Estimation of Non-Gaussian Probability Density by Dynamic Bayesian Networks

Hyun C. Cho*, Sami M. Fadali**, and Kwon S. Lee***

* Electrical Engineering Dept./260, University of Nevada-Reno, NV, 89557, USA
  (Tel: 1-775-324-2509, E-mail: hyun@unr.nevada.edu)
** Electrical Engineering Dept./260, University of Nevada-Reno, NV, 89557, USA
  (Tel: 1-775-784-6951, E-mail: fadali@ieee.org)
*** Division of Electrical, Electronic, and Computer Eng., Dong-A Univ., Busan, 604-714, Korea
  (Tel: +82-51-200-7739, E-mail: kslee@dau.ac.kr)

Abstract: A new methodology for discrete non-Gaussian probability density estimation is investigated in this paper based on a dynamic Bayesian network (DBN) and kernel functions. The estimator consists of a DBN in which the transition distribution is represented with kernel functions. The estimator parameters are determined through a recursive learning algorithm according to the maximum likelihood (ML) scheme. A discrete-type Poisson distribution is generated in a simulation experiment to evaluate the proposed method. In addition, an unknown probability density generated by nonlinear transformation of a Poisson random variable is simulated. Computer simulations numerically demonstrate that the method successfully estimates the unknown probability distribution function (PDF).

Keywords: Probability density estimate, Discrete non-Gaussian distribution, Dynamic Bayesian networks, Kernel function

1. INTRODUCTION

Probability estimation is an important issue for many engineering applications, especially, for pattern recognition, signal detection, artificial intelligence, etc. Several parametric and nonparametric techniques are available for the estimation [1]. Parametric methods are the simplest to estimate the parameters of a distribution of known form. A Gaussian distribution is widely used as a parametric density model due to its convenient statistical analysis. However, non-Gaussian distributions are often encountered in practical applications. A nonparametric method is more important to this case for which the form of the distribution is unknown. The histogram is the simplest nonparametric approach available. However, the method is less attractive because of its discontinuous nature and because it requires a large amount of data. 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 estimate [2] which uses Gaussian 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. Additional computation, such as smoothing, is sometimes needed.

In recent years, several approaches to PDF estimation have been proposed, such as soft computation, machine learning, information theory, etc. In [3], Fiori and Bucciarelli proposed estimating the distribution of a quasi-stationary random process by using an adaptive neural network which was trained to maximize the differential entropy. M. Srikant et al. utilized a MinMax measure to estimate a distribution, which is a quantitative measure of information contained in a given set of moment constraints [4]. A conditional distribution function (CDF) estimate was developed by A. Sarajedini et al. in [5] based on sigmoidal network learning and compared to a kernel CDF estimator for a higher dimensional problem. Miller and Horn investigated estimating both a PDF and CDF by using an information-based learning in which encoding and decoding steps were proposed [6]. A generalized Gaussian distribution was involved in [7] where an estimate of a PDF shape parameter was addressed. In [8], Modha and Fainman obtained a PDF estimate using an exponential family based on multilayer feedforward networks, and derived an unsupervised learning algorithm for ML function. Baram and Roth developed an estimation method that maximized the output entropy of a sigmoidal neural network and named the proposed approach density shaping [9].

We propose a new probability model estimation approach for discrete problems using a DBN and a set of kernel functions. A Bayesian network (BN) is a graphical reasoning system that uses a probability distribution to describe stochastic relationships between its nodes. Each node in a BN represents a random variable and edges indicate dependencies between nodes. A DBN is a dynamic extension of a BN based on an evolving probabilistic model. BNs and DBNs are typically used to solve problems with significant uncertainty.

The proposed approach utilizes a DBN in which a transition distribution is replaced with a set of kernel functions and the probability of feasible variables is expressed 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. We test the proposed approach using computer simulations that estimate a discrete-type Poisson density then a nonlinear transformation of the discrete Poisson data.

We next review kernel-based PDF estimation and the
Then we describe our proposed estimation approach and derive its learning algorithm. Finally, we present and discuss our simulation results.

2. KERNEL-BASED PROBABILITY ESTIMATE

A kernel-based estimate is typically composed of a mixture of kernel (basis) functions

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

(1)

where \( x \) is a data vector, \( P(x) \) is the probability of \( x \), \( \phi \) is a kernel function, and \( \alpha_i \) and \( \beta_i \) are parameters. Based on probability axioms, the kernel function must satisfy \( \phi \geq 0 \) and the parameters \( \alpha_i \) must be positive.

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

3. DYNAMIC BAYESIAN NETWORKS

A BN is a graphic model for stochastic relationships. It represents random variables as nodes and their dependencies as edges between the nodes. The random variables are assumed to have a finite number of states. The BN is described by a directed acyclic graph and a table of conditional probabilities. Let \( U = (A, B, C, D) \) be a universe of random variables, then a corresponding BN is depicted in Fig. 1.

In this network, the probabilities \( P(A), P(B), P(C|A), \) and \( P(D|A,B) \) must be known. A joint probability \( P(U) \) is calculated as

\[ P(U) = P(A,B,C,D) = P(A)P(B)P(C|A)P(D|A,B) \]  

(10)

Inference is the task of calculating the probability of each node by probability calculus from other probabilities. For instance, in Fig. 1, conditional probability of the variable \( A \) given \( B, C, \) and \( D \) is expressed using Bayes rule as
where the conditional probability \( P(x_k | x_{k-1}) \) is called the transition probability. A higher-order Markov model provides a more sophisticated solution but at an increasingly higher computational cost. A hidden Markov model (HMM) is a popular structure in which an observation variable depends on a hidden state and a sequence of hidden states evolves according to a Markov process. Fig. 3 illustrates a typical example of an HMM.

![Fig. 3. An example of HMM.](image)

The joint probability of the hidden states and observation variables is expressed similarly to the first Markov model as

\[
P(x_{k-1}, y_k) = P(y_k | x_k) \prod_{i=2}^{T} P(x_i | x_{i-1})P(y_i | x_i)
\]

where the state transition probability \( P(x_k | x_{k-1}) \) is formed by a single transition matrix for a time-invariant HMM. The observation probability \( P(y_k | x_k) \) is usually modeled as Gaussian, a combination of Gaussians, or a neural network. In (17), the transition state \( x_k \) and observation state \( y_k \) are decomposed into deterministic and stochastic terms as

\[
x_k = f_k(x_{k-1}) + w_k
\]
\[
y_k = g_k(x_k) + v_k
\]

where \( w_k \) and \( v_k \) are random noise vectors, and \( f \) and \( g \) are transition and observation functions, respectively. If both functions are linear and time-invariant, we have the dynamic linear time-invariant model

\[
x_k = Ax_{k-1} + w_k
\]
\[
y_k = Cx_k + v_k
\]

where \( A \) is the state transition matrix and \( C \) is the observation matrix. If the two noise vectors in (19) are modeled as zero-mean Gaussian with covariance matrices \( Q \) and \( R \), respectively, the transition and observation probabilities are

\[
P(x_k | x_{k-1}) \sim N(Ax_{k-1}, Q)
\]
\[
P(y_k | x_k) \sim N(Cx_k, R)
\]

where \( N(\cdot) \) denotes Gaussian.

After constructing the Bayesian network, a learning procedure is developed, based on given data samples...
Using (6), (7), (8) and (9) of Section II, the adjustment rules of the parameters are obtained as

\[ \alpha_i(t_k + 1) = \alpha_i(t_k) + \eta \sum_{n=1}^{N} \frac{P(x_n) \delta(x_n - z_j)}{P(z_j)} \]  

(25)

and

\[ \beta_j(t_k + 1) = \beta_j(t_k) + \eta \sum_{n=1}^{N} \frac{P(z_j) \delta(x_n - z_j)}{P(z_j)} \]  

(26)

4. DENSITY ESTIMATE BY A BN

In this Section, we propose a new probability estimation method using a kernel-based approach (Section II) and a BN (Section III). In Fig. 2, the posterior probability of a state vector \( x \) at \( k+1 \) is given by

\[ P(x(k+1)) = \sum_{i=1}^{N} P(x(k+1) \mid x(k))P(x(k)) \]  

(21)

where \( P(x_i) \) is prior probability of \( x_i \). For specific values \( z_i \), \( z_i \in \mathbb{R}^N \), of \( x_i \) in (21) we have

\[ P(x(k+1)) = \sum_{j=1}^{N} P(x(k+1) \mid z_j)P(z_j(k)) \]  

(22)

In (22), the transition distribution is obtained by a combination of kernel functions as in (1). Thus, the proposed estimate is formed as

\[ P(x(k+1)) = \sum_{j=1}^{N} \left( \sum_{i=1}^{N} \alpha_{ij} \phi(x - z_j, \beta_j) \right) P(z_j(k)) \]  

(23)

This estimate is sequentially updated for a random vector \( x \) at time \( k \). The structure of this estimate is depicted in Fig. 4.

As stated in Section II, the parameter vectors \( \alpha \) and \( \beta \) in Fig. 4 are recursively adjusted via sequential learning. The objective function is similarly given by

\[ J = \max_{\alpha, \beta} \sum_{n=1}^{N} \ln \{P(x_n)\} = \max_{\alpha, \beta} \sum_{n=1}^{N} \ln \left\{ \sum_{i=1}^{N} \beta_{ij} \phi(x_n - z_j, \beta_j) P(z_j) \right\} \]  

(24)

5. EXAMPLES AND RESULTS

We simulate the proposed method to estimate the discrete probability vector of a non-Gaussian signal. In (25) and (26), a kernel function is used with the Gaussian model

\[ \phi(x_n - z_j, \beta_j) = \frac{1}{\sqrt{2\pi} \beta_j} \exp \left( -\frac{(x_n - z_j)^2}{\beta_j^2} \right) \]  

(27)

For simplicity, the prior probabilities of the values \( z_i, i = 1, \ldots, N \), in (23) are assumed equal, i.e. \( P(z_i) = 1/N \).

Example I: First, we simulate the estimation of the Poisson distribution

\[ P(x = k) = \exp(-\lambda) \frac{\lambda^k}{k!} \]  

(28)

where \( \lambda \) is a parameter and \( k = 0, 1, \ldots, \infty \). Fig. 5 shows \( N=100 \) Poisson distributed data samples for \( \lambda=10 \) which are generated using the MATLAB© command `poissrnd`.

![Fig. 5. Poisson random number (Example I).](image)
\[ e = \frac{1}{N} \sum_{n=1}^{N} \left| \hat{P}(x) - P(x) \right| \]  

(29)

In the training simulation, 100 training data samples at each time \( k \) were temporally generated. Fig. 6 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 II: We non-linearly transform Poisson random data in this Example. To simulate this scenario, a random input is fed to a non-linear system and the probabilities of the system output are estimated. A block diagram for this process is depicted in Fig. 7.

![Fig. 7. Transformation of a random variable.](image)

The random input is Poisson distributed with the same parameters as Example I. Because the data is non-linearly transformed, the output probabilities are not Poisson and the output distribution is of an unknown form. We define the non-linear function as \( y = cx^3(k) \) where \( c \) is constant and generate 500 random input values. A time history of the output is shown in Fig. 8.

![Fig. 8. Poisson random number (Example II).](image)

6. DISCUSSION

We propose a probability estimation approach using DBN. The method was successfully used to estimate discrete non-Gaussian signals. From the simulation results, we conclude that the performance of the approach is acceptable.

In future work, we will address the estimation of more complex distributions such as multi-variable non-Gaussian or non-stationary stochastic models. To this end, we will utilize and compare well-known algorithms such as the Viterbi algorithm [13].

ACKNOWLEDGEMENTS

The authors are grateful for the support of NRL, National Research Laboratory in Dong-A University, Korea.

REFERENCES


