Online Probability Density Estimation of Nonstationary Random Signal using Dynamic Bayesian Networks

Hyun Cheol Cho, M. Sami Fadali, and Kwon Soon Lee*

Abstract: We present two estimators for discrete non-Gaussian and nonstationary probability density estimation based on a dynamic Bayesian network (DBN). The first estimator is for off-line computation and consists of a DBN whose transition distribution is represented in terms of kernel functions. The estimator parameters are the weights and shifts of the kernel functions. The parameters are determined through a recursive learning algorithm using maximum likelihood (ML) estimation. The second estimator is a DBN whose parameters form the transition probabilities. We use an asymptotically convergent, recursive, on-line algorithm to update the parameters using observation data. The DBN calculates the state probabilities using the estimated parameters. We provide examples that demonstrate the usefulness and simplicity of the two proposed estimators.

Keywords: Dynamic bayesian networks, nonstationary signal, online estimation, probability density function.

1. INTRODUCTION

Probability density function (pdf) estimation is an important problem for many engineering applications including pattern recognition, signal detection, artificial intelligence, etc. Several parametric and nonparametric techniques are available for pdf estimation [1]. Parametric methods are the simplest to estimate the parameters of a pdf of known form. A Gaussian distribution is widely used as a parametric density model to simplify statistical analysis. In many practical applications, non-Gaussian distributions of unknown form are encountered. In these applications nonparametric methods, which are more suited to problems where the form of the distribution is unknown, are more useful. The simplest nonparametric approach available is the histogram. Unfortunately, the histogram is inherently discontinuous and requires a large amount of data to obtain useful results. Another popular nonparametric method is the kernel based approach in which an estimator is constructed using a set of kernel functions. The best known kernel based approach is the Parzen-window estimation [2] which uses Gausssian kernels. For the best performance, an appropriate kernel function and its parameter values are chosen for specific data samples. However, the method is very sensitive to parameter values and it is often difficult to determine the optimal kernel. Thus, additional computation, such as smoothing [3], is sometimes required.

In recent years, advanced techniques have been used for pdf estimation, including: soft computation algorithms [4-7], machine learning [8-10], and information theory [11-13], statistical inference [14], [15], etc. Most of this work employed neural network learning trained using a given data set. Although the objective functions used are somewhat different, learning is similarly accomplished by finding the extremum of an information-based objective function. Examples of the objective function are: differential entropy, and the Kullback-Leibler function. Several neural model are used, including: multilayered back-propagation neural networks [4,6,8], the Self-Organizing Map (SOM) [9,10,16], the sigmoidal network model [5], and probability principal component analysis (PPCA) [15,17]. More recently, dynamic density estimation was developed based on mixture of Gaussian distributions for autoregressive processes in [18].

Most research to date has focused on offline or batch learning in which datasets are obtained from actual processes or experimental simulations. In

Manuscript received August 29, 2006; revised April 2, 2007 and August 1, 2007; accepted November 9, 2007. Recommended by Editor Jin Young Choi. This work was supported by the Korea Science and Engineering Foundation (KOSEF) through the National Research Lab. Program funded by the Ministry of Science and Technology (No. M103000003 0306J000030310).

Hyun Cheol Cho and Kwon Soon Lee are with the Dept. of Electrical Engneering, Dong-A University, 840, Hadan 2-dong, Saha-gu, Pusan 604-714, Korea (e-mails: hyunccho@gmail.com, kslee@dau.ac.kr).

M. Sami Fadali is with the Dept. of Electrical Engineering, Univ. of Nevada-Reno, Reno, NV 89557, USA (e-mail: fadali@unr.edu).

* Corresponding author.

addition, most methodologies are only applicable to Gaussian density functions and stationary statistics, and do not incorporate information regarding temporal causality for a dynamic system. Although the methods above give excellent results for some applications, they have their limitations that exclude their use in others. Online estimation can eliminate many of these drawbacks by adaptively updating the pdf for stationary statistical systems [19]. However, online implementation requires that the computational burden be kept low by using a simple learning formula. Unfortunately, most available approaches cannot easily be adapted for online computation since their estimation rule involves calculating the gradient of an objective function.

We estimate unknown pdfs with unknown statistical characteristics online using a DBN approach. A DBN is a graphical reasoning model for dynamic systems to statistically represent its temporal causality [20]. We develop online parameter learning for our DBN suitable avoiding the computationally costly gradient-based framework. Thus, our proposed estimation is recursively computed with a simple update rule. Two pdf estimation methods are presented using a discrete Markov Chain (MC) model, which is the simplest DBN. It expresses a posterior probability in terms of prior probabilities and transition probabilities. In the first method we represent the transition probabilities as a weighted sum of shifted kernel functions. We then express the probability of feasible variables in terms of prior probabilities. The optimal parameter values in the estimator are determined via a learning algorithm that maximizes the likelihood function using gradient descent optimization. In the second estimator, the posterior probability is alternatively expressed in vector form in order to construct a simple DBN whose parameters form the transition distribution. We estimate the DBN parameters from a given observation sequence and then compute the posterior probability density. The asymptotic behavior of the estimator is analytically investigated using a stochastic convergence theorem.

The remainder of this paper is as follows: We provide a brief review of kernel-based pdf estimation and a discrete MC model in Sections 2 and 3 respectively. In Sections 4 and 5, we propose our estimation approaches using a DBN model. Convergence analysis for the online estimation is studied in Section 6 and a simulation example is presented in Section 7. Finally, conclusion and future work are given in Section 8.

2. KERNEL-BASED PROBABILITY ESTIMATION

A kernel based estimate is typically a linear combination of kernel (basis) functions of the form:

$$P(x) = \sum_{i=1}^{N} \alpha_i \phi(x - x_i, \beta_i), \tag{1}$$

where x is a $N\times 1$ data vector, $P(x)$ is the probability of x, ϕ is a kernel function, and α_i and β_i are parameters. Based on probability axioms, the kernel functions must satisfy $\phi \geq 0$ with

$$\int \phi(u)du = 1 \tag{2}$$

and the parameters α_i must be positive. The values of the parameters α_i and β_i determine the performance of the estimator and must be chosen appropriately. A simple method for choosing the parameters is to plot several curves with different parameter values and visually examine them to search for the best fit. However, this approach is subjective, requires skill and experience, and cannot be automated.

We use ML estimation [21] to determine the optimal parameter values in (1). Assuming independent data points x_i, $i = 1, \dots, N$, the objective function is defined as

$$L_o(x \mid \alpha, \beta) = \prod_{i=1}^{N} P(x_i). \tag{3}$$

Taking the natural logarithm of (3), we obtain the log likelihood

$$\begin{aligned} L(x \mid \alpha, \beta) &= \ln(L_o) \\ &= \sum_{i=1}^{N} \ln P(x_i) \\ &= \sum_{i=1}^{N} \ln \left\{ \sum_{j=1}^{N} \alpha_j \phi(x_i - x_j, \beta_j) \right\}. \end{aligned} \tag{4}$$

To maximize likelihood, we minimize the following objective function with respect to two parameter vectors $\alpha = [\alpha_1 \cdots \alpha_N]^T$ and $\beta = [\beta_1 \cdots \beta_N]^T$

$$J = \min_{\alpha,\beta} \left\{ -\sum_{i=1}^{N} \ln \left\{ \sum_{j=1}^{N} \alpha_j \phi(x_i - x_j, \beta_j) \right\} \right\}. \tag{5}$$

It is popular to use gradient descent optimization for determining the two parameter vectors because of its high initial convergence rate [21]. For gradient descent, we express the update rules for the parameters as

$$\alpha_j(k+1) = a_j(k) - \eta \frac{\partial J}{\partial \alpha_j} \tag{6}$$

and

$$\beta_j(k+1) = \beta_j(k) - \eta \frac{\partial J}{\partial \beta_j}, \tag{7}$$

where k is discrete time and $\eta < 0$ is the learning rate. We expand the partial derivatives using the chain rule as

$$\frac{\partial J}{\partial \alpha_j} = \frac{\partial J}{\partial P(x_i)} \frac{\partial P(x_i)}{\partial \alpha_j} = \sum_{i \neq j}^{N} \frac{1}{P(x_i)} \phi(x_i - x_j, \beta_j), \quad (8)$$

and

$$\frac{\partial J}{\partial \beta_j} = \frac{\partial J}{\partial P(x_i)} \frac{\partial P(x_i)}{\partial \beta_j} = \sum_{i \neq j}^{n} \frac{1}{P(x_i)} \left\{ \alpha_j \frac{\partial \phi(x_i - x_j, \beta_j)}{\partial \beta_j} \right\}. \quad (9)$$

The parameter values are determined using the recursions (6) and (7) with the derivatives expressed as (8) and (9).

3. DYNAMIC BAYESIAN NETWORKS

In this section, we briefly review DBN modeling of a MC for a random signal. MC models have been popular in engineering and science for decades. The primary reason is that many practical stochastic systems can be represented by a MC. The basic structure of a first-order MC is depicted in Fig. 1.

In Fig. 1, a random variable $X(k)$ at discrete time k with N distinct states is temporally dependant on its state at time k–1. The state probability $p(X = i)$, $i = 1, \cdots, N$ in the model sequentially evolves. Based on the Markov property [22], the joint probability of $X(k)$ of the model for a finite time interval $k = [0, T]$ is given by

$$p(X(0), \cdots, X(T)) = p(X(0)) \prod_{i=1}^{T} p(X(i) \mid X(i-1)), \quad (10)$$

where $p(X(0)) \in \Re^N$ is the initial state probability and the conditional probability represents a sequential transition state for the variable. In practice, the initial state probability is obtained from a specified random distribution, while the conditional probability must be estimated based on observations. The conditional probability as a model parameter is defined as

$$a_{ij} = p(X(k) = i \mid X(k-1) = j), i, j = 1, \cdots, N \quad (11)$$

subject to the probability constraint

$$\sum_{i=1}^{N} a_{ij} = 1, \qquad i = 1, \ldots, N. \quad (12)$$

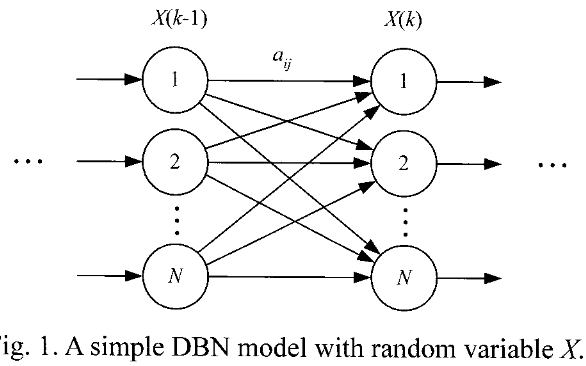


Fig. 1. A simple DBN model with random variable X.

4. DBN KERNEL-BASED DENSITY ESTIMATION

In this section, we propose a new probability estimation method using a kernel based approach and a simple DBN model shown in Fig. 1. In this model, the posterior probability of an $N\times1$ state vector X at k is given by

$$P(x_j(k)) = \sum_{i=1}^{N} P(x_j(k) \mid x_i(k-1)) P(x_i(k-1)), \quad (13)$$

where for simplicity, $x_j(k) \equiv \{X(k)=j\}$ and $P(x_i(k$-1$))$ is the prior probability of x_i whose initial condition is usually based on a subjective decision or random distribution [20]. For specific values $z_i \in \Re^N$ of x_i in (13), we have

$$P(x_j(k)) = \sum_{i=1}^{N} P(x_j(k) \mid z_i(k-1)) P(z_i(k-1)). \quad (14)$$

In (11), the transition probabilities are written as a linear combination of kernel functions as in (1). Substituting from (1) we have the posterior probability is given by

$$P(x_j(k)) = \sum_{i=1}^{N} \left\{ \sum_{j=1}^{N} \alpha_{ij} \phi(x - z_j, \beta_j) \right\} P(z_i(k-1)). \quad (15)$$

This posterior probability is updated for a random vector X at time k as depicted in Fig. 2.

As stated in Section 2, the parameter sets α and β in Fig. 2 are recursively adjusted via a learning

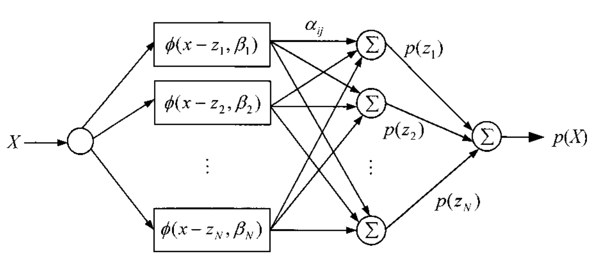


Fig. 2. Posterior probability update using kernel functions.

algorithm. The objective function for the learning algorithm is given by

$$J = \min_{\alpha,\beta}\left\{-\sum_{n=1}^{T}\ln\{P(X(n))\}\right\}$$
$$= \min_{\alpha,\beta}\left\{-\sum_{n=1}^{T}\ln\left\{\sum_{i=1}^{N}\left[\sum_{j=1}^{N}\alpha_{ij}\phi(x_n - z_j,\beta_j)\right]P(z_i)\right\}\right\}. \tag{16}$$

Using (6), (7), (8), and (9) of Section 2, the adjustment rules of the parameters are obtained as

$$\alpha_{ij}(k+1) = \alpha_{ij}(k) - \eta\sum_{n=1}^{N}\left\{\frac{1}{P(x_n)}\phi(x_n - z_j,\beta_j)P(z_i)\right\} \tag{17}$$

and

$$\beta_j(k+1) = \beta_j(k) - \eta\sum_{n=1}^{N}\frac{1}{P(x_n)}\left\{\sum_{i=1}^{N}\alpha_{ij}P(z_i)\frac{\partial\phi(x_n - z_j,\beta_j)}{\partial\beta_j}\right\}. \tag{18}$$

The following examples illustrate the use of our kernel-based method for pdf estimation. In the first example, we correctly estimate a Poisson pdf from data. In the second example we estimate the pdf for nonlinearly transformed Poisson data and compare the results to those of the histogram approach.

Example I-1: We estimate the discrete probability vector of a non-Gaussian signal. We use (17) and (18) with the Gaussian kernel function:

$$\phi(x_n - z_j,\beta_j) = \frac{1}{\sqrt{2\pi}\beta_j}\exp\left\{-\frac{(x_n - z_j)^2}{\beta_j^2}\right\}. \tag{19}$$

For simplicity, the prior probabilities of the values z_i $i=1,\ldots,N$, in (15) are assumed equal, i.e., $P(z_i) = 1/N$. First, we simulate the estimation of the Poisson distribution:

$$P(x = k) = \exp(-\lambda)\frac{\lambda^k}{k!}, \tag{20}$$

where λ is a parameter and $k = 0, 1, \ldots, \infty$. Fig. 3 shows N=100 Poisson distributed data samples for λ=10 generated using the MATLAB© command *poissrnd*.

The initial values of α and β were randomly selected as uniformly distributed in [0,1] with the learning rate η=0.5. Recursive learning continued until a specified error tolerance is reached. We define the error function is the average absolute difference between a reference probability p^* and the estimated probability

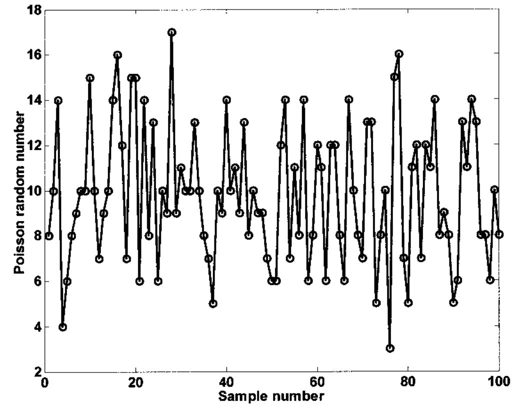


Fig. 3. Poisson random number (Example I-1).

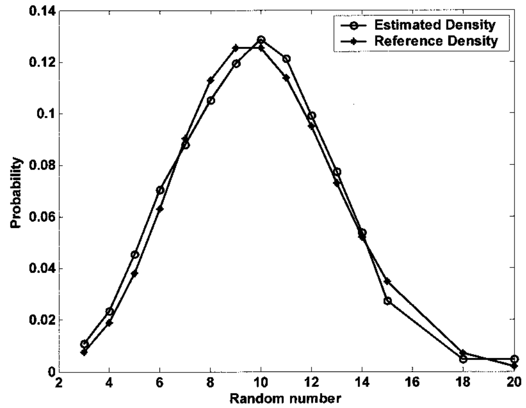


Fig. 4. Estimated probability (Example I-1).

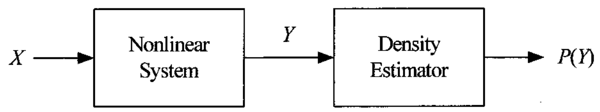


Fig. 5. Nonlinear transformation of Poisson distributed data.

P i.e.,

$$e = \frac{1}{N}\sum_{n=1}^{N}\left|P^*(x) - P(x)\right|. \tag{21}$$

In the training simulation, 100 training data samples at each time k were temporally generated. Fig. 4 illustrates simulation result of the estimated probabilities along with the reference values. The plot shows that the estimation errors are less than 0.01 at all data point with an average error value of 0.0089.

Example I-2: We nonlinearly transform the Poisson random data of Example I-1. To simulate this scenario, a random input is fed to a nonlinear system and the probabilities of the system output are estimated. A block diagram for this process is depicted in Fig. 5.

The random input is Poisson distributed with the same parameters as Example I-1. Because the data is nonlinearly transformed, the output probabilities are

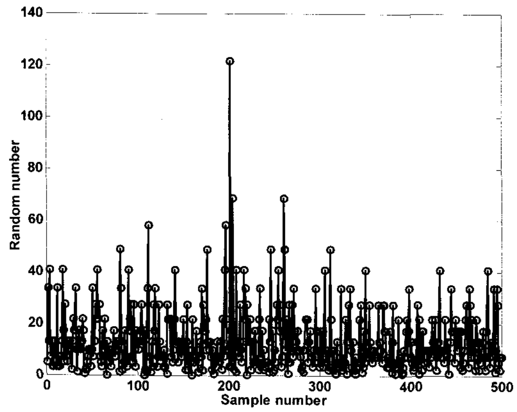


Fig. 6. Poisson random number (Example I-2).

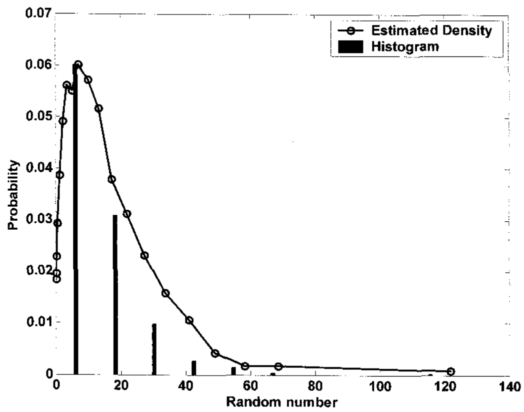


Fig. 7. Estimated probability (Example I-2).

not Poisson and the form of the output distribution is unknown. We define the nonlinear function in Fig. 5 as $y = c\,x^3(k)$ where c is constant and generate 500 random input values. The time history of the output is shown in Fig. 6.

Training is accomplished as in Example I-1. For comparison, we obtain an estimate of the pdf from the data using the histogram method and the MATLAB command *hist*. The simulation results are given in Fig. 7. Fig. 7 shows that our results are very similar to those of the histogram approach.

5. ONLINE ESTIMATION OF PROBABILITY DENSITY BY DBN

We present use of a DBN in pdf estimation in Section 4, but it is hard to construct an online estimation algorithm due to the gradient-type update rule. This Section proposes a novel online pdf estimation with a DBN whose structure is identical in Section 4.

A linear state-space model with the posterior probability vector as the state vector and with state matrix A is given by

$$P(X(k+1)) = A(k)P(X(k)), \tag{22}$$

where $P(X(k)) = [P(X(k)=1), \cdots, P(X(k)=N)]^T$ is a stochastic vector and $A(k)=[a_{ij}(k)]$, $i, j = 1, \cdots, N,$ is a time-dependent transition matrix. As estimates of the transition matrix are obtained from observations, the state vector of probabilities is recursively updated. The linear equation is relatively simple and the low computational load makes online estimation feasible.

To derive the recursive estimation rule, we first define the transition probability as

$$a_{ij}(k) = \rho\, m_{ij}(k), \qquad i,j = 1,\cdots,N, \tag{23}$$

where ρ is a normalizing factor to satisfy the constraint in (12), and m_{ij} is the time-average likelihood of transition between the ith state at k and the jth state at k–1. This parameter is recursively computed to reflect the related transition density using observation data. We express the parameter in the recursive form:

$$\begin{aligned} m_{ij}(k) &= \frac{1}{k}\sum_{n=1}^{k}\zeta_{ij}(n) \\ &= \left(\frac{k-1}{k}\right)m_{ij}(k-1) + \left(\frac{1}{k}\right)\zeta_{ij}(k), \end{aligned} \tag{24}$$

where $\zeta_{ij}(k)$ is a random variable chosen as zero or as equal to a positive constant c. Specifically, if the observation data at k and k–1 are $X(k\text{-}1) = j$ and $X(k) = i$ (or $X(k) = i \mid X(k-1) = j$), then $\zeta_{ij}(k) = c$, otherwise $\zeta_{ij}(k) = 0$, i.e.,

$$\zeta_{ij}(k) = \begin{cases} c, & \text{if } X(k) = i \text{ given } X(k-1) = j \\ 0, & \text{otherwise.} \end{cases} \tag{25}$$

The dynamic relationship for the posterior probability in (22) is graphically modeled as a simple DBN as shown in Fig. 8 where the dark circles are the two observation states at k and k–1. Thus, for this case ζ_{ij} is selected as c based on the rule (25). All other parameters ζ are zero and are represented by dotted lines in Fig. 8. As a result, the average likelihood m_{ij} is increased while other likelihoods are decreased. Finally, we normalize the likelihoods to obtain the

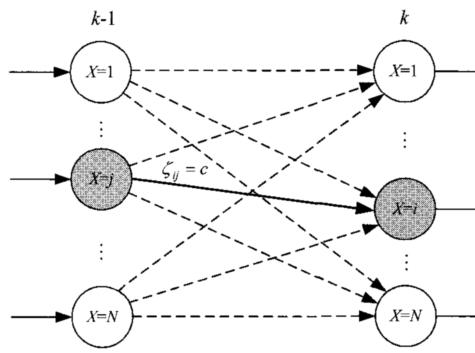


Fig. 8. Update of parameter a_{ij} based on observations.

new transition probability a_{ij}. All other entries of the transition matrix A are similarly updated. The estimated transition matrix is then used to update the stochastic vector in (22). The estimation algorithm is simple and efficient because of its recursive nature which eliminates the need to calculate earlier likelihoods. As illustrated by the following example, the estimation algorithm is suited to a large data set and real-time implementation.

Example II: This simulation example extends the simulation scenario of Example I-1 in which we generate 1000 nonstationary Poisson distributed data with random mean in [2,4] using the MATLAB© command *unidrnd*. Fig. 9 shows the random data for this example discretely ranging from 0 to 14. To construct our pdf estimate, we set up 21 states in (22), i.e., $P(X) = [P(X=0), \cdots ,P(X=20)]^T$ and selected the random initial probability vector $P(0)$ as uniformly distributed. We ran our estimation algorithm under these simulation conditions and plotted the trajectories of the state probabilities as shown in Fig. 10. We observe that the trajectories generally show large oscillations for the first 200 data points, then small sustained oscillations around a steady level. We infer that the sustained oscillations are caused by the nonstationary statistics of the observation data.

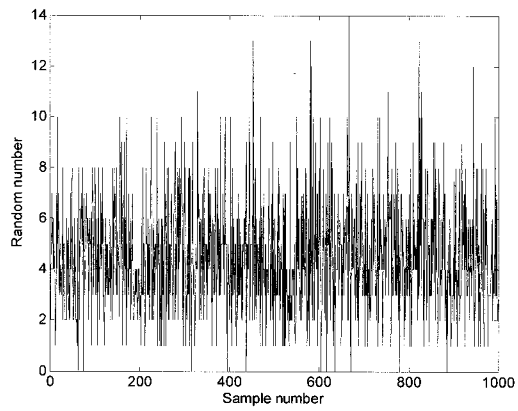


Fig. 9. Nonstationary Poisson random data (Example II).

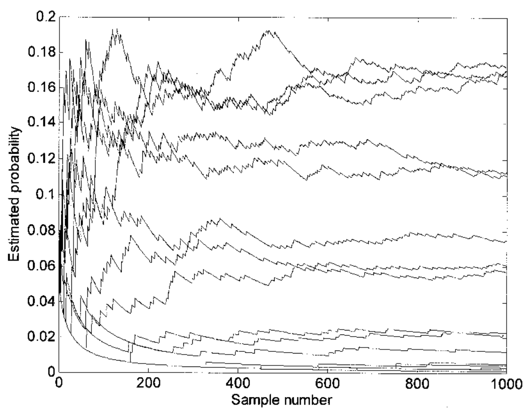


Fig. 10. Estimated state probabilities (Example II).

6. CONVERGENCE PROPERTY OF THE ESTIMATION

We study the asymptotic behavior of the proposed estimation algorithm of Section 5. We show that the estimator asymptotically converges under mild assumptions. We formulate our parameter update rule as a dynamic recursion and apply stability criteria for discrete-time systems to evaluate its stability.

6.1. Related theorems

In this Section, several definitions and theorems related to stochastic convergence are first introduced for later use to prove stochastic convergence.

Definition 1 [21]**:** Let $X(k)$ be a scalar sequence of random variables defined on a probability space Ω. We say that $X(k)$ converges with probability one (or strongly, almost surely, almost everywhere) to a random variable X if and only if

$$P\left\{ \omega \in \Omega : \lim_{k\to\infty} X(k,\omega) = X(\omega) \right\} = 1 \tag{26}$$

and we denote this by $X(k) \xrightarrow{WP1} X$. □

Definition 2 [21]**:** For a sequence of random variables X on a probability space Ω, $X(k)$ converges in probability to X^*, if, for $\varepsilon > 0$,

$$\lim_{k\to\infty} P\{| X(k) - X^* | > \varepsilon\} = 0. \tag{27}$$

This statement is denoted by $X(k) \xrightarrow{P} X^*$. □

These two types of convergence sometimes impose tough requirements on a stochastic process by constraining the behaviors of individual sample trajectories. Convergence in mean square imposes different constraints that relax these requirements.

Definition 3 [22]**:** A sequence of random variables $X(k)$ converges to X^* in a mean-squared sense, if

$$\lim_{k\to\infty} E\{(X(k) - X^*)^2\} = 0. \tag{28}$$

□

Theorem 1 [21]**:** If a sequence of random variables $X(k)$ converges to X^* in mean square, then it converges to X^* in probability.

Definition 4: Let $F(k)$ be a sequence of probability distribution function. If there is a distribution function F such that

$$\lim_{k\to\infty} F(k,X) = F(X) \tag{29}$$

at every point X in which F is continuous, we say that $F(k)$ converges in distribution to F. If $\{X(k)\}$ is a sequence of random variables and $\{F(k)\}$ is the corresponding sequence of distribution functions, we say that $X(k)$ converges in distribution to X if there is a random variable X with distribution function F such that $F(k)$ converges in distribution to F. We write $X(k) \xrightarrow{d} X$. □

Corollary 1 [21,23]**:** Convergence with probability one implies convergence in probability, but the converse is not true. Convergence in mean square implies convergence in probability, but the converse is not true. Convergence in probability implies convergence in distribution, but the converse is not true. □

Because a direct proof of convergence in probability or convergence with probability one is often difficult, we typically prove convergence in mean square, which by Theorem 1 implies convergence in probability.

6.2. Convergence of the estimation

We rewrite the estimation rule in Section 5 as

$$m_{ij}(k+1) = \alpha(k)m_{ij}(k) + \beta(k)\zeta_{ij}(k), \quad (30)$$
$$a_{ij}(k) = \rho\, m_{ij}(k),$$

where time-varying parameters are given by

$$\alpha(k) = (k+1)^{-1}k,$$
$$\beta(k) = k^{-1}. \quad (31)$$

Since ζ_{ij} = 1 or 0, we model it as a random variable with Bernoulli distribution, i.e.,

$$p(\zeta = \varsigma_{ij}) = q^{\varsigma_{ij}}(1-q)^{1-\varsigma_{ij}}, \qquad \varsigma_{ij} = 0,1, \quad (32)$$

where

$$q = p(\zeta_{ij} = 1) \in (0,1). \quad (33)$$

Recall that for the Bernoulli distribution, the mean and mean square are

$$E(\zeta) = E(\zeta^2) = q. \quad (34)$$

In the DBN model, the posterior probability of the state variable is given by

$$p_i(k+1) = a_i^T p(k), \quad (35)$$

where $p_i(k+1)$ is ith element of the stochastic vector, $p(k)$ is a prior probability vector, and the parameter vector $a_i = col\{a_{i1},\cdots,a_{iN}\}$, $i = 1,\cdots, N$. Substituting (30) in (35), we have

$$p_i(k+1) = (\rho\, m_i^T)p(k), \quad (36)$$

where $m_i = col\{m_{i1},\cdots, m_{iN}\}$, $i, j = 1,\cdots,N$. The expression shows that the posterior probability is a linear function of the estimate $m \in \{m_{i1},\cdots, m_{iN}\}$, i, j = 1, ⋯, N. By Theorem 2, we can therefore examine the asymptotic behavior of the posterior probability through the asymptotic behavior of m. We prove mean square convergence of m, which by Theorem 1 is sufficient to conclude convergence in probability. Thus, we show that $p_i(k+1)$ converges in probability.

Lemma 1: The sequence of the random variable m_{ij} in (30) asymptotically converges in mean-square to q.

Proof: We seek to prove that

$$\lim_{k\to\infty} E\left\{\left(m_{ij}(k+1) - q\right)^2\right\} = 0, \quad (37)$$

where $q = E\{\zeta\}$. Using (7), we expand the limit as

$$\lim_{k\to\infty} E\left\{\left(\frac{1}{k^2}\left[\sum_{n=1}^{k}\varsigma_n\right]^2 - \frac{2q}{k}\sum_{n=1}^{k}\varsigma_n + q^2\right)\right\}$$
$$= \lim_{k\to\infty} E\left\{\left(\frac{1}{k^2}\left[\sum_{n=1}^{k}\varsigma_n^2 + \sum_{n=1}^{k}\sum_{\substack{l=1\\ l\neq n}}^{k}\varsigma_n\varsigma_l\right] - \frac{2q}{k}\sum_{n=1}^{k}\varsigma_n + q^2\right)\right\}. \quad (38)$$

For i.i.d. Bernoulli trials, the expression becomes

$$\lim_{k\to\infty}\left(\frac{q}{k^2} + \frac{k(k-1)}{k^2}q^2 - 2q^2 + q^2\right) = 0. \quad (39)$$

□

Remark: From Definition 2 and Theorem 1, we conclude that $p_i(k+1)$ in (36) converges in probability. Therefore, the estimator in the second equation of (30) stochastically converges such that the state probability vector of the DBN model is converges asymptotically.□

6.3. Stability of the online estimation

We discuss the stability of the time-varying dynamic systems to our learning algorithm. The estimation rules of (30) are rewritten in vector form as

$$m(k+1) = F(k)m(k) + G(k)\zeta(k), \quad (40)$$
$$a(k) = C\, m(k),$$

where m, ζ, $a \in \Re^{N^2}$ are nonnegative vectors and F, G, $C \in \Re^{N^2\times N^2}$ are the corresponding nonnegative matrices. Note that $F(k)$ and $G(k)$ are time-varying and diagonal with elements less than unity, expressed by

$$F(k) = (k - 1/k)I_{N^2},$$
$$G(k) = (1/k)I_{N^2}. \quad (41)$$

Similarly, C in (40) is a diagonal and nonnegative matrix whose elements are less than unity.

Theorem 2 [24]**:** Consider an unforced linear discrete time-varying system as $x(k+1) = F(k)x(k)$. Its solution vector is $x(k) = \phi(k,k_0)x(k_0)$, $k_0 < k$, where the state-transition matrix $\phi(k,k_0) = F(k)F(k\text{-}1)\cdots F(k_0)$. If a norm of the solution $\|x(k)\| \to 0$ as $k \to \infty$ for any initial state $x(k_0)$, this system is asymptotically stable. This is equivalent to the condition $\|\phi(k,k_0)\| \to 0$ as $k \to \infty$. □

Lemma 2: The dynamic equation (40) is asymptotically stable for any initial state $m(k_0)$ at initial time k_0.

Proof: The state-transition matrix from k_0 to k for (40) is

$$\phi(k,k_0)=\left[\prod_{i=k_0}^{k}\left(\frac{i-1}{i}\right)\right]I_{N^2}=\left(\frac{k_0-1}{k}\right)I_{N^2}, \tag{42}$$

where $k_0 \geq 1$ and $k_0 << k$. Applying a limit to (42), we have

$$\lim_{k\to\infty}\phi(k,k_0)=\left[\lim_{k\to\infty}\left(\frac{k_0-1}{k}\right)\right]I_{N^2}=\mathbf{0}. \tag{43}$$

From Theorem 4, we conclude that the recursion (40) is asymptotically stable. □

7. SIMULATION EXAMPLE

We consider a first-order dynamic system with random input and noise, and time-varying parameters in [25] for our simulation study. The system model is given by

$$\zeta(k)=a(k)\zeta(k-1)+b(k)u(k)+\omega(k). \tag{44}$$

This equation is a single-input-single-output system model relating an input sequence $u(k)$, an output sequence $\zeta(k)$, and a random noise term $\omega(k)$. In (44), the density functions of the two time-varying parameters $a(k)$ and $b(k)$, and the noise $\omega(k)$ are defined as

$$\begin{aligned}&\gamma_a(x)=1-|4x-1|, \quad x\in[0,1],\\&\gamma_b(x)=\frac{3}{2}(1-x^2), \quad x\in[0,1],\\&\gamma_\omega(x)=1, \quad x\in[0,1],\end{aligned} \tag{45}$$

where a random variable x is uniformly distributed, and $u(k) \sim N(0,1)$. The initial values of the input and output $u(0)$=1 and $\zeta(0)$=0, respectively. We adopt the simulation environments of [25]. Fig. 11 illustrates time-histories of the observations for the system. We first define a discrete random variable for the output sequence as

$$\begin{aligned}&x_6^-=\{\zeta\mid\zeta\in[-5,-\infty]\}, \quad x_5^-=\{\zeta\mid\zeta\in[-4,-5)\},\\&x_4^-=\{\zeta\mid\zeta\in[-3,-4)\}, \quad x_3^-=\{\zeta\mid\zeta\in[-2,-3)\},\\&x_2^-=\{\zeta\mid\zeta\in[-1,-2)\}, \quad x_1^-=\{\zeta\mid\zeta\in[0,-1)\},\\&x_1^+=\{\zeta\mid\zeta\in(0,1]\}, \quad x_2^+=\{\zeta\mid\zeta\in(1,2]\},\\&x_3^+=\{\zeta\mid\zeta\in(2,3]\}, \quad x_4^+=\{\zeta\mid\zeta\in(3,4]\},\\&x_5^+=\{\zeta\mid\zeta\in(4,5]\}, \quad x_6^+=\{\zeta\mid\zeta\in(5,\infty]\}.\end{aligned} \tag{46}$$

We apply our estimation algorithm to this sequence and plot the probability of each variable in Fig. 12. Most of the curves are not stationary in the steady-state region due to the nonstationay random nature of the system. We also compare our results to those obtained using the approach recently proposed in [18] for the same dataset. The authors of [18] developed a dynamic density estimator based on a Bayesian nonparametric prior for a set of distributions which is constructed by defining the distribution at any time point as a Dirichlet process. For quantitative analysis, we define the error as the logarithm of the norm of the difference

$$e(k)=\log\|p(X(k))-\hat{p}(X(k))\|, \tag{47}$$

where $p(X)$ is an estimation using the method of [18].

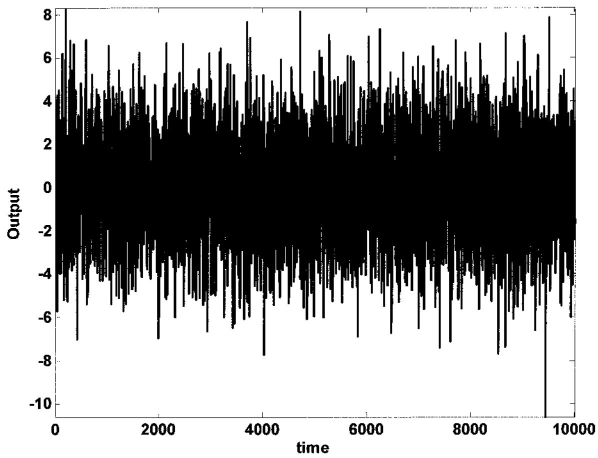


Fig. 11. Observation sequence.

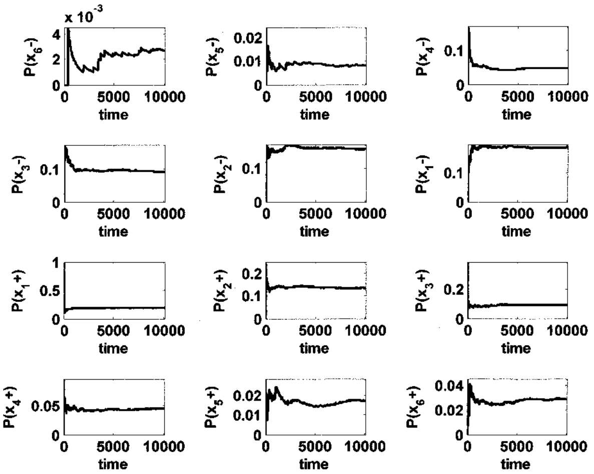


Fig. 12. Probability estimation.

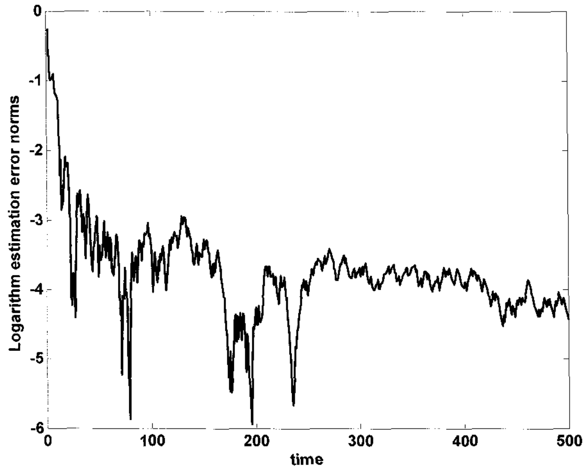


Fig. 13. Logarithm error norm.

The error trajectory is plotted in Fig. 13. The error trajectory shows that the logarithmic error progressively decreases. Thus, the two estimates are asymptotically identical.

8. CONCLUSION

We present a pdf estimation approach using a simple DBN model for adaptive online estimation given a large data set. Simulation examples illustrate the good performance of the algorithm in online computation. Future work will involve the application of the algorithm to problems in signal processing, pattern recognition and control.

REFERENCES

[1] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, Wiley-Interscience Publication, New York, 2001.

[2] E. Parzen, "On estimation of a probability density function and mode," *Analysis of Mathematical Statistics*, vol. 33, pp. 1065-1076, 1962.

[3] B. W. Silverman, *Density Estimation for Statistics and Data Analysis*, Chapman & Hall/CRC, 1986.

[4] S. Fiori and P. Bucciarelli, "Probability density estimation using adaptive activation function neurons," *Neural Processing Letters*, vol. 12, pp. 31-42, 2001.

[5] A. Sarajedini, R. Hecht-Nielsen, and P. M. Chau, "Conditional probability density function estimation with sigmoidal neural networks," *IEEE Trans. on Neural Networks*, vol. 10, no. 2, pp. 231-238, 1999.

[6] Y. Baram and Z. Roth, "Forecasting by density shaping using neural networks," *IEEE/IAFE Proc. of Computational Intelligence for Financial Engineering*, pp. 57-71, 1995.

[7] A. Likas, "Probability density estimation using artificial neural networks," *Computer Physics Communications*, vol. 135, no. 2, pp. 167-175, 2001.

[8] D. S. Modha and Y. Fainman, "A learning law for density estimation," *IEEE Trans. on Neural Networks*, vol. 5, no. 3, pp. 519-523, 1994.

[9] H. Yin and N. M. Allinson, "Self-organizing mixture networks for probability density estimation," *IEEE Trans. on Neural Networks*, vol. 12, no. 2, pp. 405- 411, 2001.

[10] T. Kostiainen and J. Lampinen, "On the generative probability density model in the self-organizing map," *Neurocomputing*, vol. 48, pp. 217-228, 2002.

[11] M. Strikanth, H. K. Kesavan, and P. H. Roe, "Probability density function estimation using the MinMax measure," *IEEE Trans. on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, vol. 30, no. 1, pp. 77-83, 2000.

[12] G. Miller and D. Horn, "Maximum entropy approach to probability density estimation," *Proc. of Int. Conf. on Knowledge-Based Intelligent Electronic Systems*, vol. 1, pp. 225-230, 1998.

[13] J. N. Kapur, G. Baciu, and H. K. Kesavan, "The MinMax information measure," *Int. J. of System Science*, vol. 26, no. 1, pp. 1-12, 1995.

[14] K. Kokkinakis, "Exponent parameter estimation for generalized Gaussian probability density functions with application to speech modeling," *Signal Processing*, vol. 85, no. 9, pp. 1852-1858, 2005.

[15] C. Wang and W. Wang, "Links between PPCA and subspace methods for complete Gaussian density estimation," *IEEE Trans. on Neural Networks*, vol. 17, no. 3, pp. 789-792, 2006.

[16] T. Kohonen, "The self-organization map," *Proc. of the IEEE*, vol. 78, no. 9, pp. 1464-1480, 1990.

[17] M. E. Tipping and C. M. Bishop, "Probability principle component analysis," *J. Roy. Statist. Soc., Ser. B*, vol. 21, no. 3, pp. 611-622, 1999.

[18] A. Rodriguez and E. T. Horst, "Dynamic density estimation with financial applications," *http://ftp.stat.duke.edu/WorkingPapers/06-21.pdf*, 2006.

[19] C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, 1995.

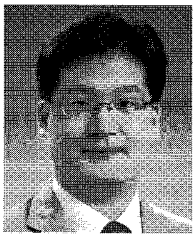
[20] K. Murphy, *Dynamic Bayesian networks: Representation, Inference and Learning*, Ph.D. Dissertation, UC Berkeley, 2002.

[21] J. M. Mendel, *Lessons in Estimation Theory for Signal Processing, Communications, and Control*, Prentice Hall, New Jersey, 1995.

[22] A. Papoulis and S. U. Pillai, *Probability, Random Variables and Stochastic Processes*, McGraw Hill, 2002.

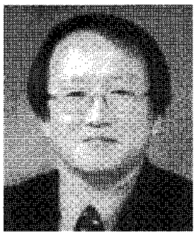
[23] R. J. Serfling, *Approximation Theorems of*

Mathematical Statistics, Wiley & Sons, New York, 1980.
[24] W. J. Rugh, *Linear System Theory*, Prentice Hall, 1996.
[25] H. Wang, A. Wang, and Y. Wang, "Online estimation algorithm for the unknown probability density functions of random parameters in auto-regression and exogenous stochastic systems," *IEEE Proc.-Control Theory Applications*, vol. 153, no. 4, pp. 462-468, 2006.



Hyun Cheol Cho received a B.S. from the Pukyong National University in 1997, the M.S. from the Dong-A University, Korea in 1999, and the Ph.D. from University of Nevada-Reno, USA in 2006. He is currently a Post-doc Researcher in the Dept. of Electrical Engineering, Dong-A University. His research interests are in the areas of control systems, neural networks, stochastic process, and signal processing.

M. Sami Fadali earned the B.S. in Electrical Engineering from Cairo University in 1974, an the M.S. from the Control Systems Center, UMIST, England, in 1977 and the Ph.D. from the University of Wyoming in 1980. He was an Assistant Professor of Electrical Engineering at the University of King Abdul Aziz in Jeddah, Saudi Arabia 1981-1983. From 1983-85, he was a Post Doctoral Fellow at Colorado State University. In 1985, he joined the Electrical Engineering Dept. at the University of Nevada, Reno, where he is currently a Professor of Electrical Engineering. In 1994 he was a Visiting Professor at Oakland University and GM Research and Development Labs. He spent the summer of 2000 as a Senior Engineer at TRW, San Bernardino. His research interests are in the areas of robust control, robust stability, fault detection, Bayesian networks and fuzzy logic control.



Kwon Soon Lee received the B.S. degree from Chungnam National University, Daejeon, and the M.S. degree in Electrical Engineering from Seoul National University, Seoul, Korea, in 1977 and 1981, respectively, and the Ph.D. degree in Electrical and Computer Engineering from Oregon State University, USA, in 1990. He joined the Department of Electrical Engineering, Dong-A University, Busan, Korea, as an Assistant Professor from 1982 to 1994. Since October 1, 1994, he has been with Division of Electrical, Electronic, and Computer in Dong-A University, Busan, Korea, where he is currently a Professor. He has authored or coauthored over 130 articles in archival journals and conference proceedings. His research interests include all aspects of port automation systems, intelligent control theory, and application of immune algorithm, etc. Prof. Lee is a responsible person of National Research Laboratory nominated by the Korean Ministry of Science & Technology, the Team Leader of New University for Regional Innovation (NURI) in Dong-A University, and the director of international cooperation department of Regional Research Center (RRC) in Korea.