$$

where $f(x_i, x_j)$ is the fitness of the individual having value (x_i, x_j). And the marginal conditional probability on a set of variables is written as

$$\hat{p}(i|J) = \sum_{\geqslant \bar{f}} f(x_i, x_{j0}, \ldots, x_{j(k-1)}) \Big/ \sum_{\geqslant \bar{f}} f(x_{j0}, \ldots, x_{j(k-1)}) \tag{15}$$

where $(x_{j0}, \ldots, x_{j(k-1)})$ is a subset of k variables, and $f(x_i, x_{j0}, \ldots, x_{j(k-1)})$ is the fitness of the individual having value $(x_i, x_{j0}, \ldots, x_{j(k-1)})$.

The proposed model is based on two feedback adaptive functions, in which the population fitness is used as the feedback information to control the thresholds of the selected population size and constraint size. The notations used in the adaptive functions are listed in Table 1.

The adaptive function to select the constraints is defined as

$$p_{\lambda,t} = p_{\lambda\min} + \frac{\exp(\bar{f}_{s,t} - f'_{\min})(p_{\lambda\max} - p_{\lambda\min})}{\exp(f'_{\max} - f'_{\min})} \tag{16}$$

Table 1
The adaptive functions notation

Notation	Description
$f_{\max}$	Maximum scaled fitness value
$f_{\min}$	Minimum scaled fitness value
$\bar{f}_t$	Average fitness at generation t
$\bar{f}s, t$	Average fitness of selected population at generation t
$f'_{\max}$	Maximum scaled fitness in selected population at generation t
$f'_{\min}$	Minimum scaled fitness in selected population at generation t
p_λ	Threshold weight of constraints
p_η	Threshold weight of selected population
$p_{\lambda\max}$	Maximum threshold value of p_λ
$p_{\lambda\min}$	Minimum threshold value of p_λ
$p_{\eta\max}$	Maximum threshold value of p_η
$p_{\eta\min}$	Minimum threshold value of p_η

Similarly, the adaptive function to select the population size is defined as

$$p_{\eta,t} = p_{\eta\min} + \frac{\exp(\bar{f}_t - f_{\min})(p_{\eta\max} - p_{\eta\min})}{\exp(f_{\max} - f_{\min})} \tag{17}$$

The exponential functions in Eqs. (16) and (17) capture the dynamic relationship between the population fitness and the constraint (or population) size. That is, a larger improvement of fitness leads to a higher percentage of selected constraints.

4.2. The adaptive MEED model

Several variables need to be initialized in the two adaptive functions. These variables include the maximum and minimum scaled fitness values, i.e. $f_{\max}$ and $f_{\min}$, the maximum and minimum are scaled fitness of vector $\boldsymbol{x}$ in the selected population at the tth generation, i.e. $f'_{\max}$ and $f'_{\min}$, the maximum and minimum value of the threshold weight value for the constraint, i.e. $p_{\lambda\max}$ and $p_{\lambda\min}$, and the maximum and minimum value of threshold weight value for the selected population, i.e. $p_{\eta\max}$ and $p_{\eta\min}$.

After the random generation of the initial population, the adaptive selection is performed to shortlist the promising individuals according to Eq. (17). The selected population is denoted as D^S_{n-1} where $n-1$ is the index of iteration. Next, the constraint size is determined by adaptive function according to Eq. (16).

Next, the model estimates the maximum-entropy probability distribution from the dynamically selected constraints. This is done in the same way as in the simple model by updating *Lagrangian* multipliers $\bar{\lambda}r, i$. The procedure of the adaptive model is summarized in Fig. 2.

The adaptive features are shown at Step 2.1 and Step 2.2 of Fig. 2. The adaptive operation mostly ignores non-essential constraints when the population fitness is low. This greatly reduces the number of constraints in estimating the maximum-entropy distribution function.

Step 1: $D_0 \leftarrow$ Generate initial population with size N.
Step 2: Repeat for k = 1,2, … until a stop criterion is met.
  Step 2.1: $D^S_{k-1} \leftarrow$ Adaptively select $S <= N$ individuals from D_{k-1} according to equation (17). And apply global randomization operation.
  Step 2.2: Determine $p_{\lambda,k-1}$ according to (16).
  Step 2.3: $\bar{\lambda}_{r',i} \leftarrow$ Learning model parameter vector $\bar{\lambda}_{r'} << \bar{\lambda}_r$ for $\hat{p}_i$ from D^S_{k-1} according to an optimization method.
  Step 2.4: $\hat{P}_k(X \mid D^S_{k-1}) \leftarrow$ Estimate the conditional probability distribution of selected individuals.
  Step 2.5: $D_n \leftarrow$ Sample N individuals from $\hat{P}_k(X \mid D^S_{k-1})$.

Fig. 2. The adaptive MEED procedure.

4.3. Complexity analysis of the adaptive MEED

Assume that the computational cost of parameter learning is much larger than the aggregated cost of global sampling, selection, and local sampling. The complexity analysis of the Adaptive MEED model is simplified, particularly in the cost of parameter learning.

In the Adaptive MEED model, the constraint size M and the selected population size T are dynamically reduced. After applying the adaptive operation, the constraint size has a form, $M \cdot p_\lambda$, where p_λ is constraint weight and $p_\lambda \in (0,1)$. And the population size has a form, $T \cdot p_\eta$, where p_η is population weight and $p_\eta \in (0,1)$. Assuming that the active constraints are evenly distributed (regardless of their fitness values), the slowing factor f_{GIS} is improved by a factor of $p_\lambda \cdot p_\eta$ when applying the adaptive operation.

In one iteration, the Adaptive MEED model has the time complexity of $\mathbf{O}\left(\frac{M \cdot N \cdot T \cdot p_\lambda^2 \cdot p_\eta^2}{f_{\mathrm{GIS}}}\right)$. Considering the convergence factor f_{EDA}, the model has the time complexity of $\mathbf{O}\left(\frac{M \cdot N \cdot T \cdot p_\lambda^2 \cdot p_\eta^2}{f_{\mathrm{GIS}} \cdot f_{\mathrm{EDA}}}\right)$. Therefore, the adaptive operation reduces the time complexity by $1-\beta^2$, where β is a threshold value associated with the population fitness and $\beta = p_\lambda \cdot p_\eta$.

5. Markov chain analysis

The Markov chain has been successfully applied to model the operation of GA [9,16,38]. This section extends Markov chain to model the operation of the proposed model. The Adaptive MEED model is in essence a stochastic process consisting of the selection, the global sampling, and the local sampling. This section shows that the proposed model can be modelled as an ergodic Markov chain. In other words, after a sufficiently long run, the Adaptive MEED model converges to a steady-state distribution of population.

The following section firstly introduces the basic definitions of Markov chain. Let the state variable $\mathbf{X}$ be a set of mutually exclusive states, and its observations (or feature vectors) at different time are recorded as $\{\mathbf{x}^{(t)}\}$. A discrete-time finite-state Markov chain is a time series of state variable $\{\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(t)}\}$, in which the conditional probability of any future event given any past events and present state depends only on the present state [22]. Formally, that is, $p(\mathbf{x}^{\{t+1)}|\mathbf{x}^{(0)}, \mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(t-1)}, \mathbf{x}^{(t)} = p(\mathbf{x}^{(t+1)}|\mathbf{x}^{(t)})$.

The conditional probabilities $p(\mathbf{x}^{(t+1)}|\mathbf{x}^{(t)})$ are called (one-step) transition probabilities. If the transition probabilities do not change over time, these probabilities are called stationary transition probabilities, i.e. $p(\mathbf{x}^1|\mathbf{x}^{(0)}) = p(\mathbf{x}^{(t+1)}|\mathbf{x}^{(t)})$. In general, the stationary transition probabilities from state x at time t to state x' at time $t+n$ are called n-step stationary transition probabilities, i.e. $p(\mathbf{x}^n = x|\mathbf{x}^{(0)} = x') = p(\mathbf{x}^{(t+n)} = x|\mathbf{x}^{(t)} = x')$. For simplicity, the n-step transition probabilities from state x to state x' are denoted as $T^n(x,x')$.

A Markov chain is ergodic if the probabilities of the state variable converge to an equilibrium distribution π when the time step t goes to infinite, regardless of the initial distribution $p^{(0)}$. That is, $\lim_{t\to\infty} p^{(t)}(x) = \pi(x)$, where $\pi(x)$ is called steady-state (or invariant) probability, and it uniquely satisfies $\pi(x) = \sum_{x' \in M} \pi(x') T(x',x)$ and $\sum_{x \in M} \pi(x) = 1$. In other words, after a long run, the probability of state variable x becomes steady, and it is the only steady-state in the chain.

Clearly, the global sampling operator can be modelled with a stochastic process. In fact, it is well known that the Metropolis sampling procedure can be modelled with an ergodic Markov chain [36]. In addition, Rudolph [41] proved that selection without mutation and crossover can be modelled with a homogeneous Markov chain, but the chain is not ergodic. However, it is not obvious that the stochastic procedure of adaptive MEED is ergodic. The rest of this section extends the Markov modelling of canonical genetic algorithms proposed by Rudolph [41] to the Adaptive MEED model. The classification of the transition matrix follows the definitions used in Rudolph's notation.

Definition 1. A square matrix $\mathbf{A} : n \times n$ is called positive if $a_{ij} > 0$ for all $i,j \in \{1,\ldots,n\}$. A non-negative square matrix $\mathbf{A}$ is called primitive if there exists a $k \in N$ such that $\mathbf{A}^k > 0$. A non-negative square matrix $\mathbf{A}$ is stochastic if $\sum_{j \in [1,n]} a_{ij} = 1$ for all $i \in [1,n]$. A stochastic matrix $\mathbf{A}$ is column-allowable if it has at least one positive entry in each column.

Theorem 1 (cf. [25, p.123]). *Let* $\mathbf{P}$ *be a primitive stochastic matrix, and k be the time step. Then* $\mathbf{P}^k$ *converges to a positive stable stochastic matrix* $\mathbf{P}^\infty$ *as* $k \to \infty$, *and* $\mathbf{P}^\infty$ *has positive entries and is unique regardless of the initial distribution.*

Lemma 1 (cf. [41]). *Let* $\mathbf{S}$, $\mathbf{M}$, *and* $\mathbf{PS}$ *be stochastic matrices, where* $\mathbf{M}$ *is positive and* $\mathbf{PS}$ *is column-allowable. Then the product* $\mathbf{S} \cdot \mathbf{M} \cdot \mathbf{PS}$ *is positive.*

Theorem 2. *The MEED transition matrix with local sampling, global sampling, proportional selection, and parameter learning is primitive.*

Proof. Each operator is represented with a square matrix. Let **S**, **M, PS**, and **PL** denote the square matrices for operators of local sampling, global sampling, proportional selection, and parameter learning, respectively. Each operator may be considered as a probabilistic mapping function from one state to another state. Let state space be denoted as $S = \{s_1, \ldots s_n\}$. Let a_{ij} for $i, j \in \{1, \ldots, n\}$ represent the probability of state s^i to state s^j, and $\sum_{j\in[1,n]} a_{ij} = 1$ for all $i \in [1, n]$. Thus all four square matrices, **S**, **M**, **PS**, and **PL**, are stochastic matrices.

In global sampling, the probability of generating an individual with state x^i is $m_i = P(x_i) = p_m > 0, \forall i \in [1, n]$, where p_m is the probability of global sampling. As the global sampling operator is applied to all bits in the population, $m_i > 0$ for every $i \in \{1, \ldots, n\}$. Therefore, **M** is positive.

In proportional selection, the probability of selecting an individual x^i is $Selection_i = P(X_i) = f(X_i)/\sum_{i\in[1,N]} f(X_i)$. The probability of selecting the same state i is, $ps_{ii} = f(x_i)/\sum_{i\in[1,N]} f(x_i) = f(X_i)/\sum_{i\in[1,N]} f(X_i) > 0$. Since stochastic matrix **PS** has at least one positive entry in each column, i.e. $ps_{ii} > 0$, **PS** is column-allowable.

In parametric learning, all individuals are represented in a form of probabilistic distribution. The probability of a state i that is mapped into the distribution is $pl_{ii} = freq(x_i)/N > 0$. Since stochastic matrix **PL** has at least one positive entry in each column, i.e., $pl_{ii} > 0$, **PL** is column-allowable.

According to Lemma 1, the product $\mathbf{S} \cdot \mathbf{M} \cdot \mathbf{PS}$ is positive, because **S** is stochastic matrix, **M** is positive, and **PS** is column-allowable. Similarly, Let $(s \cdot m \cdot ps \cdot pl)_{ij} = \sum_{k=1}^{n} [(s \cdot m \cdot ps)_{ik} \cdot pl_{kj}] > 0$ for all $i, j \in \{1, \ldots, n\}$ because **PL** is column-allowable. Therefore transition matrix $\mathbf{S} \cdot \mathbf{M} \cdot \mathbf{PS} \cdot \mathbf{PL}$ is positive. As all positive matrices are primitive, the transition matrix $\mathbf{S} \cdot \mathbf{M} \cdot \mathbf{PS} \cdot \mathbf{PL}$ is primitive. □

Notice that the proposed model does not make use of elitism. However, when elitism is included in the model, one additional elitism matrix should be included. The elitism matrix is sparse and contains only a few entries in the matrix that have non-zero probabilities. The matrix would be added to the matrix of proportional selection, which results in a positive matrix. Therefore, the product matrix of the model with elitism is still primitive.

Theorem 3. *Markov chain of the MEED is ergodic.*

Proof. A Markov chain is ergodic if the probabilities of state variable converge to an equilibrium distribution π when time step t goes to infinite, regardless of initial distribution $p^{(0)}$, i.e. $\lim_{t\to\infty} p^{(t)}(x) = \pi(x)$. According to Theorems 1 and 2, Markov chain of the MEED is ergodic. □

Theorem 4. *The ergodic Markov chain of EDA has global asymptotic convergence.*

Proof. This is a property of ergodic Markov chain. □

The above section modelled the MEED model without an adaptive operation. The following shows that an adaptive model can also be modelled as an ergodic Markov chain.

Theorem 5. *Markov chain of the adaptive MEED is ergodic.*

Proof. Adaptive operation is based on the change of domain fitness, and it applies to two parameters: (i) constraint size, and (ii) selected population size. In the case of population size, individuals with a fitness value below thresholds will be lost.

Fortunately, this will not affect the properties of the transition matrix of proportional selection, and the transition matrix **PS** is still column-allowable with adaptive operation. In this case, the probability of selecting the same state i is, $ps_{ii} = f(x_i)/\sum_{i\in[1,N']} f(x_i) = f(X_i)/\sum_{i\in[1,N']} f(X_i) > 0$, where N' is the number of individuals after adaptive selection.

Similarly, in the case of constraint size, the loss of constraints below threshold will not change the properties of the transition matrix of parametric learning. Specifically, the probability of a state i mapped into underlying distribution is $pl_{ii} = freq(x_i)/N' > 0$. So **PL** after adaptive selection is also column-allowable. According to Theorems 1 and 2, therefore the Adaptive MEED algorithm can be modelled as an ergodic Markov chain. □

The above discussed the ergodic properties of the proposed model. Being ergodic, the proposed model can be used as a global optimization technique. The next section presents the applications of the Adaptive MEED model to several optimization problems in data mining.

6. Applications in data mining

This section illustrates several examples of applying the proposed MEED model to improve linear discriminant functions, association rules, and clustering in data mining.

The first example shows that the proposed model helps in learning linear discriminant functions. A system of linear functions defines a set of decision surfaces which may separate k classes [14]. It is well known that the decision regions are convex in the system of linear discriminant functions. Thus, the system is suitable only for the classification domains where the class conditional probabilities are unimodal. In order to deal with the multimodal problems, the generalized linear discriminant function has been suggested [14], which has the form, $f(\mathbf{x}) = \sum_{i=1}^{k} a_i y_i(\mathbf{x}) = \mathbf{a}'\mathbf{y}$, where k is the order of the discriminant function, $\mathbf{a}$ is a k-component weight vector, and $y_i(\mathbf{x})$ is an arbitrary function of $\mathbf{x}$. However, the generalized linear discriminant function requires an exponential number of polynomial coefficients to be solved. In general, there are $\frac{(d+k)!}{d!k!}$ polynomial terms for the d-dimensional problem classified with the k-order discriminant function.

In multimodal domains, the MEED model can help the linear discriminant functions in reducing the number of polynomial coefficients. Specifically, the MEED model searches for polynomial coefficients in the linear discriminant functions. Let the feature space, class space, and parametric space of polynomial coefficients be denoted as X, T_1, and T_2 respectively. The classification of the generalised linear function is the second-order projection, i.e. $X \to T_1 \to T_2$. It firstly transforms the feature space X into the parametric space T_1, and then it further transforms T_1 into the class space T_2. In order to find the optimal values of the polynomial coefficients, the linear discriminant function maximizes a classification criterion, e.g. maximum likelihood. Then the polynomial coefficients can be solved by the proposed MEED model.

The second example concerns the association rules. It is well known that the performance bottleneck of the association rules lies in searching for the frequent item-sets [1–4]. It is an *NP* hard problem to search for the association rules in a complete set of candidates. The number of candidate frequent item-sets has a form of $\sum_{k=1}^{d} \binom{d}{k}$, where d is the number of attributes, and k is the size of frequent item-sets. Clearly, the form of the complete item-sets is exponential.

The strength of the MEED-based association rule is the power of global search when the candidate set is extremely large. This approach is different from the Apriori and its variations [1,10,23,44,45], which are instances of the breadth-first search. The Apriori examines the rules with ïitem-set first, and then examines the rules with $i + 1$ item-sets, and so on. The Apriori is an exhaustive search method and it is inefficient when the search space is exponential. In contrast, the MEED-based association rule searches for rules at all levels at the same time. And it looks for an optimal set of rules from a very large set of candidate frequent item-sets.

The last example is about finding globally optimal solutions in clustering problems. It is known that the clustering tasks are non-convex discrete optimization problems [26,42,46], in which the deterministic approaches are likely to be trapped by local optima. Thus the MEED model can be used to search for the optimal number of clusters subject to the given data constraints. The next section describes the implementation details of the MEED-based clusterer.

7. Empirical study

This section presents the implementation details of the clusterer based on the Adaptive MEED model. The MEED-based clustering algorithm employs the representation scheme where each individual represents a set of clusters. This representation scheme is also known as the Pittsburgh approach [15]. The Pittsburgh-based scheme has the advantage of evaluating the quality of a cluster set as a whole, so that different clusters represent different data sets. In this implementation, a cluster is represented by its medoid, which has the advantage of being insensitive to extreme values.

All individuals are measured with the objective function of *F*-statistics. *F*-statistics is defined as the ratio of the variance between classes and the variance within classes, which is written as $F_{stat} = \frac{\alpha \sum_k n_k (\bar{\mathbf{x}}_k - \bar{\mathbf{x}})^2}{\sum_k \sum_i (\mathbf{x}_i - \bar{\mathbf{x}}_k)^2}$. The numerator is the measure of the variation between class means $\bar{\mathbf{x}}_k$ and grand mean $\bar{\mathbf{x}}$, and the denominator is the measure of the variation between data $\boldsymbol{x}^i$ within a class and its class mean $\bar{\mathbf{x}}_k$. The constant α is defined as $\alpha = df_W/df_B$, where df_B is the degree of freedom between classes and $df_B = k - 1$, df_W is the degree of freedom within classes and $df_W = n - k$, n is the total number of data, and k is the number of classes.

The MEED clusterer starts with a random population where each individual contains randomly assigned k data instances as medoids. For each individual, it partitions the data to their nearest medoids by executing one iteration of k-means algorithm. Then all individuals are evaluated, and those which are selected to create the new population are labelled. The maximum-entropy probability function is then calculated based on the selected individuals. The probability function provides a mapping between the attributes of k medoids and the fitness classes.

The fitness classes are user-defined. For example, a coarse class set is defined as low (L), medium (M), and high (H). And the more elaborate class set can be defined as LL, LM, ..., HH, and so on. To sample from the probability function, one fitness class is randomly selected based on its likelihood of occurrence, i.e. $\theta_k = n_k / \sum n_k$, and that class is then taken as a random source to generate an individual. Then a global sampling is performed with a low probability to increase population diversity. After that, the adaptive learning is performed in two levels. First, adaptive individual selection is used to select individuals based on the population fitness. The set of distinct values of medoids shrinks when the selected portion of the population decreases. Second, the active attribute selection takes into account those attribute values whose occurrence is greater than a user-defined threshold.

For comparison, a GA-based clusterer is implemented. The GA clusterer uses a binary coding and the same Pittsburgh representation of cluster sets. It makes use of elitism and the roulette wheel selection. The crossover operator is implemented with the awareness of cluster-boundary. In other words, the crossover operator skips the boundary bits (i.e. the leading and ending bits) of all clusters. All experiments use the crossover rate of 0.8 and the mutation rate of 0.03. In addition, a Gaussian-based EDA clusterer (GauED) is also implemented for comparison. The GauED clusterer uses a Gaussian density function to represent a cluster.

A number of data sets from the UCI machine learning repository [37] are used for the performance study. The summary of the data sets are listed in Table 2. These data sets are used for training without any pre-processing, apart from the two data sets marked with "*". The "vowel" has its three leading attributes removed (i.e. train or test, speaker name, and sex), and the "zoo" has its leading attribute removed (i.e. animal).

The first experiment aims to study the performance of the evolutionary algorithms by varying the number of generations. The information on the number of classes is provided to all algorithms. All experiments have the same population size. All algorithms use the *F-stat* metric as the objective function. The training set is the same as the testing set, except that the class attribute is used for the evaluation during the testing. The performance evaluation is based on the procedure shown in Fig. 3, and the misclassification error is based on the accuracy rate, i.e. $(TP + TN)/(TP + FP + FN + TN)$. The simple evaluation procedure is based on finding a single nearest data instance, and it can be extended to k nearest data instances (also known as k nearest neighbor).

The best fitness values found by the evolutionary algorithms are shown in Fig. 4. The horizontal axis refers to the number of generations, and the vertical axis refers to the absolute *F-stat* value. The error rates of the three algorithms are shown in Fig. 5. Each data point in the figures is the average value over five runs. Two data sets varying in the number of classes, i.e. "glass" having 7 classes and "credit-a" having 2 classes, have been used to make comparison, and they are referred to the (a) and (b) in the figures.

Table 2
The UCI data sets

Data sets	Instances	Attributes	Classes
contact-lenses	24	4	3
credit-a	690	15	2
diabetes	768	8	2
glass	214	9	7
iris	150	4	3
labour	57	16	2
mushroom	8124	22	2
segment	2310	19	7
sonar	208	60	2
soybean	683	35	19
splice	3190	62	3
vote	435	16	2
vowel*	990	10	11
weather	14	4	2
zoo*	101	17	7

```
For each data instance Data[i] {
        Classify the Data[i] to the Cluster[j];

        Get the predicted class value of the nearest data instance for Cluster[j];

        Get the observed class value of Data[i];

        If the observed class value is NOT the predicted class value,
                misClassified ++;
}
Error = misClassified / totalDataNum;
```

Fig. 3. Evaluation function.

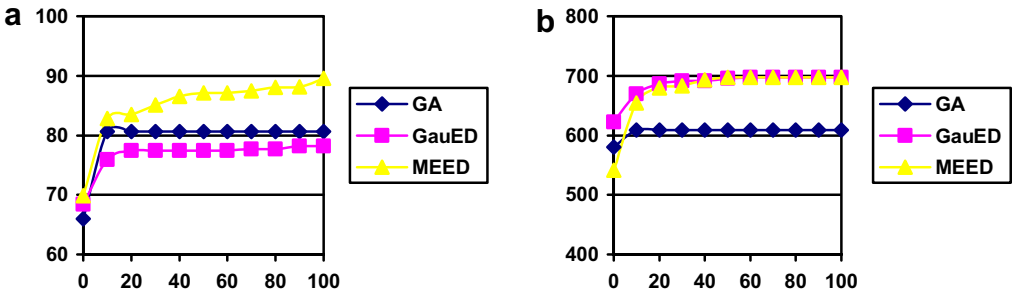

Fig. 4. The effect of generation on the best fitness: (a) "glass" and (b) "credit-a".

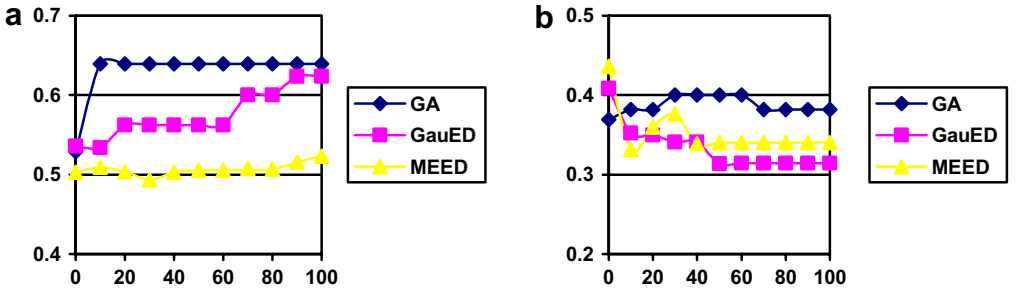

Fig. 5. The effect of generation on the error rate: (a) "glass" and (b) "credit-a".

Fig. 4 shows that all algorithms increase their fitness values when the number of generation increases. For the "glass" data set, the MEED clusterer has a better fitness value than both the GauED and the GA in Fig. 4a, and its lower error rate is recorded correctly in Fig. 5a. For the "credit-a" data set, both the MEED and the GauED have a better fitness value than the GA in Fig. 4b, and their lower error rates are shown in Fig. 5b.

By cross-examining the fitness value and the error rate in both data sets, it is interesting to notice that the *F-stat* metric does not have a positive linear relationship with the accuracy rate. For instance, GauED has the lowest *F-stat* value in Fig. 4a but it does not have the highest error rate in Fig. 5a; an increment in the *F-stat* value in Fig. 4a does not correspond to a reduction in the error rate in Fig. 5a.

The second experiment is for the purpose of studying the performance of the evolutionary algorithms by varying the population size. The same two data sets are used in the study. The performance results of best fitness, error rate, and run-time are shown in Figs. 6–8 respectively.

Again, for the "glass" data set, MEED outperforms GauED and GA in terms of the best fitness (in Fig. 6a). For the "credit-a" data set, MEED achieves a similar performance to GauED on the fitness as shown in Fig. 6b, but MEED scores a higher accuracy than GauED when the population size increases as shown in Fig. 7b. In Fig. 8, all evolutionary algorithms have a linear run-time when the population size increases. The run-time of MEED is lower than for the other two algorithms.

The third experiment aims to study the performance of the evolutionary algorithms by using different data sets. The experiment is based on the maximum 50 generations with the population size of 50. Table 3 shows the best fitness value and run-time of three algorithms across 15 different data sets. MEED clusterer scores the highest fitness value in 11 data sets, the GauED clusterer scores the highest in 3 data sets, and the GA clusterer scores the highest in 3 data sets. In terms of run-time, MEED clusterer is a clear winner in 14 data sets.

The final experiment is to compare the performance of the evolutionary algorithms with *k*-means algorithm. The evolutionary algorithms are based on the maximum 50 generations with the population size of 50. Table 4 lists the error rates with their mean and standard deviation of the four algorithms.

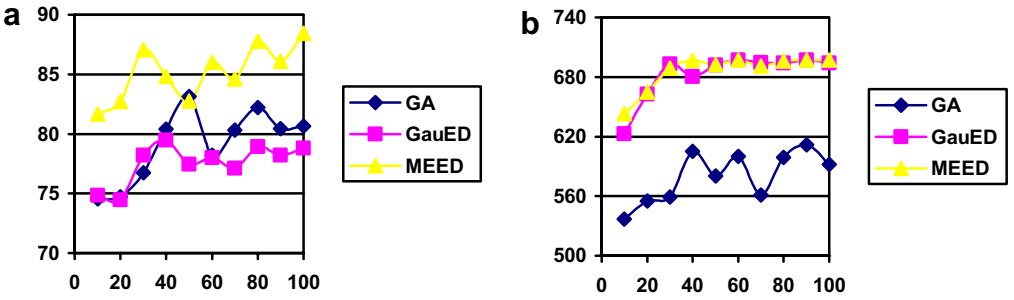

Fig. 6. The effect of population size on the best fitness: (a) "glass" and (b) "credit-a".

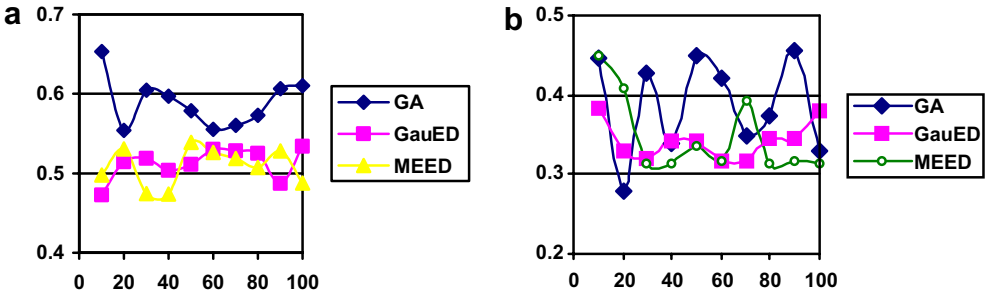

Fig. 7. The effect of population size on the error rate: (a) "glass" and (b) "credit-a".

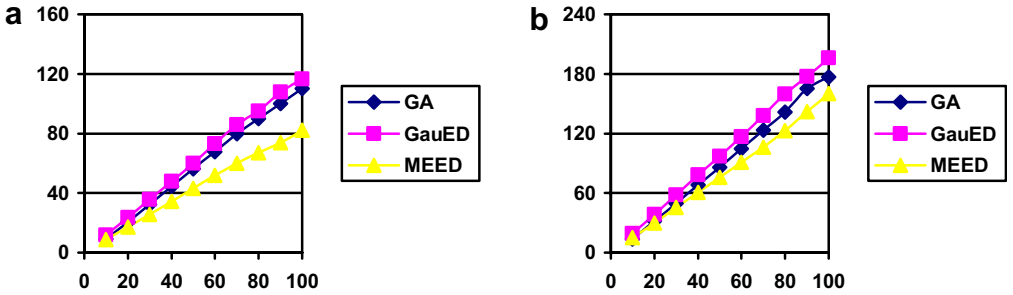

Fig. 8. The effect of population size on the run-time: (a) "glass" and (b) "credit-a".

Table 3
The best fitness and the run-time on 15 data sets

Data sets	GA		GauED		MEED	
	Fitness	Time (s)	Fitness	Time (s)	Fitness	Time (s)
contact-lenses	22.94	1.89	24.67	2.16	24.81	1.85
credit-a	562.44	104.70	693.18	119.57	695.94	92.91
diabetes	354.43	76.02	387.06	85.95	394.85	67.08
glass	81.50	64.39	80.80	71.37	85.92	51.33
iris	355.18	11.23	361.90	12.80	363.19	9.94
labour	33.47	9.35	41.44	9.98	41.44	6.46
mushroom	7314.78	1546.86	9016.59	1745.88	9113.99	1403.09
segment	1428.47	1235.30	1307.66	1287.73	1384.82	916.87
sonar	113.97	113.41	129.41	123.23	129.25	94.62
soybean	65.37	1261.37	63.73	1329.66	69.11	907.45
splice	1762.93	1973.88	1859.11	2292.89	1903.68	2107.80
vote	914.35	59.22	914.97	73.03	914.74	50.87
vowel	140.74	500.57	122.24	567.34	154.77	383.02
weather	20.13	0.86	20.02	0.83	20.13	0.12
zoo	55.68	39.08	55.28	39.34	55.62	28.68

Table 4
The error rate on 15 data sets

Data sets	*k*-means		GA		GauED		MEED	
	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.	Mean	Std. dev.
contact-lenses	0.27	0.14	0.38	0.08	0.43	0.02	0.38	0.04
credit-a	0.68	0.06	0.36	0.07	0.31	0.00	0.31	0.01
diabetes	0.95	0.00	0.38	0.08	0.35	0.00	0.41	0.13
glass	0.31	0.00	0.60	0.06	0.55	0.09	0.51	0.08
iris	0.69	0.00	0.18	0.10	0.12	0.01	0.12	0.00
labour	0.29	0.02	0.29	0.22	0.09	0.00	0.09	0.00
mushroom	0.60	0.02	0.40	0.14	0.29	0.13	0.29	0.13
segment	0.99	0.01	0.55	0.13	0.37	0.06	0.40	0.05
sonar	0.98	0.01	0.53	0.02	0.49	0.04	0.50	0.04
soybean	0.06	0.05	0.54	0.10	0.38	0.04	0.34	0.04
splice	0.73	0.03	0.56	0.07	0.51	0.07	0.52	0.06
vote	0.14	0.00	0.35	0.18	0.18	0.11	0.13	0.00
vowel	0.99	0.00	0.75	0.03	0.72	0.03	0.73	0.04
weather	0.17	0.13	0.37	0.08	0.36	0.00	0.36	0.00
zoo	0.20	0.02	0.36	0.20	0.12	0.03	0.12	0.05

As shown in Table 4, GauED scores the highest accuracy in 10 data sets, MEED scores the highest in 6 data sets, and the *k*-means scores the highest in 4 data sets. The higher accuracy rate of GauED than MEED can be explained by the use of the *F-stat* metric in the experiments. We have reasons to believe that the accuracy rate of MEED can be further improved by using appropriate classification metrics [43]. From the above preliminary empirical experiments, it can be concluded that the MEED's ability in global optimization is evident from the encouraging results on small and medium data sets.

8. Conclusion

This paper presents a global optimization technique called the *Adaptive Maximum-Entropy Estimated Distribution* (Adaptive MEED) model. By focusing on the model parameter learning, we extended the existing dimension reduction techniques and proposed an adaptive approach to effectively reduce the problem dimension while maintaining the search process. The potential dimension reduction is much larger than those of the existing one-dimensional approaches. The ergodic properties of the proposed model are discussed through the Markov chain analysis. The preliminary empirical results for the clustering problems show that the proposed model outperforms genetic algorithms on a number of small and medium data sets.

Acknowledgement

The authors would like to thank anonymous reviewers for their insightful comments and helpful suggestions.

References

[1] R. Agrawal, R. Srikant, Fast algorithm for mining association rules, Proceedings of VLDB Conference, September 1994, pp. 487–499.
[2] M.Z. Ashrafi, D. Taniar, K.A. Smith, A new approach of eliminating redundant association rules, in: Database and Expert Systems Applications, Lecture Notes in Computer Science, vol. 3180, Springer-Verlag, 2004, pp. 65–474.
[3] M.Z. Ashrafi, D. Taniar, K.A. Smith, An efficient compression technique for frequent itemset generation in association rule mining, in: Advances in Knowledge Discovery and Data Mining, PAKDD 2005, Lecture Notes in Computer Science, vol. 3518, Springer-Verlag, 2005, pp. 125–135.
[4] M.Z. Ashrafi, D. Taniar, K.A. Smith, Redundant association rules reduction techniques, in: AI 2006: Advances in Artificial Intelligence, Lecture Notes in Computer Science, vol. 3809, Springer-Verlag, 2005, pp. 254–263.
[5] S. Baluja, S. Davies, Using optimal dependency tree for combinatorial optimization: learning the structure of search space, Technical Report No. CMU-CS-97-107, Carnegie Mellon University, Pittsburgh, PA, 1997.
[6] S. Baluja, Population based incremental learning: a method for integrating genetic search based function optimization and competitive learning, Technical Report No. CMU-CS-94-163, Carnegie Mellon University, Pittsburgh, PA, 1994.
[7] E. Bengoetxea, P. Larrañaga, I. Bloch, A. Perchant, C. Boeres, Learning and simulation of Bayesian networks applied to inexact graph matching, Pattern Recognition 35 (12) (2002) 2867–2880.
[8] P.A.N. Bosman, D. Thierens, An algorithmic framework for density estimation based evolutionary algorithms, Technical Report UU-CS-1999-46, the Department of Information and Computing Sciences, Utrecht University, 1999.
[9] E. Cantú-Paz, Efficient and Accurate Parallel Genetic Algorithms, Kluwer Academic Publishers, Boston, MA, 2000, pp. 68–70.
[10] O. Daly, D. Taniar, Exception rules mining based on negative association rules, in: Computational Science and Applications, Lecture Notes in Computer Science, Part IV, vol. 3046, Springer-Verlag, 2004, pp. 543–552.
[11] J. Darroch, D. Ratcliff, Generalized iterative scaling for log-linear models, Annals of Mathematical Statistics 43 (1972) 1470–1480.
[12] J.S. De Bonet, C.L. Isbell, P. Viola, MIMIC: Finding optima by estimating probability densities, in: M. Mozer, M. Jordan, T. Petsche (Eds.), Advances in Neural Information Processing Systems, vol. 9, MIT Press, Cambridge, MA, 1997, pp. 424–431.
[13] S. Della Pietra, V. Della Pietra, J. Lafferty, Inducing features of random fields, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (4) (1997) 380–393.
[14] R.O. Duda, P.E. Hart, Pattern Classification and Scene Analysis, John Wiley & Sons, New York, NY, 1973.
[15] A.A. Freitas, Data Mining and Knowledge Discovery with Evolutionary Algorithms, Springer-Verlag, Berlin, 2002, pp. 108–109.
[16] D.E. Goldberg, P. Segrest, Finite Markov chain analysis of genetic algorithms, in: J. Grefenstette (Ed.), Genetic Algorithms and Their Applications: Proceedings of the Second International Conference on Genetic Algorithms, Lawrence Erlbaum Associates, Hillsdale, NJ, 1987, pp. 1–8.
[17] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley Longman Publishing Co., Boston, MA, 1989.
[18] J. Goodman, Classes for fast maximum entropy training, in: Proceedings of International Conference on Acoustics, Speech, and Signal Processing, IEEE CS Press, 2001, pp. 557–560.

[19] J. Goodman, Sequential conditional generalized iterative scaling, in: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics, Philadelphia, PA, July 2002, pp. 9–16.
[20] G.R. Harik, Linkage learning via probabilistic modelling in the ECGA, Illegal Report No. 99010, Illinois Genetic Algorithm Laboratory, University of Illinois, Urbana, IL, 1999.
[21] G.R. Harik, F.G. Lobo, D.E. Goldberg, The Compact genetic algorithm, in: Proceedings of IEEE Conference on Evolutionary Computation, IEEE Press, 1998, pp. 523–528.
[22] F.S. Hillier, G.J. Lieberman, Introduction to Operations Research, seventh ed., McGraw-Hill, Boston, MA, 2001.
[23] Y. Huang, H. Xiong, W. Wu, P. Deng, Z. Zhang, Mining maximal hyperclique pattern: a hybrid search strategy, Information Sciences 177 (3) (2007) 703–721.
[24] I. Inza, P. Larrañaga, B. Sierra, Estimation of distribution algorithms for feature subset selection in large dimensionality domains, in: H.A. Abbas, R.A. Sarker, C.S. Newton (Eds.), Data Mining: A Heuristic Approach, Idea Group Publishing, Hershey, PA, 2002, pp. 97–116.
[25] M. Iosifescu, Finite Markov Processes and Their Applications, John Wiley, New York, NY, 1980.
[26] A.K. Jain, R.P.W. Duin, J. Mao, Statistical pattern recognition: a review, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (1) (2000) 4–37.
[27] E.T. Jaynes, Information theory and statistical mechanics, Physical Reviews 106 (1957) 620–630.
[28] J.N. Kapur, H.K. Kesavan, Entropy Optimization Principles with Applications, Academic Press, Boston, MA, 1992.
[29] P. Larrañaga, A review on estimation of distribution algorithms, in: P. Larrañaga, J.A. Lozano (Eds.), Estimation of Distribution Algorithms: A New Tool for Evolutionary Optimization, Kluwer Academic Publishers, Boston, MA, 2001, pp. 57–100.
[30] P. Larrañaga, R. Etxeberria, J.A. Lozano, J.M. Peña, Combinatorial optimization by learning and simulation of Bayesian networks, in: The Proceeding of the Sixteenth Conference on Uncertainty in Artificial Intelligence, Standford, 2000, pp. 343–352.
[31] P. Larrañaga, R. Etxeberria, J.A. Lozano, J.M. Peña, Optimization by learning and simulation of Bayesian and Gaussian networks, Technical Report KZZA-IK-4-99, Department of Computer Science and Artificial Intelligence, University of the Basque Country, 1999.
[32] P. Larrañaga, R. Etxeberria, J.A. Lozano, J.M. Peña, Optimization in continuous domain by learning and simulation of Gaussian networks, in: The Proceeding of the 2000 Genetic and Evolutionary Computation Conference Workshop Program, Las Vegas, Nevada, 2000, pp. 201–204.
[33] R. Malouf, A Comparison of algorithms for maximum entropy parameter estimation, in: Proceedings of Sixth Conference on Natural Language Learning, 2002, pp. 49–55.
[34] H. Mühlenbein, T. Mahnig, The Factorized distribution algorithm for additively decomposed functions, in: Proceedings of Congress on Evolutionary Computation, IEEE Press, 1999, pp. 752–759.
[35] H. Mühlenbein, G. Paaß, From combination of genes to the estimation of distributions: binary parameters, in: H.M. Voigt (Ed.), Parallel Problem Solving from Nature – PPSN IV, Lecture Notes in Computer Science, vol. 1411, Springer-Verlag, Berlin, 1996, pp. 178–187.
[36] R.M. Neal, Probabilistic inference using Markov chain Monte Carlo methods, Technical Report CRG-TR-93-1, Department of Computer Science, University of Toronto, 1993.
[37] D.J. Newman, S. Hettich, C.L. Blake, C.J. Merz, UCI Repository of machine learning databases, University of California, Department of Information and Computer Science, Irvine, CA, 1998. <http://www.ics.uci.edu/~mlearn/MLRepository.html>.
[38] A. Nix, M.D. Vose, Modeling genetic algorithms with Markov chains, Annals of Mathematics and Artificial Intelligence 5 (1992) 27–34.
[39] T.K. Paul, H. Iba, Linear and combinatorial optimizations by estimation of distribution algorithms, in: The Proceeding of the 9th MPS Symposium on Evolutionary Computation, Japan, 2002, pp. 99–106.
[40] M. Pelikan, D.E. Goldberg, E. Cantú-Paz, Linkage problem, distribution estimation and Bayesian networks, Evolutionary Computation 8 (3) (2000) 311–340.
[41] G. Rudolph, Convergence analysis of canonical genetic algorithms, IEEE Transactions on Neural Networks 5 (1) (1994) 96–101.
[42] L. Tan, D. Taniar, K.A. Smith, A clustering algorithm based on an estimated distribution model, International Journal of Business Intelligence and Data Mining 1 (2) (2006) 229–245.
[43] L. Tan, D. Taniar, K.A. Smith, Maximum-entropy estimated distribution classification model, International Journal of Hybrid Intelligence Systems 3 (1) (2006) 1–10.
[44] H.C. Tjioe, D. Taniar, Mining association rules in data warehouses, International Journal of Data Warehousing and Mining 1 (3) (2005) 28–62.
[45] Y-J. Tsay, Y-W. Chang-Chien, An efficient cluster and decomposition algorithm for mining association rules, Information Sciences 160 (1–4) (2004) 161–171.
[46] J. Wang, X. Wu, C. Zhang, Support vector machines based on *K*-means clustering for real-time business intelligence systems, International Journal of Business Intelligence and Data Mining 1 (1) (2005) 54–64.
[47] A. Wright, R. Poli, C. Stephens, W.B. Landgon, S. Pulavarty, An estimation of distribution algorithm based on maximum entropy, in: Proceedings of GECCO 2004, Lecture Notes in Computer Science, LNCS, vol. 3103, Springer-Verlag, 2004, pp. 343–354.
[48] L. Yan, D.J. Miller, General statistical inference for discrete and mixed spaces by an approximate application of the maximum entropy principle, IEEE Transactions on Neural Networks 11 (3) (2000) 558–573.
[49] S.X. Yu, P. Scheunders, Feature selection for high-dimensional remote sensing data by maximum entropy principle based optimization, Proceedings of Geoscience & Remote Sensing Symposium, vol. 7, IEEE Press, 2001, pp. 3303–3305.