An Ensemble Learning algorithm for Blind Signal Separation Problem

Yan Li[1] and Peng Wen[2]
[1]Department of Mathematics and Computing, [2]Faculty of Engineering and Surveying
The University of Southern Queensland, Queensland, Australia, QLD 4350
{liyan, pengwen}@usq.edu.au

***Abstract* –The framework in Bayesian learning algorithms is based on the assumptions that the quantities of interest are governed by probability distributions, and that optimal decisions can be made by reasoning about these probabilities together with the data. In this paper, a Bayesian ensemble learning approach based on enhanced least square backpropagation (LSB) neural network training algorithm is proposed for blind signal separation problem. The method uses a three layer neural network with an enhanced LSB training algorithm to model the unknown blind mixing system. Ensemble learning is applied to estimate the parametric approximation of the posterior probability density function (pdf). The Kullback-Leibler information divergence is used as the cost function in the paper. The experimental results on both artificial data and real recordings demonstrate that the proposed algorithm can separate blind signals very well.**

I. INTRODUCTION

The problem of blind signal separation (BSS) has drawn a great attention from many researchers in the past two decades. BSS is to extract the sources $s(t)$ that have generated the observations $x(t)$.

$$x(t) = F[s(t)] + n(t) \qquad (1)$$

where $F: R^m \rightarrow R^m$ is the unknown nonlinear mixing function and $n(t)$ is additive noise.

The objective is to find a mapping that yields components

$$y(t) = g(x(t)) \qquad (2)$$

So that $y(t)$ are statistically independent and as close as possible to $s(t)$. This must be done from the observed data in a blind manner as both the original sources and the mixing process are unknown. Many different approaches to BSS have been attempted by numerous researchers [1]. In this paper, we explore a new blind separation method using a Bayesian estimation technique and an enhanced LSB neural network training algorithm to model the system.

Bayesian ensemble learning, also called Variational Bayesian learning [2], utilizes an approximation which is fitted to be posterior distribution of the parameter(s) to be estimated. The approximative distribution is often chosen to be Gaussian because of its simplicity and computational efficiency. The mean of this Gaussian distribution provides a point estimate for the unknown parameter considered, and its variance gives a measure of the reliability of the point estimate. The approximative posterior distribution is fitted to the posterior distribution estimated from the data using the Kullback-Leibler information divergence. This measures the difference between two probability densities and is sensitive to the mass of the distributions.

One problem in Bayesian estimation methods is that their computational load is high in problems of realistic size in spite of the efficient Gaussian approximation. Another problem is that the Bayesian ensemble learning procedure may get stuck to a local minimum and requires careful initialization [3]. These obstacles have prevented their applications to real unsupervised or blind learning problems where the number of unknown parameters to be estimated grows very large.

To combat these problems, we use, in this paper, a LSB neural network to model the blind mixing process and apply the Bayesian ensemble learning to estimate original sources. The experimental results are presented in the paper and demonstrate the technique works very well.

The rest of the paper is organized as follows: the enhanced least square neural network model and its training method are introduced in the next section. The network parameters and parametric approximation of the posterior pdf are presented in Section 3. Section 4 introduces ensemble learning and the cost function used in this paper. The experimental results are given in Section 5 to demonstrate the performance of the method. Finally, Section 6 concludes the paper.

II. THE LEAST SQUARE NEURAL MODEL

In 1993, Konig and Barmann [4] separated neural networks into linear parts and non-linear parts. The linear parts sum up the weighted inputs to the neurons and none-linear parts pass through the signals with the non-linear activity functions (such as sigmoidal activation). While solving the linear parts optimally, they used the inverse of the activation to propagate the remaining error back into the previous layer of the neural networks. Therefore, the learning error is

Proceedings of the 2005 International Conference on Computational Intelligence for Modelling, Control and Automation, and International Confe
Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'05)
0-7695-2504-0/05 $20.00 © 2005 IEEE

minimised on each layer separately from the output layer to the hidden and input layers by using least square back propagation (LSB) method. The convergence of the algorithm is much faster than that of classical Back Propagation (BP) algorithm. However, the drawback of the LSB algorithm is that the training error can not be further reduced after the beginning two or three iterations [4]. In fact, the training error has been significantly reduced at the first and second iterations, which is good enough for the most of the practical applications.

The model structure used in this paper is a three layer neural network with an enhanced LSB training algorithm [5]. Fig. 1 shows the structure of network. The LSB training algorithm optimises the network weights through an iterative process layer by layer. The training algorithm takes, firstly, outputs of nodes in the hidden layer into consideration. It not only adjusts the weights of the network but also adjusts the outputs of the hidden layer. The network works like a RNN, but it can reach its steady state very quick because of its novel training algorithm. Please refer to [5] for the details about this algorithm.

The neurons in the first layer are linear. They pass through the input signals to all the neurons in the hidden layer. The activation function used in the neurons in the hidden layer and the output layer is the inverse hyperbolic sine, sin^{-1}, which is a sigmoidal function but not saturating for large values of its inputs.

The original algorithm is a supervised learning algorithm. Inspired by [6], it can be adapted for BSS problem (with unknown inputs). During the learning process, we generate a set of random source variables to play the role of inputs. The first data vector is passed through the neural network, and the outputs of the network are produced. The observation data play the role of the outputs. The enhanced LSB algorithm is applied to find an optimal source signals which produce the observed data. The initial weights of the network are set randomly.

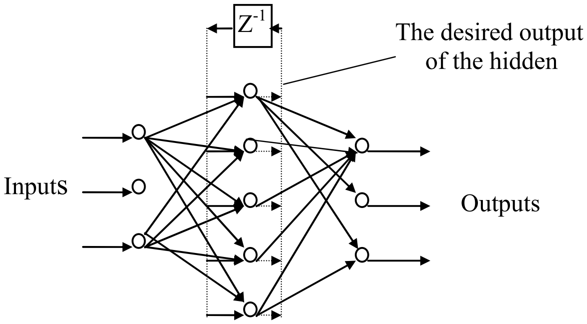

Fig. 1 The Network Structure

Once the optimal source signals are found, the inputs of the network are known and the learning process is the same as the supervised learning: the weights are adapted. It makes the best matching model vector be moved even closer to the true inputs. Then the next input data vector are taken to pass through the network, to find the source variables that best describe the data, to adapt the weights and so on.

Unlike the method used in [6], which applied the traditional BP algorithm, the algorithm does not need to be iterated many times to find an optimal original source signals as one iteration is good enough for the enhanced LSB training algorithm to reach the equivalent training error or even better.

It is expected that the training process is much faster than the approach using BP algorithm as the convergence of the enhanced LSB algorithm is nearly orders of magnitude faster than the classical BP.

III. NETWORK PARAMETERS AND PARAMETRIC APPROXIMATION

A. Network Parameters

Let $x(t)$ denote the observed data vector at time t; $s(t)$ the vectors of the source variables at the time t; $W_1(t)$ and $W_2(t)$ the matrices containing the weights on the first and the second layers, respectively. All the biases for the network are set to 0.5, and $f(.)$ is the vector of nonlinear activation functions (sin^{-1}). As all real signals contain noise, we shall assume that observations are corrupted by Gaussian noise denoted by $n(t)$. Using this notation, the model for the observations passes through the network described below;

$$x(t) = f(W_2(t)[f(W_1(t)\, s(t)]) + n(t) \qquad (3)$$

The sources are assumed to be independent and Gaussian. The Gaussianity assumption is realistic as the network has nonlinearities which can transform the Gaussian distributions to virtually any other regular distributions.

The weight matrices $W_1(t)$ and $W_2(t)$, and the parameters of the distributions of the noise, source variables and column vectors of the weight matrices are the main parameters of the network.

For simplicity, all the parameterised distributions are assumed to be Gaussian.

B. Parametric approximation of the posterior pdf

Exact treatment of the posterior pdfs of the models is impossible in practice and posterior pdfs need to be approximated. In this paper, we apply a computationally efficient parametric approximation which usually yields satisfactory results.

A standard approach for parametric approximation is the Laplace's method. MacKay introduces a variation method called the evidence framework. In his neural network approach, one first finds a (local) maximum point of the posterior pdf and then applies a second order Taylor's series approximation for the logarithm of the posterior pdf. This is equivalent as to applying the Gaussian approximation to the posterior pdf.

C. *Ensemble Learning and the Cost Function*

The ensemble learning [7], a well developed method for parametric approximation of posterior pdfs, is used in this paper. The basic idea is to minimize the differences between the posterior pdf and its parametric approximation.

Let P denote the exact posterior pdf and Q is parametric approximation. Assume that θ is the parameters of the model H and X is the set of the observed data. It is assumed that we have independent priors of each parameter, thus

$$P(\theta \mid H) = \prod_i P(\theta_i \mid H) \tag{4}$$

The Ensemble learning cost function, C_{Kl}, is the misfit measured by the Kullback-Keibler information divergence between P and Q.

$$\begin{aligned} C_{KL} &= E_\theta\{\log(\frac{Q(\theta)}{P(X \mid \theta, H)P(\theta \mid H)})\} \\ &= E_\theta\{\sum_i \log\frac{Q(\theta_i)}{P(\theta_i \mid H)} - \log P(X \mid \theta, H)\} \end{aligned} \tag{5}$$

If the marginalization is performed over all the parameters, with the exception of θ_i, we have:

$$C_{KL} = \int Q(\theta_i)(\log Q(\theta_i \mid H) - E_{Q|\theta_i}\{\log P(X \mid \theta, H)\})d\theta_i + c \tag{6}$$

where c is a constant.

Differentiating the above equation with respect to $Q(\theta_i)$, we obtain

$$\begin{aligned} \frac{\partial C_{kl}}{\partial Q(\theta_i)} &= \log Q(\theta_i) - \log P(\theta_i \mid H) \\ &= -E_{Q\backslash\theta_i}\{\log P(X \mid \theta, H)\} + 1 + \lambda_i \end{aligned} \tag{7}$$

where λ_i is a Lagrange multiplier introduced to ensure that $Q(\theta_i)$ is normalized. The optimal distribution $Q(\theta_i)$ is

$$Q(\theta_i) = \frac{1}{Z_i} P(\theta_i \mid H)\exp(E_{Q\backslash\theta_i}\{\log P(X \mid \theta_i H)\}) \tag{8}$$

where Z_i is the partition function:

$$Z_i = \int P(\theta_i \mid H)\exp(E_{Q|\theta_i}\{\log P(X \mid \theta, H)\})d\theta \tag{9}$$

This procedure leads to an iterative algorithm for the update of each distribution. Simple Gaussian distributions are used to approximate the posterior pdf.

Note that the Kullback-Leibler divergence involves an expectation over a distribution and, consequently, is sensitive to probability mass rather than probability density. The Kullback-Leibler divergence is used as the cost function in this paper.

For mathematical and computational simplicity, the approximation of Q needs to be simple.

The cost function C_{KL} is a function of the posterior means and variances of the source variables and the parameters of the network. This is because instead of finding a point estimate, a whole distribution will be estimated for the source variables and the parameters during learning. The end result of the learning is therefore not just an estimate of the unknown variables, but a distribution over the variables.

IV. *EXPERIMENTAL RESULTS*

Two experiments are presented in this section. In the first experiment, we use a set of artificial data; however, in the second one, real speech recordings are used to test the performance of the proposed approach.

A. Experiment 1: Artificial data

There are eight sources, four super-Gaussians and four sub-Gaussians, generated by Matlab functions. The observation data are generated from these sources through a nonlinear mapping neural network. The network is a randomly initialized three-layer feedforward neural network with 30 hidden neurons and eight output neurons. A Gaussian noise having a standard deviation of 0.1 is also added to the data.

The results are shown in Fig. 2. It shows eight scatter plots, each of them corresponding to one of the eight sources. The original source is on the x-axis and the estimated source on the y-axis of each plot, with each point corresponding to one data vector. An optimal result is a straight line presenting that the estimated values of the sources are the same as the true values.

The number of hidden neurons is changed to optimize the results. There are 20 neurons used in the hidden layer in the enhanced LSB neural network and only two iterations (the data set is going through the neural network twice) used in the results shown in Fig. 2. Further more iterations do not bring better results rather than more training time, which is consistent with the characteristic of LSB algorithm. Fig. 3 shows the results after 500 training iterations, which gives no better perceivable results than those in Fig. 2. The scatter plots present the differences between the sources and the estimated signals.

B. Experiment 2: Real speech signal separation

The observed signals were taken from Dr Te-Won Lee's home page at the Salk Institute on the website http://www.cnl.salk.edu/~tewon/[8]. One signal is a recording of the digits from one to ten spoken in English. The second microphone signal is the digits spoken in Spanish at the same time. The proposed algorithm is applied to the signals. Figs 4 and 5 show the real signals and the separated results (only half of the signals are presented here for clarity). It is hard to compare the results with Lee's results in a quantitative way due to the different methodologies, but comparable results can be identified when the signals are listened to.

V. CONCLUSION

In this paper, we develop a new approach based on Bayesian ensemble learning and LSB neural network training algorithm for BSS problem. A three layer neural network with an enhanced LSB training algorithm is used to model the unknown blind mixing system. The network works like a RNN, but it can reach its steady state very quick because of its enhanced LSB training algorithm. Ensemble learning is applied to estimate the parametric approximation of the posterior pdf.

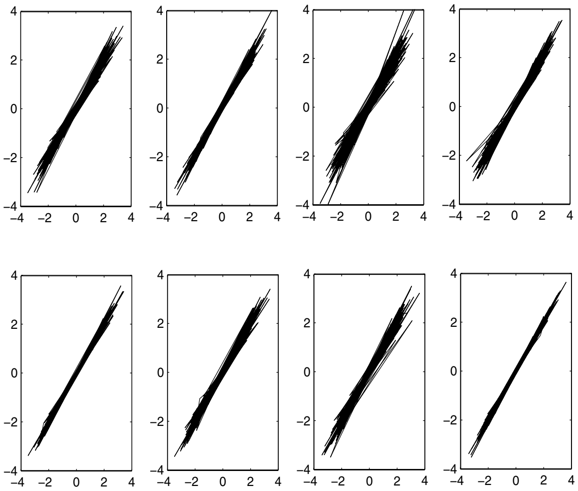

Fig. 2 The scatter plots, with the original sources on the x-axis of each scatter plot and the sources estimated by the proposed algorithm on the y-axis, after 2 iterations.

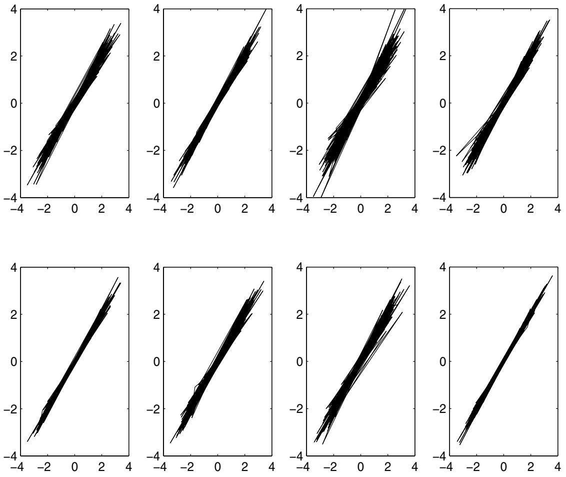

Fig. 3 The scatter plots, with the original sources on the x-axis of each scatter plot and the sources estimated by the proposed algorithm on the y-axis, after 500 iterations.

The Kullback-Leibler information divergence is used as the cost function in the paper. It is a measure suited for comparing probability distributions and it can be computed efficiently in practice if the approximation is chosen to be simple enough. Kullback-Leibler information is sensitive to probability mass and therefore the search for good models focuses on the models which have large probability mass as opposed to probability density. The drawback is that in order for ensemble learning to be computationally efficient, the approximation of the posterior needs to have a simple factorial structure.

The experiments have been carried out using both artificial data and real recordings. The results show the success of the proposed algorithm.

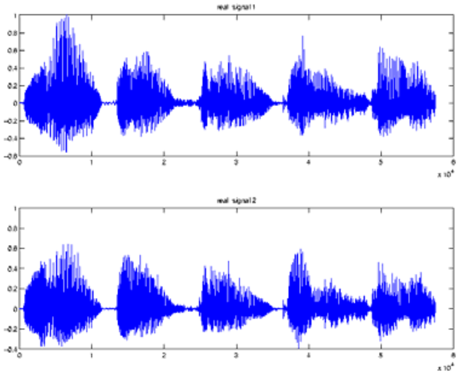

Fig. 4 The real signals

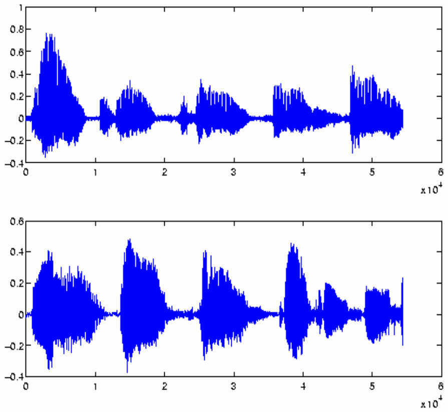

Fig. 5 The separated signals

REFERENCES

[1] Li, Yan, Peng Wen and David Powers, Methods For The Blind Signal Separation Problem, The proceeding of the IEEE International Conference on Neural Networks & Signal Processing (ICNNSP'03), Nanjing, China, December, 14 - 17, 2003, pp. 1386-1389.

[2] Lappalainen, H., Ensemble Learning", in Advances in Independent Component Analysis, M. Girolami, Ed. Berlin: Springe Verlag, 2000, pp. 75-92.

[3] Jutten, C. and J. Karhunen, Advances in Nonlinear Blind Source Separation, 4th International Symposium on Independent Component Analysis and Blind Signal Separation (ICA2003), April 2003, Nara, Japan, pp. 245-256.

[4] Biegler-Konig, F. B. and F Barman, 1993, A learning algorithm for multilayered neural networks based on linear least square problems, Neural Networks, Vol. 6, pp. 127-131.

[5] Li, Yan, A. B. Rad and Wen Peng, An Enhanced Training Algorithm for Multilayer Neural Networks Based on Reference Output of Hidden Layer, Neural Computing & Applications, Vol. 8, 1999, pp. 218-225.

[6] Lappalainen, H. and Xavier Giannakopoulos, Multi-Layer Perceptrons as Nonlinear Generative Models for Unsupervised Learning: a Bayesian Treatment, ICANN'99, pp. 19-24, 1999.

[7] Geoffery E. Hinton and Drew van Camp, Keeping neural networks simple by minimizing the description length of the weights In Proceedings of the COLT'93, pp. 5-13, Santa Cruz, California, 1993.

[8] The website http://www.cnl.salk.edu/~tewon/.