Title	Fast ML estimation for the mixture of factor analyzers via an ECM algorithm
Author(s)	Zhao, JH; Yu, PLH
Citation	Ieee Transactions On Neural Networks, 2008, v. 19 n. 11, p. 1956-1961
Issued Date	2008
URL	http://hdl.handle.net/10722/125405
Rights	Creative Commons: Attribution 3.0 Hong Kong License

Brief Papers

Fast ML Estimation for the Mixture of Factor Analyzers via an ECM Algorithm

Jian-Hua Zhao and Philip L. H. Yu

***Abstract*—In this brief, we propose a fast expectation conditional maximization (ECM) algorithm for maximum-likelihood (ML) estimation of mixtures of factor analyzers (MFA). Unlike the existing expectation–maximization (EM) algorithms such as the EM in Ghahramani and Hinton, 1996, and the alternating ECM (AECM) in McLachlan and Peel, 2003, where the missing data contains component–indicator vectors as well as latent factors, the missing data in our ECM consists of component–indicator vectors only. The novelty of our algorithm is that closed-form expressions in all conditional maximization (CM) steps are obtained explicitly, instead of resorting to numerical optimization methods. As revealed by experiments, the convergence of our ECM is substantially faster than EM and AECM regardless of whether assessed by central processing unit (CPU) time or number of iterations.**

***Index Terms*—Alternating expectation conditional maximization (AECM), expectation conditional maximization (ECM), expectation maximization (EM), maximum-likelihood estimation (MLE), mixture of factor analyzers (MFA).**

I. Introduction

Mixture factor analysis (MFA) models [1]–[3] and mixture probabilistic principal component analysis (MPCA) models [4] are two widely applied in recent years mixture models (MM). This can be attributed to their appealing capability to elegantly perform clustering and dimensionality reduction simultaneously and their tempting flexibility in density estimation for high-dimensional data to provide an appropriate tradeoff between usually overfitting full covariance MM and underfitting diagonal (spherical) MM. Despite their similarity, MPCA searches for directions with maximal variances while MFA looks for directions with maximal interesting correlations within each cluster. If the explained correlation is the main concern, MFA is a better choice and vice versa.

Expectation maximization (EM) [5] has been suggested for fitting MFA [1], which is easy to implement and converges stably since its M-step is in closed form. However, its missing data contains both component–indicator vectors and latent factor vectors. Given so much *missing information* is introduced, convergence of the EM for MFA can be painfully slow due to the well-known fact that the rate of convergence of EM is determined by the portion of *missing information* in complete data [5]. To accelerate a basic EM algorithm, perhaps the most natural consideration is to decrease the amount of missing data. Unfortunately, it is typically much more difficult to obtain the closed-form expressions in the resulting (C)M-steps and thus numerical methods with less simplicity or stability often have to be employed. One compromise strategy aiming to attain a suitable tradeoff between simplicity or stability and convergence is to decrease the amount of missing data only in some CM-steps, e.g., alternating expectation conditional maximization (AECM) [6]. An application of AECM to MFA with all its CM-steps in closed form is given in [2]. Similarly, for MPCA, a corresponding algorithm to AECM for MFA is presented in [4]. Nevertheless, [4] further provides a much more efficient algorithm, which can make use of eigendecompostion of *local* covariance matrices to determine loading matrices and isotropic noise variances in each iteration. This algorithm is an ECM algorithm [6] in which only component–indicator vectors are treated as missing data in the E-step followed by a sequence of CM expected complete data log-likelihood (CMQ) steps. Its high efficiency can be ascribed to the fact that all its CM-steps are in closed form.

However, to our knowledge, such an appealing ECM is vacant for MFA as it is a challenging problem to obtain closed-form (C)M-steps, and hence, at present, the EM [1] and the AECM [2] are two commonly used algorithms for fitting MFA. In this brief, we will propose an ECM algorithm for MFA to fill up this vacancy. In Section II, we first describe a CM algorithm for FA with all its CM-steps in closed form, which we recently proposed in [7]. In Section III, we extend the work to MFA model. In our ECM, factor loading matrices can be (indirectly) obtained via eigendecompostion of *normalized* local covariance matrices. In Section IV, we conduct experiments to compare the performance among the EM, the AECM, and our proposed ECM.

II. FA Model and a CM Algorithm

A. FA Model

Suppose that each d-dimensional data vector $\mathbf{x}_n$ in the i.i.d sample $\mathbf{X} = \{\mathbf{x}_n\}_{n=1}^N$ follows a q-factor model

$$\begin{cases} \mathbf{x}_n = \mathbf{A}\mathbf{y}_n + \boldsymbol{\mu} + \boldsymbol{\epsilon}_n \\ \mathbf{y}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \qquad \boldsymbol{\epsilon}_n \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Psi}), \end{cases} \tag{1}$$

where $\boldsymbol{\mu}$ is a d-dimensional mean vector, $\mathbf{A}$ is a $d \times q$ factor loading matrix, $\mathbf{y}_n$ is a q-dimensional latent factor vector, and $\boldsymbol{\Psi} = \operatorname{diag}\{\psi_1, \psi_2, \ldots, \psi_d\}$ is a positive diagonal matrix. We use $\mathbf{I}$ to denote an unit matrix whose dimension should be apparent from the context.

Under model (1), $\mathbf{x}_n \sim \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, where $\boldsymbol{\Sigma} \triangleq \boldsymbol{\Psi} + \mathbf{A}\mathbf{A}^T$. The global maximal estimation [maximum likelihood (ML)] of $\boldsymbol{\mu}$ in FA model is trivially the sample mean $\bar{\mathbf{x}}$. Thus, in this section, without loss of generality, we will set $\boldsymbol{\mu}$ equal to $\bar{\mathbf{x}}$. Then, up to a constant, the log-likelihood function of $\boldsymbol{\theta} = (\mathbf{A}, \boldsymbol{\Psi})$ is

$$\mathcal{L}(\boldsymbol{\theta}) = -\frac{N}{2}\{\ln|\boldsymbol{\Sigma}| + \operatorname{tr}(\boldsymbol{\Sigma}^{-1}\mathbf{S})\} \tag{2}$$

where $\mathbf{S}$ is the sample covariance matrix given by $\mathbf{S} = (1/N)\sum_{n=1}^N (\mathbf{x}_n - \bar{\mathbf{x}})(\mathbf{x}_n - \bar{\mathbf{x}})^T$. It is well known that there is no closed-form analytic solution if we take the derivative of log-likelihood function $\mathcal{L}$ in (2) with respect to $\boldsymbol{\theta}$, and hence, iterative procedures have to be employed for fitting FA. EM has been suggested in [8]. However, its convergence may be impractically slow, especially in low-noise case [9]. In the literature,

Manuscript received October 19, 2007; revised February 27, 2008; accepted July 25, 2008. First published September 26, 2008; current version published November 5, 2008. This work was supported by the Research Grants Council of the Hong Kong Special Administrative Region, China, under Project HKU 7176/02H.

J.-H. Zhao is with the Department of Statistics and Actuarial Science, The University of Hong Kong, Shek Tong Tsui, Hong Kong and also with the Department of Statistics, Yunnan University, Kunming 650091, China (e-mail:jhzhao.ynu@gmail.com).

P. L. H. Yu is with the Department of Statistics and Actuarial Science, The University of Hong Kong, Shek Tong Tsui, Hong Kong (e-mail: plhyu@hku.hk).

Digital Object Identifier 10.1109/TNN.2008.2003467

a quasi-Newton–Raphson algorithm recommended in [10] has been found empirically to converge faster than EM and has become the standard algorithm for fitting FA so far. In Section II-B, we develop a more tempting CM algorithm that consists of a sequence of CM log-likelihood steps, as follows: 1) its convergence is quadratic and monotone [11] (convergence of EM is simply linear) and 2) like EM for FA, it is easy to implement (the method in [10] is not the case).

B. The CM Algorithm

Let $\mathbf{\Psi}^{(t)} = \text{diag}(\psi_1^{(t)}, \psi_2^{(t)}, \ldots, \psi_d^{(t)})$ and $\mathbf{\Psi}_i^{(t)} \triangleq \text{diag}(\psi_1^{(t+1)}, \ldots, \psi_{i-1}^{(t+1)}, \psi_i, \psi_{i+1}^{(t)}, \ldots, \psi_d^{(t)})$. Given an initial $\mathbf{\Psi}^{(0)}$, the CM algorithm recursively performs the following two steps for $t \geq 0$:

- **CM-step 1:** Given $\mathbf{\Psi}^{(t)}$, maximize $\mathcal{L}$ with respect to (w.r.t.) $\mathbf{A}$.
- **CM-step 2:** Given $\mathbf{A}^{(t+1)}$ and $\mathbf{\Psi}_i^{(t)}$, maximize $\mathcal{L}$ w.r.t. ψ_i, for $i = 1, 2, \ldots, d$.

Due to space limitations, here we simply give a sketch of the above two CM-steps. Further technical details can be found in our work [7].

1) The Maximization in CM-Step 1: Given $\mathbf{\Psi}^{(t)}$, premultiply by $[\mathbf{\Psi}^{(t)}]^{-1/2}$ on both sides of (1). Define $\widetilde{\mathbf{A}} \triangleq [\mathbf{\Psi}^{(t)}]^{-1/2}\mathbf{A}$ and $\mathbf{z}_n \triangleq [\mathbf{\Psi}^{(t)}]^{-1/2}(\mathbf{x}_n - \bar{\mathbf{x}})$. Then, the FA model (1) becomes

$$\begin{cases} \mathbf{z}_n = \widetilde{\mathbf{A}}\mathbf{y}_n + \boldsymbol{\epsilon}_n \\ \mathbf{y}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \qquad \boldsymbol{\epsilon}_n \sim \mathcal{N}(\mathbf{0}, \mathbf{I}). \end{cases} \tag{3}$$

Model (3) appears very similar to the probabilistic PCA (PPCA) model [4], since both are *isotropic* noise models. The only difference is that isotropic noise variance in model (3) is known and equal to 1. Define $\widetilde{\mathbf{S}} \triangleq [\mathbf{\Psi}^{(t)}]^{-1/2}\mathbf{S}[\mathbf{\Psi}^{(t)}]^{-1/2}$. Let $(\lambda_i, \mathbf{u}_i)$ denote the eigenvalue–eigenvector pairs of $\widetilde{\mathbf{S}}$ so that $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_d$. It can be shown that the closed-form expression of $\widetilde{\mathbf{A}}$, which globally maximizes $\mathcal{L}(\mathbf{A}, \mathbf{\Psi}^{(t)})$, is given by

$$\widetilde{\mathbf{A}} = \mathbf{U}_{q'}\left(\mathbf{\Lambda}_{q'} - \mathbf{I}\right)^{1/2}\mathbf{R}. \tag{4}$$

Here, q' is defined as follows. If $\lambda_q > 1$, then $q' = q$; if $\lambda_q \leq 1$, q' is the unique integer satisfying $\lambda_{q'} > 1 \geq \lambda_{q'+1}$. $\mathbf{\Lambda}_{q'} = \text{diag}(\lambda_1, \lambda_2, \ldots, \lambda_{q'})$, $\mathbf{U}_{q'} = (\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_{q'})$, and $\mathbf{R}$ can be arbitrarily selected except for the requirement of $\mathbf{R}\mathbf{R}^T = \mathbf{I}$. If desired, $\mathbf{A}^{(t+1)}$ can be obtained by setting $\mathbf{A}^{(t+1)} = [\mathbf{\Psi}^{(t)}]^{1/2}\widetilde{\mathbf{A}}$.

2) The Maximization in CM-Step 2: Because $\mathcal{L}(\mathbf{A}^{(t+1)}, \mathbf{\Psi}_i^{(t)})$ is a function of ψ_i, it is written as $\mathcal{L}(\psi_i)$ for simplicity. From the model assumption that $\mathbf{\Psi}$ is positive, we can pick an arbitrarily very small number $\eta > 0$ and assume $\psi_i \geq \eta$. Let $\mathbf{e}_i$ be the ith column of the $d \times d$ identity matrix and

$$\mathbf{B}_i = \sum_{k=1}^{i-1} \omega_k^{(t+1)} \mathbf{e}_k \mathbf{e}_k' + \mathbf{I} + \widetilde{\mathbf{A}}\widetilde{\mathbf{A}}^T. \tag{5}$$

Let $\mathbf{b}_k$ be the kth column vector of $\mathbf{B}_i^{-1}$ and let b_{kk} stand for the kkth element of $\mathbf{B}_i^{-1}$. Then, the closed-form expression of ψ_i is given by (see [7] for the proof)

$$\psi_i^{(t+1)} = \max\left\{\left[b_{ii}^{-2}\left(\mathbf{b}_i^T\widetilde{\mathbf{S}}\mathbf{b}_i - b_{ii}\right) + 1\right]\psi_i^{(t)}, \eta\right\} \tag{6}$$

and the required $\omega_i^{(t+1)}$ in (5) is

$$\omega_i^{(t+1)} = \psi_i^{(t+1)}(\psi_i^{(t)})^{-1} - 1. \tag{7}$$

Note (6) can always be computed as from (6) $\psi_k^{(t+1)} \geq \eta$ and from (7) $\omega_k^{(t+1)} > -1$, $k = 1, \ldots, i-1$, and thus $\mathbf{B}_i^{-1}$ (see (5)) can be performed. Because it has been shown in [7] that $\mathcal{L}(\psi_i)$ is unimodal in the interval $\psi_i \geq \eta$ and it reaches its maximal point at $\psi_i^{(t+1)}$, using the general property of (E)CM proved in [6], the above CM is guaranteed to converge a stationary point of $\mathcal{L}$. $\mathbf{B}_i^{-1}$, $i = 1, \ldots, d$ can be recursively computed as detailed in Appendix I.

III. MFA Model and an ECM Algorithm

A. The MFA Model

The MFA model is a mixture of M FA submodels with mixture proportions α_j's. In detail, first pick a component label j according to the multinomial distribution $p(j) = \alpha_j, j = 1, \ldots, M$ with the constraint $\sum_{j=1}^M \alpha_j = 1$. Then, given label j, generate $\mathbf{x}_n$ from a q_j-factor FA model with parameter $\boldsymbol{\theta}_j = (\boldsymbol{\mu}_j, \mathbf{A}_j, \mathbf{\Psi}_j)$

$$\begin{cases} \mathbf{x}_n | j = \mathbf{A}_j\mathbf{y}_{nj} + \boldsymbol{\mu}_j + \boldsymbol{\epsilon}_{nj} \\ \mathbf{y}_{nj} \sim \mathcal{N}(\mathbf{0}, \mathbf{I}), \qquad \boldsymbol{\epsilon}_{nj} \sim \mathcal{N}(\mathbf{0}, \mathbf{\Psi}_j) \end{cases} \tag{8}$$

where $\boldsymbol{\mu}_j$ is a d-dimensional mean vector; $\mathbf{A}_j$ is a $d \times q_j$ factor loading matrix; $\mathbf{y}_{nj}$ is a q_j-dimensional factor vector, and $\mathbf{\Psi}_j$ is a positive diagonal matrix.

Before we propose our ECM in Section III-D, for completeness, we briefly review the EM [1] in Section III-B and the AECM [2] in Section III-C and give an insight to their speed of convergence in Section III-E.

B. The EM Algorithm

Let $\mathbf{Y} = \{\mathbf{y}_n\}_{n=1}^N$ and $\mathbf{Z} = \{\mathbf{z}_n\}_{n=1}^N$, where $\mathbf{z}_n = (z_{n1}, \ldots, z_{nM})$ and z_{nj} is an indicator variable taking the value 1 if $\mathbf{x}_n$ comes from the jth component, and 0 otherwise. In the EM for MFA, latent label vectors and latent factor vectors are treated as missing data and the augmented complete data is $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$. The corresponding complete data log likelihood is

$$\mathcal{L}_1(\mathbf{X}, \mathbf{Y}, \mathbf{Z}|\boldsymbol{\theta}) = \sum_{n=1}^N \sum_{j=1}^M z_{nj} \ln\{\alpha_j p(\mathbf{x}_n, \mathbf{y}_n|\boldsymbol{\theta}_j)\}$$

where $\boldsymbol{\theta} = (\alpha_j, \boldsymbol{\theta}_j; j = 1, \ldots, M)$. The EM performs an E-step followed by a M-step.

- **E-step:** Compute the expected $\mathcal{L}_1$ given $\mathbf{X}$ and $\boldsymbol{\theta}^{(t)}$

$$Q_1(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = \mathbb{E}(\mathcal{L}_1|\mathbf{X}, \boldsymbol{\theta}^{(t)}). \tag{9}$$

- **M-step:** Maximize Q_1 w.r.t. $\boldsymbol{\theta}$.

C. The AECM Algorithm

Unlike the EM, the AECM for MFA consists of two cycles: cycle 1 followed by cycle 2, each of which has its own E-step and CM-step. Its salient feature is that the augmented complete data is allowed to vary between E-steps. Specifically, the complete data in cycle 1 is $(\mathbf{X}, \mathbf{Z})$ and its log-likelihood is

$$\mathcal{L}_2(\mathbf{X}, \mathbf{Z}|\boldsymbol{\theta}) = \sum\nolimits_{n=1}^N \sum\nolimits_{j=1}^M z_{nj} \ln\{\alpha_j p(\mathbf{x}_n|\boldsymbol{\theta}_j)\}.$$

- **E-step of cycle 1:** Compute the expected $\mathcal{L}_2$ given $\mathbf{X}$ and $\boldsymbol{\theta}^{(t)}$

$$Q_2(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t)}) = \sum\nolimits_{j=1}^M \text{II}_j\left(\boldsymbol{\theta}_j|\boldsymbol{\theta}^{(t)}\right) \tag{10}$$

where

$$\text{II}_j\left(\boldsymbol{\theta}_j|\boldsymbol{\theta}^{(t)}\right) = \sum\nolimits_{n=1}^N R_{nj}(\boldsymbol{\theta}^{(t)}) \ln\{\alpha_j p(\mathbf{x}_n|\boldsymbol{\theta}_j)\} \tag{11}$$

depending only on $(\alpha_j, \boldsymbol{\theta}_j)$ of component j.

Here, $R_{nj}(\boldsymbol{\theta}^{(t)})$ is the posterior possibility of data point $\mathbf{x}_n$ belonging to component j given $\boldsymbol{\theta}^{(t)}$

$$R_{nj}(\boldsymbol{\theta}^{(t)}) \triangleq \frac{\alpha_j^{(t)} p\left(\mathbf{x}_n|\boldsymbol{\theta}_j^{(t)}\right)}{\sum_{k=1}^{M} \alpha_k^{(t)} p\left(\mathbf{x}_n|\boldsymbol{\theta}_k^{(t)}\right)}. \tag{12}$$

- **CM-step of cycle 1:** Maximize Q_2 w.r.t. α_j's and $\boldsymbol{\mu}_j$'s given $(\mathbf{A}_j^{(t)}, \boldsymbol{\Psi}_j^{(t)})$'s under the restriction $\sum_{i=1}^{M} \alpha_j = 1$, which is easy and similar to that in conventional Gaussian mixture model, leading to the following updating equations:

$$\begin{aligned} \alpha_j^{(t+1)} &= \frac{1}{N}\sum\nolimits_{n=1}^{N} R_{nj}(\boldsymbol{\theta}^{(t)}), \\ \boldsymbol{\mu}_j^{(t+1)} &= \frac{1}{N\alpha_j^{(t+1)}}\sum\nolimits_{n=1}^{N} R_{nj}(\boldsymbol{\theta}^{(t)})\mathbf{x}_n. \end{aligned} \tag{13}$$

  In cycle 2, the complete data is $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$ and its log-likelihood is $\mathcal{L}_1$, which is the same as that in the EM given in Section III-B.
- **E-step of cycle 2:** Compute the expected $\mathcal{L}_1$ given $\mathbf{X}$ and $(\alpha_j^{(t+1)}, \boldsymbol{\mu}_j^{(t+1)}, \mathbf{A}_j^{(t)}, \boldsymbol{\Psi}_j^{(t)})$'s.
- **CM-step of cycle 2:** Maximize Q_1 w.r.t. $\mathbf{A}_j$'s and $\boldsymbol{\Psi}_j$'s.

Compared with the EM, the AECM utilizes less data augmentation $(\mathbf{X}, \mathbf{Z})$ to update $\boldsymbol{\mu}_j$'s, but the step to update $\mathbf{A}_j$'s and $\boldsymbol{\Psi}_j$'s still requires larger one $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$ as in the EM.

D. The ECM Algorithm

Rather than turning to larger data augmentation $(\mathbf{X}, \mathbf{Y}, \mathbf{Z})$ as in the AECM, our ECM persists in utilizing the smaller one $(\mathbf{X}, \mathbf{Z})$ even if in the step to update $\mathbf{A}_j$'s and $\boldsymbol{\Psi}_j$'s as we find the maximization of Q_2 w.r.t. $\mathbf{A}_j$'s or $\boldsymbol{\Psi}_j$'s can be solved analytically. In detail, the ECM performs an E-step followed by three successive CM-steps.

- **E-step:** The same as the E-step of cycle 1 in AECM.
- **CM-step 1:** The same as the CM-step of cycle 1 in AECM.
- **CM-step 2:** Maximize Q_2 w.r.t. $\mathbf{A}_j$'s given $(\alpha_j^{(t+1)}, \boldsymbol{\mu}_j^{(t+1)}, \boldsymbol{\Psi}_j^{(t)})$'s.
- **CM-step 3:** Maximize Q_2 w.r.t. $\boldsymbol{\Psi}_j$'s given $(\alpha_j^{(t+1)}, \boldsymbol{\mu}_j^{(t+1)}, \mathbf{A}_j^{(t+1)})$'s.

Obviously, maximizing Q_2 w.r.t. $\mathbf{A}_j$ or $\boldsymbol{\Psi}_j$ equals to maximizing $\amalg_j$ w.r.t. $\mathbf{A}_j$ or $\boldsymbol{\Psi}_j$. Corresponding to the covariance matrix $\mathbf{S}$ in FA, we define the *local* covariance matrix for MFA

$$\mathbf{S}_j \triangleq \frac{1}{N\alpha_j^{(t+1)}}\sum_{n=1}^{N} R_{nj}(\boldsymbol{\theta}^{(t)})\left(\mathbf{x}_n - \boldsymbol{\mu}_j^{(t+1)}\right)\left(\mathbf{x}_n - \boldsymbol{\mu}_j^{(t+1)}\right)^T.$$

The expected log-likelihood of jth submodel $\amalg_j$ up to a constant is now given by

$$\amalg_j(\mathbf{A}_j, \boldsymbol{\Psi}_j) = -\frac{N\alpha_j^{(t+1)}}{2}\left\{\ln|\boldsymbol{\Sigma}_j| + \mathrm{tr}\left(\boldsymbol{\Sigma}_j^{-1}\mathbf{S}_j\right)\right\} \tag{14}$$

where $\boldsymbol{\Sigma}_j = \mathbf{A}_j\mathbf{A}_j + \boldsymbol{\Psi}_j$. Clearly, (14) is similar to (2), and therefore, the procedure for single FA model developed in Section II-B can be directly used here. Define $\widetilde{\mathbf{S}}_j \triangleq \boldsymbol{\Psi}_j^{(t)^{-1/2}}\mathbf{S}_j\boldsymbol{\Psi}_j^{(t)^{-1/2}}$. Let $(\lambda_{ji}, \mathbf{u}_{ji})$ be its eigenvalue–eigenvector pair in the order $\lambda_{j1} \geq \lambda_{j2} \geq \ldots \geq \lambda_{jd}$. Then, the solution for CM-step 2 is given by

$$\widetilde{\mathbf{A}}_j^{(t+1)} = \mathbf{U}_j\left(\boldsymbol{\Lambda}_j - \mathbf{I}\right)^{1/2}.$$

Here, $q_j' = \sum_{i=1}^{q}\mathcal{I}(\lambda_{ji} > 1)$ ($\mathcal{I}(\cdot)$ is an indicator function), $\mathbf{U}_j = (\mathbf{u}_{j1}, \mathbf{u}_{j2}, \ldots, \mathbf{u}_{jq_j'})$, and $\boldsymbol{\Lambda}_j = \mathrm{diag}(\lambda_{j1}, \lambda_{j2}, \ldots, \lambda_{jq_j'})$.

For CM-step 3, define

$$\mathbf{B}_i = \sum\nolimits_{k=1}^{i-1}\omega_{jk}^{(t+1)}\mathbf{e}_k\mathbf{e}_k^T + \mathbf{I} + \widetilde{\mathbf{A}}_j^{(t+1)}\left[\widetilde{\mathbf{A}}_j^{(t+1)}\right]^T.$$

Then, we have

$$\begin{aligned} \omega_{ji}^{(t+1)} &= \max\left[b_{ii}^{-2}\left(\mathbf{b}_i^T\widetilde{\mathbf{S}}_j\mathbf{b}_i - b_{ii}\right), \eta(\psi_i^{(t)})^{-1} - 1\right] \\ \psi_{ji}^{(t+1)} &= \left(\omega_{ji}^{(t+1)} + 1\right)\psi_{ji}^{(t)}. \end{aligned} \tag{15}$$

E. On Speed of Convergence

In this section, we compare the speed of the EM with our proposed ECM. The relationship with the AECM can be obtained similarly. Let I_{obs} be the observed data information matrix. Denote $\mathrm{com1} \triangleq (\mathbf{X}, \mathbf{Y}, \mathbf{Z})$ and $\mathrm{com2} \triangleq (\mathbf{X}, \mathbf{Z})$ and their corresponding complete data information matrices I_{com1} and I_{com2}. These information matrices are closely related to the speeds of convergence or, more exactly, matrix speeds of convergence of EM, AECM, and ECM. Meng and Rubin [12] show that the speed of the EM is given by

$$\text{speed of EM} = I_{\mathrm{obs}}I_{\mathrm{com1}}^{-1} \tag{16}$$

"the fractions of observed information" in data augmentation $\mathrm{com1}$. Let T_{com2} be the upper block part of I_{com2} with respect to the partition of parameters used in three CM steps, i.e., [$(\alpha_j, \boldsymbol{\mu}_j)$'s, $\mathbf{A}_j$'s, $\boldsymbol{\Psi}_j$'s]. From [11, eq. (15)], we have

$$\text{speed of ECM} = I_{\mathrm{obs}}T_{\mathrm{com2}}^{-1} \tag{17}$$

which is called "the sequential fractions of observed information" [11] in data augmentation $\mathrm{com2}$. By (16) and (17), we have

$$\text{speed of ECM} = \text{speed of EM} \times T_{\mathrm{com2}}I_{\mathrm{com1}}^{-1} \tag{18}$$

where $T_{\mathrm{com2}}I_{\mathrm{com1}}^{-1}$ can be called "the sequential fractions of relatively observed information" in data augmentation $\mathrm{com1}$ because $\mathrm{com2}$ is nested in $\mathrm{com1}$. Therefore, the ECM for MFA converges typically faster than the EM.

IV. Experiments

The theoretical analysis in Section III-E gives us some insight into why the ECM has faster convergence. However, the analysis does not tell us to what extent the ECM will be faster than the EM. In addition, because the ECM requires more computation (see Table IV), we are also interested in seeing whether its performance in central processing unit (CPU) time deserves its higher computation cost. We hence empirically examine the performance of EM, AECM ,and ECM in terms of CPU time and number of iterations using synthetic and real data. All computations are carried out by Matlab. For EM, we use the code,[1] which implements the algorithm described in [1]. Unless otherwise stated, we use the following in the experiments.

- *Initialization:* Perform k-means followed by deciding $\boldsymbol{\theta}^{(0)}$ by PCA, which is suggested in [2] for AECM.
- *Convergence criterion:* Stop algorithms if $t > K_{\max}$ or $|1 - \mathcal{L}^{(t)}/\mathcal{L}^{(t+1)}| < \mathrm{tol}_1$ with $\mathrm{tol}_1 = 10^{-8}$ and the maximal number of iterations: $K_{\max} = 5000$.
- *Constraint:* $\eta = 0.005$.

[1]http://lear.inrialpes.fr/~verbeek/code/MoFA.tar.gz

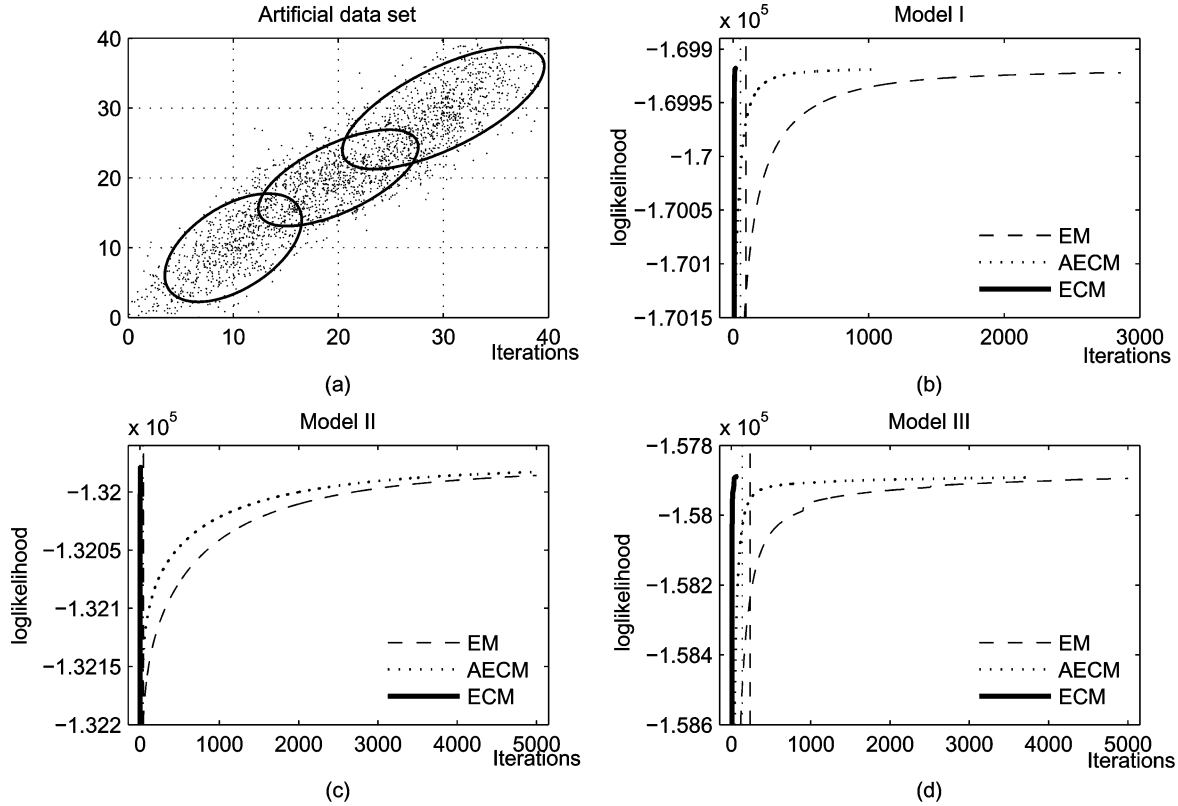

Fig. 1. (a) Artificial data. (b)–(d) Typical evolvements of log-likelihood $\mathcal{L}$ by EM (dashed line), AECM (dotted line), and ECM (solid line) when the data set is fitted by MFA models with different M and q. The vertical line with the corresponding line type signals the number of iterations that first meets $\mathrm{tol}_2 = 10^{-5}$.

TABLE I
AVERAGED CPU TIME SHOWN IN SECONDS AND NUMBER OF ITERATIONS FOR CONVERGENCE

Model	Time			Number of iterations		
	EM	AECM	ECM	EM	AECM	ECM
Model I	27.4	14.4	0.8	2853	1056	25
Model II	133.1	143.3	0.3	5000	5000	10
Model III	143.5	120.6	6.5	5000	3122	77

TABLE II
AVERAGED REQUIRED CPU TIME (IN SECONDS) AND NUMBER OF ITERATIONS

Model	Time			Number of iterations		
	EM	AECM	ECM	EM	AECM	ECM
Model I	287	357	9	5000	5000	44
Model II	607	707	51	5000	5000	125

TABLE III
AVERAGED MSE [MEAN (STD.)] AND LOG-LIKELIHOOD ($\mathcal{L}$) FROM TEN RUNS. CONVERGENCE CRITERION IS DENOTED BY CC ($\mathrm{tol}_2 = 10^{-5}\mathrm{tol}_1 = 10^{-8}$)

Model	CC		EM	AECM	ECM
Model I	tol_2	MSE	87.3 (0.7)	85.5 (0.4)	82.1 (1.1)
		$\mathcal{L}(\times 10^5)$	-8.412	-8.360	-8.314
	tol_1	MSE	86.6 (0.2)	84.5 (0.8)	82.2 (1.0)
		$\mathcal{L}(\times 10^5)$	-8.338	-8.316	-8.313
Model II	tol_2	MSE	75.3 (2.0)	73.8 (2.1)	68.6 (1.7)
		$\mathcal{L}(\times 10^5)$	-8.246	-8.230	-8.169
	tol_1	MSE	73.5 (1.5)	72.3 (0.9)	68.6 (1.8)
		$\mathcal{L}(\times 10^5)$	-8.186	-8.170	-8.165

A. Artificial Data

An artificial data set of $N = 2400$ points in $\mathbb{R}^d (d = 30)$ is generated from an MFA model with $M = 3, q_j = 8, j = 1, \ldots, M$, and

$$\boldsymbol{\mu}_j = 10j\mathbf{1}_{d\times 1} \quad \mathbf{A}_j = \mathrm{Unif}(d, q_j) \quad \boldsymbol{\Psi}_j = \mathrm{diag}\{\mathrm{Unif}(1, d)\}$$
$$\text{except that } \psi_{jl} = 100\mathrm{Unif}(1,1), \qquad l = 2, 12, 22, \qquad \pi_j = \frac{1}{3}.$$

Here, $\mathrm{Unif}(a, b)$ is an $a \times b$ random number matrix drawn from a uniform distribution on the unit interval. The first 2-D view of the data is shown in Fig. 1(a). We fit the data set using different MFA models: Model I ($M = 2, q = 3$), Model II ($M = 3, q = 8$), and Model III ($M = 6, q = 3$). Fig. 1(b)–(d) plots the typical evolution of log-likelihood $\mathcal{L}$. All algorithms are run ten times using different initializations from k-means but the same initialization is used for all algorithms in a single run. Table I shows the averaged CPU time and number of iterations. When the *convergence criterion* is met, ECM can generally obtain higher final $\mathcal{L}$ than EM and AECM.

B. Real Data

As a dimensionality reduction tool, an MFA model can be used to perform image compression. In this brief, we focus on comparing the performance of an MFA model trained by EM, AECM, and ECM, rather than the performance between an MFA model and other competing models such as PCA and MPCA for this task. We, in general, follow the compression algorithm detailed in [13] to conduct an image compression experiment, using EM, AECM, and ECM respectively, but in our implementation, the component label of data point $\mathbf{x}$ is decided by maximizing posterior probability $p(j|\mathbf{x})$ instead of minimizing reconstruction error.

A 512×512 gray image "Lena" is subdivided into $N(= 4096)$ nonoverlapping blocks of 8×8 pixels and each block is taken as a $64(= 8 \times 8)$-dimensional random vector $\mathbf{x}$. Then, the obtained data is fitted by different common $q(= 4)$ MFA models: Model I ($M = 4$) and Model II ($M = 8$). Fig. 2(a) and (b) plots the typical convergence curve of $\mathcal{L}$ for Models I and II. Table II lists the averaged required CPU time and number of iterations for convergence. When the *convergence*

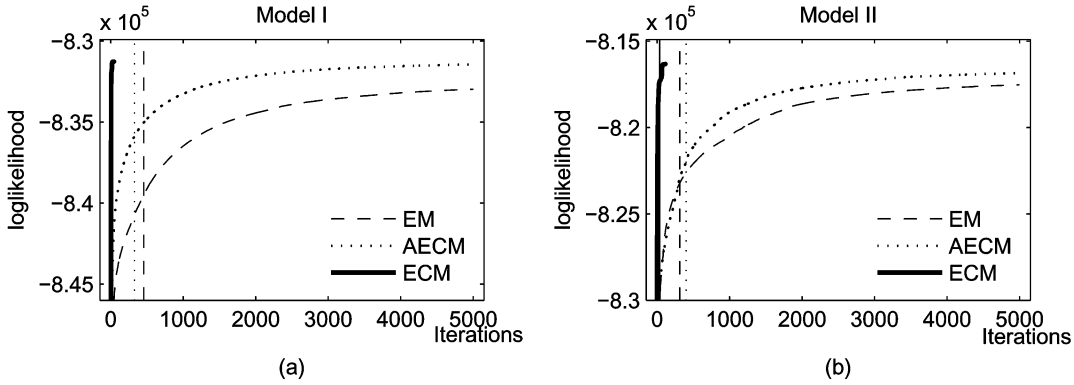

Fig. 2. Typical evolvements of log-likelihood $\mathcal{L}$ by EM (dashed line), AECM (dotted line), and ECM (solid line). The vertical line with the corresponding line type signals the number of iterations that first meets $\mathrm{tol}_2 = 10^{-5}$.

TABLE IV
COMPUTATIONAL COST OF DIFFERENT ALGORITHMS FOR EACH COMPONENT

Algorithm	E-step	$\boldsymbol{\mu}_j$ and $\mathbf{A}_j$	$\boldsymbol{\Psi}_j$
EM	$O(Nd[q+1])$	$O(Nd[q+1])$ + $O(d[q+1]^2)$	$O(Nd)+O(dq)$
AECM	$O(Nd[q+2])$	$O(Nd[q+2])$ + $O(dq^2)$	$O(2Nd)+O(dq)$
ECM	$O(Nd[q+1])$	$O(Nd^2)+O(d^3)$	$O(d^3)$

criterion is satisfied, ECM can usually obtain higher final $\mathcal{L}$ than EM and AECM.

To measure estimators from EM, AECM, and ECM, besides log-likelihood, we calculate mean squared error (MSE)[2] of the reconstruction image. Table III shows the averaged MSE and negative log-likelihood from ten runs for tol_1, and particularly, for $\mathrm{tol}_2 = 10^{-5}$. Because the slow convergence suffered by EM and AECM (see Figs. 1 and 2), one usually chooses to stop the algorithms early or to use a loose convergence criterion. The use of tol_2 here aims to check what cost it may bring if we do so. It can be observed from Table III that: 1) both EM and AECM are worse than ECM in terms of averaged MSE and log-likelihood whether tol_1 or tol_2 is used and 2) EM and AECM in tol_2 have larger MSE and lower log-likelihood than those in tol_1. Unlike EM and AECM, the MSEs and log-likelihood values of ECM in tol_1 and tol_2 are almost the same. These observations imply that ECM provides a much faster and more accurate estimator than EM and AECM.

V. CONCLUDING REMARKS

For fitting MFA, we have proposed an ECM algorithm, which, unlike existing EM and AECM, treats the component–indicator vectors as missing data only. In our empirical studies, we focus on the case that sample size N is large relative to dimension d. In this case, our ECM is computationally much more efficient than EM and AECM and thus our proposed ECM is very useful for ML estimation of MFA. It would be interesting to investigate more carefully why our ECM performs so much better than the other EMs. Notice that when N is small relative to d and the number of factors is extremely small (e.g., face recognition), our proposed ECM will be costly both in time and space and EM and AECM could become computationally more efficient, in particular, when there is no rigorous requirement on the accuracy of estimators.

[2]$\mathrm{MSE} = \left(\sum_{i=1}^{512}\sum_{j=1}^{512}(x_{ij}-\hat{x}_{ij})^2\right)/512^2$

APPENDIX I
RECURSIVE COMPUTATION OF THE MATRIX $\mathbf{B}_i^{-1}$

Proposition 1: Suppose $\mathbf{C} \succ 0$, ω is a real number such that $1+\omega\mathbf{e}_i'\mathbf{C}^{-1}\mathbf{e}_i \neq 0$. Then

$$\left(\omega\mathbf{e}_i\mathbf{e}_i' + \mathbf{C}\right)^{-1} = \mathbf{C}^{-1} - \frac{\omega\mathbf{C}^{-1}\mathbf{e}_i\mathbf{e}_i'\mathbf{C}^{-1}}{1+\omega\mathbf{e}_i'\mathbf{C}^{-1}\mathbf{e}_i}. \tag{19}$$

Proposition 1 can be viewed as a special case of a generalized Woodbury's formula [14, p. 90]. From (5), we have $\mathbf{B}_1 = \mathbf{I} + \widetilde{\mathbf{A}}^{(t+1)}[\widetilde{\mathbf{A}}^{(t+1)}]'$ and the following recursive relation:

$$\mathbf{B}_{i+1} = \omega_i^{(t+1)}\mathbf{e}_i\mathbf{e}_i' + \mathbf{B}_i, \qquad i = 1, 2, \ldots, d-1. \tag{20}$$

Because $\mathbf{B}_1 \succ 0$ and $\omega_i^{(t+1)} > -1$ as detailed in Section II-B, from (20), we have $\mathbf{B}_i \succ 0, i = 1, \ldots, d$. Furthermore, it can be shown in [7] that $\omega_i^{(t+1)} > -1$ guarantees $1+\omega_i^{(t+1)}\mathbf{e}_i'\mathbf{B}_i^{-1}\mathbf{e}_i > 0$. We hence can use Proposition 1 and obtain

$$\mathbf{B}_{i+1}^{-1} = \mathbf{B}_i^{-1} - \omega_i^{(t+1)}\mathbf{b}_i\mathbf{b}_i^T\left(1+\omega_i^{(t+1)}b_{ii}\right)^{-1}. \tag{21}$$

Thus, $\mathbf{B}_i^{-1}$ can be recursively computed. The first one is computed by using (4)

$$\mathbf{B}_i^{-1} = \mathbf{U}_{q'}\left(\mathbf{\Lambda}_{q'}^{-1} - \mathbf{I}\right)\mathbf{U}_{q'}' + \mathbf{I}. \tag{22}$$

Note that (4) and (22) simply require at most the first q eigenvalues and eigenvectors of $\widetilde{\mathbf{S}}$.

APPENDIX II
COMPUTATIONAL COMPLEXITY

We analyze the cost of ECM for each component j in each iteration. To obtain posterior probability R_{nj} in E-step, it is required from (12) to evaluate probability density $p(x_n|\boldsymbol{\theta}_j)$, cost of which is $O(d(q+1))$. Cost of E-step is, therefore, $O(Nd(q+1))$. For CM-step 1, from (13), the cost is $O(Nd)$. For CM-step 2, it is required to calculate *local* covariance matrix $\mathbf{S}_j$, whose cost is $O(Nd^2)$, and its eigendecomposition, cost of which is $O(d^3)$. For CM-step 3, as $\mathbf{S}_j$ has been calculated, the main computation in each step of d substeps is in (21) and (15), which is $O(d^2)$, and hence, the total cost of d steps is $O(d^3)$. For comparison, Table IV summarizes the computational cost for EM, AECM, and ECM.

ACKNOWLEDGMENT

The authors would like to thank the reviewers for their valuable comments and suggestions.

REFERENCES

[1] Z. Ghahramani and G. E. Hinton, "The EM algorithm for mixtures of factor analyzers," Dept. Comput. Sci., Univ. Toronto, Toronto, ON, Canada, 1996, Tech. Rep. CRG-TR-96-1.

[2] G. J. McLachlan, D. Peel, and R. W. Bean, "Modelling high-dimensional data by mixtures of factor analyzers," *Comput. Statist. Data Anal.*, vol. 41, pp. 379–388, Jan. 2003.

[3] G. McLachlan, R. Bean, and L. B.-T. Jones, "Extension of the mixture of factor analyzers model to incorporate the multivariate t distribution," *Comput. Statist. Data Anal.*, vol. 51, pp. 5327–5338, 2007.

[4] M. E. Tipping and C. M. Bishop, "Mixtures of probabilistic principal component analyzers," *Neural Comput.*, vol. 11, pp. 443–482, 1999.

[5] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data using the EM algorithm (with discussion)," *J. Roy. Statist. Soc. B*, vol. 39, no. 1, pp. 1–38, 1977.

[6] X. L. Meng and D. A. van Dyk, "The EM algorithm—An old folk-song sung to a fast new tune," *J. Roy. Statist. Soc. B*, vol. 59, no. 3, pp. 511–567, 1997.

[7] J. Zhao, P. L. H. Yu, and Q. Jiang, "ML Estimation for factor analysis: EM or non-EM?," *Statist. Comput.*, vol. 18, no. 2, pp. 109–123, 2008.

[8] D. B. Rubin and T. T. Thayer, "EM Algorithms for factor ML analysis," *Psychometrika*, vol. 47, pp. 69–76, 1982.

[9] K. B. Petersen, O. Winther, and L. K. Hansen, "On the slow convergence of EM and VBEM in low-noise linear models," *Neural Comput.*, vol. 17, no. 9, pp. 1921–1926, 2005.

[10] K. G. Jöreskog, "Some contributions to maximum likelihood factor analysis," *Psychometrika*, vol. 32, no. 4, pp. 433–482, 1967.

[11] C. Liu, "The ECME algorithm: A simple extension of EM and ECM with faster monotone convergence," *Biometrika*, vol. 81, pp. 633–648, 1994.

[12] X.-L. Meng and D. B. Rubin, "Maximum likelihood estimation via the ECM algorithm: A general framework," *Biometrika*, vol. 80, no. 2, pp. 267–278, 1993.

[13] N. Ueda, R. Nakano, Z. Ghahramani, and G. E. Hinton, "SMEM algorithm for mixture models," *Neural Comput.*, vol. 12, no. 9, pp. 2109–2128, Sep. 2000.

[14] K. Lange, *Numerical Analysis for Statisticians*. New York: Springer-Verlag, 1999.

A-Optimality Orthogonal Forward Regression Algorithm Using Branch and Bound

Xia Hong, Sheng Chen, and Chris J. Harris

***Abstract*—In this brief, we propose an orthogonal forward regression (OFR) algorithm based on the principles of the branch and bound (BB) and A-optimality experimental design. At each forward regression step, each candidate from a pool of candidate regressors, referred to as $\mathcal{S}$, is evaluated in turn with three possible decisions: 1) one of these is selected and included into the model; 2) some of these remain in $\mathcal{S}$ for evaluation in the next forward regression step; and 3) the rest are permanently eliminated from $\mathcal{S}$. Based on the BB principle in combination with an A-optimality composite cost function for model structure determination, a simple adaptive diagnostics test is proposed to determine the decision boundary between 2) and 3). As such the proposed algorithm can significantly reduce the computational cost in the A-optimality OFR algorithm. Numerical examples are used to demonstrate the effectiveness of the proposed algorithm.**

***Index Terms*—Branch and bound (BB), experimental design, forward regression, structure identification.**

I. INTRODUCTION

A large class of nonlinear models and neural networks can be classified as a linear-in-the-parameters model [1], [2]. The linear-in-the-parameters models are well structured for adaptive learning, have provable learning and convergence conditions, have the capability of parallel processing, and have clear applications in many engineering applications [3]–[5]. A basic principle in practical nonlinear data modeling is the parsimonious principle that ensures the smallest possible model for the explanation of the observational data. For the linear-in-the-parameters models, the forward orthogonal least squares (OLS) algorithm efficiently constructs parsimonious models [6], [7], and has been a popular tool in associative neural networks such as fuzzy/neurofuzzy systems [8], [9] and wavelet neural networks [10], [11]. The algorithm has also been utilized in a wide range of engineering applications, e.g., aircraft gas turbine modeling [12], fuzzy control of multiple-input–multiple-output (MIMO) nonlinear systems [13], power system control [14], and fault detection [15].

In optimum experimental design [16], the model adequacy is evaluated by design criteria that are statistical measures of goodness of experimental designs by virtue of design efficiency and experimental effort. Quantitatively, model adequacy is measured as function of the eigenvalues of the design matrix. In order to produce a model with good generalization capabilities, the A-optimality composite cost function has been used as the model selection criterion in the A-optimality-based orthogonal forward regression (OFR) algorithms [17].

Note that the nonlinear system identification is an intractable optimization problem of mixed integer programming that involves both continuous variables, e.g., model parameters and discrete variables, e.g., enumeration of possible model terms. The principle of branch-

Manuscript received March 31, 2008; revised June 23, 2008; accepted July 25, 2008. First published September 30, 2008; current version published November 5, 2008.

X. Hong is with the School of Systems Engineering, University of Reading, Reading RG6 61Y, U.K. (e-mail: x.hong@reading.ac.uk).

S. Chen and C. J. Harris are with the School of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, U.K. (e-mail: sqc@ecs.soton.ac.uk; cjh@ecs.soton.ac.uk).

Color versions of one or more of the figures in this brief are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TNN.2008.2003251