SPACE ALTERNATING DATA AUGMENTATION: APPLICATION TO FINITE MIXTURE OF GAUSSIANS AND SPEAKER RECOGNITION

Arnaud Doucet[†] - Stéphane Sénécal[‡] - Tomoko Matsui[‡]
†Cambridge University Engineering Department, Cambridge, UK.
‡Institute of Statistical Mathematics, Tokyo, Japan.
Email: ad2@eng.cam.ac.uk - steph@ism.ac.jp - tmatsui@ism.ac.jp

ABSTRACT

The SAGE (Space-Alternating Generalized Expectation-Maximization) algorithm [2] is one of the most elegant and popular extensions of the EM (Expectation Maximization) algorithm for performing ML (Maximum Likelihood) or MAP (Maximum A Posteriori) parameter estimation. This algorithm updates parameter components by subblocks by alternating missing data spaces. Its efficiency has been reported in numerous simulation studies. We propose here a MCMC (Markov chain Monte Carlo) strategy named SADA (Space-Alternating Data Augmentation) which relies on the same principle in order to sample efficiently from (posterior) distributions and we discuss its application to finite mixtures of Gaussians. For this model, we also present an original implementation of the SAGE algorithm. In Monte Carlo simulations and in an application to speaker recognition, these methods which are straightforward modifications of the standard EM and DA (Data augmentation) algorithms consistently outperform them.

Keywords: Expectation-Maximization algorithm, Finite mixture distributions, Latent variable models, Markov chain Monte Carlo.

1 Introduction

In Bayesian inference, we are often interested in computing expectations with respect to posterior distributions which do not admit any closed-form expression. In these cases, the tools of choice to approximate these expectations are MCMC algorithms; i.e. we simulate an ergodic Markov chain whose stationary distribution corresponds to the target posterior distribution of interest. In practice, we are interested in devising Markov chain transition kernels whose convergence to the stationary distribution is fast. There is no general method available to build such kernels and one has to use specificities of the statistical model under study to obtain efficient algorithms.

In many problems of interest, it is however possible to introduce so-called missing data to facilitate the design of such algorithms. The introduction of such missing data is at the core of the very popular EM algorithm for performing ML/MAP parameter estimation [4]. Similarly, when we are not interested in point estimates but in sampling from the whole posterior distribution, then the introduction of missing data often allows us to develop a simple MCMC algorithm known as DA [7]; DA is the simplest form of the popular Gibbs sampler.

Although these popular methods are elegant and can provide satisfactory results, they can also converge slowly. Loosely speaking, the more informative the missing data introduced, the slower the convergence rate; see for example [2], [4]. There have been numerous attempts to devise more efficient methods; see [4] for a recent survey of the literature. One of the most effective and useful extension of the EM algorithm is known as SAGE, as been proposed by Hero and Fessler [2]. The basic principle of SAGE is to update parameter components by subblocks. A specific missing data space is associated with each subblock such that complete data spaces which are less informative can be used and the convergence rate improved.

To quote [4, pp. 226-227] "It is expected that the approaches of the SAGE (and AECM) algorithms will give rise to a more flexible formulation of the Gibbs sampler... Such work has not yet been done". In this paper, we show that it is possible to adapt the SAGE idea to obtain an efficient MCMC algorithm for sampling from posterior distributions. Similarly to SAGE, we update the parameter components by subblocks and each subblock of parameters is sampled conditional on a specific missing data set. The resulting algorithm is named SADA for Space Alternating Data Augmentation. Wherever SAGE has been used, it should be relatively easy to devise a SADA version if we are interested in sampling from the posterior. We present an application to finite mixtures of multivariate Gaussian distributions. In this case, we develop an original SAGE algorithm and its associated SADA version. These two new algorithms appear as straightforward modifications of the standard EM and DA algorithms. In Monte Carlo simulations on simulated data and in an application to speech recognition, we demonstrate that SAGE and SADA consistently outperform them.

2 EM and SAGE Algorithms

To facilitate the presentation and comparison to DA and SADA, we introduce the EM and SAGE algorithm in the Bayesian framework; i.e. we are interested in obtaining the MAP estimate of the random variable X given a realization of $Y = y$ which satisfies

$$x_{\text{MAP}} = \arg\max \; p\left(x \middle| y\right)$$

where
$$p\left(x|\,y\right) \propto p\left(y|\,x\right) p\left(x\right).$$

We will further assume that X is a random vector whose components can be partitioned into n subsets $X = X_{1:n} = (X_1, \ldots, X_n)$. We use the notation $Z_{i:j} = (Z_i, Z_{i+1}, \ldots, Z_j)$ and $X_{-k} = X_{1:n} \backslash \{X_k\} = (X_1, \ldots, X_{k-1}, X_{k+1}, \ldots, X_n)$.

To maximize $p\left(x|\,y\right)$, the EM algorithm introduces missing data Z with a given conditional distribution $p\left(z|\,y, x\right)$. Based on this missing data, EM proceeds as follows, at iteration i, to maximize $p\left(x|\,y\right)$:

$$x^{(i)} = \underset{x}{\arg\max} \int \log\left(p\left(x, z|\,y\right)\right) p\left(z|\,y, x^{(i-1)}\right) dz.$$

To maximize $p\left(x|\,y\right)$, the SAGE algorithm introduces not one but n missing data sets $Z_{1:n}$ [2]. With each random variable or vector Z_k a conditional distribution $p\left(z_k|\,y, x_{1:n}\right)$ is associated where

$$p\left(y|\,x_{1:n}, z_k\right) = p\left(y|\,x_{-k}, z_k\right).$$

The SAGE algorithm proceeds as follows to maximize the posterior distribution at iteration i. Select an index $k \in \{1, \ldots, n\}$, set $x_k^{(i)}$ as

$$\underset{x}{\arg\max} \int \log\left(p\left(x_{-k}^{(i-1)}, x_k, z \middle|\, y\right)\right) p\left(z_k|\,y, x^{(i-1)}\right) dz_k.$$

and $x_{-k}^{(i)} = x_{-k}^{(i-1)}$. Typically we update the components cyclically; i.e. at iteration i we update the component $k = (i \bmod n) + 1$.

3 DA and SADA Algorithms

In MCMC, the objective is not to maximize $p\left(x|\,y\right)$ but to obtain random samples $\left\{X^{(i)}\right\}$ distributed according to $p\left(x|\,y\right)$ [7]. Based on these samples, it is easy to approximate the MMSE estimate

$$x_{\text{MMSE}} = \int x p\left(x|\,y\right) dx \text{ by } \widehat{x}_{\text{MMSE}} = \frac{1}{N}\sum_{i=1}^{N} X^{(i)}.$$

It is also possible to compute posterior variances, confidence intervals or predictive distributions. Constructing efficient MCMC algorithms to sample from $p\left(x|\,y\right)$ is typically difficult and the introduction of missing data can substantially ease this task.

Similarly to the EM algorithm, the DA algorithm introduces some missing data Z to sample from $p\left(x|\,y\right)$ and we have the joint posterior distribution

$$p\left(x, z|\,y\right) = p\left(x|\,y\right) p\left(z|\,y, x\right)$$

The DA algorithm proceeds as follows at iteration i given $X^{(i-1)}$:

- Sample $Z^{(i)} \sim p\left(\cdot|\,y, X^{(i-1)}\right)$
- Sample $X^{(i)} \sim p\left(\cdot|\,y, Z^{(i)}\right)$.

The transition kernel associated to $\left\{X^{(i)}, Z^{(i)}\right\}$ admits $p\left(x, z|\,y\right)$ as an invariant distribution. Under weak additional assumptions (irreducibility and aperiodicity), it can be shown that the instantaneous distribution of $\left(X^{(i)}, Z^{(i)}\right)$ converges towards $p\left(x, z|\,y\right)$ as i goes to infinity [7].

Similarly to SAGE, the SADA algorithm introduces not one but n missing data sets $Z_{1:n}$. With each random variable Z_k a distribution $p\left(z_k|\,y, x_{1:n}\right)$ is associated and we define the following joint posterior distribution

$$p\left(x_{1:n}, z_{1:n}|\,y\right) = p\left(x_{1:n}|\,y\right) \prod_{k=1}^{n} p\left(z_k|\,y, x_{1:n}\right). \quad (1)$$

Typically, $p\left(y|\,x_{1:n}, z_k\right) = p\left(y|\,x_{-k}, z_k\right)$ and all z_k are independent given y and $x_{1:n}$ but this is not necessarily the case. To sample from (1), the SADA algorithm proceeds at iteration i given $X_{1:n}^{(i-1)}$ with $k = (i \bmod n) + 1$ as follows.

- Sample $Z_k^{(i)} \sim p\left(\cdot|\,y, X^{(i-1)}\right)$
- Sample $X_k^{(i)} \sim p\left(\cdot|\,y, Z_k^{(i)}, X_{-k}^{(i-1)}\right)$.
- Set $X_{-k}^{(i)} = X_{-k}^{(i-1)}$.

To establish the validity of this algorithm, i.e. that it generates a Markov chain $\left\{X_{1:n}^{(i)}, Z_{1:n}^{(i)}\right\}$ with invariant distribution given by (1), it is sufficient to notice that it could be rewritten as:

- Sample $Z_k^{(i)}, Z_{-k} \sim p\left(\cdot|\,y, X_{1:n}^{(i-1)}\right)$
- Sample $X_k^{(i)}, Z_{-k} \sim p\left(\cdot|\,y, Z_k^{(i)}, X_{-k}^{(i-1)}\right)$.
- Set $X_{-k}^{(i)} = X_{-k}^{(i-1)}$.

At each time step, we can think of the previous algorithm as not only simulating Z_k and X_k but also Z_{-k} at each iteration. Because these updates are performed according to full conditional distributions $p\left(z_{1:n}|\,y, x_{1:n}\right)$ and $p\left(x_{1:n}|\,y, z_{1:n}\right)$ they admit (1) as an invariant distribution. However, as Z_{-k} is not necessary, it is discarded.

4 Finite Mixture of Gaussians

In the finite mixture distributions context, the EM and DA algorithms are routinely used to perform ML/MAP parameter estimation, and to sample the posterior respectively. Here we propose a new version of the SAGE algorithm which improves over [1] and a SADA algorithm; these algorithms can be straightforwardly extended to hidden Markov chains with Gaussian observations.

Assume we have T i.i.d. $\mathbb{R}^d$-valued observations $Y_{1:T}$ distributed according to a finite mixture of s Gaussians

$$Y_t \sim \sum_{j=1}^{s} \pi_j \mathcal{N}\left(\mu_j; \Sigma_j\right).$$

The parameters $X = \left\{\left(\mu_j, \Sigma_j, \pi_j\right); j = 1, \ldots, s\right\}$ are unknown, random and distributed according to the following conjugate prior distributions [3], [5]

$$\mu_j|\,\Sigma_j \sim \mathcal{N}\left(\alpha_j, \Sigma_j/\lambda_j\right),\ \Sigma_j^{-1} \sim \mathcal{W}\left(r_j, C_j\right),$$

$$\left(\pi_1, \ldots, \pi_s\right) \sim \mathcal{D}\left(\zeta_1, \ldots, \zeta_s\right).$$

The notation $\Sigma_j^{-1} \sim \mathcal{W}(r_j, C_j)$ denotes a Wishart distribution which has density proportional to

$$\left|\Sigma_j^{-1}\right|^{\frac{1}{2}(r-d-1)} \exp\left(-\frac{1}{2}\text{tr}\left(\Sigma_j^{-1} C_j^{-1}\right)\right).$$

The notation $(\pi_1, \ldots, \pi_s) \sim \mathcal{D}(\zeta_1, \ldots, \zeta_s)$ denotes a Dirichlet distribution which has a density restricted to the simplex proportional to $\prod_{k=1}^{s} \pi_k^{\zeta_k - 1}$. Here, the hyperparameters $\{(\alpha_j, \lambda_j, r_j, C_j, \zeta_j); j = 1, \ldots, s\}$ are assumed fixed but they could also be estimated from the data in a hierarchical Bayes model.

4.1 EM and DA

EM and DA introduce the i.i.d. missing data $Z_t \in \{1, \ldots, s\}$ such that

$$Y_t |\, Z_t = j \sim \mathcal{N}(\mu_j; \Sigma_j), \ \Pr(Z_t = j) = \pi_j.$$

The EM algorithm is standard and we omit the update equations for it. The DA algorithm proceeds as follows. We sample the discrete latent variables according to

$$Z_t^{(i)} \sim p\left(\cdot |\, y_t, X^{(i-1)}\right)$$

and compute the sufficient statistics $n_j^{(i)} \triangleq \sum_{t=1}^{T} \delta_{Z_t^{(i)}, j}$,

$$n_j^{(i)} \overline{y}_j^{(i)} \triangleq \sum_{t=1}^{T} \delta_{Z_t^{(i)}, j} y_t, \ \overline{S}_j^{(i)} \triangleq \sum_{t=1}^{T} \delta_{Z_t^{(i)}, j} y_t y_t^{\mathrm{T}}.$$

Then we sample the parameters according to

$$\Sigma_j^{-1(i)} \sim \mathcal{W}\left(r_j + n_j^{(i)}, \widetilde{\Sigma}_j^{-1(i)}\right) \qquad (2)$$

where

$$m_j^{(i)} = \frac{\lambda_j \alpha_j + n_j^{(i)} \overline{y}_j^{(i)}}{\lambda_j + n_j^{(i)}}$$

and

$$\widetilde{\Sigma}_j^{(i)} = C_j^{-1} + \lambda_j \alpha_j \alpha_j^{\mathrm{T}} + \overline{S}_j^{(i)} - \left(\lambda_j + n_j^{(i)}\right) m_j^{(i)} m_j^{(i)\mathrm{T}}$$

then

$$\mu_j^{(i)} \Big|\, \Sigma_j^{(i)} \sim \mathcal{N}\left(m_j^{(i)}, \frac{\Sigma_j^{(i)}}{\lambda_j + n_j^{(i)}}\right). \qquad (3)$$

Finally the weights are sampled according to

$$\left(\pi_1^{(i)}, \ldots, \pi_s^{(i)}\right) \sim \mathcal{D}\left(n_1^{(i)} + \zeta_1, \ldots, n_s^{(i)} + \zeta_s\right). \qquad (4)$$

4.2 SAGE and SADA

We follow the idea introduced in [1] for designing less informative missing data. Assume that we are interested in updating only (μ_j, Σ_j), the other components being fixed. The idea involves introducing binary missing data $Z_{t,j} \in \{0, j\}$ such that

$$\Pr(Z_{t,j} = j) = \pi_j;$$

i.e. these missing data tell us whether an observation is coming from component j which is less informative than knowing from which particular component it is derived. As outlined in [1], we cannot update π_j using this strategy because of the constraint $\sum_{j=1}^{s} \pi_j = 1$. Hence the authors in [1] propose updating the weights $(\pi_1, \ldots, \pi_s)$ using the standard EM approach. In our experiments we found that this can significantly reduce the rate of convergence of the algorithm. Here we follow an alternative approach. We propose updating the parameters of two components say j and k at the same time; i.e. we introduce the missing data $Z_{t,j,k} \in \{0, j, k\}$ such that

$$\Pr(Z_{t,j,k} = j) = \pi_j, \ \Pr(Z_{t,j,k} = k) = \pi_k$$

and

$$\begin{aligned} Y_t |\, Z_{t,j,k} &= j \sim \mathcal{N}(\mu_j; \Sigma_j), \ Y_t |\, Z_{t,j,k} = k \sim \mathcal{N}(\mu_k; \Sigma_k), \\ Y_t |\, Z_{t,j} &= 0 \sim \frac{\sum_{l \neq j, l \neq k} \pi_l \mathcal{N}(\mu_l; \Sigma_l)}{\sum_{l \neq j, l \neq k} \pi_l}. \end{aligned}$$

It follows that the SAGE update for(μ_j, Σ_j, π_j) (and similarly for (μ_k, Σ_k, π_k)) is given at iteration i by

$$\mu_j^{(i)} = \frac{\lambda_j \alpha_j + \sum_{t=1}^{T} y_t p\left(Z_{t,j,k} = j |\, y_t, X^{(i-1)}\right)}{\lambda_j + \sum_{t=1}^{T} p\left(Z_{t,j,k} = j |\, y_t, X^{(i-1)}\right)},$$

$$\begin{aligned} \Sigma_j^{(i)} = & \left(r_j - d - 1 + \lambda_j + \sum_{t=1}^{T} p\left(Z_{t,j,k} = j |\, y_t, X^{(i-1)}\right)\right)^{-1} \\ & \times \left(C_j^{-1} + \lambda_j \left(\mu_j^{(i)} - \alpha_j\right)\left(\mu_j^{(i)} - \alpha_j\right)^{\mathrm{T}}\right. \\ & \left. + \sum_{t=1}^{T} \left(y_t - \mu_j^{(i)}\right)\left(y_t - \mu_j^{(i)}\right)^{\mathrm{T}} p\left(Z_{t,j,k} = j |\, y_t, X^{(i-1)}\right)\right), \end{aligned}$$

$$\begin{aligned} \pi_j^{(i)} = & \left(1 - \sum_{l \neq j, l \neq k} \pi_l^{(i-1)}\right) \\ & \times \left(1 + \frac{\sum_{t=1}^{T} p\left(Z_{t,j,k} = k | y_t, X^{(i-1)}\right) + (\zeta_k - 1)}{\sum_{t=1}^{T} p\left(Z_{t,j,k} = j | y_t, X^{(i-1)}\right) + (\zeta_j - 1)}\right)^{-1}. \end{aligned}$$

The SADA algorithm proceeds similarly. To sample (μ_j, Σ_j, π_j) (and (μ_k, Σ_k, π_k)), we first sample the discrete latent variables

$$Z_{t,j,k}^{(i)} \sim p\left(\cdot |\, y_t, X^{(i-1)}\right)$$

and compute the sufficient statistics $n_j^{(i)} \triangleq \sum_{t=1}^{T} \delta_{Z_{t,j,k}^{(i)}, j}$,

Table 1: Log-posterior values for final iteration EM/SAGE and average log-posterior values for DA/SADA.

s	EM	SAGE	DA	SADA
5	-915.8	-671.5	-873.7	-886.0
6	-929.6	–603.2	-877.3	-886.7
7	-941.4	-576.5	-893.9	-906.9
8	-965.7	-559.2	-904.9	-875.0
9	-968.9	-503.0	-898.8	-882.5
10	-983.2	-478.1	-924.0	-906.6

$$n_j^{(i)}\overline{y}_j^{(i)} \triangleq \sum_{t=1}^{T} \delta_{Z_{t,j,k}^{(i)},j}\, y_t,\ \overline{S}_j^{(i)} \triangleq \sum_{t=1}^{T} \delta_{Z_{t,j,k}^{(i)},j}\, y_t y_t^{\mathrm{T}}$$

and similarly $n_k^{(i)}$, $n_k^{(i)}\overline{y}_k^{(i)}$ and $\overline{S}_k^{(i)}$. Then we sample the parameters $\left(\mu_j^{(i)},\Sigma_j^{(i)}\right)$ and $\left(\mu_k^{(i)},\Sigma_k^{(i)}\right)$ according to (2) and (3). Finally we sample $\left(\pi_j^{(i)},\pi_k^{(i)}\right)$ according to

$$\left(\pi_j^{(i)},\pi_s^{(i)}\right) \sim \left(1-\sum_{l\neq j,l\neq k}\pi_l^{(i-1)}\right)\mathcal{D}\left(n_j^{(i)}+\zeta_j, n_k^{(i)}+\zeta_k\right).$$

Note that the above algorithms extended for finite mixtures of Gaussians are classified to the expectation/conditional maximization[6].

4.3 Simulations

We generate a mixture of 5 10-dimensional Gaussians with components whose parameters were sampled according to the prior. For this dataset, we run 200 iterations of standard EM and SAGE 50 times fitting s components using the same initial random initializations. We have $d = 10$, $T = 100$ and the prior parameters were set to $\zeta_j = 1$, $\alpha_j = 0$, $\lambda_j = 0.01$, $r_j = d+1$, $C_j = 0.01I$. We also run 5000 iterations of DA and SADA 10 times. In Table 1, for EM and SAGE, we present the mean of the log-posterior values at the final iteration. For SA and SADA, we present the mean of the average log-posterior values of the last 1000 iterations. In terms of computational complexity, these comparisons are favourable to EM and DA. Indeed one iteration of EM/DA is more expensive than one iteration of SAGE/SADA because parameters for all components are updated in the first case whereas parameters for only two components are updated in the second case. The results are displayed in the table below. In all simulations SAGE outperforms EM consistently and significantly. For a larger number of components than eight, SADA performed better than DA. The difference between SADA and DA is not as impressive but the correlations (not presented here) of the SADA chain are reduced compared to the DA chain.

The performance of the SAGE algorithm is compared with that of the conventional EM algorithm for text-independent speaker identification experiments using a finite mixture of Gaussians as a speaker model. The data for training and testing were collected from 10 male speakers. A feature vector of 26 components, consisting of 12 mel-frequency cepstral coefficients plus normalized log energy and their first derivatives, is derived once every 10 ms over a 25.6 ms Hamming-windowed speech segment. A mixture of 16 full covariance Gaussians is used as a speaker model. For the EM and SAGE algorithms, the parameters are initialized with the same random value and used on 3 utterances from each speaker. The averaged speaker identification rates with the confidence intervals at a 90% confidence level for the SAGE and EM algorithms are respectively 83.3 [78.7, 88.0] and 81.3 [76.0, 86.0]. The confidence intervals are calculated based on the assumption that the accuracy rates follow the binomial distribution. The log-posterior value reached by SAGE was significantly superior to EM but this does not translate to major improvement in terms of speaker recognition. However, SAGE is competitive with the EM algorithm and is computationally cheaper to implement. In Figure 1, we display the speaker identification rates for each speaker. The dispersion in the rates for the SAGE algorithm is smaller than that for the EM algorithm. It can be considered that SAGE obtains more stable estimates for a wide range of speakers.

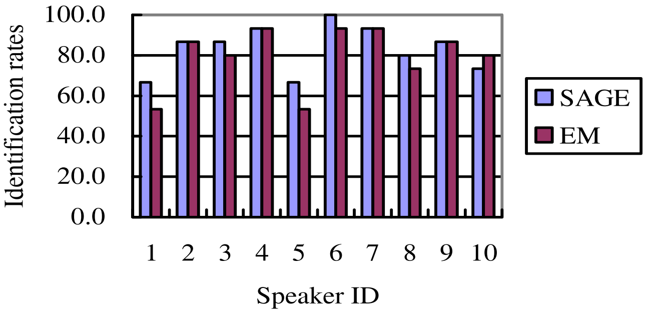

Figure 1: Speaker identification rates

5 REFERENCES

[1] G. Celeux, S. Chrétien, F. Forbes and A. Mkhadri, A component-wise EM algorithm for mixtures, *J. Comp. Graph. Stat.*, 10, 699-712, 2001.

[2] J.A. Fessler and A.O. Hero, Space-alternating generalized expectation-maximization algorithm, *IEEE Trans. Sig. Proc.*, 42:2664–2677, 1994.

[3] J.L. Gauvain and C.H. Lee, Maximum a Posteriori estimation for multivariate Gaussian mixture observations of Markov chains, *IEEE Trans. Speech Audio Proc.,* 2:291-298, 1994.

[4] G.J. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*, Wiley Series in Probability and Statistics, 1997.

[5] G.J. McLachlan and D. Peel, *Finite Mixture Models*, Wiley Series in Probability and Statistics, 2000.

[6] X. -L. Meng and D. B. Rubin, Maximum likelihood estimation via the ECM algorithm: A general framework, *Biometrika,*80, 2:267–278, 1993.

[7] C.P. Robert and G. Casella, *Monte Carlo Statistical Methods*, Springer-Verlag, 1999.