Proceedings of International Joint Conference on Neural Networks, Orlando, Florida, USA, August 12-17, 2007

Bayesian Signal Classifier

Chi Kin CHOW and Shiu Yin YUEN

Abstract-**This article points out the limitations of vectoral input pattern on density estimation and Bayesian classification. A continuous Bayesian classifier is proposed to tackle these limitations. The classifier accepts signal as input pattern; thus the problem of optimal description length selection is avoided. The algorithm is evaluated on signal clustering and distribution classification.**

I. Introduction

A *Bayesian classifier* can be viewed as a 4-tuple <**C**, **F**, X, $P_{c,\mathbf{F}}$>, where **C** is a finite set of class labels and $\mathbf{F} = [\mathbf{f_1}, \mathbf{f_2}, \ldots, \mathbf{f_n}]$ is a finite set of class feature, $X = \Pi R_i$ with R_i defining the set of possible observations of $\mathbf{f_i}$, i.e. $\mathbf{f_i} \in R_i$, and $P_{c,\mathbf{F}}$ denotes the probability of observation $\mathbf{F} \in X$ given class label $c \in \mathbf{C}$. The objective of Bayesian classifier is to correctly predict the class label of any given feature vector in $\mathbf{F} \in X$. Denoting by $P(c)$ the prior distribution of class labels, its posterior distribution is computed by Bayes rule:

$$P(c \mid \mathbf{F}) = \frac{P(\mathbf{F} \mid c)P(c)}{\sum_{e \in \mathbf{C}} P(\mathbf{F} \mid e)P(e)}$$

Recently, numerous articles reported modifications on Bayesian classifier. H. Li et al. in [14] formulated the classifier as a partially observable Markov decision process, which is responsible for selecting the feature. As a result, the cost of observing features is minimized while the classification performance is maximized. Z. Shi et al. in [15] presented a mapping of attribute set according to the information geometry and Fisher score, in order to relax the condition independence assumption. Larsen R. [16] described extensions of 2D contextual classification algorithm to 3D based on the simultaneous distribution of a pixel and its nearest neighbors. Yaniv and Boaz [17] smoothed the density by a spline, thus enabling simple implementation of the Bayesian classifier without sacrificing the classification accuracy. Balaji et al. [18] adopted a Bayesian approach to simultaneously learn both an optimal nonlinear classifier and a subset of predictor variables. It uses heavy-tailed priors to promote sparsity in the utilization of both basis functions and features. Examples of Bayesian classifier based applications include the following: scene classification with a visual grammar [19], biometric recognition [20], image segmentation [21] [22], appearance based tracking [23] and foot pressure lesions [24].

Chi Kin CHOW and Shiu Yin YUEN are with the department of electronic engineering, City University of Hong Kong, Hong Kong, PRC (tel: 852-27887717; e-mail: {chowchi, kelviny.ee}@cityu.edu.hk).

The posterior distributions of Bayesian classifier are normally estimated by a set of training samples: a pair of *Cause* and *Effect*. The form of training sample is application dependent and different applications have different representations. *Causes* normally appear as continuous signals containing huge amount of redundant information. In order to simplify the training process, they are quantized as vectors; namely vectoral *Causes* (feature vector). For example, *Cause* is represented as a high resolution grid (image or depth map) in object recognition. A 128 by 128 image forms a 16384-dimensional feature vector under raster scanning order, in which the corresponding network involves the determination of several thousands weights. Down-sampling and prior-knowledge-based feature extraction [10] [11] are common methods to reduce the dimension of *Cause* (simplify the network topology).

The description length of a feature vector is predefined and non-adjustable during training and prediction phases. Since the size of an accurate description changes from application to application and from *Cause* to *Cause*, fixed-length description is always insufficient to represent all possible *Causes* of an application. A high-dimension feature provides a detailed description of a *Cause* but leads to a complicated network with poor generalization ability. On the other hand, a low-resolution feature gives an inaccurate *Cause* description but a more generalized network. The determination of the optimal description length remains a challenging problem. Despite that Minimum Description Length (MDL) [5] [6] is a promising guideline to select the size of feature; MDL can only return the optimal length among the training set. The length may not be true to the testing set and the problem of description length is not solved.

Sometimes *Cause* is described as a parametric distribution or simply as a set of scattered point. Unlike the grid-form *Cause* whose order can be represented by the raster scanning order, the point set *Cause* cannot be converted to a vector directly. Meanwhile, neither feature extraction nor down sampling is applicable to it. Though that any distribution can be modeled as a Gaussian mixture model (GMM) by the EM algorithm [7], the problem of description length still persists in the parametric distribution *Cause*.

Motivated by the limitations of vectoral *Cause*, this article proposes a novel Bayesian classifier called Bayesian signal classifier (BSC), which performs a classification on the continuous domain. In BSC, *Cause* is represented as a continuous function $f(\mathbf{x})$ where $\mathbf{x} \in \Re^n$. It is proven in [8] [9] that any real function can be approximated as a GMM by giving sufficient number of samples. This leads to a unified expression of BSC when all functional *Causes* are modeled as GMMs. Chow and Lee [12] showed that GMM can highly compress the energy of grid-form data. This infers a simpler topology of BSC by comparing with those of vectoral *Causes* In addition; continuous *Cause* avoids the description length selection problem as no quantization is involved. Since BSC does not involve the concept "feature", the quality of feature extraction need not be considered in the performance measurement of BSC. The EM algorithm makes BSC also valid in the pattern recognition of scattered-point (distribution) form *Cause*.

Though *Cause* appears as a continuous signal in the real world, it is commonly discretized as a scattered point set by sensors like cameras and microphone. Thus, a regression is necessary to convert the received samples back to a continuous form. Unlike feature extraction and down-sampling that reduce pattern input dimension, regression performs a nearly lossless conversion of pattern input representation.

The most relevant work to this article is the Volterra series expansion model. It was firstly introduced by Vito Volterra [2] in 1959. Volterra series describe the output $y(t)$ of a nonlinear system of $x(t)$ as the sum of the responses of a 1st order, 2nd order, 3rd order operators and so on:

$$y(t) = \sum_{n=1}^{\infty} \frac{1}{n!} \int_{-\infty}^{\infty} du_1 \cdots \int_{-\infty}^{\infty} du_n g_n(u_1, \ldots, u_n) \prod_{r=1}^{n} x(t - u_r)$$

Every operator is described in time or frequency domain with a transfer function called the Volterra kernel. A Volterra representation can be regarded either as a black box or a circuit level description. Black box Volterra presentation with memory effects is described by Le Gallou et al. [13]. Though a reasonably good correlation between modeled and measured memory is reported, the Volterra description is empirical and must be characterized at the desired operating point.

Zyla and Figueiredo [3] extended the idea of Volterra series to predict the output of a given continuous function. Since [3] does not specify the form of the input function, the formulation is an application dependent framework. In addition, it may lead to an undetermined model formulation as the integration of certain functions has no analytic solution. Even the model of Zyla and Figueiredo is derived in continuous domain, Panagiotopoulos et al. [4] rewrote it in discrete time so as to simplify the complicated expression of functional weights.

By comparing with the model of Zyla and Figueiredo, the proposition of BSC is motivated by different reasons. Firstly, the model of Zyla and Figueiredo is derived from the Volterra series that mainly works on nonlinear circuit analysis. Though the concept of continuous input has been proposed in [3], the actual implementation in [4] still keeps in discrete domain for simplicity. Secondly, [3] performs regression while the proposed model targets on pattern classification.

The rest of this article is organized as follows: In Section II, we define a similarity measure between two functions. By approximating the integration of a Gaussian function as a sigmoid function, the analytical expression of the functional similarities is formulated. A signal clustering algorithm is presented in Section III. In section IV, we propose a signal density estimation method and hence a signal Bayesian classifier is constructed. Section V demonstrates the performance of signal clustering and BSC by two sets of simulations including signal clustering and signal classification. A conclusion is drawn in Section VI.

II. From Vector to Function

A. Functional Distance

Similarity plays an important role in density estimation and classification, and the Euclidean distance is the common measurement. In this section, the Euclidean distance is extended to continuous domain, namely *functional distance*, which is a methodology for measuring similarity between functions. The two functions $g(\mathbf{x})$ and $f(\mathbf{x})$ are represented as GMMs (Eq. (1) and Eq. (2) respectively) in the following derivation.

$$g(\mathbf{x}) = \sum_j m_j \exp\left(-\frac{\|\mathbf{x} - \boldsymbol{\beta}_j\|}{2\alpha_j^2} \right) + A \qquad (1)$$

and

$$f(\mathbf{x}) = \sum_i w_i \exp\left(-\frac{\|\mathbf{x} - \boldsymbol{\mu}_i\|}{2\sigma_i^2} \right) + B \qquad (2)$$

where $\{w_i\}$, $\{m_i\}$ are the kernel weights, $\{\beta_j\}$ and $\{\mu_j\}$ are the kernel mean vectors, $\{\alpha_j^2\}$ and $\{\sigma_i^2\}$ are the kernel variances, A and B are the kernel biases and $\mathbf{x} \in X = \Pi_i[X^-_k, X^+_k] \subset \Re^n$.

The similarity between two functions $f(\mathbf{x})$ and $g(\mathbf{x})$, namely functional distance $\|f(\mathbf{x}) - g(\mathbf{x})\|_X$, is defined as the integration of the squared difference over the range X:

$$\|f(\mathbf{x}) - g(\mathbf{x})\|_X = \int_X (f(\mathbf{x}) - g(\mathbf{x}))^2 d\mathbf{x} \qquad (3)$$

As we represent any function as a Gaussian mixture model, in addition to that the product of two Gaussian functions remains Gaussian, eq. (3), the functional distance can be simplified as an integration of a GMM:

$$= \sum_a C_a \int_X \exp\left(-\frac{\|\mathbf{x} - \mathbf{l}_a\|}{2q_a^2} \right) d\mathbf{x} + \int_X (A - B)^2 d\mathbf{x}$$

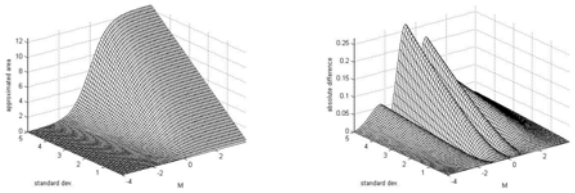
$$= \sum_a C_a \prod_k \int_{X_k^-}^{X_k^+} \exp\left(-\frac{(x_k - l_{a,k})^2}{2q_a^2} \right) dx_k + (A - B)^2 \prod_k (X_k^+ - X_k^-) \qquad (4)$$

B. Approximation of Gaussian function integration

Since the functional distance involves Gaussian function integration which has no analytical solution, we propose to approximate it as a sigmoid function:

$$\int_a^b \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)dx \approx V_a(\mu,\sigma,[a,b]) = \left.\frac{\sigma\sqrt{2\pi}}{1+\exp\left(-\frac{1.72254(x-\mu)}{\sigma^{0.9772}}\right)}\right|_a^b$$

To illustrate the accuracy of the approximation, we study the difference between the approximation and the integration derived by numerical method V_T (the trapezoidal rule is employed in this section). As the interval is divided into 10^5 divisions, the precision of V_T is sufficient to represent the actual integration. Fig. 1(a) shows the approximation of a 1D Gaussian integration by the proposed expression where $\sigma \in [0.1, 5]$, $M \in [-4, 4]$, $a = M\sigma$ and $b = -4\sigma$. Fig. 1(b) shows the absolute differences between the approximations of the proposed expression and the trapezoidal rule. The differences are kept at a low level (max. difference is less than 0.25).



(a) (b)

Fig. 1. The integration of Gaussian function (a) by the proposed method and (c) the corresponding error.

III. Signal Clustering

The k-means algorithm is a common clustering method which clusters patterns based on attributes into k partitions. Its objective is to determine the k means of data generated from gaussian distributions. In this section, the k-means algorithm is extended to continuous domain for clustering signal sets. Different from the conventional representation, the cluster; namely functional cluster mean $\mu(\mathbf{x})$; describes the averaged signal within it. Suppose that $\mathbf{T}$ is a signal set consisting of m signals $\{f_i(\mathbf{x})\}_{i\in[1,m]}$ where $\mathbf{x} \in X \subset \Re^n$. The proposed algorithm aims at partitioning $\mathbf{T}$ into k signal clusters such that the total intra-cluster variance E is minimized:

$$E = \sum_{i=1}^{m}\sum_{j=1}^{k}\left\|f_i(\mathbf{x}) - \mu_j(\mathbf{x})\right\|_X$$

The details of the algorithm are summarized as in the following:

1. The functional cluster means $\mu_i^{(1)}(\mathbf{x})$ for $i \in [1, k]$ is initialized as $f_j(\mathbf{x})$. The superscript (i) represents the value at the i^{th} iteration. The $f_j(\mathbf{x})$ is randomly selected and all $\mu_i^{(1)}(\mathbf{x})$ are distinct, i.e. $\mu_i^{(1)}(\mathbf{x}) \neq \mu_j^{(1)}(\mathbf{x})$ for $i \neq j$.
2. Compute the signal set $\mathbf{U_i} = \{g_{i,a}(\mathbf{x})\} \subset \mathbf{T}$ for the i^{th} cluster. $f_j(\mathbf{x})$ is an element of $\mathbf{U_i}$ if:

$$i = \arg\min_a \left\|f_j(\mathbf{x}) - \mu_a^{(t)}(\mathbf{x})\right\|_X$$

3. The signal cluster means are updated as:

$$\mu_i^{(t+1)}(\mathbf{x}) = \sum_{a=1}^{n} g_{i,a}(\mathbf{x})$$

4. Repeat step 2 until all signal cluster means remain unchanged in two consecutive iterations, i.e. $\mu_i^{(t+1)}(\mathbf{x}) = \mu_j^{(t)}(\mathbf{x})$ for all i.

IV. Signal Density Estimation

Many problems in computer vision involve obtaining the probability density function (p.d.f.) describing an observed random quantity. In general, the forms of the underlying density functions are not known and one can model it as a Gaussian mixture model:

$$D(f(\mathbf{x})) = \sum_{j=1}^{k}\frac{\alpha_j}{\sigma_j\sqrt{2\pi}}\exp\left(-\frac{\left\|f(\mathbf{x}) - \mu_j(\mathbf{x})\right\|_{X_I}}{2\sigma_j^2}\right) \qquad (5)$$

where $\mu_j(\mathbf{x})$, σ_j^2 and α_j are the mean vector, variance and weight of the j^{th} kernel respectively. The Expectation-Maximization (EM) algorithm is a common algorithm to determine the maximum likelyhood parameters of a mixture of k Gaussian kernels in the feature space. We briefly describe the basic steps of the extension of EM algorithm for Gaussian mixture model of a signal set. The distribution of a random signal $f(\mathbf{x})$ is a mixture of k Gaussian kernels if its density function is:

$$p(f(\mathbf{x})\,|\,\boldsymbol{\theta}) = \sum_{j=1}^{k}\frac{\alpha_j}{\sigma_j\sqrt{2\pi}}\exp\left(-\frac{\left\|f(\mathbf{x}) - \mu_j(\mathbf{x})\right\|_{X_I}}{2\sigma_j^2}\right) \qquad (6)$$

where the parameter set $\theta = \{\mu_j(\mathbf{x}), \Omega_j, \alpha_j > 0\}_{j\in[1,k]}$. Given the current estimation of the parameter set θ, each iteration t of the extended EM algorithm re-estimates the parameter set according to the following steps:

Expectation step:

$$w^{(t)}_{i,j} = \frac{p(f_i(\mathbf{x})\,|\,\boldsymbol{\theta}^{(t)})}{\sum_{l=1}^{k} p(f_i(\mathbf{x})\,|\,\boldsymbol{\theta}^{(t)})} \qquad (7)$$

for $j \in [1, k]$ and $i \in [1, n]$. The term $w^{(t)}_{i,j}$ is the posterior probability that the signal $f_i(\mathbf{x})$ was sampled from the j^{th} component of the mixture distribution.

Maximization step:

$$\alpha_j^{(t+1)} = \frac{W_j^{(t)}}{n} \qquad \mu_j^{(t+1)}(\mathbf{x}) = \frac{1}{W_j^{(t)}}\sum_{i=1}^{n} w^{(t)}_{i,j} f_i(\mathbf{x})$$

$$\Omega_j^{(t+1)} = \frac{1}{W_j^{(t)}}\sum_{i=1}^{n} w^{(t)}_{i,j}\left\|f_i(\mathbf{x}) - \mu_j^{(t)}(\mathbf{x})\right\|_{X_I} \qquad (8)$$

where $W_j^{(t)} = \sum_i w_{i,j}^{(t)}$. The parameter set θ of the density function is initialized by the signal clustering algorithm in section 3.

Thus the signal density estimation algorithm extends the Bayesian classifier to handle continuous signal, the corresponding classifier is named Bayesian Signal Classifier (BSC). BSC determines the class label of a continuous signal $f(\mathbf{x})$ based on its density given that $f(\mathbf{x})$

belongs be c_i. According to the Bayesian theorem, the probability that $f(\mathbf{x})$ belongs to $c_f \in \mathbf{C}$ is:

$$P(c_f \mid f(\mathbf{x})) = \frac{P(f(\mathbf{x}) \mid c_f)P(c_f)}{\sum_{c \in \mathbf{C}} P(f(\mathbf{x}) \mid c)P(c)} \quad (9)$$

where $\mathbf{C} = \{C_i\}$ is the set of class labels. The p.d.f.s on the right hand side of (9) are formulated based on the signal training set $\mathbf{T} = [\{f_i(\mathbf{x}) \mid c_i\}]$. Thus, the input signal $g(\mathbf{x})$ is classified as class label c_b where

$$b = \arg\max_j P(c_j \mid g(\mathbf{x})) \quad (10)$$

V. Simulation results

A. Simulation 1 – Signal Clustering

In this simulation, the performance of the signal clustering algorithm is evaluated in the following: Suppose that there are three function sets:

$$\mathbf{F}_1 = \left\{ w \exp\left(-\frac{(x-\mu)^2}{2\sigma^2} \right) \right\}$$
$$\mathbf{F}_2 = \left\{ m \sin\left(2\pi a(x+0.5)\right) + 0.5 \right\}$$
$$\mathbf{F}_3 = \left\{ x^q \right\}$$

where $x \in [0, 1]$, $w \in [0.5, 1]$, $\sigma = [0.1, 0.2]$, $\mu \in [0.35, 0.75]$, $m \in [0.25, 0.375]$, $a \in [1.6, 2]$ and $q = [0.5, 2.5]$. We denote by an operator $G(.)$ as a GMM representation of a given function $f(x)$, i.e. $G(f(x))$, which is the regressive function of the point set $\{[0 \mid f(0)], [0.05 \mid f(0.05)], \ldots, [1 \mid f(1)]\}$ obtained by support vector machine [1]. The given function is described by 100 points in which $G(.)$ can fully represent $f(x)$ in the form of GMM. The sample signal set $\mathbf{T}$ to be clustered is defined as:

$$\mathbf{T} = \{r_i(x)\}_{i \in [1, 10]} \cup \{h_i(x)\}_{i \in [1, 10]} \cup \{z_i(x)\}_{i \in [1, 10]}$$

where $r_i(x) \in G(\mathbf{F}_1)$, $h_i(x) \in G(\mathbf{F}_2)$ and $z_i(x) \in G(\mathbf{F}_3)$. The values of w, σ^2, μ and q are randomly selected within the corresponding ranges under uniform distribution. Totally 30 signal patterns are clustered in this simulation. The number of signal clusters k is chosen as 3 and the clusters are namely k_1, k_2 and k_3. Fig. 2 shows the signal subsets of each cluster. In order to illustrate the generalization of the clusters, three testing signal sets: $\mathbf{E}_1 \subset G(\mathbf{F}_1)$, $\mathbf{E}_2 \subset G(\mathbf{F}_2)$ and $\mathbf{E}_3 \subset G(\mathbf{F}_3)$ are generated and partitioned by the clusters k_i. Every testing set consists of 100 signals where the signal parameters are chosen randomly. We define $\mathbf{K}_i$ as the subset of each testing set that belongs to k_i. Table I lists the clustering results of the three testing sets.

TABLE I
SIZES OF THE SIGNAL SUBSETS $\mathbf{K}_1$, $\mathbf{K}_2$ AND $\mathbf{K}_3$

	$\mathbf{K}_1$	$\mathbf{K}_2$	$\mathbf{K}_3$
$\mathbf{E}_1$	97	1	2
$\mathbf{E}_2$	1	98	1
$\mathbf{E}_3$	3	1	96

Seen from the table, each of the three clusters $\{k_i\}$ is supported by a nearly distinct function set (i.e. the cluster impurity is less than 5%).

We define n_k as the number of kernels (optimal description length) of a signal pattern. Table II lists the distributions of n_k of $\mathbf{T}$, $\mathbf{E}_1$, $\mathbf{E}_2$ and $\mathbf{E}_3$ in this simulation. Seen from the table, the values of n_k are different from patterns to patterns and hence from training set to testing set. Thus, it is expected that the conventional clustering algorithm is failed in this simulation. Moreover, it shows that the proposed clustering algorithm avoids the problem of optimal description length selection.

TABLE II
DISTRIBUTIONS OF n_k IN $\mathbf{T}$, $\mathbf{E}_1$, $\mathbf{E}_2$ AND $\mathbf{E}_3$ IN THE SIMULATION 1

n_k	1-3	4-7	8-10
$\mathbf{T}$	9	11	10
$\mathbf{E}_1$	100	0	0
$\mathbf{E}_2$	0	22	78
$\mathbf{E}_3$	0	89	11

B. Simulation 2 – Distribution Classification

In this simulation, the performance of BSC is presented for a problem in distribution classification. Given a finite set of unknown distributions $\mathbf{D} = \{p_i(\mathbf{x})\}$ where $p_i(\mathbf{x})$ is represented by a class label c_i, distribution classification aims at finding the correct class label (unknown distribution) of a given point set $\mathbf{S} = \{[x_{1,i}, x_{2,i}]\}$. The four distributions in this simulation are shown in Fig. 3. The details of distribution pattern generation for training and testing are presented in the following:

Algorithm 1: Pattern generation of simulation 2

1. Select the class label c of the current *Cause* randomly from the class label set $\mathbf{C} = \{C_1, C_2, C_3, C_4\}$.
2. Choose the size $n \in [20, 50]$ of a point set $\mathbf{S}$ randomly.
3. Generate a point set $\mathbf{S} = \{[x_{i,1}, x_{i,2}] \in [0, 1]^2\}_{i \in [1, n]}$ based on the p.d.f. $p(\mathbf{x}) \in \mathbf{D}$ of c.
4. Compute the empirical p.d.f. $f(\mathbf{x})$ of $\mathbf{S}$ by the EM algorithm [25] involving MDL. The number of kernels of $f(\mathbf{x})$ is optimal in term of MDL score.
5. The current signal sample is defined as $[f(\mathbf{x}) \mid c]$.

The training set $\mathbf{T} = \{[f_i(\mathbf{x}) \mid c_i]\}$ consists of 100 signal samples. The BSC of $\mathbf{T}$ is constructed after estimating the density functions $P(f(\mathbf{x}) \mid c_i)$ and the prior probabilities $P(c_i)$ for $i \in [1, 4]$. The $P(c_i)$ is extracted from the class labels in $\mathbf{T}$, i.e. $P(c_i) = n_i / 100$ where n_i is the number of patterns with class label c_i. The patterns of testing set $\mathbf{E} = \{[f_j(\mathbf{x}) \mid c_j]\}$ are also generated by the *Algorithm 1*. In order to illustrate the generalization of the classifier, noises are embedded to $\mathbf{S}$ at the step 3 of *Algorithm 1*, $\mathbf{S} \leftarrow \mathbf{S} + \{[\eta_{i,x}, \eta_{i,y}]\}_{i \in [1,n]}$ where $\eta_{i,x}$ and $\eta_{i,y}$ are zero mean Gaussian random variable with variance v^2. For each value of v, 100 patterns are used to evaluate the trained BSC.

Table III lists the accuracy A of the classifier on classifying the testing set **E** with different values of v. Seen from the table, the accuracy of the BSC keeps at high level (above 80%) even the standard deviation of the Gaussian noise is up to 7% of the range of x_i.

TABLE III
ACCURACY A OF THE DISTRIBUTION CLASSIFIER

	Standard deviation of noise v ($\times 10^{-2}$)				
	5	6	7	8	9
A	94.3%	91.4%	81.9%	77.5%	63.4%

Table IV lists the distributions of n_k of **T** and **E** in this simulation. Similar to the previous simulation, no unique n_k can be concluded that the existing algorithms are failed to classify the pattern set **E**.

TABLE IV
DISTRIBUTIONS OF n_k IN **T** AND **E** IN THE SIMULATION 2

n_k	5-10	11-15	16-20
T	29	45	36
E (v = 0.05)	27	47	36
E (v = 0.06)	26	46	38
E (v = 0.07)	26	47	37
E (v = 0.08)	25	49	36
E (v = 0.09)	25	48	37

VI. CONCLUSION

In the field of pattern recognition, classifiers are developed based on the existence of the relations between *Causes* and *Effects*. Bayesian classifier constructs this relation from a set of posterior distributions supported by historical *Cause* and *Effect* pairs, namely training samples. *Causes* are usually in the form of continuous functions or sets of scattered points. To simplify the training process, *Cause* is commonly represented as a vector after quantization or feature extraction. However, these simplifications lead to a sequence of the neural network limitations. During the quantization, description length (the size of quantized vector) plays an important role to the network complexity and accuracy. A long description forms a complicated and accurate network while a short description provides a simple but inaccurate network. Furthermore, as the scattered-point-form *Cause* involves no ordering, none of the quantization methods is applicable. Though point distribution can be represented by GMM through the EM algorithm, the problem of description length selection still exists.

Driven by the limitations of vectoral *Causes*, we propose a novel Bayesian classifier called Bayesian signal classifier (BSC) that determines the class label of a given continuous signal. This feature makes a more accurate description to the relation between *Cause* and *Effect*. As a result, BSC avoids the following problems: 1) the selection of optimal description length and 2) inaccurate classification of scattered-points input pattern. In this article, two algorithms: signal clustering and signal density estimation are presented for the constructing of the posterior distribution. Different from the model of Zyla and Figueiredo [3], BSC performs pattern classification but not regression. In addition, BSC has advantage over [3] in model expression: All real functions are represented as Gaussian Mixture Model (GMM) in which an exact solution is computed. In addition, the GMM representation is immune from the problem of undefined function integration occurred in [3]. The simulation results show that both the signal clustering algorithm and BSC perform well on overcoming the description length selection problem and the limitation of distribution classification of vectoral *Cause*.

ACKNOWLEDGMENT

The work described in this article was fully supported by a grant from CityU (7001707).

REFERENCES

[1] V. N. Vapnik, *Statistical Learning Theory*, New York: Wiley, 1998.

[2] Volterra V., *Theory of Functionals and of Integral and Integro-Differential equations*, New York: Dover.

[3] L. V. Zyla and R. J. P. de Figueiredo, "Nonlinear system identification based on a Fock space framework", *SIAM J. Contr. Optimization*, vol. 21, no. 6, pp. 931 – 939, Nov. 1983.

[4] A. Panagiotopoulos, R. W. Newcomb and S. K. Singh, "Planning with a Functional Neural-Network Architecture", *IEEE Transactions on Neural Networks*, vol. 10, no. 1, pp. 115- 127, Jan. 1999.

[5] M. Small and C. K. Tse, "Minimum description length neural networks for time series prediction", *Phys. Rev. E*, *American Physical Soc.*, College PK, USA, Dec. 2002, 066701.

[6] K. Judd and A. Mees, "On selecting models for nonlinear time series", *Physica D*, Elsevier Science BV, Amsterdam, Netherlands, May 1995, pp. 426 - 444.

[7] A. Dempster, N. Laird, and D. Rubin, "Maximum Likelihood Estimation from Incomplete Data via the EM Algorithm", *J. Royal Statistic Soc.*, vol. 30, no. B, pp. 1 - 38, 1977.

[8] B. Liu and J. Si, "The best approximation to C functions and its error bounds using regular-center Gaussian networks", *IEEE Trans. Neural Networks*, vol. 5, pp. 845 – 847, 1994.

[9] E. J. Hartman, J. D. Keeler, and J. M. Kowalski, "Layered neural networks with Gaussian hidden units as universal approximations", *Neural Computation*, vol. 2, no. 2, pp. 210–215, 1990.

[10] A. Burton, V. Bruce and N. Dench, "What's the Difference Between Men and Women? Evidence from Facial Measurements", *Perception*, vol. 22, pp. 153 - 176, 1993.

[11] R. Brunelli and T. Poggio, "HyberBF Networks for Gender Classification", *DARPA Image Understanding Workshop*, pp. 311 - 314, 1992.

[12] Kai Tik Chow D. and Tong Lee, "Image approximation and smoothing by support vector regression", *Proceedings of*

International Joint Conference on Neural Networks 2001, vol. 4, pp. 2427 - 2432.

[13] Le Gallou N., Ngoya E., Buret H., Barataud D. and Nebus J. M., "An improvement behavioral modeling technique for high power amplifiers with memory", *Proceeding of the IEEE International Microwave Symposium*, Phoenix, Arizona, 2000.

[14] H. Li, X. Liao, Lawrence Carin, "A Reward-Directed Bayesian Classifier"*, Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing 2006*, vol. 5, pp.613 – 616.

[15] Z. Z. Shi, Y. P. Huang, S. L. Zhang, "Fisher Score Based Naive Bayesian Classifier", *Proceedings of the International Conference on Neural Networks and Brain 2005*, vol. 3, pp. 1616 – 1621.

[16] R. Larsen, "3-D contextual Bayesian classifiers"*, IEEE Transactions on Image Processing*, vol. 9, no. 3, pp. 518 – 524, 2000

[17] Y. Gurwicz, B. Lerner, "Rapid spline-based kernel density estimation for Bayesian networks", *Proceedings of the 17th International Conference on Pattern Recognition 2004*, vol. 3, pp. 700 – 703.

[18] B. Krishnapuram, A. J. Harternink, L. Carin, M. A. T. Figueiredo, "A Bayesian approach to joint feature selection and classifier design", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1105 – 1111, 2004.

[19] S. Aksoy, K. Koperski, C. Tusk, G. Marchisio, J. C. Tilton, "Learning bayesian classifiers for scene classification with a visual grammar", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 3, pp. 581 – 589, 2005.

[20] C. E. Thomaz, D. F. Gillies, R. Q. Feitosa, "A new covariance estimate for Bayesian classifiers in biometric recognition", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 2, pp. 214 – 223, 2004.

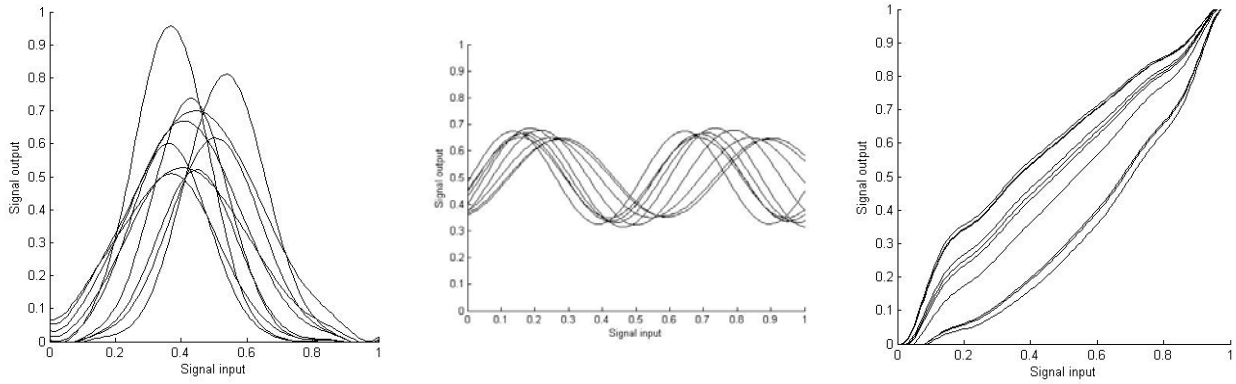
[21] S. L. Phung, A. Sr. Bouzerdoum, D. Sr. Chai, "Skin segmentation using color pixel classification: analysis and comparison", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 1, pp. 148 – 154, 2005.

[22] T. P. Weldon, "Improved Image Segmentation With A Modified Bayesian Classifier", *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing 2006*, vol. 2, pp. 697 - 700.

[23] Shu-Fai Wong, K.-Y. Wong, R. Cipolla, Robust "Appearance-based Tracking using a sparse Bayesian classifier", *Proceedings of the 18th International Conference on Pattern Recognition 2006*, vol. 3, pp. 47 – 50.

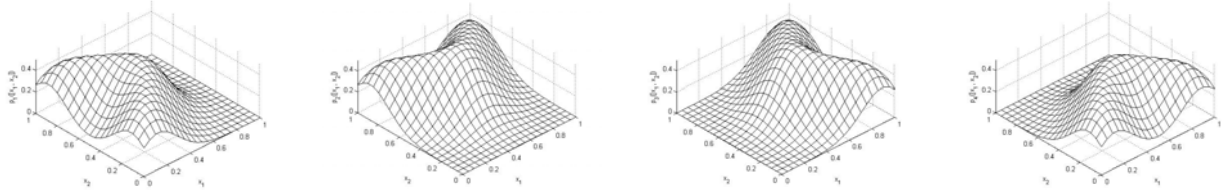
[24] J. Y. Goulermas, A. H. Findlow, C. J. Nester, D. Howard, P. Bowker, "Automated Design of Robust Discriminant Analysis Classifier for Foot Pressure Lesions Using Kinematic Data", *IEEE Transactions on Biomedical Engineering*, vol. 52, no. 9, pp. 1549 – 1562, 2005.

[25] W. Lam and F. Bacchus, "Learning Bayesian belief networks: An approach based on the MDL principle", *Computational Intelligence*, vol. 10, pp. 269 - 293, 1994.



(a) (b) (c)

Fig. 2. Clustered signals of **T** in (a) k_1 (b) k_2 and (c) k_3.



(a) (b) (c) (d)

Fig. 3. Density functions $p_i(x)$ of (a) C_1 (b) C_2 (c) C_3 and (d) C_4