Akademie věd České republiky
Ústav teorie informace a automatizace, v.v.i.

Academy of Sciences of the Czech Republic
Institute of Information Theory and Automation

RESEARCH REPORT

Jan Reichl and Kamil Dedecius

Diffusion MCMC for mixture estimation

No. 2354 January 5, 2016

ÚTIA AV ČR, P.O.Box 18, 182 08 Prague, Czech Republic
Tel: +420 266052570, Fax: +420 266052068, Url: http://www.utia.cas.cz, E-mail:
dedecius@utia.cas.cz

Abstract

Distributed inference of parameters of mixture models by a network of cooperating nodes (sensors) with computational and communication capabilities still represents a challenging task. In the last decade, several methods were proposed to solve this issue, predominantly formulated within the expectation-maximization framework and with the assumption of mixture components normality. The present paper adopts the Bayesian approach to inference of general (non-normal) mixtures via the Markov chain Monte Carlo simulation from the parameter posterior distribution. By collaborative tuning of node chains, the method allows reliable estimation even at nodes with significantly worse observational conditions, where the components may tend to merge due to high variances. The method runs in the diffusion networks, where the nodes communicate only with their adjacent neighbors within 1 hop distance.

1 Introduction

Collaborative processing of data by geographically distributed agents (sensors) has attracted considerable attention in the last decade, particularly due to the rapid development of cheap adhoc wireless sensor networks. In the domain of statistical signal processing, the goal of collaboration is to arrive at (in a sense) better estimates of the parameters of interest. The cooperation mode is mostly dictated by the logical topology of the network, and may be centralized, decentralized or run incrementally in a Hamiltonian cycle [1]. The centralized topology suffers from high computational and communication requirements imposed on the processing center — a single point of failure (SPoF). The Hamiltonian topology somewhat reduces the communication burden, but has SPoF at each node and link, and its reconfiguration is an NP-hard problem [2]. Therefore, we focus on *diffusion networks*, where the nodes collaborate only with their adjacent neighbors within 1 hop distance [1, 3]. This mode has excellent robustness to node and link failures, and does not require high computational and communication performance of the network or its parts. Principally, the diffusion data processing is similar to the running consensus strategies with a limited amount of cooperation among nodes per time instant [4, 5].

While there exists abundance of methods for distributed estimation of numerous models, decentralized estimation of mixtures – convex combinations of probability distributions – is still rather underdeveloped. The existing methods are mostly based on expectation-maximization (EM) and assume normal components. For instance, Nowak [6] proposed the decentralized and multiple-steps decentralized EM algorithms for the Hamiltonian topology. A similar algorithm for this network type is due to Safarinejadian *et al.* [7], who later alleviated the network topology constraints to point to point networks [8]. Their method allows estimation of both the model order and parameters via sequential evaluation of global sufficient statistics and averaging of their local counterparts (E step), followed by re-estimation of parameters (M step). Gu proposes a decentralized EM algorithm with local E step and consensus-based M step [9].

The probably first diffusion algorithm is due to Weng *et al.* [10], followed by Pereira *et al.* [11], whose EM-based algorithm propagates available information across the network with a faster term for information diffusion and a slower term for information averaging. However, it is again oriented on normal mixtures only. The first component distribution-independent method for diffusion mixture estimation was proposed by Towfic *et al.* [12]. Its M step is solved by an adaptive diffusion process with a Newton's recursion. Very recently, Dedecius *et al.* [13] developed a Bayesian method for online mixture estimation from sequentially incoming data that can be used for any component distributions that have conjugate priors.

1.1 Contribution

The goal of this paper is to propose an adaptive diffusion method that overcomes the limitations of the existing methods. Namely, it (i) does not require normal components, (ii) provides means for simultaneous estimation of local and global mixture parameters, and (iii) allows efficient estimation even under heterogeneous conditions across the network, when the components virtually merge. The method is inspired by the delayed rejection adaptive Metropolis (DRAM) algorithm of Haario *et al.* [14], explained and reformulated for the diffusion setting below.

2 Basic DRAM algorithm

Assume a target posterior distribution of a parameter θ given a set of observations y with an analytically intractable probability density function $\pi(\theta|y)$. Starting from an arbitrary initial point $\theta_0 \in \operatorname{supp} \pi(\theta|y)$, the traditional Metropolis-Hastings algorithm simulates a sequence of random samples $(\theta_t)_{t=1,2,\ldots}$ whose empirical distribution converges to the target. The algorithm proceeds with new candidates θ_t' sampled from a convenient user-selected proposal distribution $q(\theta_t'|\theta_{t-1})$, that are accepted as θ_t with probability

$$\alpha_1(\theta_{t-1}, \theta_t') = \min\left\{1, \frac{\frac{\pi(\theta_t'|y)}{q(\theta_t'|\theta_{t-1})}}{\frac{\pi(\theta_{t-1}|y)}{q(\theta_{t-1}|\theta_t')}}\right\}. \tag{1}$$

Under rejection, the value of θ_{t-1} is duplicated and assigned to θ_t. Symmetry of the proposal density $q(\theta_t'|\theta_t)$ makes the denominators in (1) cancel and results in the basic Metropolis algorithm considered in the sequel. For details of the Metropolis-Hastings algorithm, see, e.g., [15].

In many applications, the normal distribution $\mathcal{N}(\theta_{t-1}, \Sigma)$ is a popular choice for the proposal distribution. Under normality and known covariance of the target it is even possible to set the proposal covariance so that the algorithm performs in a weak sense optimally [16]. However, the target covariance is rarely known. Haario *et al.* [17] propose its recursive empirical estimation from the already simulated samples; this method is called *adaptive Metropolis* (AM).

The main drawback of AM inherited from the basic Metropolis algorithm is the poor exploration of the target density under specific conditions where α_1 is too low, e.g. under a local bad fit of the proposal. To resolve this issue, it is possible to extend AM by *delayed rejection* (DR) algorithm of Mira [18]. It employs a given number of normal proposals used at different stages. Whenever a candidate θ'_t is rejected according to (1) (with $q_1 \equiv q$), instead of retaining the same position, another candidate θ''_t is proposed from the higher stage proposal $q_2(\theta''_t|\theta'_t)$ and accepted with probability

$$\alpha_2(\theta_{t-1}, \theta'_t, \theta''_t) = \frac{\frac{\pi(\theta''_t|y)}{q_1(\theta'_t|\theta_{t-1})q_2(\theta'_t|\theta''_t)}}{\frac{\pi(\theta_{t-1}|y)}{q_1(\theta'_t|\theta''_t)q_2(\theta_{t-1}|\theta'_t)}} \frac{[1 - \alpha_1(\theta''_t, \theta'_t)]}{[1 - \alpha_1(\theta_{t-1}, \theta'_t)]}.$$

Often, a scaled first-stage proposal is used for the second stage. It is naturally possible to design several stages, in this paper we stick with two for simplicity. The resulting DRAM algorithm thus combines the global (AM) adaptation with the improved local (DR) exploration. Due to the proposal adaptation, DRAM does not preserve the Markovian property, however, it is ergodic [14]. In the next section, we modify the DRAM algorithm for the diffusion mixture estimation.

3 Diffusion mixture estimation with DRAM

A diffusion network is a directed or undirected graph consisting of a set of nodes $\mathcal{I} = \{1, \ldots, I\}$ representing the agents, that are connected by vertices determining their communication links [1]. Each node $i \in \mathcal{I}$ can communicate with its *adjacent* neighbors $j \in \mathcal{I}$, i.e., the nodes within 1 edge distance. These nodes form the i's neighborhood $\mathcal{I}_i$; note that $i \in \mathcal{I}_i$, too. The diffusion estimation algorithms exploit information obtained from the neighborhood $\mathcal{I}_i$ to improve the estimation performance at node i. To avoid confusion, the superscript ${}^{[i]}$ will denote the quantities related to ith node in the sequel. For instance, the symbol $y^{[i]}$ is the set of measurements taken by node i.

3.1 Diffusion DRAM algorithm

The proposed distributed mixture estimation method assumes the mixture models of the form

$$p\left(y^{[i]}\middle|\phi, \delta, \vartheta^{[i]}\right) = \sum_{k=1}^{K} \phi_k p_k \left(y^{[i]}\middle|\delta_k, \vartheta^{[i]}_k\right), \qquad (2)$$

where $i \in \mathcal{I}$ is the node index, $y^{[i]}$ is a set of observations taken by the ith node, $\phi = [\phi_1, \ldots, \phi_K]^\intercal$ is a vector of global component weights taking values in the unit K-simplex, $\delta = \{\delta_1, \ldots, \delta_K\}$ are global component parameters and $\vartheta^{[i]} = \{\vartheta^{[i]}_1, \ldots, \vartheta^{[i]}_K\}$ are local component parameters. The unknown parameters to be estimated are ϕ, θ and $\vartheta^{[i]}$. A classical example is the normal mixture model where the nodes assume identical locations (global parameters δ) with different observation noise (local parameter $\vartheta^{[i]}$). The assumption of common ϕ may be potentially restrictive if some nodes have very different observation conditions, e.g. there are missing observations connected with some components, but it may be easily relaxed.

The Bayesian estimation of global parameters $\theta = \{\phi, \delta\}$ and local $\vartheta^{[i]}$ proceeds with the prior distribution $\pi(\theta, \vartheta^{[i]})$, that being updated via the Bayes' theorem yields the posterior distribution

$$\begin{aligned} \pi\left(\theta, \vartheta^{[i]} \middle| y^{[i]}\right) &= \pi\left(\phi, \delta, \vartheta^{[i]} \middle| y^{[i]}\right) \\ &\propto p\left(y^{[i]} \middle| \phi, \delta, \vartheta^{[i]}\right) \pi\left(\phi, \delta, \vartheta^{[i]}\right). \end{aligned} \tag{3}$$

Now, the aim is to simulate samples from the clearly analytically intractable posterior distribution using the DRAM procedure with collaborative adaptation. The respective steps are described below.

3.1.1 Diffusion sampling and 1st stage DR adaptation

Each node $i \in \mathcal{I}$ starts with simulating $\left(\theta_t^{[i]}, \vartheta^{[i]}\right)_{t=1,2,\ldots}$ from the posterior density $\pi\left(\theta, \vartheta^{[i]} \middle| y^{[i]}\right)$ using the basic DRAM algorithm outlined in Section 2. Let us focus on θ, as sampling from $\vartheta_{[i]}$ follows standard procedure. The two-stage DR employs two normal proposal distributions $\mathcal{N}\left(\theta_{t-1}^{[i]}, C_{1,t}^{[i]}\right)$ and $\mathcal{N}\left(\theta_{t-1}^{[i]}, C_{2,t}^{[i]}\right)$, respectively. After certain amount of steps, chosen either *a priori* or randomly, each node $i \in \mathcal{I}$ receives the last sample $\theta_t^{[j]}$ from a randomly chosen neighbor $j \in \mathcal{N}_i$. This sample is accepted at node i with probability

$$\alpha^{[i \leftarrow j]} = \min \left\{ 1, \frac{\pi\left(\theta_t^{[j]}, \vartheta^{[i]} \middle| y^{[i]}\right)}{\pi\left(\theta_t^{[i]}, \vartheta^{[i]} \middle| y^{[i]}\right)} \right\} \tag{4}$$

leading to

$$\theta_{t+1}^{[i]} \leftarrow \theta_t^{[j]}. \tag{5}$$

Since the acceptance of neighbor's sample may mean that its proposal distribution better reflects statistical properties of the target posterior distribution, its adoption is a part of the algorithm, too:

$$C_{1,t+1}^{[i]} \leftarrow C_{1,t}^{[j]}. \tag{6}$$

This exchange should prevent the chains from being stuck in high probability areas and to explore the target support as much as possible. Robustness to invalid $C_{1,t+1}^{[i]}$ is preserved by its periodic optimization (see Algorithm 1).

3.1.2 Diffusion 2nd stage DR adaptation

The second nodes cooperation feature – recentering and fixing the second stage proposals – starts after a reasonable number t_0 of steps after initialization. Until this time instant, e.g. scaled first stage proposal can be used.

The collaborative tuning of the second stage proposal is realized either randomly or after every n_3 steps. It consists in finding the "best" neighboring node $k \in \mathcal{I}_i$ and adoption of its statistical knowledge about mean (or alternatively mode) of the posterior component distribution. Node k is determined at step t according to the following posterior likelihood maximization criterion:

$$k = \operatorname*{argmax}_{j \in \mathcal{I}_i} \pi \left(\widehat{\theta}^{[j]}, \vartheta^{[j]} \middle| y^{[j]} \right), \tag{7}$$

where $\widehat{\theta}^{[j]}$ is computed as the mean of the last $t - t_a$ samples of the chain $\left(\theta_t^{[j]}\right)$.

After finding the best node $k \in \mathcal{I}_i$, we fix the proposal at the approximate mean, leading to $\mathcal{N}(\widehat{\theta}^{[k]}, \cdot)$. The advantage of this approach lies in stabilization of the sampling process, as the worse nodes second stage proposals are relocated and fixed in the high probability regions.

3.1.3 Local optimization of proposals

Local AM optimization of the proposal covariance matrices [14, 17] relies on the sample covariance matrix

$$C_t^{[i]} = \operatorname{cov}\left(\theta_0^{[i]}, \ldots, \theta_t^{[i]}\right).$$

The resulting proposals covariance matrices

$$C_{1,t}^{[i]} = s_d \cdot C_t^{[i]}, \tag{8}$$

$$C_{2,t}^{[i]} = \varepsilon \cdot C_t^{[i]}, \tag{9}$$

where $s_d, \varepsilon > 0$ are the scaling factors. Gelman *et al.* [16] show, that under normal target distribution and normal proposal, $s_d = 2.38^2/d$, d being the dimension, optimizes in a weak sense mixing properties of the Metropolis algorithm. The other scaling factor $\varepsilon < s_d$ is used to tighten the second proposal.

```
**Algorithm 1** Distributed DRAM
**Initialization**: For all $i \in \mathcal{I}$, set the number of steps $n_1$ between AM covariance updates, set the number $n_2$ of steps between exchanges of $\theta_t^{[j]}$ and $C_{1,t}^{[j]}, j \in \mathcal{N}_i$, set the number of steps $n_3$ between exchanges of second stage DR proposals. Set constants $s_d, \varepsilon, t_a$ and $t_0$. Set initial proposals and sample initial points $\theta_0^{[i]}$.

**For** $t = 1 : T$ **do**:
    1. DR 1st stage: Propose $\theta_t^{\prime,[i]} \sim \mathcal{N}\left(\theta_{t-1}^{[i]}, C_{1,t-1}^{[i]}\right)$;
    2. DR 2nd stage (if $\theta_t^{\prime}$ rejected):
       Propose $\theta_t^{\prime\prime,[i]} \sim \mathcal{N}\left(\widehat{\theta}^{[i]}, C_{2,t-1}^{[i]}\right)$.
       If $t \leq t_0, \widehat{\theta}^{[i]} \equiv \theta_{t-1}^{[i]}$.
    3. Each $n_1$ steps update $C_t^{[i]} = \mathrm{cov}\left(\theta_{0:t}^{[i]}\right)$.
    4. Each $n_2$ steps randomly choose $j \in \mathcal{I}_i$ and set

           $\theta_t^{[i]} \leftarrow \theta_t^{[j]}$ and $C_t^{[i]} \leftarrow C_t^{[j]}$ with probability (4).

    5. **if** $t \geq t_0$:
       Each $n_3$ steps select $k \in \mathcal{I}_i$ such, that

           $k = \underset{j \in \mathcal{I}_i}{\mathrm{argmax}}\, \pi\left(\widehat{\theta}^{[j]}, \vartheta^{[j]} \middle| y^{[j]}\right)$.

       Set for all $j \in \mathcal{I}_i$ the 2nd stage proposal mean to $\widehat{\theta}^{[k]}$.
```

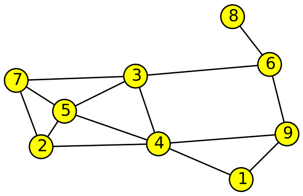

Figure 1: Diffusion network layout.

4 Simulation Results

We assume a randomly generated diffusion network consisting of 9 nodes $\mathcal{I} = \{1,\ldots,9\}$ depicted in Fig. 1. The nodes estimate the parameters of a three component normal mixture model

$$Y_\tau^{[i]} \sim \sum_{k=1}^{3} \phi_k \mathcal{N}\left(y_\tau^{[i]} \middle| \mu_k, \sigma_k^{2,[i]}\right), \; \tau = 1,2,\ldots,5000,$$

with the global parameter $\theta = \{\phi,\mu\}$, $\phi = [0.2, 0.25, 0.55]^\intercal$ and $\mu = [2, 6.5, 13.5]^\intercal$, and the local standard deviations $\sigma^{[i]} = [1.4 + 0.1i, 0.9 + 0.1i, 2.7 + 0.3i]$. Obviously, with higher (noise) variance the components tend to merge, see Fig. 2. One of the main and most challenging goals is the reliable detection of these merging components.

The aim of each node is to estimate the global vectors μ and ϕ and the local vector $\sigma^{[i]}$ using the described diffusion DRAM. For comparison, we consider the same task with isolated nodes employing the basic DRAM procedure and the (non-Bayesian) diffusion EM algorithm of Towfic *et al.* [12], whose step size is 0.01 and combiner weights are uniform.

The adopted prior distribution for DRAM scenarios is

$$\pi\left(\phi,\mu,\sigma^{[i]}\right) = \mathrm{Dir}(\phi|\mathbf{1}) \times \mathcal{N}(\mu|m, 15^2 I_{3\times 3}) \times \mathcal{U}\left(\sigma^{[i]} \middle| 0, 10\right)$$

where the vector m contains values of the k-means clustering centroids from obtained from the set of observations of the first node. The simulation from the target posterior density employed ordering constraint $\mu_1 < \mu_2 < \mu_3$ to prevent label switching [19]. The setting of Algorithm 1 is as follows: $T = 25000$, $t_a = t - 500$, $n_1 = 5$, $n_2 = 500$, $t_0 = 5000$ and $n_3 = 500$. The final inference is based on the last 15000 samples, i.e the burn-in period is 10000 draws.

The algorithm performance assessment is based on the estimates mean squared error averaged over the network (AMSE). The resulting AMSEs are given in Table 1. It is not surprising to see that collaboration leads to significantly better results. The main reason is that the non-collaborating nodes with high variances fail to properly identify the three-component mixture. Instead, they fit a two-component mixture. This is a demonstration of a well-known deeper issue: the observation noise hides the phenomenon of interest, which may in certain applications lead to (for instance physically) absurd results. The proposed method

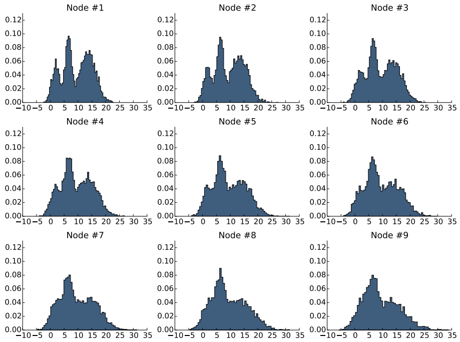

Figure 2: Nodes data sets.

yields better results than the diffusion EM algorithm [12], naturally at the cost of a higher computational burden.

5 Conclusion

The paper proposes a distributed MCMC algorithm of DRAM type, whose settings are collaboratively tuned during the estimation process in an unsupervised way. Its important feature is the ability to detect and effectively estimate merging components under spatially heterogeneous (noise) variances. Unlike the existing distributed (mostly EM-based) solutions, the method is not limited to normal mixtures. Its performance is superior to the basic non-collaborative DRAM and to the diffusion EM [12], of course at the cost of higher computational burden associated with the MCMC framework.

Acknowledgement

The work is supported by the Czech Science Foundation, grant no. 14-06678P.

References

[1] A. H. Sayed, “Adaptive networks,” *Proceedings of the IEEE*, vol. 102, no. 4, pp. 460–497, Apr. 2014.

[2] X. Zhao, S.-Y. Tu and A.H. Sayed, “Diffusion adaptation over networks under imperfect information exchange and non-stationary data,” *IEEE Transactions on Signal Processing*, vol. 60, no. 7, pp. 3460–3475, 2012.

[3] A. H. Sayed, “Diffusion adaptation over networks,” in *Academic Press Library in Signal Processing*, vol. 3, R. Chellapa and S. Theodoridis, Eds. Academic Press, Elsevier, 2014, pp. 323–454.

[4] P. Braca, S. Marano, and V. Matta, "Enforcing consensus while monitoring the environment in wireless sensor networks," *IEEE Transactions on Signal Processing* vol. 56, no. 7, pp. 3375–3380, 2008.

[5] P. Braca, S. Marano, V. Matta, and P. Willett, "Asymptotic optimality of running consensus in testing binary hypotheses" *IEEE Transactions on Signal Processing*, vol. 58, no. 2, pp. 814–825, 2010.

[6] R. Nowak, "Distributed EM algorithms for density estimation and clustering in sensor networks," *IEEE Transactions on Signal Processing*, vol. 51, no. 8, pp. 2245–2253, 2003.

[7] B. Safarinejadian, M. Menhaj and M. Karrari, "Distributed data clustering using expectation maximization algorithm," *J. Appl. Sci.*, vol. 9, no. 5, pp. 854–864, 2009.

[8] B. Safarinejadian, M. Menhaj and M. Karrari, "Distributed unsupervised Gaussian mixture learning for density estimation in sensor networks," *IEEE Trans. Instrum. Meas.*, vol. 59, no. 9, pp. 2250–2260, 2010.

[9] D. Gu, "Distributed EM algorithm for Gaussian mixtures in sensor networks," *IEEE Trans. Neural Netw.*, vol. 19, no. 7, pp. 1154–1166, Jul. 2008.

[10] Y. Weng, W. Xiao, and L. Xie, "Diffusion-based EM algorithm for distributed estimation of Gaussian mixtures in wireless sensor networks." *Sensors*, vol. 11, no. 6, pp. 6297–316, Jan. 2011.

[11] S. S. Pereira, R. Lopez-Valcarce, and A. Pages-Zamora, "A diffusion-based EM algorithm for distributed estimation in unreliable sensor networks," *IEEE Signal Process. Lett.*, vol. 20, no. 6, pp. 595–598, Jun. 2013.

[12] Z. Towfic, J. Chen and A. Sayed, "Collaborative learning of mixture models using diffusion adaptation," in *Proc. 2011 IEEE International Workshop on Machine Learning for Signal Processing*, 2011.

[13] K. Dedecius, J. Reichl and P.M.Djurić, "Sequential estimation of mixtures in diffusion networks," *IEEE Signal Process. Lett.*, vol. 22, no. 2, pp. 197–201, 2015.

[14] H. Haario, M. Laine, A. Mira and E Saksman, "DRAM: Efficient adaptive MCMC," *Statistics and Computing*, vol. 16, pp. 339–354, 2006.

[15] C. Robert and G. Casella, *Monte Carlo Statistical Methods.* Springer New York, 2004.

[16] A. Gelman, G. Roberts and W. Gilks, "Efficient Metropolis jumping rules," *Bayesian Statistics*, vol. 5, pp. 599–607, 1996.

[17] H. Haario, E. Saksman and J. Tamminen, "An Adaptive Metropolis algorithm," *Bernoulli*, vol. 7, pp. 223–242, 1998.

[18] A. Mira, "On Metropolis-Hastings algorithms with delayed rejection," *Metron*, vol. 59, pp. 3–4, 2001.

[19] J. Marin and C. Robert, *Bayesian Core: A Practical Approach to Computational Bayesian Statistics*. Springer New York 2007.

Table 1 AMSEs summary for chosen parameters and total AMSE of all parameters.

Parameter	**DRAM**	**Diff. DRAM**	**Diff. EM** [12]
μ_1	1.9612	0.0043	0.0001
μ_2	0.0166	0.0650	0.0102
μ_3	0.2600	0.0013	0.0238
σ_1	0.1495	0.0047	0.1080
σ_3	0.1154	0.0129	0.5066
σ_3	0.2838	0.0159	0.1520
ϕ_1	0.0371	0.0000	0.0002
ϕ_2	0.0025	0.0000	0.0002
ϕ_3	0.0039	0.0001	0.0001
$\sum$ AMSE	2.8300	0.1042	0.8012