Artificial Intelligence in Medicine (2006) **36**, 257—267

ELSEVIER

ARTIFICIAL
INTELLIGENCE
IN MEDICINE

http://www.intl.elsevierhealth.com/journals/aiim

An incremental EM-based learning approach for on-line prediction of hospital resource utilization

Shu-Kay Ng[a,*], **Geoffrey J. McLachlan**[a,b], **Andy H. Lee**[c]

[a] *Department of Mathematics, University of Queensland, Brisbane, Qld 4072, Australia*
[b] *Institute for Molecular Bioscience, University of Queensland, Brisbane, Qld 4072, Australia*
[c] *School of Public Health, Curtin University of Technology, Perth, WA 6845, Australia*

Received 18 May 2005; received in revised form 27 July 2005; accepted 27 July 2005

KEYWORDS
EM algorithm;
Mixture of experts;
Incremental update;
Length of stay;
Machine learning algorithm;
On-line prediction

Summary

Objective: Inpatient length of stay (LOS) is an important measure of hospital activity, health care resource consumption, and patient acuity. This research work aims at developing an incremental expectation maximization (EM) based learning approach on mixture of experts (ME) system for on-line prediction of LOS. The use of a batch-mode learning process in most existing artificial neural networks to predict LOS is unrealistic, as the data become available over time and their pattern change dynamically. In contrast, an on-line process is capable of providing an output whenever a new datum becomes available. This on-the-spot information is therefore more useful and practical for making decisions, especially when one deals with a tremendous amount of data.
Methods and material: The proposed approach is illustrated using a real example of gastroenteritis LOS data. The data set was extracted from a retrospective cohort study on all infants born in 1995—1997 and their subsequent admissions for gastroenteritis. The total number of admissions in this data set was $n = 692$. Linked hospitalization records of the cohort were retrieved retrospectively to derive the outcome measure, patient demographics, and associated co-morbidities information. A comparative study of the incremental learning and the batch-mode learning algorithms is considered. The performances of the learning algorithms are compared based on the mean absolute difference (MAD) between the predictions and the actual LOS, and the proportion of predictions with MAD $\leq$ 1 day (Prop(MAD $\leq$ 1)). The significance of the comparison is assessed through a regression analysis.
Results: The incremental learning algorithm provides better on-line prediction of LOS when the system has gained sufficient training from more examples (MAD = 1.77 days and Prop(MAD $\leq$ 1) = 54.3%), compared to that using the batch-mode learning. The regression analysis indicates a significant decrease of MAD (p- value = 0.063) and a

* Corresponding author. Tel.: +61 7 33656139; fax: +61 7 33651477.
E-mail address: skn@maths.uq.edu.au (S.-K. Ng).

0933-3657/$ — see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.artmed.2005.07.003

significant (p- value $= 0.044$) increase of Prop(MAD $\leq$1) with the incremental learning algorithm.

Conclusions: The incremental learning feature and the self-adaptive model-selection ability of the ME network enhance its effective adaptation to non-stationary LOS data. It is demonstrated that the incremental learning algorithm outperforms the batch-mode algorithm in the on-line prediction of LOS.

© 2005 Elsevier B.V. All rights reserved.

1. Introduction

Health care is a rapidly changing field that embraces information technology at all levels. The continuing development and innovative use of information technology in health care has played a significant role in contributing and advancing this active and burgeoning field. Targeting high quality and efficient health care, artificial intelligent systems such as neural networks, are required for health care professionals [1,2]. In particular, limitations in health care funding require hospitals to find effective ways to utilize hospital resources [3]. Inpatient length of stay (LOS) is an important measure of hospital activity and health care utilization [4,5]. It is also considered to be a measurement of disease severity and patient acuity [1,6]. Length of stay predictions have therefore important implications in various aspects of health care decision support systems.

Neural networks have been adopted to predict LOS for many disease states [1,3,6,7]. Neural networks are intelligent systems that attempt to simulate many abilities of the human brain, such as decision-making based on learning from experiences. They can be regarded as universal function approximators of underlying nonlinear functions that can be learned (trained) from examples of known input—output data (training set) [8]. Many existing neural networks for predicting LOS, however, only predict a broad category for the LOS, such as less than 7 days or greater than 7 days [6,7]. This binary classification does not provide hospital administrators with enough information to adequately plan for hospital resource allocation [3]. Moreover, the backpropagation learning method adopted in most of these LOS prediction networks has been criticized for failure to extrapolate from the training population [3]. The backpropagation algorithm also requires careful adjustment of data-dependent tuning constants [9]. The development of alternative efficient learning methods in neural networks, such as the expectation maximization (EM) algorithm [10,11], is amongst the latest research directions in machine learning [8,12].

The main drawback of existing neural networks for predicting LOS is the use of a batch-mode learning process. That is, the network is learned only after the entire training set is available. Such a learning method is unrealistic in the prediction of LOS as the data become available over time and the input—output pattern of data changes dynamically over time. On the other hand, an on-line process is capable of providing an output whenever a new datum becomes available. This on-the-spot information is therefore more useful and practical for adaptive training of model parameters and making decisions [13,14], especially when one deals with a tremendous amount of data.

In this paper, we propose an intelligent mixture of experts (ME) system for on-line prediction of LOS via an incremental EM-based learning process. The strength of an incremental learning process is that it enables the network to be updated when an input—output datum becomes known. These on-line and incremental updating features increase the simulation between neural networks and human decision-making capability in terms of learning from "every" experience. The computational efficiency in real-time applications is thus improved. The above features also reduce storage space and are especially useful and essential in situations where data become available over time. The rest of the paper is organized as follows: Section 2 introduces the ME network [15] for approximating underlying nonlinear mappings between the input and the output. We also describe how ME networks can be learned in batch-mode via the use of an expectation-conditional maximization (ECM) algorithm [16]. In Section 3, an incremental version of the ECM algorithm is formulated where the unknown parameters are updated whenever an input—output datum is available. In Section 4, we describe how the proposed system can provide on-line predictions of LOS. The ability of the system to "prune" or "grow" the expert networks is also investigated. The proposed intelligent system is illustrated in Section 5, using a set of gastroenteritis LOS data derived from the Western Australia hospital morbidity data system. Section 6 presents some concluding remarks and discussion.

2. Mixture of experts neural networks

2.1. Background

In ME neural networks (Fig. 1), there are m modules, referred to as expert networks. These expert networks approximate the distribution of the output y_j within each region of the input space. The expert network maps its input $\boldsymbol{x}_j = (x_{1j}, \ldots, x_{pj})^{\mathrm{T}}$ to an output, the density $f_h(y_j|\boldsymbol{x}_j; \boldsymbol{\theta}_h)$, where p is the dimension of the input vector $\boldsymbol{x}_j, \boldsymbol{\theta}_h$ is a vector of unknown parameters for the h-th expert network, and the superscript T denotes vector transpose. It is assumed that different experts are appropriate in different regions of the input space. The gating network provides a set of scalar coefficients $\pi_h(\boldsymbol{x}_j; \boldsymbol{v})$ that weight the contributions of the various experts, where $\boldsymbol{v}$ is a vector of unknown parameters in the gating network. Therefore, the final output of the ME neural network is a weighted sum of all the output produced by expert networks:

$$f(y_j|\boldsymbol{x}_j;\, \Psi) = \sum_{h=1}^{m} \pi_h(\boldsymbol{x}_j;\, \boldsymbol{v})\, f_h(y_j|\boldsymbol{x}_j;\, \boldsymbol{\theta}_h), \qquad (1)$$

where $\Psi = (\boldsymbol{v}^{\mathrm{T}}, \boldsymbol{\theta}^{\mathrm{T}})^{\mathrm{T}}$ is the vector of all the unknown parameters [15,17] and $\boldsymbol{\theta} = (\boldsymbol{\theta}_1^{\mathrm{T}}, \ldots, \boldsymbol{\theta}_m^{\mathrm{T}})^{\mathrm{T}}$. With the probabilistic interpretation (1), the expected value of the output y_j for a given input $\boldsymbol{x}_j$ under the current model $\hat{\Psi}$ is a weighted sum of the expectations of the local outputs [17]:

$$E(y_j|\boldsymbol{x}_j;\, \hat{\Psi}) = \sum_{h=1}^{m} \pi_h(\boldsymbol{x}_j;\, \hat{\boldsymbol{v}}) E(y_j|\boldsymbol{x}_j;\, \hat{\boldsymbol{\theta}}_h).$$

With the ME network, the output of the gating network is usually modeled by the softmax function [18] as

$$\pi_h(\boldsymbol{x}_j; \boldsymbol{v}) = \frac{\exp(\boldsymbol{v}_h^{\mathrm{T}} \boldsymbol{x}_j)}{1 + \sum_{l=1}^{m-1} \exp(\boldsymbol{v}_l^{\mathrm{T}} \boldsymbol{x}_j)} \qquad (2)$$
$$(h = 1, \ldots, m-1),$$

and $\pi_m(\boldsymbol{x}_j; \boldsymbol{v}) = 1/(1 + \sum_{l=1}^{m-1} \exp(\boldsymbol{v}_l^{\mathrm{T}} \boldsymbol{x}_j))$, where $\boldsymbol{v}$ contains the elements in the weight vectors $\boldsymbol{v}_h (h = 1, \ldots, m-1)$ such that $\boldsymbol{v} = (\boldsymbol{v}_1^{\mathrm{T}}, \ldots, \boldsymbol{v}_{m-1}^{\mathrm{T}})^{\mathrm{T}}$. It is implicitly assumed that the first element of $\boldsymbol{x}_j$ is one to account for the bias term. The local output densities $f_h(y_j|\boldsymbol{x}_j;\, \boldsymbol{\theta}_h)\,(h = 1, \ldots, m)$ can be assumed to belong to the exponential family of densities [17]. For regression problems such as the prediction of LOS, the local output densities are generally assumed to be Gaussian

$$f_h(y_j|\boldsymbol{x}_j;\, \boldsymbol{\theta}_h) = \frac{1}{\sqrt{(2\pi\sigma_h^2)}} \exp\left\{ \frac{-\frac{1}{2}(y_j - \boldsymbol{w}_h^{\mathrm{T}} \boldsymbol{x}_j)^2}{\sigma_h^2} \right\}, \qquad (3)$$

where $\boldsymbol{w}_h$ and σ_h^2 are, respectively, the weight vector and the variance (dispersion parameter) of the h-th expert network. Thus, we have $\boldsymbol{\theta}_h = (\boldsymbol{w}_h^{\mathrm{T}}, \sigma_h^2)^{\mathrm{T}}$. The unknown parameter vector Ψ can be estimated by

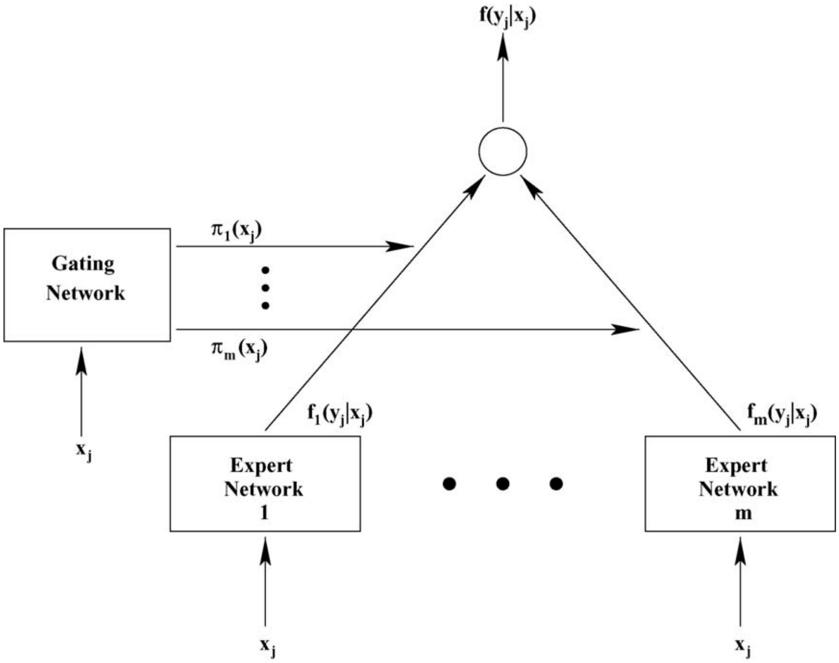

Figure 1 Mixture of experts with m modules.

the ML approach via the EM algorithm [17,19] or its more recent extensions such as the ECM algorithm [8].

2.2. Batch-mode learning via the ECM algorithm

To apply EM-based algorithms to the ME networks, we introduce the indicator variables z_{hj}, where z_{hj} is one or zero according to whether y_j belongs or does not belong to the h th expert [8]. The complete-data log likelihood for Ψ is then given by

$$\log L_c(\Psi) = \sum_{j=1}^{n}\sum_{h=1}^{m} z_{hj}\{\log \pi_h(\mathbf{x}_j; \mathbf{v}) + \log f_h(y_j|\mathbf{x}_j; \boldsymbol{\theta}_h)\},$$

where n is the total number of input—output data. On the $(t+1)$th[1] iteration of the EM algorithm, the E-step involves the computation of the so-called Q-function, which is given by the expected value of the complete-data log likelihood conditioned on the observed input—output and the current model [11]. That is

$$\begin{aligned} Q(\Psi; \Psi^{(t)}) &= E_{\Psi^{(t)}}\{\log L_c(\Psi)|\mathbf{y}, \mathbf{x}\} \\ &= \sum_{j=1}^{n}\sum_{h=1}^{m} E_{\Psi^{(t)}}(Z_{hj}|\mathbf{y}, \mathbf{x})\{\log \pi_h(\mathbf{x}_j; \mathbf{v}) + \log f_h(y_j|\mathbf{x}_j; \boldsymbol{\theta}_h)\} \\ &= \sum_{j=1}^{n}\sum_{h=1}^{m} \tau_{hj}^{(t)} \log \pi_h(\mathbf{x}_j; \mathbf{v}) + \sum_{j=1}^{n}\sum_{h=1}^{m} \tau_{hj}^{(t)} \log f_h(y_j|\mathbf{x}_j; \boldsymbol{\theta}_h), \end{aligned} \tag{4}$$

where

$$\tau_{hj}^{(t)} = E_{\Psi^{(t)}}(Z_{hj}|\mathbf{y}, \mathbf{x}) = \frac{\pi_h(\mathbf{x}_j; \mathbf{v}^{(t)})\, f_h(y_j|\mathbf{x}_j; \boldsymbol{\theta}_h^{(t)})}{\sum_{r=1}^{m} \pi_r(\mathbf{x}_j; \mathbf{v}^{(t)})\, f_r(y_j|\mathbf{x}_j; \boldsymbol{\theta}_r^{(t)})} \tag{5}$$

is the current estimated posterior probability that y_j belongs to the h-th expert $(h = 1, \ldots, m)$.

The M-step updates the estimates that maximizes the Q-function over the parameter space [11]. In (4), it can be seen that the Q-function can be decomposed into two terms with respect to $\mathbf{v}$and $\boldsymbol{\theta}$, corresponding to the gating and expert networks, respectively. It implies that separate maximizations can be performed independently [17]. For most members of the exponential family for the local output density (such as Gaussian distribution), the second term of (4) can be further decomposed into m terms corresponding to each expert network [8]. However, within the gating network, it can be seen from (2) that the parameter vector $\mathbf{v}_h$ for the h-th expert depends also on other parameter vectors $\mathbf{v}_l (l = 1, \ldots, m-1)$. It means that each parameter vector $\mathbf{v}_h$ cannot be updated independently. In this paper, we adopt the learning process via the ECM algorithm as proposed by Ng and McLachlan [8]. With the ECM algorithm, the M-step is replaced by several computationally simpler conditional-maximization (CM) steps. More importantly, each CM-step corresponds to a separable set of parameters in $\mathbf{v}_h$ for $h = 1, \ldots, m-1$. The ECM algorithm also preserves the appealing convergence properties of the EM algorithm, such as the monotone increasing of likelihood after each iteration [11,20]. A detailed formulation of the CM steps is given in Appendix A. With the gating and expert networks as specified, respectively, by (2) and (3), it follows from Appendix A that the maximization of the Q-function (4) in the M-step leads to the following updating rules:

$$\mathbf{v}_h^{(t+1)} = \mathbf{v}_h^{(t)} + [\mathbf{S}_{v,h}^{(t+h/(m-1))}]^{-1} \mathbf{E}_h^{(t+h/(m-1))}, \tag{6}$$

$$\mathbf{w}_h^{(t+1)} = [\mathbf{S}_{xx,h}^{(t)}]^{-1} \mathbf{S}_{xy,h}^{(t)}, \tag{7}$$

$$\sigma_h^{2^{(t+1)}} = \frac{(S_{yy,h}^{(t)} - \mathbf{w}_h^{(t+1)^{\mathrm{T}}} \mathbf{S}_{xy,h}^{(t)})}{S_{1,h}^{(t)}}. \tag{8}$$

In (6), we have

$$\mathbf{E}_h^{(t+h/(m-1))} = \sum_{j=1}^{n} (\tau_{hj}^{(t)} - \pi_h(\mathbf{x}_j; \mathbf{v}^{(t+h/(m-1))}))\mathbf{x}_j$$

and

$$\mathbf{S}_{v,h}^{(t+h/(m-1))} = \sum_{j=1}^{n} \pi_h(\mathbf{x}_j; \mathbf{v}^{(t+h/(m-1))})(1 - \pi_h(\mathbf{x}_j; \mathbf{v}^{(t+h/(m-1))}))\mathbf{x}_j\mathbf{x}_j^{\mathrm{T}},$$

where $\mathbf{v}^{(t+h/(m-1))}$ indicates that, in calculating $\pi_h(\mathbf{x}_j; \mathbf{v}^{(t+h/(m-1))})$, $\mathbf{v}_l (l < h)$ are fixed at $\mathbf{v}_l^{(t+1)}$ while $\mathbf{v}_l (l > h)$ are fixed at $\mathbf{v}_l^{(t)}$. In (7) and (8), we have

$$\mathbf{S}_{xx,h}^{(t)} = \sum_{j=1}^{n} \tau_{hj}^{(t)} \mathbf{x}_j \mathbf{x}_j^{\mathrm{T}},$$

$$\mathbf{S}_{xy,h}^{(t)} = \sum_{j=1}^{n} \tau_{hj}^{(t)} y_j \mathbf{x}_j,$$

$$S_{yy,h}^{(t)} = \sum_{j=1}^{n} \tau_{hj}^{(t)} y_j^2,$$

$$S_{1,h}^{(t)} = \sum_{j=1}^{n} \tau_{hj}^{(t)}.$$

[1] Here and after, the iteration number is labeled by t. We purposely do so for the consistency of labeling, as each incremental update of estimates considered in Section 3 corresponds to an input—output datum being known over time.

These equations may be referred to as the conditional expectations of the sufficient statistics, using the value $\Psi^{(t)}$ for Ψ [21].

3. Formulation of incremental learning algorithm

With the batch-mode learning described in the previous subsection, the unknown parameters are updated after the entire training set is available. In this section, we derive an incremental ECM learning algorithm where the unknown parameters are updated whenever an input-output datum becomes known. The work on speeding up the convergence of the EM algorithm with an incremental EM (IEM) algorithm [21] forms the basis for the formulation of an incremental learning process, which updates the unknown parameters via the M-step when a single input-output datum is available. Related work in the context of recursive (incremental) learning algorithms can be found in [22]. In particular, stochastic approximation procedures have been considered for the recursive estimation of parameters that can be linked to the EM algorithm.

Let $\Psi(t+1)$ be the unknown parameter vector after the *t*-th observed input—output datum $(\boldsymbol{x}_t, y_t)$. With the incremental updating version of the EM algorithm for the on-line prediction of LOS, an important feature is the introduction of a discount factor that gradually ''forgets'' the effect of the old posterior probabilities (5) obtained from earlier inaccurate estimates [17,23]. The idea is to introduce a discount parameter γ, where $0<\gamma<1$, such that the sufficient statistics required in the learning process (Eqs. (6)—(8)) are decayed exponentially with a multiplicative factor γ as the learning proceeds. For example, $\boldsymbol{S}_{xx,h}^{(t)}$in (7) can be updated incrementally as

$$\boldsymbol{S}_{xx,h}^{(t)} = \gamma \boldsymbol{S}_{xx,h}^{(t-1)} + \tau_{ht}^{(t)} \boldsymbol{x}_t \boldsymbol{x}_t^{\mathrm{T}}. \tag{9}$$

The discount parameter γ is related to the degree of discounting past examples. When γ is relatively small, the network tends to forget the past learning result and to adapt quickly to the input—output pattern [24,25]. On the other hand, a larger γ implies that a larger effect on learning the unknown parameters has imposed from past examples. When the discount parameter $\gamma$is scheduled to approach one as t tends to infinity, the updating rules so formed can be considered as a stochastic approximation for obtaining the ML estimators [22,23]. For example, Jordan and Jacobs [17] initialised γ to be 0.99 and increased a fixed fraction (0.6) of the remaining distance to 1.0 every 1000 time steps. A similar schedule was adopted in [26], where a fixed fraction of 0.0007 for every time step was used. Travén [27], on the other hand, considered the *d* most recent datapoints and specified $\gamma = (1 - 1/d)$. In this paper, we adopt the scheme in [26] with an initial value of 0.99 for γ.

Based on the incremental scheme (9), we obtain the incremental analog of the updating rules (Eqs. (6)—(8)) as follows (details are given in Appendix B):

$$\boldsymbol{v}_h^{(t+1)} = \boldsymbol{v}_h^{(t)} + [\boldsymbol{S}_{v,h}^{(t+h/(m-1))}]^{-1}(\tau_{ht}^{(t)} - \pi_h(\boldsymbol{x}_t; \boldsymbol{v}^{(t+h/(m-1))}))\boldsymbol{x}_t, \tag{10}$$

$$\boldsymbol{w}_h^{(t+1)} = \boldsymbol{w}_h^{(t)} + [\boldsymbol{S}_{xx,h}^{(t)}]^{-1} \tau_{ht}^{(t)} \boldsymbol{x}_t (y_t - \boldsymbol{x}_t^{\mathrm{T}} \boldsymbol{w}_h^{(t)}), \tag{11}$$

$$\sigma_h^{2^{(t+1)}} = \frac{(S_{yy,h}^{(t)} - \boldsymbol{w}_h^{(t+1)^{\mathrm{T}}} \boldsymbol{S}_{xy,h}^{(t)})}{S_{1,h}^{(t)}} = \frac{\gamma S_{yy,h}^{(t-1)} + \tau_{ht}^{(t)} y_t^2 - \boldsymbol{w}_h^{(t+1)^{\mathrm{T}}} \boldsymbol{S}_{xy,h}^{(t)}}{\gamma S_{1,h}^{(t-1)} + \tau_{ht}^{(t)}} \tag{12}$$

In (10) and (11), the direct calculation of the inverses of $\boldsymbol{S}_{v,h}$ and $\boldsymbol{S}_{xx,h}$ can be avoided by using the efficient updating formula [21] that specify the new matrix inverse in terms of the old one; see Appendix B. Letting $\Lambda_{v,h}^{(t+h/(m-1))} = [\boldsymbol{S}_{v,h}^{(t+h/(m-1))}]^{-1}$and $\pi_h = \pi_h(\boldsymbol{x}_t; \boldsymbol{v}^{(t-1+h(m-1))})$, we have

$$\Lambda_{v,h}^{(t+h/(m-1))} = \frac{1}{\gamma}\left(\Lambda_{v,h}^{(t-1+h/(m-1))} - \frac{\pi_h(1-\pi_h)\Lambda_{v,h}^{(t-1+h/(m-1))}\boldsymbol{x}_t\boldsymbol{x}_t^{\mathrm{T}}\Lambda_{v,h}^{(t-1+h/(m-1))}}{\gamma+\pi_h(1-\pi_h)\boldsymbol{x}_t^{\mathrm{T}}\Lambda_{v,h}^{(t-1+h/(m-1))}\boldsymbol{x}_t}\right).$$

Similarly, letting $\Lambda_{xx,h}^{(t)} = [\boldsymbol{S}_{xx,h}^{(t)}]^{-1}$, we have

$$\Lambda_{xx,h}^{(t)} = \frac{1}{\gamma}\left(\Lambda_{xx,h}^{(t-1)} - \frac{\tau_{ht}^{(t)}\Lambda_{xx,h}^{(t-1)}\boldsymbol{x}_t\boldsymbol{x}_t^{\mathrm{T}}\Lambda_{xx,h}^{(t-1)}}{\gamma + \tau_{ht}^{(t)}\boldsymbol{x}_t^{\mathrm{T}}\Lambda_{xx,h}^{(t-1)}\boldsymbol{x}_t}\right).$$

4. On-line prediction and model selection

The ME network can be incrementally learned using Eqs. (10)—(12) from known input—output data. Whenever a new input $\boldsymbol{x}_t$ becomes available, a on-line prediction of LOS is given by the expected value of the output y_t in (1), conditioned on the input $\boldsymbol{x}_t$ and the current estimates of the parameters $\Psi(t)$; see Section 2.1. A 95% confidence interval (CI) of the prediction can be formed using a resampling approach [28]. Samples of size 100 for y_t are generated independently from $f(y_t|\boldsymbol{x}_t;\,\Psi(t))$. The 3rd and the 98th ordered samples of y_t can be used to estimate the 95% confidence limits for the prediction. If the actual LOS observed is outside the 95% CI of the prediction, this datum may be considered as an "outlier" and therefore inappropriate for updating the parameters of the model. These outliers, however, should be checked frequently with new input data because they may indicate a systematic change in the input—output pattern.

The intelligent system of ME networks with incremental learning process as described in the previous section requires the initialization of unknown parameters. This can be proceeded by applying the batch-mode learning on an "initialization" data set consisting of some past examples. The initialization data set can also be used to initialize the "structural" parameters of the ME network such as the number of expert networks and the number of covariates in each input vector. These two aspects are referred to as the main subproblems in model selection in the context of neural networks [29]. In particular, the selection of the number of expert networks is relevant in modeling dynamic environments where the input—output pattern changes over time [23]. Hence, with the incremental learning process, another important aspect required is the provision of pruning and growing of expert networks for adapting with varying environments as the learning proceeds.

Within the Bayesian framework, Jacobs et al. [29] considered ways of assessing whether a given ME networks should be pruned or grown. They defined the worth index for the h-th expert, based on the indicator variables z_{hj} over the observed data, as

$$I_h = \sum_{j=1}^{n} \frac{z_{hj}}{n} \quad (h = 1, \ldots, m), \tag{13}$$

where the unknown variable z_{hj} was estimated by the average of its generated values on a specified number of simulations. Here, we consider a frequentist analog of (13) where z_{hj} is replaced by its estimated conditional expectation $\tau_{hj}^{(t)}$ with the current estimates $\Psi(t)$. If the indices for the experts are all of similar magnitudes, for example, they are within 10% of $1/m$, then a network with additional experts may be considered. Alternatively, if the index for an expert is small relative to that of other experts, this expert can be pruned from the architecture [29]. In practice, the threshold index values for pruning and growing of expert networks should be carefully designed [23]. Mixture of experts with too many free parameters (or too many experts) tend to overfit the training data and show poor generalization performance. In contrast, networks with as few free parameters as possible but are still adequate for summarizing the data tend to generalize comparatively well. A useful criterion suggested by Jacobs et al. [29] to determine the number of experts is the minimum number of experts with the largest worth indices for which the sum of their worth indices exceeds some critical value κ. For example, with a ME network of m^* modules, we select

$$\min \left\{ m : m < m^* \text{ and } \sum_{h=1}^{m} I_{(h)} > \kappa \right\}, \tag{14}$$

where $I_{(1)} \geq I_{(2)} \geq \cdots \geq I_{(m^*)}$ are ordered worth indices. All other $(m^* - m)$ expert networks can be pruned from the model. Jacobs et al. [29] suggested the value of $\kappa = 0.8$. Another criterion is proposed by Ishii and Sato [26], who suggest that expert networks with $I_h < 0.1/m^*$ are pruned. Hence, their criterion will select a larger number of experts as compared to that given by Jacobs et al. [29]. Although these threshold values are arbitrary, they have worked well in practice. With the incremental learning process, the worth indices can be updated according to the incremental scheme (9). The criterion (14) can be checked, say after each time step of 10. The flow diagram of the on-line prediction procedure of the proposed intelligent ME network is depicted in Fig. 2.

5. An example of gastroenteritic LOS data

In this section, the proposed incremental learning of ME networks is compared with the batch-mode learning to predict the LOS of infants admitted for gastroenteritis in Western Australia (WA). Gastroenteritis is an infectious disease prevalent among infants and children worldwide, especially in developing countries. The present data were extracted from a retrospective cohort study on all infants born in 1995—1997 and their subsequent admissions for gastroenteritis. Admissions by infants, who were born in other years, for gastroenteritis were not captured. Also, we focus on a single public tertiary hospital for children in WA. Thus, the present data set should be considered as a set of hospital-based LOS data observed over time for the cohort, rather than a data set retrieved from hospital admission database that also accumulates records from other patients. The total number of admissions in this data set was $n = 692$. Linked hospitalization records of

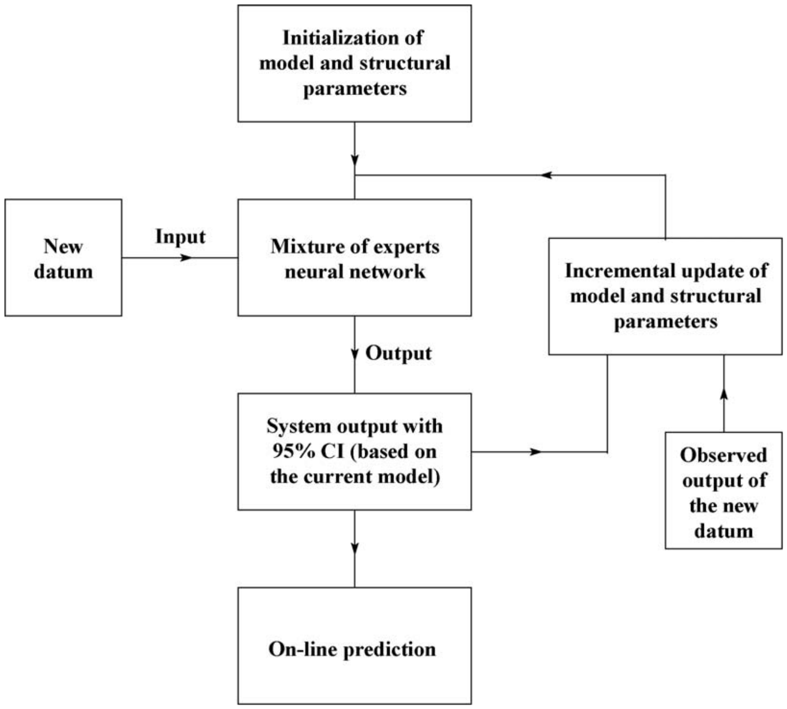
Initialization of model and structural parameters
New datum
Input
Mixture of experts neural network
Incremental update of model and structural parameters
Output
System output with 95% CI (based on the current model)
Observed output of the new datum
On-line prediction

Figure 2 The flow diagram of the proposed intelligent ME network.

the cohort were retrieved retrospectively to derive the outcome measure, patient demographics, and associated co-morbidities information [30]. The observed LOS (outcome measure) ranged from one (same day separation) to 82 days. As the empirical distribution of LOS appears to be positively skewed, log-transformation of LOS is applied [31]. Other transformations within the Box—Cox family of transformations [32] may also be considered and assessed using the profile likelihood, with reference to the local output densities (3) adopted in the ME network. In this study, concomitant information on each patient's age (in months), gender (female: 0; male: 1), indigenous status (non-Aboriginal: 0; Aboriginal: 1), place of residence (metropolitan: 0; rural: 1), and number of co-morbidities (0—5) were included as the input variables of the ME network. These variables are considered as potential determinants of LOS for gastroenteritis [5]. Table 1 presents the patient demographic and descriptive measures at admission.

The LOS input—output data are rearranged in order according to the date of admission. As described in Section 4, the first 100 ordered samples are chosen as the initialization set for the determination of initial model and structural parameters. Table 2 displays the results for the model selection. We here take $\kappa = 0.9$ in (14) because an initial value

Table 1 Patient demographic and descriptive characteristics

Number of gastroenteritis admissions	692
Average (standard deviation) LOS in days	4.5 (5.9)
Average (standard deviation) patient age in months	8.4 (6.9)
Proportion of admissions (%)	
Male	55.3
Aboriginal	17.3
Rural residence	8.7
Average (standard deviation) number of co-morbidities	0.6 (0.8)
Co-morbidity (%)	
Dehydration	29.9
Gastrointestinal sugar intolerance	10.3
Failure to thrive	6.5
Iron deficiency anaemia	4.2
Infection (genitourinary/scabies/otitis media)	8.5

Table 2 Model selection

Number of experts	Worth indices (I)	Remark
2	(0.69, 0.31)	$I_{(1)} < 0.9$
3	(0.42, 0.37, 0.21)	$I_{(1)} + I_{(2)} < 0.9$
4[a]	(0.30, 0.27, 0.23, 0.20)	$I_{(1)} + I_{(2)} + I_{(3)} < 0.9$
5	(0.46, 0.26, 0.16, 0.10, 0.02)	$I_{(1)} + I_{(2)} + I_{(3)} + I_{(4)} > 0.9$

[a] The number of experts selected.

Table 3 Prediction of LOS for gastroenteritis

Learning method	Data set $D_1 (n_1 = 500)$		Data set D_2 $(n_2 = 92)$	
	MAD	Prop(MAD $\leq$ 1)	MAD	Prop(MAD $\leq$ 1)
Batch mode learning	2.03 days	49.2%	2.03 days	42.4%
Incremental learning	2.13 days	46.2%	1.77 days	54.3%

for m is determined based on a small initialization set. A ME network with $m = 4$ experts is selected and the initial estimates of the parameters Ψ are obtained. The incremental learning algorithm (Eqs. (10)—(12)) are applied to the remaining data $(n = 592)$. In this study, the criterion (14) is checked after each time step of 10 during the incremental learning process to determine whether the pruning or growing of the ME network is required. For comparison, we also include the predictions of LOS obtained by the batch-mode learning algorithm (Eqs. (6)—(8)). With the batch-mode learning process, the first 500 ordered samples of the remaining data ($n = 592$) is used as the training set D_1 to train the ME network. The relative performance of the incremental and batch-mode learning algorithms is compared separately on data sets D_1 and D_2 (the remaining 92 samples). In Table 3, the mean absolute difference (MAD) between the predictions and the actual LOS along with the proportion of predictions with MAD less than or equal to 1 day (Prop(MAD $\leq$ 1)) are presented. It can be seen that the batch-mode learning algorithm performs slightly better than the incremental learning algorithm in the data set D_1. However, when the intelligent system has gained sufficient learning from more examples, the incremental learning algorithm provides better online prediction as indicated in the data set D_2. This improvement is quantified in Fig. 3, where the MAD and Prop(MAD $\leq$ 1) for each 30 input—output examples are plotted against the time frame. Outliers that are outside the 95% CI of the predictions are excluded in this regression analysis. The regression fitted lines in Fig. 3 indicate a significant decrease of MAD (p- value $= 0.063$) and a significant (p-value $= 0.044$) increase of Prop(MAD $\leq$ 1), as time proceeds.

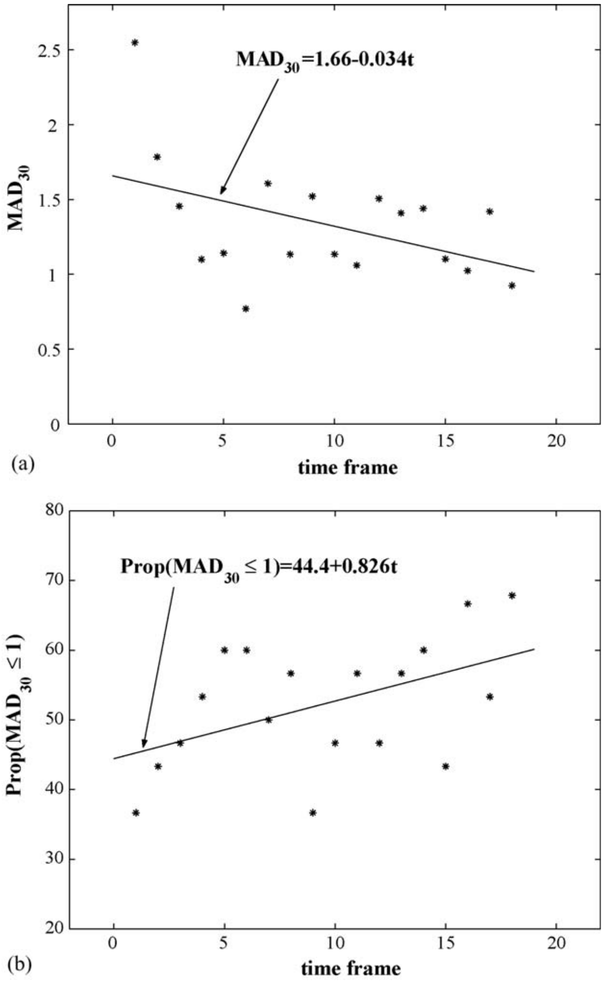

Figure 3 The performance of the incremental learning algorithm as time proceeds: (a) mean absolute difference for each 30 input—output examples (MAD$_{30}$) along the time frame; (b) proportion of predictions with MAD$_{30}$ less than or equal to 1 day (Prop(MAD$_{30}$$\leq$1)) along the time frame.

6. Concluding remarks

Intelligent decision support systems such as neural networks and machine learning algorithms have been of much interest in recent years in their applications to health care studies. Different initiatives propel the development and use of intelligent systems in health care leading to significant advances in the field. As LOS is an important measure of hospital activity and health care utilization, an accurate prediction of LOS has important implications in various aspects of health care management and decision-making.

In this paper, we have developed an intelligent ME network that "reads" the input data that become available over time and enables the on-line prediction of LOS. The strength of the proposed intelligent system is that it provides on-the-spot information for determining appropriate policies in hospital resource utilization management and making financial decisions for improving economic efficiency [1]. The on-line information also contributes towards an early prediction of patients who will require longer hospital care [6] and the achievements of artificial intelligence scheduling in clinical management concerning the minimization of patient stay in hospital [33]. At the same time, the comparison of LOS predictions can be taken into consideration for the assessment of the global influence of preventive and therapeutic interventions or the allocation of resources [7]. Another important feature of the intelligent system is that the architecture of the ME network can be determined from the data itself via pruning or growing the ME network. The sequential discounting feature and the self-adaptive model-selection ability of the system enhance its effective adaptation to non-stationary LOS data, in terms of memory and computational efficiency (cf. [21,24]). It is noted that the learning behavior changes according to the scheduling of the discount parameter γ(Section 3). With non-stationary input-output pattern of data, the MAD and the number of experts m should be tracked during the incremental updating process. A drastic increase of m may indicate that the value of γ is too large and the network adapts to the change of the input distribution without forgetting past examples by growing of expert networks [23].

The example presented in Section 5 shows that the performance of the batch-mode algorithm in LOS predictions deteriorates in the test set. It means that the batch-mode algorithm fails to adapt to the changing input-output distribution of data over time. An additional disadvantage of the batch-mode learning is the heavy burden of storing the training data. On the other hand, it is observed that the performance of the incremental learning algorithm improves when the system learns from more examples as time proceeds. The results from the example demonstrate that the incremental learning on-line system outperforms the batch-mode learning algorithm in the prediction of LOS for gastroenteritis. In related work, the uses of incremental algorithms have been successfully applied to the motion-based tracking of objects [13], robot dynamics problems [23], and the reconstruction of nonlinear dynamics [26].

Mixture of experts networks are useful in modelling nonlinear mappings between the input—output patterns, mainly due to their wide applicability [17,29], generalization capability, and advantage of fast learning via EM-based algorithms [8,17]. A theoretical analysis of the convergence of the EM algorithm for the ME networks is provided in [19]. Jiang and Tanner [34] have obtained conditions for the identifiability of the ME network, which they showed held for some commonly used expert networks such as Poisson, gamma, and Gaussian experts. For the specification of the local output density $f_h(\boldsymbol{y}_j|\boldsymbol{x}_j;\,\boldsymbol{\theta}_h)$ within the ME network, instead of Gaussian distribution, other members of the exponential family of densities may be adopted in views of the skewness of the LOS data. An example is given by Lee et al. [4], where a gamma mixture model is used for the analysis of maternity LOS data. In addition, the normal inverse Gaussian distribution [35], which also belongs to the exponential family of densities, may be adopted to handle skewed and heavy-tailed data. The assumption of a Gaussian distribution for $f_h(\boldsymbol{y}_j|\boldsymbol{x}_j;\,\boldsymbol{\theta}_h)$ in (3) is motivated by its simple form and well-established theoretical framework within the context of ML estimation via EM-based algorithms. Although the focus of this paper is on the prediction of LOS, the proposed intelligent ME system should be readily applicable to predict other clinical outcomes or health care resource measures. For example, the system can be used to predict disease severity or prognostic indices [6], a patient's acuity measure, health insurance payment, and payment rate for each hospital stay [36].

Acknowledgements

The authors are indebted to Diana Rosman and William Pyper of the Department of Health, Western Australia for providing the linked data and coding information. Thanks are also due to the editor and three anonymous reviewers for their helpful comments and suggestions. This work was supported in part by grants from the Australian Research Council and the National Health and Medical Research Council of Australia.

Appendix A. Learning via the ECM algorithm

With the ECM algorithm, the parameter vector $\boldsymbol{v}$ is partitioned as $(\boldsymbol{v}_1^{\mathrm{T}}, \ldots, \boldsymbol{v}_{m-1}^{\mathrm{T}})^{\mathrm{T}}$. On the $(t+1)$th iteration of the ECM algorithm, the M-step is replaced by $(m-1)$ computationally simpler CM-steps:

- CM-step 1: Calculate $\boldsymbol{v}_1^{(t+1)}$ by maximizing Q_v with $\boldsymbol{v}_l (l = 2, \ldots, m-1)$ fixed at $\boldsymbol{v}_l^{(t)}$.
- CM-step 2: Calculate $\boldsymbol{v}_2^{(t+1)}$ by maximizing Q_v with $\boldsymbol{v}_1$ fixed at $\boldsymbol{v}_1^{(t+1)}$ and $\boldsymbol{v}_l (l = 3, \ldots, m-1)$ fixed at $\boldsymbol{v}_l^{(t)}$.
- $\vdots$
- CM-step $(m-1)$: Calculate $\boldsymbol{v}_{(m-1)}^{(t+1)}$ by maximizing Q_v with $\boldsymbol{v}_l (l = 1, \ldots, m-2)$ fixed at $\boldsymbol{v}_l^{(t+1)}$,

where

$$Q_v = \sum_{j=1}^{n} \sum_{h=1}^{m} \tau_{hj}^{(t)} \log \pi_h(\boldsymbol{x}_j; \boldsymbol{v})$$

is the term of the Q-function in (4) for the gating network. As the CM maximizations are over smaller dimensional parameter space, they are often simpler and more stable than the corresponding full maximization called for on the M-step of the EM algorithm, especially when iteration is required [16]. More importantly, each CM-step above corresponds to a separable set of the parameters in $\boldsymbol{v}_h$ for $h = 1, \ldots, m-1$, and can be obtained using the iterative reweighted least squares (IRLS) algorithm of Jordon and Jacobs [17]; see [8].

Let $\boldsymbol{v}^{(t+h/(m-1))} = (\boldsymbol{v}_1^{(t+1)^{\mathrm{T}}}, \ldots, \boldsymbol{v}_{h-1}^{(t+1)^{\mathrm{T}}}, \boldsymbol{v}_h^{(t)^{\mathrm{T}}}, \ldots, \boldsymbol{v}_{m-1}^{(t)^{\mathrm{T}}})^{\mathrm{T}}$, at the h-th CM-step on the $(t+1)$th iteration of the ECM algorithm $(h = 1, \ldots, m-1)$, it follows from (2) that the IRLS updating rule for $\boldsymbol{v}_h$ is given by

$$\boldsymbol{v}_h^{(t+1)} = \boldsymbol{v}_h^{(t)} + \left[-\frac{\partial^2 Q_v}{\partial \boldsymbol{v}_h \boldsymbol{v}_h^{\mathrm{T}}} \right]_{(t+h/(m-1))}^{-1} \left[\frac{\partial Q_v}{\partial \boldsymbol{v}_h} \right]_{(t+h/(m-1))},$$

where

$$\left[\frac{\partial Q_v}{\partial \boldsymbol{v}_h} \right]_{(t+h/(m-1))} = \boldsymbol{E}_h^{(t+h/(m-1))} = \sum_{j=1}^{n} (\tau_{hj}^{(t)} - \pi_h(\boldsymbol{x}_j; \boldsymbol{v}^{(t+h/(m-1))})) \boldsymbol{x}_j$$

and

$$\left[-\frac{\partial^2 Q_v}{\partial \boldsymbol{v}_h \boldsymbol{v}_h^{\mathrm{T}}} \right]_{(t+h/(m-1))} = \boldsymbol{S}_{v,h}^{(t+h/(m-1))} = \sum_{j=1}^{n} \pi_h(\boldsymbol{x}_j; \boldsymbol{v}^{(t+h/(m-1))}) \times (1 - \pi_h(\boldsymbol{x}_j; \boldsymbol{v}^{(t+h/(m-1))})) \boldsymbol{x}_j \boldsymbol{x}_j^{\mathrm{T}}.$$

This IRLS loop is referred to as the inner loop of the EM algorithm [17]. It is terminated when the algorithm has converged or after some prespecified number of iterations, say 10 iterations. A least squares algorithm has been considered by Jordan and Jacobs [17] to replace the ML learning approach. In that case, the gating network parameters $\boldsymbol{v}_h (h = 1, \ldots, m-1)$ can be fit by a one-pass of the least squares algorithm. Although this algorithm was found to work reasonably well in practice, even in the early stages of fitting when the residuals can be large, there is no guarantee that the appealing convergence properties of the EM algorithm can be preserved.

Appendix B. Incremental updating rules

As described in Section 3, incremental updating rules can be formulated based on Scheme (9) that gradually discounts the effect of previous sufficient statistics. Here we focus on $\boldsymbol{w}_h (h = 1, \ldots, m)$ as an example. The derivation for $\boldsymbol{v}_h$ can be obtained similarly. By replacing $\boldsymbol{S}_{xx,h}^{(t)}$ and $\boldsymbol{S}_{xy,h}^{(t)}$ in (7) with their incremental analogs such as (9), an incremental analog of (7) is given by

$$\begin{aligned} \boldsymbol{w}_h^{(t+1)} &= [\boldsymbol{S}_{xx,h}^{(t)}]^{-1} \boldsymbol{S}_{xy,h}^{(t)} = [\gamma \boldsymbol{S}_{xx,h}^{(t-1)} + \tau_{ht}^{(t)} \boldsymbol{x}_t \boldsymbol{x}_t^{\mathrm{T}}]^{-1} \\ &\quad \times (\gamma \boldsymbol{S}_{xy,h}^{(t-1)} + \tau_{ht}^{(t)} y_t \boldsymbol{x}_t) = [\gamma \boldsymbol{S}_{xx,h}^{(t-1)} + \tau_{ht}^{(t)} \boldsymbol{x}_t \boldsymbol{x}_t^{\mathrm{T}}]^{-1} \\ &\quad \times (-\tau_{ht}^{(t)} \boldsymbol{x}_t \boldsymbol{x}_t^{\mathrm{T}} \boldsymbol{w}_h^{(t)} + \tau_{ht}^{(t)} y_t \boldsymbol{x}_t) + \boldsymbol{w}_h^{(t)} \\ &= \boldsymbol{w}_h^{(t)} + \tau_{ht}^{(t)} [\gamma \boldsymbol{S}_{xx,h}^{(t-1)} + \tau_{ht}^{(t)} \boldsymbol{x}_t \boldsymbol{x}_t^{\mathrm{T}}]^{-1} \boldsymbol{x}_t \\ &\quad \times (y_t - \boldsymbol{x}_t^{\mathrm{T}} \boldsymbol{w}_h^{(t)}). \end{aligned} \tag{15}$$

It can be seen from (15) that direct calculation of the inverse of $\boldsymbol{S}_{xx,h}^{(t)}$ for each t is required. Fortunately, this computation can be avoided by using the updating formula given in Ng and McLachlan [21]. Let

$$\Lambda_{xx,h}^{(t)} = [\boldsymbol{S}_{xx,h}^{(t)}]^{-1} = [\gamma \boldsymbol{S}_{xx,h}^{(t-1)} + \tau_{ht}^{(t)} \boldsymbol{x}_t \boldsymbol{x}_t^{\mathrm{T}}]^{-1}.$$

From [21], the inverse is given by

$$
\begin{aligned}
\Lambda^{(t)}_{xx,h} &= [\gamma \boldsymbol{S}^{(t-1)}_{xx,h} + \tau^{(t)}_{ht}\boldsymbol{x}_t\boldsymbol{x}_t^{\mathrm{T}}]^{-1} \\
&= \frac{1}{\gamma}\Lambda^{(t-1)}_{xx,h} - \frac{\tau^{(t)}_{ht}(\frac{1}{\gamma})^2\Lambda^{(t-1)}_{xx,h}\boldsymbol{x}_t\boldsymbol{x}_t^{\mathrm{T}}\Lambda^{(t-1)}_{xx,h}}{1+\frac{1}{\gamma}\tau^{(t)}_{ht}\boldsymbol{x}_t^{\mathrm{T}}\Lambda^{(t-1)}_{xx,h}\boldsymbol{x}_t} \\
&= \frac{1}{\gamma}\left(\Lambda^{(t-1)}_{xx,h} - \frac{\tau^{(t)}_{ht}\Lambda^{(t-1)}_{xx,h}\boldsymbol{x}_t\boldsymbol{x}_t^{\mathrm{T}}\Lambda^{(t-1)}_{xx,h}}{\gamma+\tau^{(t)}_{ht}\boldsymbol{x}_t^{\mathrm{T}}\Lambda^{(t-1)}_{xx,h}\boldsymbol{x}_t}\right).
\end{aligned}
$$

The use of this updating formula avoids the direct calculation of the inverse of $\boldsymbol{S}^{(t)}_{xx,h}$ in the incremental updating rule for $\boldsymbol{w}^{(t+1)}_h$ and reduces the amount of computation time.

References

[1] Dombi GW, Nandi P, Saxe JM, Ledgerwood AM, Lucas CE. Prediction of rib fracture injury outcome by an artificial neural-network. J Trauma 1995;39:915—21.

[2] Park J, Edington DW. A sequential neural network model for diabetes prediction. Artif Intell Med 2001;23:277—93.

[3] Walczak S, Pofahl WE, Scorpio RJ. A decision support tool for allocating hospital bed resources and determining required acuity of care. Decis Support Syst 2003;34:445—56.

[4] Lee AH, Ng SK, Yau KKW. Determinants of maternity length of stay: a gamma mixture risk-adjusted model. Health Care Manage Sci 2001;4:249—55.

[5] Wang K, Yau KKW. Lee AH. Factors influencing hospitalisation of infants for recurrent gastroenteritis in Western Australia. Meth Inform Med 2003;42:251—4.

[6] Pofahl WE, Walczak SM, Rhone E, Izenberg SD. Use of an artificial neural network to predict length of stay in acute pancreatitis. Am Surgeon 1998;64:868—72.

[7] Lowell WE, Davis GE. Predicting length of stay for psychiatric diagnosis-related groups using neural networks. J Am Med Inform Assoc 1994;1:459—66.

[8] Ng SK, McLachlan GJ. Using the EM algorithm to train neural networks: misconceptions and a new algorithm for multiclass classification. IEEE Trans Neural Netw 2004;15:738—49.

[9] Ripley BD. Statistical aspects of neural networks. In: Barndorff-Nielsen OE, Jensen JL, Kendall WS, editors. Networks and chaos — statistical and probabilistic aspects. London: Chapman & Hall; 1993. p. 40—123.

[10] Dempster AP, Laird NM, Rubin DB. Maximum likelihood from incomplete data via the EM algorithm (with discussion). J R Stat Soc B 1977;39:1—38.

[11] McLachlan GJ, Krishnan T. The EM algorithm and extensions. New York: Wiley, 1997.

[12] Aitkin M, Foxall R. Statistical modelling of artificial neural networks using the multi-layer perceptron. Stat Comput 2003;13:227—39.

[13] Jepson AD, Fleet DJ, El-Maraghi TF. Robust online appearance models for visual tracking. IEEE Trans Pattern Anal 2003;25:1296—311.

[14] Lai SH, Fang M. An adaptive window width/center adjustment system with online training capabilities for MR images. Artif Intell Med 2005;33:89—101.

[15] Jacobs RA, Jordan MI, Nowlan SJ, Hinton GE. Adaptive mixtures of local experts. Neural Comput 1991;3:79—87.

[16] Meng XL, Rubin DB. Maximum likelihood estimation via the ECM algorithm: a general framework. Biometrika 1993;80:267—78.

[17] Jordan MI, Jacobs RA. Hierarchical mixtures of experts and the EM algorithm. Neural Comput 1994;6:181—214.

[18] Bridle JS. Probabilistic interpretation of feedforward classification network outputs, with relationships to statistical pattern recognition. In: Soulié FF, Hérault J, editors. Neurocomputing: algorithms, architectures and applications. Berlin, Germany: Springer; 1990. p. 227—36.

[19] Jordan MI, Xu L. Convergence results for the EM approach to mixtures of experts architectures. Neural Netw 1995;8:1409—31.

[20] Meng XL. On the rate of convergence of the ECM algorithm. Ann Stat 1994;22:326—39.

[21] Ng SK, McLachlan GJ. On the choice of the number of blocks with the incremental EM algorithm for the fitting of normal mixtures. Stat Comput 2003;13:45—55.

[22] Titterington DM. Recursive parameter estimation using incomplete data. J R Stat Soc B 1984;46:257—67.

[23] Sato M, Ishii S. On-line EM algorithm for the normalized Gaussian network. Neural Comput 2000;12:407—32.

[24] Neal RM, Hinton GE. A view of the EM algorithm that justifies incremental, sparse, and other variants. In: Jordan MI, editor. Learning in graphical models. Dordrecht: Kluwer; 1998. p. 355—68.

[25] Ng SK, McLachlan GJ. Speeding up the EM algorithm for mixture model-based segmentation of magnetic resonance images. Pattern Recogn 2004;37:1573—89.

[26] Ishii S, Sato M. Reconstruction of chaotic dynamics by on-line EM algorithm. Neural Netw 2001;14:1239—56.

[27] Travén HGC. A neural network approach to statistical pattern classification by "semiparametric" estimation of probability density functions. IEEE Trans Neural Netw 1991;2:366—77.

[28] Efron B. The Jackknife the bootstrap and other resampling plans. Philadelphia: SIAM, 1982.

[29] Jacobs RA, Peng F, Tanner MA. A Bayesian approach to model selection in hierarchical mixtures-of-experts architectures. Neural Netw 1997;10:231—41.

[30] Lee AH, Flexman J, Wang K, Yau KKW. Recurrent gastroenteritis among infants in Western Australia: a 7-year hospital-based cohort study. Ann Epidemiol 2004;14:137—42.

[31] Leung KM, Elashoff RM, Rees KS, Hasan MM, Legorreta AP. Hospital- and patient-related characteristics determining maternity length of stay: a hierarchical linear model approach. Am J Public Health 1998;88:377—81.

[32] Box GEP. Cox DR. The analysis of transformations (with discussion). J R Stat Soc B 1964;26:211—52.

[33] Spyropoulos CD. AI planning and scheduling in the medical hospital environment. Artif Intell Med 2000;20:101—11.

[34] Jiang W, Tanner MA. On the identifiability of mixtures-of-experts. Neural Netw 1999;12:1253—8.

[35] Øigård TA, Hanssen A, Hansen RE, Godtliebsen F. EM-estimation and modeling of heavy-tailed processes with the multivariate normal inverse Gaussian distribution. Signal Process 2005;85:1655—73.

[36] Quantin C, Sauleau E, Bolard P. Modeling of high-cost patient distribution within renal failure diagnosis related group. J Clin Epidemiol 1999;52:251—8.