Real-coded Crossover as a Role of Kernel Density Estimation

Jun Sakuma
Tokyo Institute of Technology
4259, G5-21, Nagatsuta-cho
Midori-ku, Yokohama-city, Kanagawa, Japan
jun@fe.dis.titech.ac.jp

Shigenobu Kobayashi
Tokyo Institute of Technology
4259, G5-21, Nagatsuta-cho
Midori-ku, Yokohama-city, Kanagawa, Japan
kobayasi@dis.titech.ac.jp

ABSTRACT

This paper presents a kernel density estimation method by means of real-coded crossovers. Estimation of density algorithms (EDAs) are evolutionary optimization techniques, which determine the sampling strategy by means of a parametric probabilistic density function estimated from the population. Real-coded Genetic Algorithm (RCGA) does not explicitly estimate any probabilistic distribution, however, the probabilistic model of the population is implicitly estimated by crossovers and the sampling strategy is determined by this implicit probabilistic model. Based on this understanding, we propose a novel density estimation algorithm by using crossovers as nonparametric kernels and apply this kernel density estimation to the Gaussian Mixture modeling. We show that the proposed method is superior in the robustness of the computation and in the accuracy of the estimation by the comparison of conventional EM estimation.

Categories and Subject Descriptors

F.2.1 [**Theory of Computation**]: Analysis of Algorithms and Problem Complexity numerical algorithms and problems

General Terms

Algorithms

Keywords

Real-coded GA, crossover, kernel density estimation, Gaussian Mixture Model, Expectation Maximization

1. INTRODUCTION

In applying Genetic Algorithms to function optimization in the continuous search space, Real-coded Genetic Algorithms (RCGAs) which use the real-number vector representation have been proposed and reported to show better performance than bit-string GAs[4][10]. In the background of the development of RCGAs, great efforts have been made not only in the improvement of the algorithms but also in the establishment of guidelines for designing real-coded crossovers.

In [6], a guideline for the design of real-coded crossover, "Preservation of statistics" has been proposed. This guideline means that the distribution of offspring generated by crossovers should preserve the statistics such as the mean vector and the covariance matrix of the parental population. Several real-coded crossovers have been proposed following this guideline and the validity thereof has been verified[6][5]. This guideline regards that a real-coded crossover is an operator which maps the probability distribution of parents to that of offspring.

As another evolutionary optimization techniques considering the probabilistic distribution of population, Estimation of Distribution Algorithms (EDAs) have also been developed. In EDAs, the probabilistic distribution is explicitly estimated by various methods and the sampling strategy is determined by the distribution. In this paper, we focus on the fact that the search framework of EDAs are mainly designed by the following two steps, (1) the explicit estimation of a parametric or semi-parametric probabilistic density distribution, (2) the sampling from the estimated distribution. To the contrary, the search algorithm of RCGAs is mainly designed by the following two steps, (1) the implicit estimation of non-parametric probabilistic density distribution, (2) the sampling from the estimated distribution.

In comparison of the two evolutionary algorithms, they are totally designed under a similar principal. Although great attentions have ever been paid only to the search performance of algorithms in conventional studies, we focus on the estimation performance of real-coded crossovers in this paper. Then, we show that real-coded crossovers satisfies the following facts:

- Real-coded crossovers can be interpreted as a nonparametric kernel density estimator (KDE)
- If a crossover satisfies the preservation of statistics, the KDE using the crossover corresponds to the approximation of the maximum likelihood estimation of Gaussian distribution

Then we show that KDE can be actually designed from crossover Unimodal Normal Distribution Crossover (UNDX-m)[6] and Simplex Crossover (SPX) [5].

GECCO'05, June 25–29, 2005, Washington, DC, USA.

In [13], a KDE using Constructive Elliptical Basis Function (CEBF) have been introduced and an EM algorithm using CEBF have been proposed. We show that CEBF and a kernel based on UNDX are geometrically equivalent in this paper. Then, we propose an EM algorithm using KDE based on crossovers for the Gaussian Mixture Model (GMM) by generalizing conventional EM algorithm.

Experiments show that EM estimation using KDE of crossovers is superior in robustness and the estimation accuracy. Add to this, it performs better than conventional EM in the sense that the estimation result does not tend to be captured in local optima. The effectiveness of EM using crossovers is shown by experiments of artificial data modeling problem and letter recognition problem.

RCGAs are widely used for large-scale applications and the performance thereof is superior even in the optimization of high dimension function. Though RCGAs do not estimate the density model in the procedure, it is considered that the performance of implicit density estimation does not degrades even in high dimensionality.

Based on the viewpoint shown above, our motivation in this paper is to show the applicability of RCGAs not only to optimization but also to density estimation. The probabilistic density estimation is often needed in the field of data analytics such as classification algorithms, clustering techniques and data compression methods.

Another motivation is to show an unified framework for RCGAs and EDAs based on the understanding that the probabilistic model estimated by real-coded crossovers is approximately corresponds to the maximum likelihood estimation of Gaussian model in a nonparametric manner. This framework contributes to bridge a gap between RCGAs and EDAs in continuous optimization.

In section 2, we clarify the role of real-coded crossovers by comparing the difference of the sampling strategy of RCGAs and EDAs in terms of probabilistic density function. Then, we show that real-coded crossovers can be interpreted as a kernel density estimator. In section 3, we explain algorithms of nonparametric kernels using two real-coded crossovers, UNDX-m and SPX. Section 4 describes a procedure of EM algorithm for the Gaussian Mixture Model using KDE of crossovers. Section 5 is experiments using benchmark problems and a letter recognition problem. Section 6 concludes this paper.

2. UNIFIED UNDERSTANDING OF RCGA AND EDA

2.1 Guidelines for Real-coded Crossovers

In [6], a guideline for designing real-coded crossovers that offspring generated by crossover operators should preserve the statistics such as the mean vector and the covariance matrix of the parental population. This guideline is called *Preservation of statistics*. This is because if the offspring generated by a crossover distribute narrower than the parents, it may let the optimum escape. To the contrary, if the offspring generated by a crossover distribute wider, it wastes computation time in searching hopeless region. Hence, sampling new points in the region where the parents reside will be an appropriate choice. Here, the population of RCGA is regarded as a probabilistic density function (pdf).

Actually, the population of GA does not always cover the interesting region, therefore, mutations or exploration-directed search operators[12] are optionally used in order to make the performance robustly. [1] has also mentioned about this point. However, in this paper, we regard real-coded crossovers as sampling operators that satisfy the preservation of statistics and do not add any extrapolative search bias to the offspring distribution.

2.2 The Search Framework of EDA

EDA is evolutionary algorithms that generate offspring randomly following a pdf[7]. Although many variants of EDAs have been proposed not only for the continuous optimization problem but the combinatorial optimization problem, we only focus on continuous EDAs in this paper. A basic framework of continuous EDA for cost minimization is shown in [9].

Let $P_\theta(\boldsymbol{x})$ be a pdf where the density is uniform in the region such that $f(\boldsymbol{x}) < \theta$ and the density is zero in the other region. Initial distribution is normally a uniform distribution over the whole search space. Then, the procedure of EDA is described as follows:

[EDA]

1. Generate sample set $X_{\theta(0)}$ drawn from the initial distribution $\hat{P}_{\theta(0)}(\boldsymbol{x})$
2. Estimate pdf $\hat{P}_{\theta(t)}(\boldsymbol{x})$ from sample set $X_{\theta(t)}$
3. Generate sample set $Z_{\theta(t)}$ drawn from $\hat{P}_{\theta(t)}(\boldsymbol{x})$
4. Choose new sample set $X_{\theta(t+1)}(\theta(t+1) < \theta(t))$ from sample set $X_{\theta(t)}$ and $Z_{\theta(t)}$
5. $t \leftarrow t+1$ and jump to 2.

Here, $\hat{P}_{\theta(t)}(\boldsymbol{x})$ is an estimation of $P_{\theta(t)}(\boldsymbol{x})$. When the estimation of $\hat{P}_{\theta(t)}(\boldsymbol{x})$ is accurate in all steps, then the optimum $\boldsymbol{x}^*$ can be obtained from $\hat{P}_{\theta(t)}(\boldsymbol{x})$. That is, when $\boldsymbol{z} \sim \hat{P}_{\theta(t)}$ and $t \to \infty$, then $\boldsymbol{z} \to \boldsymbol{x}^*$. Although many variants of EDAs using different estimation methods of $P_{\theta(t)}(\boldsymbol{x})$ and different generation alternation methods of new population $X_{\theta(t+1)}$ have been proposed, BEA(Bayesian Evolutionary Algorithm)[16], BOA(Bayesian Optimization Algorithm) [11] and BGA(Breeder Genetic Algorithm)[9] have basically close framework.

In the step 2, the probabilistic model $\hat{P}_{\theta(t)}(\boldsymbol{x})$ that decides the sampling strategy of the next generation is estimated from the current population $X_{\theta(t)}$. This means that $\hat{P}_{\theta(t)}(\boldsymbol{x})$ should be estimated such that Z and X distributes in the similar region under the assumption that the current population X distributes in the promising region to search for optimum. This search strategy of $\hat{P}_{\theta(t)}(\boldsymbol{x})$ is almost the same as the preservation of statistics shown in previous section.

2.3 The Search Strategy of RCGA

In this section, we compare RCGAs and EDAs in terms of the estimation of the pdf and clarify the role of crossovers. The procedure of RCGAs is described as follows:

[RCGA]

1. Generate parental population(initial population)X_0(t=0)
2. Generate offspring Z_t from the parental population X_t(crossover)
3. Choose new population X_{t+1} from X_t and Z_t(generation alternation)
4. $t \leftarrow t+1$ and jump to 2.

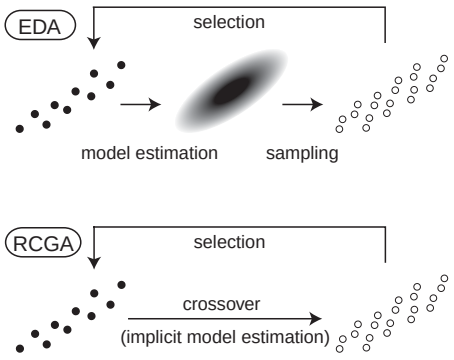

Figure 1: Probabilistic density estimation of EDA and RCGA.

Here, we can see that both RCGAs and EDAs have similar framework, however, the manner of model estimation is different.

EDAs use parametric and semi-parametric models as P, for example, Gaussian distribution[7] or Latent variable models[2] and so on.. On the other hand, RCGA does not explicitly estimate any probabilistic model in the procedure and the offspring Z is generated directly from X by using crossovers. However, from the viewpoint of the preservation of statistics, the sampling strategy of crossovers is determined considering a pdf. This indicate that algorithm of crossovers implicitly include the estimation of a pdf. Fig. 1 shows the conceptual difference between RCGAs and EDAs.

2.4 Probabilistic Density Estimation of Crossover

In this section, the implicit estimation of probabilistic density functions by crossovers are formulated and we show that crossovers are interpreted as a kind of nonparametric kernel density estimation(KDE).

Given a finite number of data points $X = \{\boldsymbol{x} \mid \boldsymbol{x} \in \boldsymbol{R}^d\}$ drawn from an unknown probabilistic density function $P(\boldsymbol{x})$, we consider an estimation problem of an underlying $P(\boldsymbol{x})$. Suppose that $X^l = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_m\}$ is a set of parents used for a crossover chosen randomly from X. m is the number of the parent. We regard the generation of offspring by a crossover C as sampling from a pdf. Then the crossover is formulated in the form,

$$\boldsymbol{z} \sim C(\boldsymbol{x}; X^l), \tag{1}$$

where C is a pdf which is characterized by parents X^l and $\boldsymbol{z}$ is an offspring drawn from C. The set of offspring Z is generated by the iteration of the sampling from C. When we assume that Z is drawn from a pdf, the pdf is described as a mixture model of C as follows:

$$\boldsymbol{z} \sim \frac{\sum_{l=1}^{K} C(\boldsymbol{x}; X^l)}{K}, \tag{2}$$

where K is the number of the iteration of the crossover C. When C satisfies the preservation of statistics, the following equations are valid.

$$\langle \boldsymbol{x} \rangle = \langle \boldsymbol{z} \rangle \tag{3}$$

$$\langle (\boldsymbol{x} - \langle \boldsymbol{x} \rangle)^T (\boldsymbol{x} - \langle \boldsymbol{x} \rangle) \rangle = \langle (\boldsymbol{z} - \langle \boldsymbol{z} \rangle)^T (\boldsymbol{z} - \langle \boldsymbol{z} \rangle) \rangle \tag{4}$$

RHS of (2) can be considered as a kind of kernel density estimation(KDE). In KDE, the data density is estimated by overlapping of large number of kernels whose location and the form is different dependent on the given data point. For instance, Radial Basis Function (RBF) is defined as a normal distribution with a fixed variance and its mean vector is set at the location of each data point in the data set. When we regard the crossover C as a kind of kernel, the pdf in eq. (2) can be considered as KDE from data set X. We call the pdf of crossovers C call *crossover kernel*.

The estimation accuracy of the KDE is normally measured by mean squared error (MSE) between the estimation density and the true density. MSE is normally minimized by some iterative procedure. To the contrary, KDE by crossover kernel preserves the mean vector and the covariance matrix of data set X as shown in eq. 3 and 4 instead of the minimization of MSE without any iteration procedure. In this sense, KDE by crossover estimate the underlying distribution $P(\boldsymbol{x})$ as an approximation of the maximum likelihood estimation of Gaussian distribution. Please notice that the estimated density can be obtained by the model but the model parameters, that is, the mean vector and the covariance matrix, cannot be obtained from the model.

The discussion in this section is mentioned about the optimization on continuous domain, however, the same discussion is also applicable to the combinatorial optimization.

3. CROSSOVER AS KERNEL DENSITY ESTIMATOR

In this section, we show how to design crossover kernels based on conventional crossovers, UNDX-m and SPX.

3.1 UNDX Kernel

Firstly, the algorithm of UNDX-m is shown briefly. Let the parents be $X^l = \{\boldsymbol{x}_1, ..., \boldsymbol{x}_{m+2}\}$. We call X^l *kernel construction set*. Let the center of parental vectors $\boldsymbol{x}^1, ..., \boldsymbol{x}^{m+1}$ be $\boldsymbol{p}$, the difference vector of $\boldsymbol{x}^i$ and $\boldsymbol{p}$ be $\boldsymbol{d}^i = \boldsymbol{x}^i - \boldsymbol{p}$. Let D be the length of the component of $\boldsymbol{d}^{m+2}$. Let $\boldsymbol{e}^1, ..., \boldsymbol{e}^{n-m}$ be an orthogonal bases of the subspace orthogonal to the subspace spanned by $\boldsymbol{d}^1, ..., \boldsymbol{d}^m$. Then, offspring vector $\boldsymbol{x}^c$ is generated as follows:

$$\boldsymbol{z} = \boldsymbol{p} + \sum_{i=1}^{m} w_i \boldsymbol{d}^i + \sum_{i=1}^{n-m} v_i D \boldsymbol{e}^i, \tag{5}$$

where w_i, v_i is a random number drawn from a normal distribution $N(0, \sigma_\xi^2), N(0, \sigma_\eta^2)$ respectively. Fig. 2 shows the distribution of offspring generated by UNDX-m. When $m = n$, the 3rd term in eq. 5 is zero and rewritten in the form

$$\boldsymbol{z} = \boldsymbol{p} + \sum_{i=1}^{n} w_i \boldsymbol{d}^i. \tag{6}$$

When $\sigma_\xi = \alpha/\sqrt{m}, \alpha = 1$, it is theoretical proved that the mean vector and the covariance is preserved[6].

Conversely, we formulate the pdf of UNDX C_{UNDX}. A basis transform matrix D^l of kernel construction set X^l is defined as follows:

$$\boldsymbol{D}^l = \left(\frac{\boldsymbol{d}^1}{h|\boldsymbol{d}^1|^2}, ..., \frac{\boldsymbol{d}^d}{h|\boldsymbol{d}^d|^2}\right). \tag{7}$$

Here, h is called a smoothing parameter. Then, the crossover kernel of UNDX is described as follows:

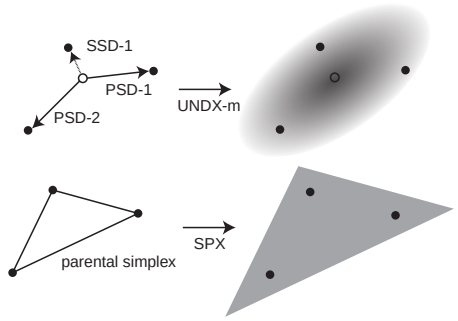

Figure 2: The density distribution of crossover UNDX-m and SPX.

$$C_{\mathrm{UNDX}}(\boldsymbol{x}; X^l) = \frac{1}{2\pi^{d/2}} \times \exp\{-\frac{(\boldsymbol{D}^{l^{-1}}(\boldsymbol{x}-\boldsymbol{p}))^T(\boldsymbol{D}^{l^{-1}}(\boldsymbol{x}-\boldsymbol{p}))}{2}\}. \quad (8)$$

We call eq. (8) *UNDX kernel*. In data set X, the probabilistic model estimated by UNDX kernel is written in the form

$$\tilde{P}_{\mathrm{UNDX}}(\boldsymbol{x}; X) = \frac{\sum_l C_{\mathrm{UNDX}}(\boldsymbol{x}; X^l)}{K}. \quad (9)$$

When the center is computed as the center of the population $\boldsymbol{\mu} = \sum_i \boldsymbol{x}^i/N$ instead of the center of the parents $\boldsymbol{p}$, the crossover is written in the form,

$$\boldsymbol{z} = \boldsymbol{\mu} + \sum_{i=1}^{n} w_i \boldsymbol{d}^i. \quad (10)$$

The preservation of the statistics is similarly shown in this crossover and this probabilistic density function is written in the form

$$C_{\mathrm{CEBF}}(\boldsymbol{x}; X^l, \boldsymbol{\mu}) = \frac{1}{2\pi^{d/2}} \exp\{-\frac{(\boldsymbol{D}^{l^{-1}}(\boldsymbol{x}-\boldsymbol{\mu}))^T(\boldsymbol{D}^{l^{-1}}(\boldsymbol{x}-\boldsymbol{\mu}))}{2}\}, \quad (11)$$

This pdf is proposed as a kernel called Constructive Elliptical Basis Function (CEBF)[13]. The estimation model from X by CEBFs is written in the form,

$$\tilde{P}_{\mathrm{CEBF}}(\boldsymbol{x}; X) = \frac{\sum_l C_{\mathrm{CEBF}}(\boldsymbol{x}; X^l)}{K}. \quad (12)$$

On C_{UNDX} and C_{CEBF}, the following theorem is valid.

Theorem 1 : *Let X^l be a kernel construction set, which is chosen uniform randomly from X. Let Z be a sample set generated from $\tilde{P}(\boldsymbol{x})$, which is estimated by C_{UNDX} or C_{CEBF}. When $h = \alpha/\sqrt{m}$ and $\alpha = 1$, $\boldsymbol{z} \in Z$ holds the following statistical characters,*

$$\langle \boldsymbol{x} \rangle = \langle \boldsymbol{z} \rangle \quad (13)$$

$$\langle (\boldsymbol{x} - \langle \boldsymbol{x} \rangle)(\boldsymbol{x} - \langle \boldsymbol{x} \rangle)^T \rangle = \langle (\boldsymbol{z} - \langle \boldsymbol{z} \rangle)(\boldsymbol{z} - \langle \boldsymbol{z} \rangle)^T \rangle, \quad (14)$$

The proof is shown in appendix and this is obtained by a slight modification shown in [6]. By setting at $h = \alpha/\sqrt{m}, \alpha = 1$, X and Z have the same statistics. Please notice that $\tilde{P}(\boldsymbol{x})$ is estimated without estimating the mean vector and the covariance matrix of X.

Usually, the smoothing parameter h must be estimated by some iterative procedure in KDE, however, h is automatically obtained in crossover kernel due to the preservation of statistics.

$C_{\mathrm{UNDX}}(\boldsymbol{x}; X^l)$ and $C_{\mathrm{CEBF}}(\boldsymbol{x}; X^l)$ are sorts of a normal distribution kernel. As a normal distribution kernel function, Radial Basis Function(RBF) is well known, which is written in the form,

$$C_{\mathrm{RBF}}(\boldsymbol{x}; \boldsymbol{x}^i) = \frac{1}{2\pi^{d/2}} \times \exp\{-\frac{(\boldsymbol{x}-\boldsymbol{x}^i)^T(\boldsymbol{x}-\boldsymbol{x}^i)}{2}\}. \quad (15)$$

Then, estimation by RBFs written in the form

$$\tilde{P}_{\mathrm{RBF}}(\boldsymbol{x}; \boldsymbol{x}_i) = \frac{\sum_i C_{\mathrm{RBF}}(\boldsymbol{x}; \boldsymbol{x}^i)}{K}, \quad (16)$$

where $\tilde{P}_{\mathrm{RBF}}(\boldsymbol{x}; X)$ preserves the mean vector of X but does not the covariance matrix. In this sense, $C_{\mathrm{UNDX}}(\boldsymbol{x}; X^i)$ and $C_{\mathrm{CEBF}}(\boldsymbol{x}; X^i)$ is interpreted as an extension of RBF to preserve the covariance matrix. Fig. 3 shows conceptual illustrations KDE of UNDX kernel, CEBF and RBF.

3.2 SPX Kernel

Next, we design a kernel based on crossover SPX. The offspring of SPX are drawn from a uniform distribution in the simplex which is spanned by all $\boldsymbol{d}^i$. Then, SPX kernel is written in the form,

$$C_{\mathrm{SPX}}(\boldsymbol{x}; X) = \begin{cases} 1/V_S(\epsilon) & \text{if } x \in S\ , \\ 0 & \text{otherwise,} \end{cases} \quad (17)$$

where S is the simplex and $V_S(\epsilon)$ is the volume of S. See [5] in detail.

When the center is computed as the center of the population $\boldsymbol{\mu} = \sum_i x_i/N$ instead of the center of the parents $\boldsymbol{p}$, this kernel also preserves the statistics of X. In this paper, we call this kernel *Constructive Simplex Basis Function(CSBF)*.

On data set X, the probabilistic model is similarly estimated as well as eq. (9) and (12). By setting smoothing parameter of SPX at $\epsilon = \sqrt{d+2}$ in C_{SPX} and C_{CSBF}, it is also proven that X and Z have approximately the same statistics by using the same technique shown in the theorem 1 and [5].

$C_{\mathrm{SPX}}(\boldsymbol{x}; X^i)$ and $C_{\mathrm{CSBF}}(\boldsymbol{x}; X^i)$ is sorts of a uniform kernel based on data points. As a kernel based on uniform distribution, uniform kernel is written in the form,

$$C_{\mathrm{Uni}}(\boldsymbol{x}; \boldsymbol{x}_i) = \begin{cases} 1/h^d & \text{if } \boldsymbol{x} \in [x_i - h/2, x_i + h/2]^d \\ 0 & \text{otherwise.} \end{cases} \quad (18)$$

Nonparametric model of X estimated by uniform kernel is represented in the form

$$\tilde{P}_{\mathrm{Uni}}(\boldsymbol{x}; X) = \frac{\sum_i C_{\mathrm{Uni}}(\boldsymbol{x}; \boldsymbol{x}^i)}{K}, \quad (19)$$

where $\tilde{P}_{\mathrm{Uni}}(\boldsymbol{x}; X)$ preserves the mean vector of X but does not the covariance matrix. In this sense, $C_{\mathrm{SPX}}(\boldsymbol{x}; X^i)$ and $C_{\mathrm{CSBF}}(\boldsymbol{x}; X^i)$ is interpreted as an extension of the uniform kernel to preserve covariance matrices.

In this paper, we call crossovers whose basis vectors are computed as the difference vectors between the parent and the center of the parent *parent centric crossovers*. To the Contrary, crossovers whose basis vectors are computed as the difference vectors between the parent and the center of the population *population centric crossovers*.

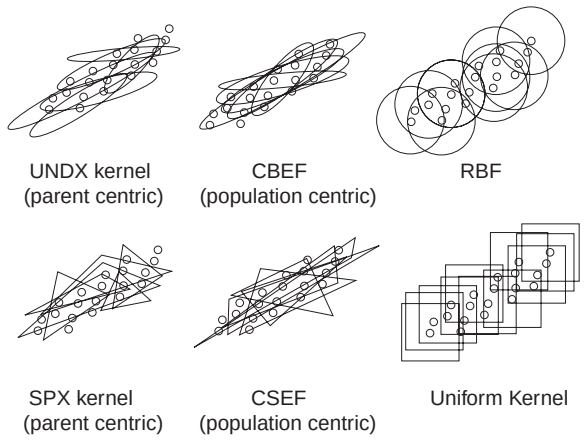

Figure 3: Density estimation by crossover kernels and conventional kernels.

The nonparametric estimation of $P(\boldsymbol{x})$ by a kernel $C(\boldsymbol{x}; X^i)$, is not a Gaussian distribution but has the same statistics as $P(\boldsymbol{x})$. Strictly speaking, this fact does not necessarily means that $\tilde{P}(\boldsymbol{x})$ is an approximation of $P(\boldsymbol{x})$, however, $\tilde{P}(\boldsymbol{x})$ well preserves the statistical character of $P(\boldsymbol{x})$. Therefore, it is expected that $\tilde{P}(\boldsymbol{x})$ is an approximation of $P(\boldsymbol{x})$ as Gaussian.

4. MIXTURE MODELING USING CROSSOVER KERNELS

4.1 Our Viewpoint

In previous section, we explain that the design of crossover kernels and clarify that the nonparametric KDE using crossover kernels is a kind of an approximation of Gaussian distribution while the model parameters does not have to be computed explicitly. The benefit to use crossover kernels for the estimation of Gaussian model is not large because maximum likelihood estimator of the mean vector and the covariance matrix is easily obtained even in high dimension.

On the other hand, in the estimation of Gaussian Mixture Model (GMM), crossover kernels contribute to the robust estimation. GMM is a probabilistic model that consists of a linear combination of Gaussian distributions.

Expectation Maximization (EM) algorithm is often utilized for the estimation of GMMs[3]. EM algorithms estimate the probabilistic model from incomplete data set, which includes unobservable variables. EM estimation in high dimensional GMM suffers from the instability of the computation. In addition, though EM algorithm theoretically assures the local convergence property, the landscape of the likelihood has many local optima and the estimation result is likely to be captured in one of them[14]. EM estimation based on crossover kernels is expected to be stable in computation and to be able to escape from local optima even in high dimensionality because the explicit parameter estimation is not required.

In this section, we show that the procedure of an EM algorithm using crossover kernels.

4.2 The Gaussian Mixture Model and the EM algorithm

We consider a problem of modeling a probabilistic density function, $P(\boldsymbol{x})$ from X again. Gaussian Mixture Model (GMM) is described in the form,

$$P(\boldsymbol{x}; \Theta) = \sum_{i=1}^{k} \alpha^i n^i(\boldsymbol{x}; \boldsymbol{\mu}^i, \boldsymbol{\Sigma}^i), \tag{20}$$

where n^i is a multivariate Gaussian distribution with mean vector $\boldsymbol{\mu}^i$ and covariance matrix $\boldsymbol{\Sigma}^i$, k denotes the number of mixed component, α^i denotes a weight parameter of each components and $\boldsymbol{\Theta} = (\alpha^i, \boldsymbol{\mu}^i, \boldsymbol{\Sigma}^i)_{i=1}^k$ denotes the model parameter.

The EM algorithm is generally used to determine the model parameters $\boldsymbol{\Theta}$ from X. Let $w(i|\boldsymbol{x}_j)(i = 1, ..., k, j = 1, ..., N)$ be class conditional probabilities with which the j-th data x_j is generated from the i-th component , $n^i(\boldsymbol{x}_j; \boldsymbol{\mu}^i, \boldsymbol{\Sigma}^i)$.

Estimation of the mixing parameter α^i, the mean vector $\boldsymbol{\mu}^i$ and the covariance matrix $\boldsymbol{\Sigma}^i$ is carried out through iterations of sequential parameter updating by the following equations for each component i. Through these operations, the log likelihood of the model $\sum_{j=1}^{N} \log P(X|\boldsymbol{\Theta})$ converges to the local maximum[15].

$$w(i|\boldsymbol{x}_j) = \frac{\hat{\alpha}^i n^i(\boldsymbol{x}_j|\hat{\boldsymbol{\mu}}^i, \hat{\boldsymbol{\Sigma}}^i)}{\sum_{l=1}^{k} \hat{\alpha}^l n^l(\boldsymbol{x}^i|\hat{\boldsymbol{\mu}}_l, \hat{\boldsymbol{\Sigma}}_l)} \tag{21}$$

$$\hat{\alpha^i} = \frac{1}{N} \sum_{j=1}^{N} w(i|\boldsymbol{x}_j) \tag{22}$$

$$\hat{\boldsymbol{\mu}^i} = \frac{\sum_{j=1}^{N} w(i|\boldsymbol{x}_j)\boldsymbol{x}_j}{\sum_{j=1}^{N} w(i|\boldsymbol{x}_j)} \tag{23}$$

$$\hat{\boldsymbol{\Sigma}^i} = \frac{\sum_{j=1}^{N} w(i|\boldsymbol{x}_j)(\boldsymbol{x}_j - \boldsymbol{\mu}^i)^T(\boldsymbol{x}_j - \boldsymbol{\mu}^i)}{\sum_{j=1}^{N} w(i|\boldsymbol{x}_j)}. \tag{24}$$

EM estimation in high dimensional GMM suffers from the instability of the computation mainly because of the computation of the inverse of ill-conditioned weighted covariance and the divergence of the likelihood.

4.3 EM Based on Crossover Kernels

By using crossover kernels instead of maximum likelihood estimation shown in eq. (23) and (24), the parameters $\hat{\boldsymbol{\mu}}_i, \hat{\boldsymbol{\Sigma}}_i$ do not have to be estimated explicitly and the density of the Gaussian is computed directly. However, please notice that the mean vector and the covariance matrix estimated here is weighted estimator. Unfortunately, parent centric crossovers cannot be used for the estimation of weighted mean vector and the covariance matrix, however, following theorem is valid for population centric crossovers.

Theorem 2 : *Let $w_j(j = 1, ..., N)$ be a weight parameter which is assigned to each data $\boldsymbol{x}_j(j = 1, ..., N)$. Let $X_w^l(i = 1, ..., s)$ be a kernel construction set, which is chosen randomly from X with the probability proportional to w_j. Let Z be a sample set generated from $\tilde{P}(\boldsymbol{x})$, which is estimated by population centric crossover kernels, CEBF or CSBF constructed from X_w^l. When smoothing parameter is set to satisfy the preservation of statistics, $\boldsymbol{z} \in Z$ holds following statistical characters,*

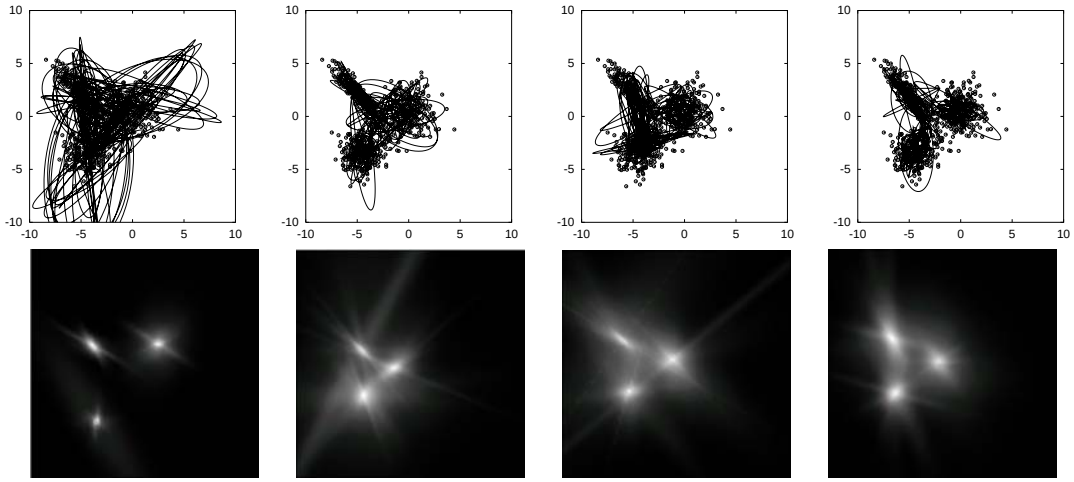

Figure 4: Top: the shape of CEBFs, bottom: the contour of the estimated density function ($d = 2, k = 3$, from left to right $t = 1, 5, 10, 25$).

$$\langle \boldsymbol{x} \rangle_w = \langle \boldsymbol{z} \rangle \tag{25}$$

$$\langle (\boldsymbol{x} - \langle \boldsymbol{x} \rangle_w)(\boldsymbol{x} - \langle \boldsymbol{x} \rangle_w)^T \rangle_w = \langle (\boldsymbol{z} - \langle \boldsymbol{z} \rangle)(\boldsymbol{z} - \langle \boldsymbol{z} \rangle)^T \rangle, \tag{26}$$

where $\langle \cdot \rangle_w$ stands for weighted expectation operation of w, that is, $\langle \boldsymbol{x} \rangle_w = \sum_{j=1}^N w_j \boldsymbol{x}^j / \sum_{j=1}^N w_j$. The proof of this theorem is shown in appendix.

Now, we can compute the density $n^i(\boldsymbol{x}|\hat{\boldsymbol{\mu}}^i, \hat{\boldsymbol{\Sigma}}^i)$ with a weighted mean vector and a weighted covariance by crossover kernels without computing the weighted covariance. By choosing kernel construction sets X^l randomly from X with the probability proportional to w_j, the weighted density estimation is performed by crossover kernels as follows:

$$\tilde{P}(\boldsymbol{x}) = \frac{\sum_l C(\boldsymbol{x}|X_w^l, \boldsymbol{\mu})}{K}. \tag{27}$$

Then, the update equations of EM algorithm using crossover kernels can be modified as follows:

$$\tilde{n}^i(\boldsymbol{x}_j) = \frac{\sum_{l=1}^s C(\boldsymbol{x}|X^l, \boldsymbol{\mu}^i)}{s}, \tag{28}$$

$$w(i|\boldsymbol{x}_j) = \frac{\hat{\alpha}^i \tilde{n}^i(\boldsymbol{x}_j)}{\sum_{l=1}^k \hat{\alpha}^l \tilde{n}^l(\boldsymbol{x}_j)}, \tag{29}$$

$$\hat{\alpha^i} = \frac{1}{N} \sum_{j=1}^N w(i|\boldsymbol{x}_j), \tag{30}$$

$$\hat{\boldsymbol{\mu}^i} = \frac{\sum_{j=1}^N w(i|\boldsymbol{x}_j)\boldsymbol{x}_j}{\sum_{j=1}^N w(i|\boldsymbol{x}_j)}. \tag{31}$$

From these equations, EM algorithm using crossover kernels is designed as follows:

[EM based on KDE]

1. Initialize $w^{(0)}(i|\boldsymbol{x}_j)$ randomly ($t = 0$).
2. Estimate $\hat{\alpha}^l$ by Eq. 30.
3. Estimate $\boldsymbol{\mu} = \sum_i w(i|\boldsymbol{x}_j)\boldsymbol{x}_j/N$ by Eq. 31..
4. Choose X_w^i randomly with the probability proportional to $w^{(t)}(i|\boldsymbol{x}_j)$.
5. Construct crossover kernel $C^i(\boldsymbol{x}|\boldsymbol{\mu}, X_w^i)$ and obtain density $\tilde{n}^i(\boldsymbol{x}_j)$ by Eq. 28.
6. Estimate $w^{(t)}(i|\boldsymbol{x}_j)$ by Eq. 29.
7. If terminate conditions are satisfied, then terminate. Else $t = t + 1$ and jump to step 2.

Although the local optimal convergence is assured theoretically in EM, escaping from local maxima is impossible since the update is carried out in a deterministic manner. Conversely, in EM using crossover kernels, it is possible to escape from local maxima taking account of the perturbation of $\tilde{n}^i(\boldsymbol{x}_j)$ because of the randomness in selecting kernel construction set.

If update in Eq. 29 is carried out on every iteration, the estimation results vibrates because the kernel construction set is chosen randomly from the data set. Therefore, by bounding the difference of $w^{(t)}(i|\boldsymbol{x}_j)$, the fluctuation of $w^{(t)}(i|\boldsymbol{x}_j)$ is decreased and the estimation converges stably.

Fig. 4 show the shapes of CEBFs and the contours of the estimated density function in the estimation of EM using CEBF. As the iteration number grows, the region CEBFs cover separates to each Gaussian data cluster. The number of kernel used in this experiment is 90 and the iterations are terminated at $t = 30$.

5. EXPERIMENTS

In this section, we compare the estimation results of conventional EM and EM using crossover kernels for Gaussian Mixtures. Experiment 1 is the estimation of Gaussian mixture from artificially generated data sets, where the estimation accuracy and the computation time are compared. Experiment 2 is a letter recognition problem. EM algorithms estimate the classification model as Gaussian mixtures and the training error and the test error are compared. In both experiments, CEBF is used as a crossover kernel.

5.1 Experiment 1 : Artificial Dataset

The data set is generated as follows. The mean vector $\boldsymbol{\mu}^i$ and the covariance matrix $\boldsymbol{\Sigma}^i$ of each component is chosen randomly satisfying the following inequality

$$|\boldsymbol{\mu}^i - \boldsymbol{\mu}^j| < c\sqrt{n}\max\{\sqrt{\mathrm{trace}\boldsymbol{\Sigma}^i}, \sqrt{\mathrm{trace}\boldsymbol{\Sigma}^j}\}, \qquad (32)$$

for all $i, j (i \neq j)$. Parameter c determines the degree of overlapping of each components. $10d$ data points per one component were generated. For instance, 10-dimension 3-component problem holds 300 data points. The number of components k is given in advance for all algorithms. In EM using crossover kernel, K, the number of overlapped kernels, is set at $30k$ by preliminary experiments.

We compare conventional EM and EM using crossover kernel (we call this crossover kernel EM) in terms of the estimation accuracy and the time to convergence. Here, the estimation accuracy is the difference between the log likelihood of the true model and the estimated model. Since the probabilistic model obtained by crossover kernel EM is a nonparametric one, we convert the model to a Gaussian mixture model by computing the weighted mean vectors and weighted covariance matrices from $\hat{w}(i|\boldsymbol{x}_j)$ of the last iteration. Then we compare the log likelihood. Please notice that crossover kernel EM does not require the construction of the GMM and computation of the likelihood on every update. On each trial, the Gaussian mixture satisfying Eq.32($c = 3.0$) is newly generated randomly. Both algorithms are implemented by GNU-C++ and the experiments are carried out on Pentium3-1Ghz.

In this experiment, the number of component is $k = 2, 4$ and the dimension is $d = 10, 20, 30, 40$. Figure 5 left and right show the estimation accuracy and the computation time, respectively.

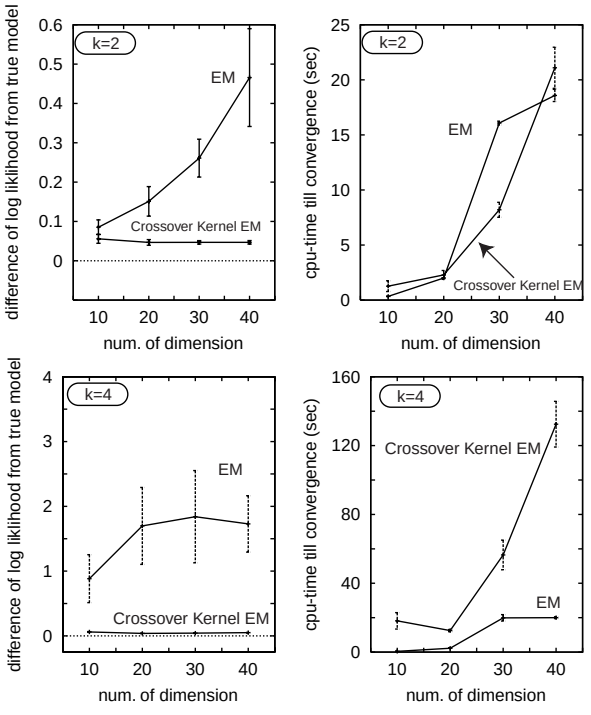

Figure 5: Left: The difference between the log likelihood of the true model and the estimation model, right: CPU time till convergence. The data is the average over 20 iterations and the error bars show the standard deviation/3. d=10,20,30,40, k=2(top),4(bottom).

Method	training err.	test err.
Crossover Kernel EM($k = 3$)	0.035	0.059
Crossover Kernel EM($k = 4$)	0.024	**0.051**
Crossover Kernel EM($k = 5$)	**0.016**	**0.042**
EM($k = 3$)	0.033	0.059
EM($k = 4$)	0.023	0.053
EM($k = 5$)	**0.015**	**0.046**
ALLOC80	0.065	0.064
k-nearest neighbor	**0.000**	0.068
Kohonen Net	0.057	0.079
Quadratic Discriminant	0.101	0.113
CN2	0.021	0.115
C4.5	0.042	0.132
Neural Net	0.323	0.327

Table 1: Training error and test error of each learning method in letter-recognition problem. The bold letter is the best three learning method.

On both cases($k = 2, 4$), the estimation accuracy by crossover kernel EM is almost zero in high dimensionality. On the other hand, it degrades as the dimensionality increases in EM. This is because many local optima arise in high dimensionality and EM tends to be captured in them. Crossover kernel EM can escape from the local maxima utilizing the randomness of the selection of the kernel construction set.

The computation complexity is $O(Nd^3)$ in EM and $O(KNd^3)$ in crossover kernel EM per iteration. In $k = 2$, since crossover kernel EM converges in small number of iteration, the computation time is eventually almost the same. In $k = 4$, EM converges much faster than crossover kernel EM though the estimation results are not good.

5.2 Experiment 2: Letter Recognition

Letter-recognition benchmark problem (letter) in UCI Machine Learning Repository is applied to the proposal method. The data vectors are 16 numerical attributes which is scaled and discretised into $[0, 15]$. The detail can be seen in [8]. The objective is to predict the alphabet (26 capital letters) from the data vector. The classifier is learned using 15000 labeled data out of 20000. The training error is measured using the 15000 data and the test error is measured using the rest 5000 data without label.

Data models are estimated as Gaussian mixture from data having the same label, that is, 26 models are trained from "A " to "Z". For instance, let a data model of "A" be $P("A"|\boldsymbol{x})$. The label which gives the maximum density in $P("A"|\boldsymbol{x}), ..., P("Z"|\boldsymbol{x})$ is the output of the classifier.

For comparison, the training and test error of k-nearest neighbor, quadratic discriminant, Kohonen networks and learning vector quantizers, CN2, ALLOC80, C4.5, neural networks learning with back propagation(NN) is also shown in the same experimental conditions[8]. Results show that the both EM and crossover kernel EM show better classification accuracy than traditional classifier and the test error of crossover kernel EM is slightly better than EM.

6. CONCLUSION

We propose a kernel function CEBF that approximate Gaussian distribution in a nonparametric manner and construct crossover kernel EM that estimate the Gaussian Mix-

ture. The appealing point of our proposal is that (1) the estimation results can escape form the local optima using the randomness in the selection of kernel construction set, (2) the procedure does not include the unstable computation and it gives a robust estimation result even in high dimension. In density modeling using the EM, the dimensionality is at most 10 in tradition. However, the dimensionality of the real world problem often reaches more than hundreds. Applying the proposal method to real-world application is also our future work.

The disadvantage of our algorithm is that the convergence property to the local optimum is not assured theoretically. Though we use a stopping condition heuristically, the time to convergence is highly dependent on the parameter, therefore, the improvement of the algorithm that theoretically assures local optimal convergence is our future work.

Our method can be applied to the optimization using the same framework of EDA in a straightforward manner. In the optimization of the multiple-peak function, the population forms several clusters and our method is expected to be the efficient sampler on these kinds of problem.

7. REFERENCES

[1] Bayer, H.-G. and Deb, K., On the Desired Behaviors of Self-adaptive Evolutionary Algorithm, *Proc. of PPSN VI*, pp. 59-68 (2000).

[2] Cho, D. Y. and Zhang, B. T., Continuous Estimation of Distribution Algorithms with Probabilistic Principal Component Analysis, *Proc. of Congress on Evolutionary Computation*, pp. 521-526, (2001).

[3] Depmster, A. P., Laird, N.M. and Rubin, D. B. , Maximum likelihood from incomplete data via the EM algorithm, *J. Roy. Statist. Soc. B*, 39:1-38 (1977).

[4] Eshelman, L. J. and Schaffer, J. D., Real-coded Genetic Algorithm and Interval Schemata, *Foundation of Genetic Algorithms 2*, pp. 375-382, (1993).

[5] Higuchi, T., Tsutsui, S., and Yamamura, M, Theoretical analysis of simplex crossover for real-coded Genetic Algorithms, *Proc. of PPSN VI*, pp. 365-374 (2000).

[6] Kita, H., Ono, I. and Kobayashi, S. (1999), Multi-parental Extension of the Unimodal Normal Distribution Crossover for Real-coded Genetic Algorithms, *Proc. of CEC1999*, pp. 1581-1587(1999) .

[7] Larrañaga, R. Etxberria, J. A. Lozano and J. M. Peña, A review of cooperation between evolutionary computation and probabilistic graphical models, *Proc. of the Second Symposium on Artificial Intelligence*, CIMAF 99, pp.314-324, (1999).

[8] D. Michie, D. J. Spiegelhalter and Taylor, C. C., Machine Learning, Neural and Statistical Classification, (1994).

[9] H. Mühlhenbein, J. Bendisch, H.-M. Voigt, From recombination of genes to the estimation of distributions, *Proc. of PPSN IV*, pp.178-187 (1996).

[10] I. Ono and S. Kobayashi, A Real Coded Genetic Algorithm for Function Optimization Using Unimodal Normal Distributed Crossover , *Proc. 7th ICGA*, pp246-253 (1997).

[11] M. Pelikan, D. E. Goldberg and F. Lobo, A Survey of Optimization by Building and Using Probablistic Models, *Computational Optimization and Applications*, Kluwer, (2000).

[12] Sakuma, J. and Kobayashi, S., Extrapolation-Directed Crossover for Real-coded GA: Overcoming Deceptive Phenomena by Extrapolative Search, *Proc of Congress on Evolutionary Computation*, pp. 655-662 (2001).

[13] Sakuma, J. and Kobayashi, S., Non-parametric Expectation-Maximization for Gaussian Mixtures, *Proc of Int'l Conf. on Neural Information Processing*, pp. 517-522. (2002).

[14] Ueda, N., Nakano, R., Ghahramani, Z. and Hinton, G. E. (2000), SMEM Algorithm for Mixture Models. *Advances in Neural Information Processing Systems 12*.

[15] Verbeek, J. J., Vlassis, N. and Krose, B. J. A., Efficient Greedy Learning of Gaussian Mixture Models, *Neural Computation* 15(2), pp. 469-485, (2003).

[16] Zhang, B-T., Paaβ, G. and Mühlenbein, H., Convergence Properties of Incremental Bayesian Evolutionary Algorithms with Single Markov Chains, *Proc. of CEC2000*, vol. 2, pp. 938-945, (2000).

Appendix : Proof of Theorem 1 and 2

We show the proof of theorem 1 of CEBF. Proof of CSBF is omitted here. When a sample $z \in Z$ is generated from CEBF, z can be expressed as follows:

$$\boldsymbol{z}^l \sim \boldsymbol{\mu} + \sum_{i=1}^{d} \frac{1}{h} \xi \boldsymbol{d}^{l,i}, \tag{33}$$

where ξ stands for the univariate normal distribution whose mean parameter is zero and variance parameter is 1. Here, the data included in each kernel construction set X^l is chosen randomly from X, so we can consider the expectation according to X^l as the expectation according to X. As for the mean vector of $\boldsymbol{z}$, we obtain

$$\begin{aligned}
\langle \boldsymbol{z}^l \rangle_{X^l} &= \langle \boldsymbol{z} \rangle \\
&= \boldsymbol{\mu} + \langle \sum_{i=1}^{d} \frac{1}{h} \xi \boldsymbol{d}^i \rangle \\
&= \boldsymbol{\mu} + \langle \sum_{i=1}^{d} \frac{1}{h} \xi (\boldsymbol{x}^i - \boldsymbol{\mu}) \rangle = \boldsymbol{\mu} = \langle \boldsymbol{x} \rangle.
\end{aligned} \tag{34}$$

As for covariance matrix $\boldsymbol{\Sigma}$ of $\boldsymbol{z}$, we obtain

$$\begin{aligned}
\boldsymbol{\Sigma} &= \langle (\boldsymbol{z}^l - \boldsymbol{\mu})(\boldsymbol{z}^l - \boldsymbol{\mu})^T \rangle_{X^l} \\
&= \langle (\boldsymbol{z} - \boldsymbol{\mu})(\boldsymbol{z} - \boldsymbol{\mu})^T \rangle \\
&= \langle \boldsymbol{z}\boldsymbol{z}^T \rangle - \boldsymbol{\mu}\boldsymbol{\mu}^T \\
&= \frac{1}{h^2} \sum_{k=1}^{n} \langle \xi^2 (\boldsymbol{x}^k - \boldsymbol{\mu})(\boldsymbol{x}^k - \boldsymbol{\mu})^T \rangle \\
&= n \frac{1}{h^2} \langle \xi^2 \boldsymbol{x}^k (\boldsymbol{x}^k)^T \rangle - 2n \frac{1}{h^2} \langle \xi^2 \boldsymbol{\mu} \boldsymbol{x}^k \rangle + n \frac{1}{h^2} \langle \xi^2 \boldsymbol{\mu}\boldsymbol{\mu}^T \rangle \\
&= n \frac{1}{h^2} \langle \ \boldsymbol{x}\boldsymbol{x}^T \rangle - n \frac{1}{h^2} \boldsymbol{\mu}\boldsymbol{\mu}^T \\
&= n \frac{1}{h^2} \langle (\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^T \rangle
\end{aligned} \tag{35}$$

Therefore, when $h = \sqrt{n}$,

$$\begin{aligned}
\langle (\boldsymbol{z} - \boldsymbol{\mu})(\boldsymbol{z} - \boldsymbol{\mu})^T \rangle &= \langle (\boldsymbol{z} - \langle \boldsymbol{z} \rangle)(\boldsymbol{z} - \langle \boldsymbol{z} \rangle)^T \rangle \\
&= \langle (\boldsymbol{x} - \boldsymbol{\mu})(\boldsymbol{x} - \boldsymbol{\mu})^T \rangle \\
&= \langle (\boldsymbol{x} - \langle \boldsymbol{x} \rangle)(\boldsymbol{x} - \langle \boldsymbol{x} \rangle)^T \rangle. \ Q.E.D.
\end{aligned}$$

In UNDX kernel, because the parents are chosen independently, the expectation of the parent center is the expectation of the population. Therefore, the preservation of the mean vector is proved similarly. About the covariance, the proof is the same in CEBF. By extending the expectation operation $\langle \cdot \rangle$ to $\langle \cdot \rangle_w$, theorem 2 is easily proved. Please notice that this is not valid for parent centric crossovers in theorem 2.