Intelligent Information Management, **2011, 3, 87-103**
doi:10.4236/iim.2011.33011 Published Online May 2011 (http://www.SciRP.org/journal/iim)

Supervised Fuzzy Mixture of Local Feature Models

Mingyang Xu, Michael Golay
Massachusetts Institute of Technology, Cambridge, USA
E-mail: *xumy@alum.mit.edu, golay@mit.edu*
Received January 17, 2011; revised April 7, 2011; accepted April 10, 2011

Abstract

This paper addresses an important issue in model combination, that is, model locality. Since usually a global linear model is unable to reflect nonlinearity and to characterize local features, especially in a complex system, we propose a mixture of local feature models to overcome these weaknesses. The basic idea is to split the entire input space into operating domains, and a recently developed feature-based model combination method is applied to build local models for each region. To realize this idea, three steps are required, which include clustering, local modeling and model combination, governed by a single objective function. An adaptive fuzzy parametric clustering algorithm is proposed to divide the whole input space into operating regimes, local feature models are created in each individual region by applying a recently developed feature-based model combination method, and finally they are combined into a single mixture model. Correspondingly, a three-stage procedure is designed to optimize the complete objective function, which is actually a hybrid Genetic Algorithm (GA). Our simulation results show that the adaptive fuzzy mixture of local feature models turns out to be superior to global models.

Keywords: Adaptive Fuzzy Mixture, Supervised Clustering, Local Feature Model, PCA, ICA, Phase Transition, Fuzzy Parametric Clustering, Real-Coded Genetic Algorithm

1. Introduction

Models are useful in both explaining systems and predicting their future behavior. Usually, for a specific system there are many different competing models. In this situation, the question arises of how to improve model performance given all available information. One strategy is to select a single best model among the group of competing models. An alternative is to combine multiple competing models. In both theoretical and empirical work, it has been shown that model combination can lead to better models than does selecting a single model. However, how to aggregate information contained in candidate models and new observations in an efficient way is still an open research problem. To this end, Xu and Golay [1] proposed a feature-based model combination method, which, compared to other methods mentioned above, makes more efficient use of information embedded in candidate models and extra data.

In this feature-based model combination approach, it is assumed the true model can be expressed as

$$M_T(x) = \sum_{i=1}^{N} w_i(x)\, f_i(x) \tag{1}$$

where $f_i(x) \in F$, the factor set, and $w_i(x)$ is its corresponding weight, intensity, or factor loading as in factor analysis [2], N is the number of total factors, x is an input variable. In a linear case, $w_i(x)$ is constant, independent of x. Correspondingly, a candidate model, as an approximate representation of the system, can be expressed in a similar way, for example, the kth candidate model

$$M_k(x) = \sum_{i=1}^{N_k} w_{ki}(x)\, f_{ki}(x) \tag{2}$$

where $f_{ki}(x) \in F_k$ with F_k the set of factors for kth model and N_k is the number of factors in F_k. Note that F_k is a subset of feature space F.

Assuming the features $f_i(x)$, $i = 1, \cdots, N$, are either uncorrelated or independent conditional on input x, [1] proposes to apply either Principal Component Analysis (PCA) [3] or Independent Component Analysis (ICA) [4] to extract the features and then aggregates them through regression into a new optimal composite model based on data

$$M(x) = \alpha + \sum_{i \in S_c} w_i f_i(x) \tag{3}$$

where S_c denotes the set of selected factors, and the factor weights and constant α can be determined based upon data.

However, there are some weaknesses inherent in this approach that limit its application. Among them two major issues are the model locality and nonlinearity. As shown later it can be unjustified to use a single global linear model to describe a complex system over its entire domain. In fact, this limitation is inherent in the application of the global linear PCA or ICA for feature extraction, whose weaknesses have been investigated by Karunen and Malaroiu [5] and other authors.

In order to mitigate these limitations, we propose a mixture model of local feature models. Basically, local models are built to approximate the complex system within operating regimes and then are combined by smooth interpolation into a complete global model. The split of the input space enable us to characterize the nonlinearity of the system although somewhat coarsely, and local component analysis produces different sets of local features, which lead to different local models.

The outline of this paper is as follows: in section 2, the idea of mixture of local models is proposed and justified in terms of different arguments; an adaptive fuzzy parametric clustering algorithm is presented in order to split the entire input space into sub-domains; and then local PCA or ICA is used to extract local features, which constitute local models; finally, a method is proposed to piece local models together into a mixture model. In section 3, a three-stage optimization algorithm is used to implement the procedure. Both an artificial example and a physical case are presented in section 4 to demonstrate the performance of this new approach. The last section summarizes the paper.

2. Mixture of Local Models

In this section, we introduce a mixture of local models in order to characterize model nonlinearity and locality. Local models are built to approximate the complex system locally and then are combined by smooth interpolation into a complete global model. In order to describe model locality, we introduce the concept of phase transition.

2.1. Why Local Models

In general, it is usually preferred to build a single global model to describe a system's behavior over its entire input space. However, in reality a model might not be able to cover the full range of the input space accurately due to high complexity. This can happen because of the need to describe the interactions between a large number of phenomena that appear within the domain. Rather a well-defined model may only be appropriate over a specific prescribed subspace, namely its operating regime. For example, a model, which is fitted quite well to the data in a specific region, may be inaccurate when extrapolated to other regions, for example, because some assumption underlying a model can only be met in a certain range of inputs. Typically for a complicated system the true model is nonlinear and highly complex and a global linear model is far from adequate.

It is also conceivable that a physical system may undergo different phase changes that are governed by different underlying laws. This phenomenon is regarded as phase transition. Meanwhile, the set of important features of the system can vary over different domains or the same set of features but with varying influences. Both cases result in varied local behavior patterns. Correspondingly, this situation can lead to variation of model structures, that is, different models over different regions. Thus, in the presence of phase transition, it can be difficult to incorporate the properties of distinct phases into a single global model.

Therefore, a single global linear model sometimes cannot describe a nonlinear system adequately or capture all the local features accurately. In order to deal with this nonlinearity and locality, it might be possible to identify missing hidden variables needed to characterize nonlinearity or phase transition phenomena and create a comprehensive complex global model. Such variables are called Arrhenius-type terms by Johansen and Foss [6]. However, it is usually difficult to find such extra hidden variables, not only because it requires greater knowledge concerning the system under investigation, but also because even if we do so, we may obtain an overly complicated global model or even an intractable one. Following the philosophy of divide-and-conquer, an alternative simpler way to capture the locality is to divide the whole input space into several small regions and to perform local analysis in each local region, for example, classification and regression tree (CART) and use of a hierarchical mixture of experts (HME). Local analysis leads to simple local models, which try to characterize a complicated physical system over a specific regime called an operating regime. Then local models are combined into a global composite model. In contrast to a global model that is valid over the full range of the input space, a local model is valid only in a predefined operating region smaller than the input space. Typically, local models can be considerably simpler because a smaller number of phenomena are relevant and their interactions are simpler [6]. Therefore, this divide-and-conquer principle simplifies the modeling problem by transforming the task of modeling a complex system into one of simpler modeling local results can be combined relatively easily to yield a satisfactory overall model.

Typically, in local learning models are constructed as linear functions of local features. Then, they are pieced

together somehow to form a global linear mixture model [7]. Although simpler, this mixture model improves the model accuracy because it reduces model bias by specifying the model more properly. By examining the bias/variance tradeoff for local and global learning, Murray-Smith and Johansen [8] show that local learning can be viewed as a simple form of regularization, which produces models with higher accuracy and greater robustness than do global learning methods.

However, as noted by Jordan and Jacobs [9], divide-and-conquer tends to increase the model variance. A remedy to it is to employ a soft split of the input space. A simple version of soft partition is to overlap the operating regimes of the local models. Doing this help smooth the switching between local models and thereby can reduce variance. This will be discussed in more details as we proceed.

It is noteworthy that local models here are different from those used in local modeling [10] where a parametric function is fitted to data in the neighborhood around a query point x. This can be done by locally weighted regression [11] where the weights in Weighted Least Squares (WLS) depend upon the distance from a data point to the query point x, or by mixtures of local experts [12] where local experts are fitted to all data but not equally well in some local regions. A common drawback of these local learning methods is the complete lack of interpretability of the resultant models.

2.1.1. Phase Transition

Usually, local models are combined into a global model by smooth superposition. The main motivation for this is that the system often has some smoothness properties between regions, *i.e.* with the operating point changing the phenomena or behavior changes smoothly. However, one may occasionally come across processes that change abruptly. For example, in fluid dynamics a phase transition or flow pattern changes can occur [13]. Below a mixture-of-phases model is introduced to characterize a phase transition. This leads to use of a mixture of overlapping local models.

In general, phase transition means a system undergoes a discontinuous change in association with continuously changing parameters, transforming from one phase to another. For example, a liquid flow changes from laminar flow to turbulent flow as the Ronald number increases. In the current case, by phase transition we mean a system undergoes a qualitative change in its underlying local features, thereby leading to the need for different local models.

According to the modern classification scheme, phase transitions fall into two broad categories, namely the first and second-order phase transitions. Under this scheme, phase transitions were labeled by the lowest derivatives of the free energy that is discontinuous at the transition. First-order transitions exhibit a discontinuity in the first derivative of the free energy, or the value of the response variable. In contrast, second-order transitions are continuous in the value of the response variable but not in the second derivative of the free energy.

In this scheme, first-order transitions are associated with "mixed-phase regimes", where some parts of the system have completed the transition and others have not. A typical example of this class of transitions is water boiling where with the temperature increasing the water does not instantly turn into a gas but forms a turbulent mixture of water and water vapor. Mixed-phase systems are unstable and difficult to study, because their dynamics are violent and hard to control. However, many important phase transitions fall in this category, including the solid/liquid/gas transitions.

On parallel with soft partition, in the present case we can adopt soft phase boundaries, where multiple phases coexist. This is consistent with overlapping operating regime concept mentioned earlier. In the coexistence region, the system can be considered to be a random mixture of multiple phases governed by different local models. This stochastic mixture model of phase transitions helps explain instability within the transitional regime, because inside this coexistence region the system behaves like one of the distinct phases with specific probabilities and where distinct phases are quite different in behavior. Therefore, the expected behavior of the system is simply a probability-weigthed average of those of distinct phases. At the sametime, outside the coexitence regime the system is dominated by a single phase. Thus a local model can be applied deterministically. This statistical mixture model of phase transition is not short of physical evidence. For example, Harrington et al. [14] reported in liquid-liquid phase transition, the coexistence of two different phases inside an unstable region.

In addition, generally we have no idea in advance where and how wide the transitonal regions. Thus their domains have to be estiamted based upon data. Conveniently, each transitional region can be represented by two parameters, the outset and the end-point of a phase transition region. In fact, this parametric model is able to characterize both categories of phase transitions, namely first-order or second-order transition. If the width of the coexistence regime turns out to be nil, it is a first-order phase transition; otherwise, it is a second-order phase transition.

Based upon the above argument, the framework of mixture of local models is able to model the phase transition phenomena via dynamic identification of operating regimes.

2.2. Adaptive Fuzzy Parametric Clustering

In our local model scheme, the input space is split into partitions, which overlap, reflecting the effect of a coexistence region. Each partition corresponds to a distinct operating regime, in which the system can be described by a local model. In implementing this scheme, the first and a key problem is that of how to split the input space. This is actually a problem of clustering. For this purpose, we propose a supervised clustering algorithm.

Clustering can be considered the most important unsupervised learning problem, which is intended to organized objects into groups whose members are similar in some way. Clustering involves the task of dividing data points into homogeneous classes or clusters so that items in the same class are as similar as possible and items in different classes are as dissimilar as possible. Thus, the goal of clustering is to find common patterns or similarity.

The measure of similarity plays an essential role in clustering algorithms. Usually, distance is employed as the similarity criterion. However, this seems inappropriate for the current situation. In our scheme, if two points in the input space belong to the same phase, they are thought of behaving similarly and hence as being in the same cluster. From the perspective of local features, points in the same cluster have the same underlying local features. Furthermore, since currently local features are extracted from candidate models through either PCA or ICA, similarity in local features is equivalent to similarity in the separating matrix [15]. Therefore, it is more reasonable to cluster data based on the similarity of the mixing matrix W, which is equal to the inverse of the separating matrix. Because of the similarity of points in the same cluster, it is reasonable to treat any cluster independent in the course of feature analysis and building local models.

In trying to meet different needs, many clustering algorithms have been proposed. They can be roughly grouped into two classes, namely hard clustering and soft clustering.

Hard clustering assumes exclusive assignment of each datum to respective clusters, which means that a specific datum belonging to a definite cluster cannot be included in another cluster simultaneously. So, hard clustering results in crisp clusters, where each data point belongs to exactly one cluster. An example of this class is the K-means clustering algorithm. Its application is mainly in pure local piecewise models such as CART [16].

On the contrary the soft clustering, also called overlapping clustering, allows each point to belong to two or more clusters simultaneously. Using the two existing uncertain reasoning techniques, fuzzy set theory and probability theory, this class of clustering algorithms can be further divided into fuzzy clustering and probabilistic clustering. In fuzzy clustering, fuzzy clusters are identified and the data points can belong to more than one cluster associated with membership grades, which indicate the degree to which the data points belong to the different clusters. The fuzzy c-means algorithm is one of the most widely used fuzzy clustering algorithms, which is developed by Dunn [17] and improved by Bezdek [18].

Similar to the fuzzy clustering, in probabilistic clustering each data point has specific probability of belonging to a particular cluster. Use of probabilistic reasoning is implied by availability of only a restricted amount of evidence. A well used probabilistic clustering algorithm is the mixture of Gaussians, where the well-known Expectation-Maximization algorithm is applied to estimate parameters.

As pointed out in [9], the divide-and-conquer technique tends to increase the variance. A simple remedy to this problem is to apply soft partition. This can be done in the current case; and thus, we favor soft clustering. Another fact making soft clustering appealing is that many systems change behaviors smoothly as a function of inputs and soft transition between regimes introduced by the fuzzy set representation characterizes this feature in an elegant fashion. However, both fuzzy clustering and probabilistic clustering cannot be applied directly, because fundamentally they measure the similarity based on distance and are separated from the modeling process, which is inappropriate in our case. Meanwhile, in comparison to probabilistic algorithms, fuzzy clustering is more natural and flexible in the current case. Consequently, we propose a new adaptive fuzzy clustering algorithm below to identify different phases over the entire input space. The characteristics of the new fuzzy clustering algorithm are described in detail in the following discussion.

1) Fuzzy clustering

A natural way to interpret overlapping operating regimes is to apply fuzzy set theory, where an operating point can fall into two or more operating regime simultaneously. According to our scheme, in the overlapping regions multiple local models might be relevant while outside the coexistence regions only one local model, called the dominant local model, is valid. The simple trapezoidal fuzzy membership function is specifically suitable for characterizing such operating regimes. Furthermore, the choice of a trapezoidal shape membership function produces more interpretable local models than other functions like Gaussians [15].

In order to be consistent with the concept of mixture modeling and superposition, a constraint on the fuzzy membership functions is imposed, which requires that at

any point x, $\sum_{m=1}^{M}\mu_m(x)=1$, where $\mu_m(x)$ is membership degree of the **mth** cluster. This results in smooth transition between operating regimes.

Clusters or operating regimes are represented by fuzzy sets. Typical fuzzy clustering with three overlapping operating regimes is depicted in Figure 1, where [a,b] represents the overlapping region between the cluster I and II and likewise [c, d] represents the overlapping region between the cluster II and III. Any point in the input space can belong to multiple clusters with simulta- neous memberships. From another practical angle, the membership can be interpreted as describing how possi- ble an observation can be generated by a specific local model.

This type of member functions is consistent with the nature of phase transition and able to model both classes of phase transition. It also keeps the interpretability of each local model corresponding to one cluster. Thus, we can argue that this type of member function is a natural choice for physical models.

2) Supervised clustering

In our fuzzy clustering scheme, the fuzzy membership functions are parameterized by the number of clusters used and the locations of the splitting points.

As usual, the main task of fuzzy clustering is to identify fuzzy sets characterized by parametric fuzzy membership functions. For each cluster, the parameters include the location of boundaries and their widths. The locations of the boundaries can be chosen such that the similarity within a cluster is maximized, while the patterns of different clusters should be as dissimilar as possible. Only so, in each cluster can the system be better represented by a local model and the bias decreased.

The overlap or the width of coexistence region plays a major role in smoothing the transition between local models. [8] further argues that overlap has a regularizing effect in the ill conditioning in a learning problem and the level of overlap determines the amount of regularization. A high level of overlap leads to a high level of correlation between neighboring local models and decreased transparency of the local models, *i.e.* compatibility with the understanding of a system [6]. A low level of overlap results in an abrupt transition between local models. Hence, the optimal degree of overlap and softness depends upon the modeling problem through the objective function.

This algorithm is called adaptive in the sense that in addition to the separators the number of regions is determined based upon the data. If the number of clusters is not large enough, the nonlinearity of the system cannot be described adequately. On the other hand, an increasing number of operating regimes increase the model complexity. The overall effect of an increasing number of local models depends upon whether the decrease in bias is more significant than the increase in variance.

Thus, the number, location and overlap of the operating regimes should be so tuned dynamically as to reach optimal values. These are determined by objective functions. In so doing, it can be ensured that there are adequate amount of data within each operating regime to get a good local model.

3) Single global objective function

The goal of modeling processes is to minimize the predictive error. Likewise, local modeling also aims to minimize the generalization error across the entire input space. Both the splitting of the input space and the building of local models should be determined by this overall goal. Nevertheless, in most previous work such as local PCA [19] or local ICA [20], and use of local models [7], clustering is treated as a separate optimization problem from local learning and obtaining a global mixture. This causes sub-optimality. In this paper, we optimize both problems jointly by incorporating them within a single objective function, which reflects the overall goal of minimizing the global generalization error. This goal can be realized by two steps, namely estimation of clustering parameters given the number of clusters and estimation of the number of clusters. The first step can be done by minimizing the empirical error.

Until now, we have not discussed how to create local models and the global mixture model. Suppose the global mixture model to be $f_g(x, \alpha, \beta)$, where α denotes the clustering parameters, β refers to other local model parameters and x is the model input. Therefore, the partial objective can be expressed as

$$\arg\min_{\alpha,\beta}\sum_{i=1}^{n}\left(y_i - f_g\left(x_i,\alpha,\beta\right)\right)^2 \qquad (4)$$

where (x_i, y_i) denotes a pair of observations. Note that the squared loss is applied, which is equivalent to use of MLE method under the assumption of a normal distribution of the data.

From the partial objective function in Equation (1), it is seen that the input space is so split as to minimize the empirical error. In this sense, this clustering algorithm is supervised.

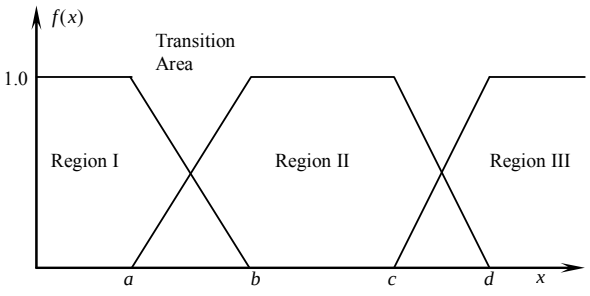

Figure 1. Illustration of the fuzzy clustering concept.

Therefore, through a same objective function fuzzy clustering is closely connected to the modeling process. This is exactly in agreement with [6] that the creation of local models should not be separated from the choice of operating regimes. In this aspect, it is also in spirit similar to Mixture of Experts (MoE), where probabilistic clustering is mixed with learning.

Nevertheless, the partial objective function in Equation (1) does not involve the number of clusters, which is in fact another important part of our fuzzy clustering. This is because a different number of operating regimes lead to varied model structures, which cannot be reflected in the empirical error. Following the principle of parsimony in model selection, we use a complete objecttive function, which incorporates the effect of the number of clusters.

The divide-and-conquer principle reduces the model bias by specifying the model structure more properly, but the variance is increased at the same time, because with an increasing number of local models (increasing model structure) more parameters need to be estimated. This leads to larger variance on the parameter estimate. This phenomenon is well known in the literature of statistics as bias/variance tradeoff (see e.g. [21]). On one hand, if the input space is split into too many regions, overfitting will occur; on the other hand, use of too few regions may not capture the structure in enough detail and thus could lead to underfitting. The task here is to find out the optimal balance point within the bias/variance tradeoff given a finite number of samples. To this end, generally two different strategies can be employed, namely model selection or regularization. In the current case, our purpose is to choose the optimal number of operating regimes, and thus model selection appears more appropriate.

With an increasing number of operating regimes local models can fit the data much better, but the model complexity is also increased, which usually leads to deteriorating generalization. Generally, the higher the model complexity is, the smaller the bias but the larger the variance. Most model selection criteria realize Occam's razor by penalizing the goodness-of-fit with increasing model complexity, thereby minimizing the generalization error. Among all model selection methods, the information theoretical criteria like Akaike's Information Criterion (AIC) [22] or Bayesian Information Criterion (BIC) [23] have a close connection to the maximum likelihood method, which seems to many statisticians an advantage. As a result, these information criteria can be easily applied in many circumstances without any additional computation. In addition, some of these information criteria including AIC and BIC can be justified under a Bayesian framework, which is also viewed by many statisticians as another big advantage (see [22] and [23]). Therefore, in this paper we will only utilize the BIC for the purposes of demonstration.

BIC was first derived by Schwarz in a Bayesian context with a uniform prior probability on each competing model and priors with everywhere positive densities on the model parameters θ in each model. Choosing the model dimensionality with the highest posterior probability leads to the BIC criterion of Schwarz [23],

$$\mathrm{BIC} = -2\log L\left(x\middle|\hat{\theta}\right) + k\log n, \tag{5}$$

where $L\left(x\middle|\hat{\theta}\right)$ is likelihood function of data x at maximum likelihood estimate $\hat{\theta}$, k is the number of parameters or model dimension, and n is the sample size. Note that the first term on the RHS comes directly from the general maximum likelihood and the second term is a complexity penalty term.

Assuming that the errors in data are Gaussian and homogeneous over the operating regimes, we can obtain the BIC formula as

$$\mathrm{BIC} = -n\left[\log\left(n/2\pi\right) - \log RSS - 1\right] + k\log n, \tag{6}$$

where *RSS* is the sum of empirical squared error, *i.e.* $RSS = \sum_{i=1}^{n}\left(y_i - f_g\left(x_i,\alpha,\beta\right)\right)^2$.

In the current situation, where any data point can fall into more than one clusters simultaneously and Gaussian errors are heterogeneous over operating regimes, similar to the Mixture of Gaussian (cf. [9]) the likelihood function can be written as

$$L\left(y,x\middle|\theta\right) = \prod_{i=1}^{M}\prod_{j=1}^{n} L_i\left(y_j,x_j\middle|\theta\right)^{\mu_{ji}}, \tag{7}$$

where M denotes the number of clusters or operating regimes, μ_{ji} refers to the membership of jth data point to ith cluster, and $L_i(\cdot,\cdot|\theta)$ denotes the likelihood function for the local model of ith cluster. Thus, the log likelihood becomes

$$\begin{aligned} l\left(y,x\middle|\theta\right) &= \log L\left(y,x\middle|\theta\right) \\ &= \sum_{i=1}^{M}\sum_{j=1}^{n}\mu_{ji}\log L_i\left(y_j,x_j\middle|\theta\right) \end{aligned} \tag{8}$$

where parameters θ, including β and σ, are estimated by MLE.

Assuming that the local models are linear, *i.e.* $f_i\left(x,\beta_i\right) = \beta_i^{\mathrm{T}}x$, and the errors are Gaussian, *i.e.* $\varepsilon_{ij} \sim N\left(f_i\left(x_j,\beta_i\right),\sigma_i\right)$, the log likelihood for a local model can be expressed as

$$\begin{aligned} & l_i\left(y,x\middle|\theta\right) \\ &= \sum_{j=1}^{n}\mu_{ji}\log L_i\left(y_j,x_j\theta\right) \\ &= \sum_{j=1}^{n}\left(-\mu_{ji}\log\sqrt{2\pi} - \mu_{ji}\log\sigma_i - \frac{\mu_{ji}\left(y_j-\beta_i^{\mathrm{T}}x_j\right)^2}{2\sigma_i^2}\right), \end{aligned} \tag{9}$$

and therefore

$$\hat{\beta}_i = \arg\min_{\beta_i} \sum_{j=1}^{n} \mu_{ji} \left(y_j - \beta_i^{\mathrm{T}} x_j \right)^2 \tag{10}$$

and the variance for the *i*th local model can be estimated as

$$\hat{\sigma}_i^2 = \frac{\sum_{j=1}^{n} \mu_{ji} \left(y_j - \beta_i^{\mathrm{T}} x_j \right)^2}{\sum_{j=1}^{n} \mu_{ji}} = \frac{WRSS_i}{\sum_{j=1}^{n} \mu_{ji}} \tag{11}$$

where the weighted RSS is obtained as

$$WRSS_i = \sum_{j=1}^{n} \mu_{ji} \left(y_j - \beta_i^{\mathrm{T}} x_j \right)^2 .$$

Substituting the likelihood function to the BIC formula produces

$$\mathrm{BIC} = \sum_{i=1}^{M} \left[\sum_{j=1}^{n} \mu_{ji} \cdot \log(2\pi) + \sum_{j=1}^{n} \mu_{ji} \cdot (\log \sigma_i + 1) \right] + k(M) \log n, \tag{12}$$

Finally, we obtain the complete objective function as

$$\arg\min_{M,\alpha,\beta} \sum_{i=1}^{M} \left[(\log(2\pi) + \log \sigma_i + 1) \cdot \sum_{j=1}^{n} \mu_{ji} \right] + k(M) \log n, \tag{13}$$

where M refers to the number of operating regimes and the model dimensionality $k(M)$ is a function of it.

Usually, the penalty term is equal to the number of free parameters that need to be estimated based upon the data. In the current case model parameters include clustering parameters and local model parameters. Therefore, the model complexity can be evaluated by

$$k(M) = 2(M-1) + \sum_{m=1}^{M} p_m, \tag{14}$$

where the first term refers to the number of clustering parameters specifying the boundary positions and the p_i denotes the model dimensionality of each local model. However, with different number of operating regimes, the complexity of the local models does not change in terms of model structure and the number of parameters. Therefore, for the purpose of choosing the number of clusters the appropriate model complexity can be expressed as

$$k(M) = 2(M-1) \tag{15}$$

Note from the above that the purpose of model selection is only to determine the optimal number of operating regimes, because the penalty term only depends upon the number of free parameters rather than upon their values.

By the complete objective function, not only does this - algorithm determine the regime location, size and overlap, but it also determines the number of regimes. The number of clusters depends upon the sample size. Its upper bound should be such that in each cluster the number of data points having non-zero membership should be greater than the number of features. Note that such an objective function favors parsimonious models rather than under or over-parameterized model structures obtained by optimizing the number of local models.

2.3. Local Analysis

Similar to the global feature-based model combination in [1], local analysis include two steps, namely, extracting local features through local component analysis and create local models by aggregation.

2.3.1. Local Component Analysis

ICA is a successful technique for reducing statistical dependence, and hence redundancy, between the candidate models. Dimension reduction is also achieved by eliminating a subset of independent components without significant loss of information.

PCA [3,25] is another effective dimension reduction technique, which only relies on second order statistics and helps remove linear dependencies. As a result, the principal components, although uncorrelated, can be highly statistically dependent. In contrast, ICA takes into account higher order statistics and thus is able to capture non-linear dependency. Therefore, ICA can produce a more compact representation of the data and outperforms PCA in terms of statistical redundancy as well as dimension reduction. However, PCA is much easier to use than ICA and in some cases where no significant nonlinear dependencies are involved, PCA can produce satisfactory results. Thus, although in the following we will mainly focus on local ICA, the arguments are also directly applicable to local PCA.

Standard ICA still has some limitations, namely its linearity and globality.

First, the standard ICA assume that the data x are a linear superposition of independent components s, *i.e.*

$$\boldsymbol{x}(t) = \boldsymbol{W}\boldsymbol{s}(t) \tag{16}$$

where $\boldsymbol{x}(t) = \left[x_1(t), \cdots, x_n(t) \right]^{\mathrm{T}}$, $\boldsymbol{s}(t) = \left[s_1(t), \cdots, s_m(t) \right]^{\mathrm{T}}$, and $\boldsymbol{W}$ is the mixing matrix.

However, it is not general to assume this linearity, even though in many cases linear ICA delivers useful results. For general nonlinear data structures, it can provide only a crude approximation and cannot describe nonlinear characteristics adequately. In view of this drawback, during the recent years researchers have attempted to generalize linear ICA to nonlinear ICA, as

$$\boldsymbol{x}(t) = f\left(\boldsymbol{s}(t) \right), \tag{17}$$

where $f(\cdot): R^m \to R^n$ can be an arbitrary nonlinear mixing function.

Second, the standard ICA tries to describe all of the data using a single group of independent component called global features. This means that the mixing matrix, $\boldsymbol{W}$, in Equation (16) and the corresponding separating matrix are assumed to be the same over the entire region. However, usually physical systems have varying characteristics; and thus, varying mixing matrices in qualitatively different domains of the entire domain. This calls for using different local features in each domain in order to obtain an efficient representation.

In order to overcome the limitation of global linearity, recently researchers have proposed nonlinear ICA as in Equation (14). However, the difficulty of nonlinear ICA arises because the solution of nonlinear ICA problem is usually highly non-unique (see e.g. [25]) and is computationally rather demanding [20]. In order to develop nonlinear extensions of ICA, we propose to use a local linear ICA, in which the data space is first partitioned into disjoint regions and then ICA is performed within each cluster.

Local linear ICA can provide an approximation of nonlinear ICA because using the Taylor expansion the nonlinear mixing function $f(\cdot)$ in Equation (14) can be approximated locally at any point by linear functions. By choosing the number of regions adaptively, the nonlinear characteristic can be represented accurately given limited observations. At the same time, linear ICA is utilized to extract local features within each more homogeneous domain. Thus, multiple sets of local features rather than global features are produced.

Therefore, local ICA can overcome some weaknesses of linear ICA while avoiding the problems associated with general nonlinear ICA. Local ICA usually works in conjunction with a suitable clustering algorithm, which is responsible for partitioning the data space into clusters. For example,[20] proposed to use *k*-means clustering algorithm, and [15] suggested using fuzzy c-varieties clustering [18], which partitions the data space based on the similarity of the mixing matrix.

In our case a different adaptive fuzzy clustering is proposed (see section 2.2), which is embedded in local modeling process rather than standalone. Given clustering, PCA or the Fast ICA algorithm [4] is applied in each cluster to extract local features. It seems more appropriate to employ weighted ICA based upon fuzzy memberships, because even intuitively the points having smaller fuzzy memberships, in the coexistence region for example, should have smaller influence upon feature extraction. However, since the coexistence regions are not so wide, for simplicity we apply the standard algorithms in each cluster.

2.3.2. Build Local Models

Once local features are extracted, we can proceed to construct local models, each pertaining to one of the overlapping operating regimes of the input space. Since each local model is only relevant to one cluster, it is reasonable to train local models using data points belonging to that cluster. Thus, before building local models, all observations need to be first assigned to their respective fuzzy clusters. This constitutes a fuzzy multi-category classification problem. Because the membership functions $\mu_j(x)$ of all fuzzy clusters are known from the step of clustering, the classification task is simply to evaluate the membership of each data point in all clusters, that is, to obtain the result

$$\mu_{ij} = \mu_j(x_i), \tag{18}$$

which specifies the influence of each data point in building local models pertaining to the different operating regimes.

The creation of local composite models follows the same method as in [1]. They are built by multiple linear regression models,

$$f_m(x) = \sum_{j=1}^{p} \beta_{mj} h_{mj}(x), \tag{19}$$

where $h_{mj}(x)$'s refer to local features in the m-th fuzzy operating regime.

Taking into account the varied influence of the data points, the parameters in local models can be estimated by weighted least squares as

$$\beta_m = \arg\min_{\beta_m} \sum_{i=1}^{n} \mu_{ij} \left(y_i - f_m(x)\right)^2, \tag{20}$$

which further encourages the locality of local models.

Local feature selection is an integral part of building local models. The purpose of feature selection is to eliminate non-informative features and noise and remove redundant information, thereby reducing model dimensionality. Since the multiple linear regression method is used in constructing local linear models, feature selection is actually a variable selection problem. Feature selection results in parsimonious models, which are known to yield improved generalization.

From the above, it is easy to see that the act of building of each of the local models is separated from that of the others, except for the effects of overlapping domains. Thus, local feature selection can be also performed separately in each operating regime. As a result, the local feature selection only depends upon the empirical errors of each of the observations falling into a certain cluster.

2.4. Combine Local Feature Models

Once local models are built for all clusters, they can be combined into a global mixture model, or so-called operating based model according to [26], based upon our phase transition model.

Based upon the fuzzy membership functions, the final global mixture model can easily be formulated as a fuzzy weighted average model as

$$f_g(x) = \sum_{m=1}^{M} \mu_m(x) f_m(x), \quad (21)$$

where the fuzzy membership function $\mu_m(x)$ characterizes the operating regime of the m-th local model $f_m(x)$. Each local ICA model can be expressed as

$$f_m(x) = \sum_{j=1}^{p} \beta_{mj} h_{mj}(x), \quad (22)$$

where $h_{mj}(x)$ denotes j-th local featur for m-th cluster and β_{mj} is its corresponding regression coefficient, and p denotes the number of features.

In the current case a simple trapezoidal membership function is applied to represent the fuzzy operating regimes and for any point x a constraint is imposed such that $\sum_{m=1}^{M} \mu_m(x) = 1$, so the mixture of local models turns out to be simple. If for given point x, which is outside the unstable phase transition regions, some $\mu_m(x) = 1$, the m-th local model $f_m(x)$ is dominant and thus $f_g(x) = f_m(x)$; otherwise, if x is inside a transition region two different phases characterized by two different local models coexist. Based upon our stochastic modeling of phase transition, the mixture model $f_g(x)$ can be constructed by a weighted linear superposition of local models having nonzero membership as,

$$f_g(x) = \mu_i(x) f_i(x) + \mu_j(x) f_j(x). \quad (23)$$

An example of two operating regimes is shown in **Figure 2**.

From the above example, it can be seen that the global mixture model is continuous. In fact, this is true as long as the width of the coexistence region is not equal to zero, because, for example,

$$f_g(x_a) = \mu_1(x_a) f_1(x_a) + \mu_2(x_a) f_2(x_a) = f_1(x_a), \quad (24)$$

where $\mu_1(x_a) = 1$ and $\mu_2(x_a) = 0$.

Nevertheless, in general the first derivates are not continuous. This is because with trapezoid membership functions we have the result

$$\begin{aligned} f_g'(x_a) &= \mu_1'(x_a) f_1(x_a) + \mu_1(x_a) f_1'(x_a) \\ &\quad + \mu_2'(x_a) f_2(x_a) + \mu_2(x_a) f_2'(x_a) \\ &= c\left(f_1(x_a) - f_2(x_a)\right) + f_1'(x_a) \end{aligned} \quad (25)$$

where $\mu_1'(x_a) = -c$ and $\mu_2'(x_a) = c$.

3. Three-Stage Optimization Algorithm

For our problem under investigation, we need to jointly optimize the number of operating regimes, the locations of boundaries as well as parameters in local models. This is analogous to use of adaptive regression splines with free knots (cf. [27,28]). Splines are piecewise polynomial functions that are constrained to join smoothly at points called knots. In particular, a free-knot spline is a spline where the knot locations are considered parameters to be estimated from the data. Freeing the knots greatly improves the spline's approximating power [29]. However, it poses a very difficult problem, that is, estimating the optimal number of knots and their locations, which is similar to our problem.

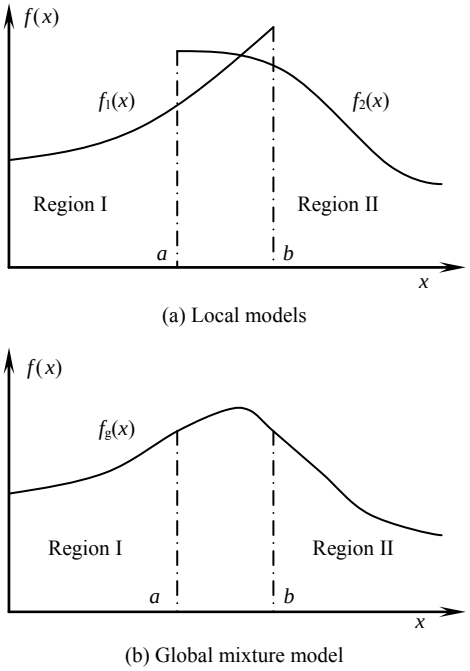

Figure 2. An example with two operating regimes with corresponding local and global models.

But, there are important differences between them. First, our model is more flexible without smoothness constraints, which, on the other hand, leads to use of more free parameters. Second, in our model the regressors nonlinearly depend upon the boundary locations while for splines regressors are fixed as polynomials. Therefore, we expected that the current optimization problem is even more difficult than that of free-knot splines.

The difficulty lies in some undesirable characteristics of the complete objective function in Equation (13). In order to illustrate its properties, we can substitute $f_g(x, \alpha, \beta)$ in Equation (21) and rewrite it as

$$\arg \min_{M} \min_{\alpha} \min_{\beta} n \log \sum_{i=1}^{n} \left(y_i - \sum_{m=1}^{M} \mu_m(x_i, \alpha) \sum_{j=1}^{p_m} \beta_{mj} h_{mj}(x_i, \alpha) \right)^2 + k(M) \log n. \quad (26)$$

First, it is a complex nonlinear function of the boundary locations, because the membership functions and the local independent components depend nonlinearly upon α and they appear inside the square.

Second, it is non-differentiable partly because of the non-differentiable membership function. Furthermore, the explicit dependence of the local features, or equivalently the separating matrices, on the fuzzy clustering parameters α cannot be known. Because of this non-differentiability, all gradient-based optimization algorithms such as steepest descent, Newton-Raphson method and conjugate gradients methods will fail. However, some numerical searching algorithms are still applicable.

Finally, the objective function is not strictly convex or concave but has many local optima. This can be seen by applying the "lethargy" theorem introduced by Jupp [27]. Similar to free-knot splines [27], the existence of multiple optima in the objective surface is related to the symmetry introduced by the exchangeability of the boundary parameters. For example, in a simple case with two clusters the objective surface is symmetric along any normal to the line defined by two equal parameters. Consequently, the derivative along the normal at the intersection to the equal-parameter line is equal to zero. This property, called "lethargy" by [27], results in many stationary points and ridges along lines or planes in the parameter space where two or more parameters coincide.

This property results in the failure of all local optimization algorithms including gradient-based methods, line search and hill climbing, because they easily fix upon local optima, the locations of which depend upon the different initialization.

In order to overcome this problem, global optimization algorithms including simulated annealing and genetic algorithms should be applied instead.

These are some interrelated reasons that make it so difficult to locate the global optimum. Since there are many local optima in the objective surface, use of good starting parameter values is essential for finding the global optimum. Unfortunately, this is usually difficult. One possible way is to construct starting values based upon data. First, we sort the inputs x and split the input range to segments s_1 through s_{n-1}. Clearly, every parameter α_i can be formulated within each one of the segments. Then, the entire parameter space of the vector α will be divided into $(n\text{-}1)^{2M-2}$ pieces of subspaces. If we pick an initial value for α within each subspace, we will eventually find a local optimum within that subspace. Comparing all of these local optima will give us an approximate global optimum. However, because we cannot make sure there is only one local optimum within each subspace, the globality of the best identified optimum is not guaranteed. Furthermore, this procedure is computationally very expensive because there are $O(n^{n/k})$ possible initial values in total.

Originally developed by Holland [30], genetic algorithm (GA) is a global stochastic search algorithm, which is less susceptible to getting fixed upon local optima than are gradient search methods. On the other hand they tend to be computationally expensive. In practice, genetic algorithms work very well on mixed (continuous and discrete), combinatorial problems. For example, Pittman [31] suggests using GAs to optimize the knot locations in adaptive splines.

Here we propose a similar hybrid genetic algorithm, which combines global optimization together with a local search. It is different from that in [31] in that a distinct genetic chromosomal representation and correspondingly different genetic operators are defined. Furthermore, in this scheme GAs are only used to identify good starting values for the local search rather than the global optimum. Doing this significantly reduces the required computational time arising from the slow convergence of GAs.

On the whole, following a strategy of problem splitting, the optimization problem can be solved through three stages.

First, we optimize the local model parameters given fuzzy clusters. In order to encourage competition and locality of local models, we can approximate the original objective function with a slightly different one

$$\begin{aligned}\hat{\beta} &= \arg\min_{\beta} \sum_{i=1}^{n} \left(y_i - f_g\left(x_i, \alpha, \beta\right)\right)^2 \\ &\approx \arg\min_{\beta} \sum_{m=1}^{M} \sum_{i=1}^{n} \mu_{im} \left(y_i - f_m\left(x, \beta, \alpha\right)\right)^2.\end{aligned} \tag{27}$$

The only difference between the two objective functions arises in the coexistence regions. For convenience, let's denote $E\left(\theta_i\right) = \sum_{m=1}^{m} \mu_{im}\theta_i$, providing the result

$$\begin{aligned}&\sum_{m=1}^{M} \mu_{im} \left(y_i - f_m\left(x, \beta, \alpha\right)\right)^2 \\ &= E\left(y_i - f_m\left(x_i, \beta, \alpha\right)\right)^2 \\ &= \left(y_i - E\left(f_m\left(x_i, \beta, \alpha\right)\right)\right)^2 \\ &\quad + E\left(f_m\left(x_i, \beta, \alpha\right) - E\left(f_m\left(x_i, \beta, \alpha\right)\right)\right)^2 \\ &= \left(y_i - f_g\left(x_i, \beta, \alpha\right)\right)^2 \\ &\quad + \sum_{m=1}^{M} \mu_{im} \left(f_m\left(x_i, \beta, \alpha\right) - f_g\left(x_i, \beta, \alpha\right)\right)^2\end{aligned} \tag{28}$$

where the second term is usually small within the coexistence regions.

In fact, another advantage of this change is that it makes the optimization problem much easier, because the membership function is moved out of the square operator. Moreover, with this change local models can be

built independently from each other (being consistent with section 2.3.2)

At this stage the local model parameters β can be optimized as functions of α, which can be easily done by WLS as described in section 2.3.

Second, we need to identify the best fuzzy clustering given the number of clusters, by rewriting the sub-optimization problem as

$$\begin{aligned}\hat{\alpha} &= \arg\min_{\alpha} \sum_{i=1}^{n}\left(y_i - \sum_{m=1}^{M}\mu_m(x_i,\alpha)\, f_m\left(x_i,\hat{\beta}(\alpha),\alpha\right)\right)^2 \\ &= \arg\min_{\alpha} F(\alpha),\end{aligned} \tag{29}$$

where $\alpha = (\alpha_1, \cdots, \alpha_{2M-2})$ with $x_{\min} < \alpha_1 \le \alpha_2 < \cdots < \alpha_{2M-1} \le \alpha_{2M-2} \le x_{\max}$.

Note that under the assumption of Gaussian errors the fuzzy clustering parameters are actually quantified by the Maximum Likelihood Estimator. Nevertheless, a well-known problem with MLE is the danger of overfitting. For a simple example, suppose we have two operating regimes and the number of data points falling in the first regime is equal to the number of candidate models and all others fall in the other. In this case, the data can be fitted perfectly in the first regime and a little better in the other regime than in the single regime case. Therefore, as a result the maximum likelihood is increased dramatically, but the resultant model most likely becomes worse.

In order to overcome this pitfall, we apply the cross-validation approach of using the testing likelihood in place of maximum likelihood. Correspondingly, the sub-optimization problem can be expressed as

$$\begin{aligned}\hat{\alpha} &= \arg\min_{\alpha} \sum_{i=1}^{n_t}\left(y_{ti} - \sum_{m=1}^{M}\mu_m(x_{ti},\alpha)\, f_m\left(x_{ti},\hat{\beta}(\alpha),\alpha\right)\right)^2 \\ &= \arg\min_{\alpha} F_t(\alpha)\end{aligned} \tag{30}$$

where the subscribe *t* denote the out-of-sample test.

The sub-objective function $F_t(\alpha)$ in Equation (28) has all of the unpleasing properties mentioned earlier. In order to address this challenge, we will propose using a hybrid optimization algorithm combining GAs with multi-dimensional hill climbing.

After performing both global and local optimization procedure, a group of good candidate solutions are obtained, from which the solution with the smallest $F_t(\alpha)$ can be easily chosen as the optimal one.

The last stage is to choose the optimal number of local models (a model selection problem). Since we require that in each operating regime the number of data points must be greater than the number of local features, therefore there exists an upper bound for the number of local models, M_m, which is much smaller than the sample size *n*. Thus, this optimization problem can be expressed as

$$\begin{aligned}\hat{M} &= \arg\min_{1\le M\le M_m} n\log\sum_{i=1}^{n_t}\left(y_i - f_g\left(x_i,\hat{\alpha},\hat{\beta}\right)\right)^2 \\ &\quad + k(M)\log n_t \\ &= \arg\min_{1\le M\le M_m} G(M)\end{aligned} \tag{31}$$

In order to determine the optimal number of clusters, we repeat the first two stages of our procedure for *M* ranging from 1 to M_m, and finally we choose the optimal value that corresponds to the minimal model selection criterion value in Equation (29). In practice, a forward stepwise process can be utilized, which increases the value of *M* from 1 and stops when the model selection criterion value increases.

3.1. Real-Coded Genetic Algorithm (GA)

In order to apply an actual-coded genetic algorithm to facilitate the search for a global optimum, we need to design problem-specific fitness functions and operators.

The first crucial issue of using a GA is to define a proper fitness function, which indicates the quality of a candidate solution. Usually, GA works by maximizing the fitness, but here we intend to minimize the objective function or equivalently to maximize the likelihood. Therefore, the fitness function can be constructed from the objective function as

$$\text{fitness}\left(\alpha^{(k)}\right) = \text{Max}\left(F_t\left(\alpha^{(j)}\right)\right) - F_t\left(\alpha^{(k)}\right), \tag{32}$$

where $F_t(\alpha)$ is the objective function in Equation (32), $\text{Max}\left(F_t\left(\alpha^{(j)}\right)\right)$ stands for the maximum $F_t(\alpha)$ in a population and $\text{fitness}\left(\alpha^{(k)}\right)$ refers to the fitness of the *k*-th individual chromosome.

The definition of fitness significantly influences the convergence behavior. For example, in the early stage few "super individuals" tend to dominate the selection process leading to premature convergence, whereas later when the population is less diverse, the simulation tends to lose focus [32]. Therefore, in practice we would like to apply a more general and flexible fitness function by scaling and shifting, *i.e.*, as

$$\text{fitness}\left(\alpha^{(k)}\right) = b + a\left(\text{Max}\left(F_t\left(\alpha^{(j)}\right)\right) - F_t\left(\alpha^{(k)}\right)\right) \tag{33}$$

where the scaling factors *a* and shifting factor *b* are so adjusted adaptively during simulation avoid premature convergence early and to encourage later convergence. In our simulation, we choose *b* to be the minimum fitness and *a* to be the reciprocal of the average fitness.

As for selection, we utilize the fitness-weighted roulette wheel method, which is conceptually equivalent to giving each individual a slice of a roulette wheel equal in

area to the individual's fitness. The wheel is spun and the ball comes to rest on the wedge shaped slice, by which means the corresponding individual is selected. Therefore, the probability for a chromosome to be chosen is proportional to its fitness. A pair of "parents" is selected by spinning the wheel two times to reproduce a pair of "children" via recombination and mutation.

Notably, the GA success is also sensitive to the nature of the recombination and mutation operators. For example, it is found that general, fixed, problem-independent recombination operators often break partial solutions and cause slow convergence. In order to avoid such problems, we design problem-specific crossover and mutation operators.

The recombination strategy that we applied is the one-point arithmetic crossover. Let the parents be $P_1 = [P_{11}, \cdots, P_{1L}]$ and $P_2 = [P_{21}, \cdots, P_{2L}]$, respectively. Then, the two offspring are obtained as

$$C_{1i} = \begin{cases} P_{1i}, & i \le t \\ P_{1t} + \dfrac{x_{\max} - P_{1t}}{x_{\max} - P_{2t}}(P_{2i} - P_{2t}), & i > t \end{cases} \quad (34)$$

and

$$C_{2i} = \begin{cases} P_{2i}, & i \le t \\ P_{2t} + \dfrac{x_{\max} - P_{2t}}{x_{\max} - P_{1t}}(P_{1i} - P_{1t}), & i > t \end{cases} \quad (35)$$

where t is a random integer number among 1, 2, ···, L.

This crossover operator is so designed that it guarantees that the resulting children are still ordered sequences of real numbers within a valid range, and the number of clusters is maintained. In fact, this treatment is especially suitable for chromosomal representations of ordered sequences of real numbers.

The crossover rate (*i.e.* the probability that crossover occurs) is generally around 0.5, and in this paper we set it at 0.6.

The mutation operator is defined as an addition of a normally distributed factor with mean value 0 (*i.e.* $D_i' = D_i + \varepsilon$) where D_i is an original parameter and D_i' is the mutated one, ε is a normally distributed random number, *i.e.* $N(0,\sigma^2)$, where the value of σ^2 is tunable. ε plays a similar role of the step size in a line search. In this study, we choose

$$\sigma = \frac{x_{\max} - x_{\min}}{3n}, \quad (36)$$

where n denotes the sample size.

[33] suggested that the optimal mutation rate, i.e. the probability that mutation occurs for a single gene in a chromosome, is approximately $(S \cdot L^{1/2})^{-1}$, where S is the population size and L is the length of the chromosome. Here we follow this rule.

Since the current optimization problem is constrained by the requirement $x_{\min} < \alpha_1 \le \alpha_2 < \cdots < \alpha_{2M-1} \le \alpha_{2M-2} \le x_{\max}$, in addition to the selection, crossover and mutation operators, we need another check operator in order to create a new valid child chromosome.

A valid chromosome has to satisfy some constraint. First, a chromosome must be an ordered sequence of real numbers within the range $[x_{\min}, x_{\max}]$. Also, the number of data points falling into each cluster must be greater than the number of local independent components.

If there is no crossover and mutation, a chromosome is simply copied to the next generation.

The stopping rule for the current iterative case is relatively simple, as our purpose is to search for promising initial inputs for a local optimization algorithm. Thus, when we observe that the convergence of GA becomes very slow we stop it.

In summary, the main steps of the GA are the following:

1) Build an initial population of S chromosomes randomly selected within $[x_{\min}, x_{\max}]$;

2) Calculate the fitness of each chromosome;

3) Select chromosomes from the parent generation to reproduce a child generation:

i) Select two parent chromosomes,

ii) Generate a random number between [0,1].(If it is smaller than the crossover rate, recombine them by one-point arithmetic crossover; otherwise, enter the next step);

iii) Generate a random number between [0,1]. (If it is smaller than the mutation rate, perform mutation on a gene in a chromosome. Repeat this for each gene in both chromosomes).

iv) Add the two resulting chromosomes to the next generation.

Repeat steps i) through iv) until S new chromosomes are reproduced.

4) Finally, if the stopping criterion is satisfied, then exit; otherwise, return to step (2).

3.2. Adaptive Multi-Dimensional Hill Climbing

By means of GA optimization, we obtain a set of global good initial guesses of the best vector of fuzzy clustering parameters, namely the last generation produced by the GA. In practice, it is also useful to keep track of the "best" chromosome throughout the whole GA simulation history. The next task is to search for the optima corresponding to these good initial guesses.

As noted, our objective functions are quite complicated and, thus, it is difficult to apply classical gradient-based optimization methods. However, this problem can

be solved numerically by a derivative-free approach, for instance, hill climbing. We propose a derivative-free method in order to optimize the parameters one by one while keeping the others fixed. Furthermore, the step size is adaptively tuned. However, since each parameter is not independent of the others, the overall optimization has to be performed iteratively. Our algorithm, described below, is actually a multi-dimensional version of adaptive hill climbing,

It consists of multiple loops, in each of which the individual parameters are optimized one at a time. Consider optimization of α_i, where its current value is $\alpha_i^{(0)}$ and the current model evaluation value is $F_t(\alpha)^{(0)}$. Let $k = 1$ and $\alpha_i^{(k)} = \alpha_i^{(k-1)} + kd$, where d is small positive number, and keep the other $p - 1$ parameters unchanged. Then recalculate the model evaluation value as $F_t(\alpha)^{(k)}$. If $F_t(\alpha)^{(1)} > F_t(\alpha)^{(0)}$, that is, the fuzzy model gets worse, then return to $\alpha_i^{(0)}$ and let $k = 1$and replace d by $-d$; otherwise, continue to search in the same direction within the interval $[x_{\min}, x_{\max}]$ until $F_t(\alpha)^{(k+1)} > F_t(\alpha)^{(k)}$. The final $\alpha_i^{(k)}$ is taken as the optimum in the current loop. Then we turn to the next parameter α_{i+1}. Each computational loop starts with α_0 and ends up with α_{2M-2}. Once a loop is computed, another one will be started depending upon the stopping criterion.

At the beginning of each loop, we calculate the resultant model's values of $F_t(\alpha)$, and we do the same for the end of each loop. If the difference between these two values is small enough, for example,

$$\frac{\left|F_t(\alpha)^{(j+1)} - F_t(\alpha)^{(j)}\right|}{F_t(\alpha)^{(j+1)}} < \delta, \tag{37}$$

where δ is very small (e.g. 10^{-5}), we say that the minimum has been reached and we stop the local searching process.

In view of the facts that i) There exists an (unknown) lower bound for $F_t(\alpha)^{(k)}$, and ii) the sequence of $F_t(\alpha)^{(k)}$ is not increasing, the convergence is guaranteed according to the Cauchy convergence criterion.

3.3. Procedure of Mixture of Local Feature Models

To this point the development of a new method for mixture of local feature models, from model structure to parameter estimation, is almost complete. In summary, the input space of the system is first decomposed into fuzzy subspaces by use of the adaptive fuzzy clustering algorithm and then in each subspace the system is approximated by a local linear feature model. This is somewhat analogous to that of the Takagi-Sugeno fuzzy model [34] within the context of predictive control.

The entire modeling procedure can be summarized as follows.

1) Set $M = 1$

2) Optimize the fuzzy clustering using a hybrid GA given the number of fuzzy clusters

a) Build an initial population of S chromosomes randomly;

b) Calculate the fitness of each chromosome;

i) Classify data points into each fuzzy cluster

ii) Perform local PCA/ICA within each fuzzy cluster

iii) Create local PCA/ICA models based upon data using the multiple regression method with fuzzy variable selection

iv) Mix local PCA/ICA models

v) Calculate the weighted residual sum of squared error

c) Generate the next generation of chromosomes by selection, crossover and mutation

d) If the "stop" criterion is met, then go to e); otherwise go to b)

e) Treat the last generation of GA as the starting parameter values and determine the local minima around them

3) Choose the best local optimum as the global optimum and then assess the resulting optimal model by means of $G(M)$ in Equation (29). If $G(M) < G(M-1)$ for $M \geq 2$, then go to step (4); otherwise, go to step (5)

4) Set $M = M + 1$ and go to step (2)

5) Return the final optimal mixture model including the optimal fuzzy clustering.

4. Numerical Simulation Study

In this section, we present results from our numerical simulation studies. In order to directly compare with a global composite model method, we apply the new method to the same examples used in [1].

It is first applied to an artificial example, where the true model is supposed to be known. This allows us to demonstrate how the method works and its advantage over global models. Then, the method is used concerning a real physical case. From our numerical simulation, we also justify our argument that a mixture of local models is suitable to situations where severe nonlinearity is involved, with the linear local models reflecting the nonlinear characteristics.

4.1. Artificial Example

In this example, artificial models and data are used to demonstrate the effectiveness of the new method in treating complex nonlinearity. Assume that the true model is expressed as

$$y(x) = 150 - 150e^{-2x} + x^2 - 0.1x^3 + 4x + 30e^{-x/3}\sin(x) + 15\sin(1.5x) - 20\ln(x+1), \quad (38)$$

Correspondingly, the data generative model can be written as

$$y = y(x) + \varepsilon, \quad (39)$$

where ε obeys a normal distribution, *i.e.* $N(0,\sigma^2)$ where σ^2 is equal to 64. From this generative model, we gathered a set of data with $n = 50$, *i.e.* (x_i, y_i), where the x_i are evenly distributed between [0,10] (see **Figure 3**) and the y_i are the corresponding noisy observations.

We also propose a class of candidate models as follows:

$$\begin{aligned}
f_1(x) &= 150 - 150e^{-2x} + 4x + 15\sin(1.5x) - 20\log(x+1);\\
f_2(x) &= 150 - 150e^{-2x} + x^2 - 0.1x^3;\\
f_3(x) &= 150 - 150e^{-2x} + x^2 - 0.1x^3 + 30e^{-x/3}\cdot\sin(x);\\
f_4(x) &= 150 - 150e^{-2x} + 6x + 30e^{-x/3}\cdot\sin(x) - 20\cdot\log(x+1);\\
f_5(x) &= 150 - 150e^{-2x} + x^2 - 0.1x^3 + 15\sin(1.5x);\\
f_6(x) &= 150 - 150e^{-2x} + 15\cos(2x) - 15 + 0.004x^2;
\end{aligned} \quad (40)$$

as shown in **Figure 3**. Thus, for the same input these candidate models give different results.

Note that each candidate model is either incomplete or erroneous, or both. As shown in **Figure 3**, these models approximate the true model to varying degrees over the input space.

Once the candidate models are formulized and data are collected, we are ready to employ our new approach to determine local domains by fuzzy clustering, to build local models within each fuzzy cluster and finally to mix local models into a global model. The results are shown in **Table 1**, compared to a single best model and a global linear model. In **Table 1**, test error refers to the cross-validation mean squared error.

The fuzzy membership functions and resultant mixture models are plotted in **Figures 4** and **5**, respectively. From the results in Table 1, we see that the new method works well as expected. This is understandable, because in this example highly non-linear dependencies among candidate models and severe nonlinearity in the data are involved. These are exactly the two primary problems that our new approach is supposed to deal with.

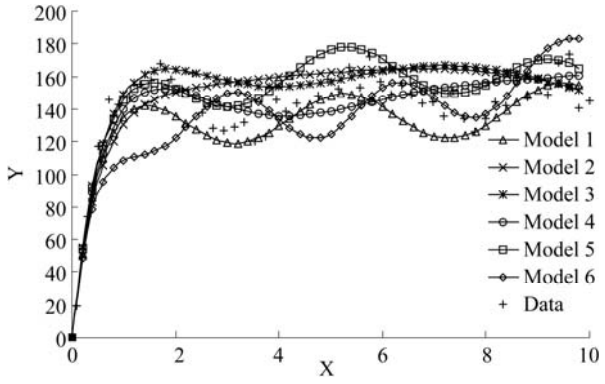

Figure 3. Artificial candidate models and generated data.

Table 1. Simulation results of the artificial example.

Number of domains	Domains	BIC	Test error
A single best model	[0, 10]	N/A	104.67
Global combination	[0, 10]	583	18.02
2-Regime local models	[0, 0.57, 4.865, 10]	55.15	11.68

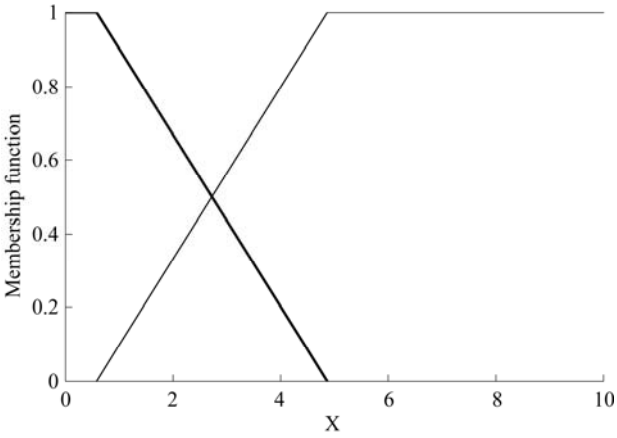

Figure 4. Membership functions.

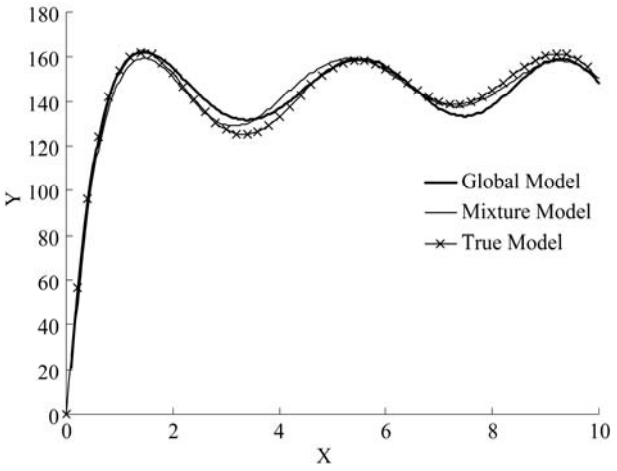

Figure 5. Local models and the mixture model.

4.2. Physical Case Study

In the above subsection, we demonstrate the effectiveness of our method using artificial data. Here, we address a physical example.

The physical example that we use here is that of the peak ground acceleration (PGA) attenuation models used in seismology. In this example, the purpose is to build a more accurate mixture of local models, which is applicable to southern California in the United States. A sample data set of size 102 is obtained from the literature [35], whose logarithms are assumed to include Gaussian noise. Correspondingly, the candidate attenuation models include those of Boore *et al*. (1997)[36], Sadigh *et al*. (1997)[37], Abrahamson and Silva (1997)[38], Campbell and Bozorgnia (1997)[39], Spudich *et al*. (1997) and Idriss(1995) [40]. All of these attenuation relations may be found in Seismological Research Letters, Volume 68, Number 1, January/February, 1997. All of these attenuation relationships were developed for shallow crustal earthquakes in active tectonic regions, and thus they should be applicable to southern California.

The candidate models are plotted together with the sample data in **Figure 6**. From **Figure 6**, it is easy to note that all the models are close to be a straight line, which means that unlike the artificial example the dependence among candidate models are mostly linear.

As in the artificial example, we apply the new method to optimize fuzzy domains, create local models and finally create a mixture model. The simulation results are shown in **Table 2**, where the test error refers to the cross-validation mean squared error, and the mixture model is plotted in **Figure 7** together with the data.

From the above results, we see that the test error of the mixture model using two operating regimes is smaller than that of the global model by about 12%. Also, the model plotted in **Figure 7** appears reasonable. First of all, the peak ground acceleration (PGA) decreases with increasing distance from the earthquake epicenter. It also shows that in two regions, namely near the epicenter and far from the epicenter, the attenuation model as a function of the distance is somewhat different. One of the possible reasons is the effect of the depth of the seismic source. Likewise, a possible explanation of the turning point shown near 23 km is that for shallow crustal earthquakes the average depth of ruptures is about 25 km [39].

5. Summary

In this paper, we propose a mixture model of local ICA models to overcome the weakness of global models in dealing with nonlinearity and locality. This method consists of three components: fuzzy clustering, local feature and model combination. The supervised adaptive fuzzy clustering algorithm is proposed in order to divide the entire input space into operating regimes, local PCA or ICA analysis is carried out and the feature-based model combination method is applied to create local feature models in each individual region. Finally the local feature models are combined into a single mixture model. Correspondingly, a three-stage optimization procedure is designed to optimize the complete objective function, which is actually a hybrid GA algorithm.

Table 2. Simulation results of physical case study.

Number of domains	Domains	BIC	Test error
Best single model	[0,120]	N/A	0.1935
Global combination	[0, 120]	–66.72	0.1567
2-Regime local model	[0, 23, 23.9, 120]	–69.37	0.1303

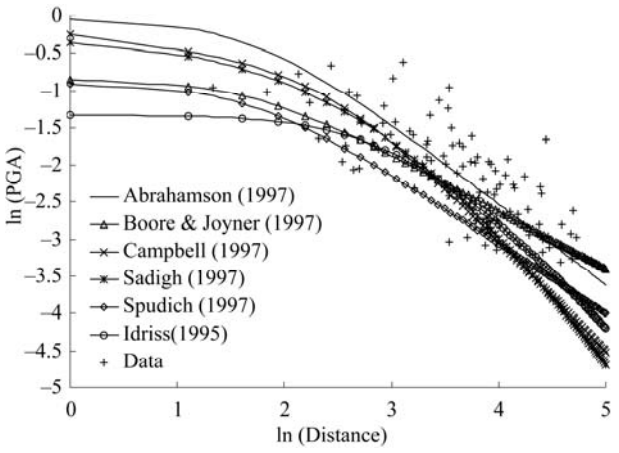

Figure 6. Candidate peak ground acceleration (PGA) attenuation models and data.

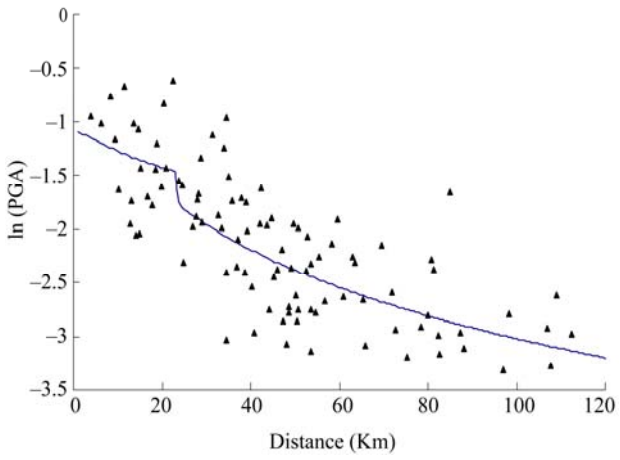

Figure 7. Local models and mixture model.

In order to demonstrate its effectiveness this new method is applied to both an artificial example and a physical case in a seismic study. Our simulation results show that the adaptive fuzzy mixture of local ICA models is superior to global models.

6. References

[1] M. Xu and M. Golay, "Data-guided Model Combination by Decomposition and Aggregation," *Machine Learning*, Vol. 63, No. 1, 2005, pp. 43-67.

[2] D. J. Bartholomew and M. Knott, "Latent Variable Models and Factor Analysis," London: Arnold; New York: Oxford University Press, 1999.

[3] I. T. Jolliffe, "Principal Component Analysis," New York: Springer-Verlag, 1986.

[4] A. Hyvärinen, "Fast and Robust Fixed-Point Algorithms for Independent Component Analysis," *IEEE Transactions on Neural Networks*, Vol. 10, No. 3, 1999, pp. 626-634.

[5] J. Karhunen and S. Malaroiu, "Locally Linear Independent Component Analysis," *International Joint Conference on Neural Networks*, 1999.

[6] T. A. Johansen and B. A. Foss, "Operating Regime Based Process Modeling and Identification," *Computers and Chemical Engineering*, Vol. 21, 1997, pp. 159-176. doi:10.1016/0098-1354(95)00260-X

[7] G. J. McLachlan and K. E. Basford, "Mixture Models: Inference and Application to Clustering," New York: Marcel Dekker, 1988.

[8] R. Murray-Smith and T. A. Johansen, "Local Learning in Local Model Networks," *Proceedings of IEE International Conference on Artificial Neural Networks*, Cambridge, UK, 1995, pp. 40-46. doi:10.1049/cp:19950526

[9] M. I. Jordan and R.A. Jacobs, "Hierarchical Mixtures of Experts and the EM Algorithm," *Neural Computation*, Vol. 6, 1994, pp. 181-214. doi:10.1162/neco.1994.6.2.181

[10] J. Fan, "Local Modelling," Encyclopidea of Statistical Science, 1995.

[11] W. S. Cleveland and S. J. Devlin, "Locally Weighted Regression: An Approach to Regression Analysis by Local Fitting," *Journal of the American Statistical Association*, Vol. 83, 1988, pp. 596-610.

[12] R. A. Jacobs, M. I. Jordan, S. J. Nowlan and G. E. Hinton, "Adaptive Mixtures of Local Experts," *Neural Computation*, Vol. 3, 1991, pp. 79-87. doi:10.1162/neco.1991.3.1.79

[13] U. Söderman, J. Top and J.-E. Strömberg, "The Conceptual Side of Mode Switching," *Proceedings of IEE International Conference on Systems, Man, and Cybernetics*, Le Touquet, France, 1993, pp. 245-250.

[14] S. Harrington, R. Zhang, P. H. Poole, F. Sciortino, and H. E. Stanley, "Liquid-Liquid Phase Transition: Evidence from Simulations," *Physical Review Letters*, Vol. 78, No. 12, 1997, pp. 2409-2412. doi:10.1103/PhysRevLett.78.2409

[15] K. Honda, H. Ichihashi, M. Ohue and K. Kitaguchi, "Extraction of Local Independent Components Using Fuzzy Clustering," *Proceedings of 6th International Conference on Soft Computing*, 2000.

[16] L. J. Breiman, H. Friedman, R. A. Olshen, and C. J. Stone, "Classification and Regression Trees," Belmont CA: Wadsworth, 1984.

[17] J. C. Dunn, "A Fuzzy Relative of the ISODATA Process and Its Use in detecting Compact Well-Separated Clusters," *Journal of Cybernetics*, Vol. 3, 1973, pp. 32-57. doi:10.1080/01969727308546046

[18] J. C. Bezdek, "Pattern Recognition with Fuzzy Objective Function Algorithms," Plenum Press, New York, 1981.

[19] N. Kambhatla and T. Leen, "Dimension Reduction by Local Principal Component Analysis," *Neural Computation*, Vol. 9, 1997, pp. 1493-1516. doi:10.1162/neco.1997.9.7.1493

[20] J. Karhunen and S. Malaroiu, "Local Independent Component Analysis Using Clustering," *Proc. First Int. Workshop on Independent Component Analysis and Signal Separation*, 1999, pp. 43-48.

[21] S. E. Geman, Bienenstock and R. Doursat, "Neural Networks and the Bias/Variance Dilemma," *Neural Computation*, Vol. 4, 1992, pp. 1-58.

[22] H. Akaike, "Information Theory and an Extension of the Maximum Likelihood Principle," *2nd International Symposium on Information Theory* (B. N. Petrov and F. Czáki, eds.), 1973, pp. 267-281.

[23] G. Schwarz, "Estimating the Dimension of a Model," *Annals of Statistics*, Vol. 6, 1978, pp. 461-464. doi:10.1214/aos/1176344136

[24] H. Hotelling, "Analysis of a Complex of Statistical Variables into Principal Components," *Journal of Educational Psychology*, Vol. 24, 1933, 417-441. doi:10.1037/h0071325

[25] A. Hyvärinen and P. Pajunen, "Nonlinear Independent Component Analysis: Existence and Uniqueness Results," *Neural Network*, Vol. 12, No. 2, 1999, pp. 209-219.

[26] R. Murray-Smith and T. A. Johansen (Eds.), "Multiple Model Approaches to Nonlinear Modeling and Control," Taylor and Francis, London, UK, 1997.

[27] D. L. B. Jupp, "Approximation to Data by Splines with Free Knots," *SIAM Journal on Numerical Analysis*, Vol. 15, No. 2, 1978, pp. 328-343. doi:10.1137/0715022

[28] J. Friedman, "Multivariate Adaptive Regression Splines (with discussion)," *Annals of Statistics*, Vol. 19, 1991, pp. 1-141. doi:10.1214/aos/1176347963

[29] H. G. Burchard, "Splines (With Optimal Knots) are Better," *Applicable Analysis*, Vol. 3, 1974, pp. 309-319. doi:10.1080/00036817408839073

[30] J. M. Holland, "Adaptation in Nature and Artificial Systems," Ann Arbor, MI: The University of Michigan Press,

1975.

[31] J. Pittman, "Adaptive Spline and Genetic Algorithms," *Journal of Computational and Graphical Statistics*, Vol. 11, No. 3, pp. 1-24.

[32] David E Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning," Kluwer Academic Publishers, Boston, MA, 1989.

[33] J. Hessner and R. Männer, "In Proceedings of the First Workshop on Parallel Problem Solving from Nature," *Lecture Notes in Computer Science*, Vol. 496, Springer-Verlag: Berlin, 1991, pp. 23-31.

[34] T. Takagi and M. Sugeno, "Fuzzy Identification of Systems and Its Application to Modeling and Control," *IEEE Transactions on Systems, Man and Cybernetics*, Vol. 15, 1985, pp. 116-132.

[35] J. H. Steidl and Y. Lee, "The SCEC Phase III Strong-Motion DataBase," *Bulletin of the Seismological Society of America*, Vol. 90, No. 6B, 2000, pp. S113-S135. doi:10.1785/0120000511

[36] D. M. Boore, W. B. Joyner and T. E. Fumal, "Equations for Estimating Horizontal Response Spectra and Peak Acceleration from Western North American Earthquakes: A Summary of Recent Work," *Seismological Research Letters*, Vol. 68, No. 1, 1997, pp. 128-153.

[37] K. Sadigh, C.-Y. Chang, J. A. Egan, F. Makdisi and R. R. Youngs, "Attenuation Relations for Shallow Crustal Earthquakes Based on California Strong Motion Data," *Seismological Research Letters*, Vol. 68, No. 1, 1997, pp. 180-189.

[38] N. A. Abrahamson, and W. J. Silva, "Empirical Response Spectral Attenuation Relations for Shallow Crustal Earthquakes," *Seismological Research Letters*, Vol. 68, No. 1, 1997, pp. 94-12.

[39] K. W. Campbell, "Empirical Near-source Attenuation Relations for Horizontal and Vertical Components of Peak Ground Acceleration, Peak Ground Velocity, and Pseudo-absolute Acceleration Response Spectra," *Seismological Research Letters*, Vol. 68, No. 1, 1997, pp. 154-179.

[40] I. M. Idriss, "An Overview of Earthquake Ground Motion Pertinent to Seismic Zonation," *5th International Conference on Seismic Zonation*, 1995, pp. 17-19.

