Bayesian Multi-topic Microarray Analysis with Hyperparameter Reestimation

Tomonari Masada, Tsuyoshi Hamada, Yuichiro Shibata, and Kiyoshi Oguri

Nagasaki University, Bunkyo-machi 1-14, Nagasaki 852-8521, Japan
masada@cis.nagasaki-u.ac.jp

Abstract. This paper provides a new method for multi-topic Bayesian analysis for microarray data. Our method achieves a further maximization of lower bounds in a marginalized variational Bayesian inference (MVB) for Latent Process Decomposition (LPD), which is an effective probabilistic model for microarray data. In our method, hyperparameters in LPD are updated by empirical Bayes point estimation. The experiments based on microarray data of realistically large size show efficiency of our hyperparameter reestimation technique.

1 Introduction

Latent Dirichlet allocation (LDA) [4], an epoch-making Bayesian multi-topic analysis method, finds its application in various research fields including natural language processing, information retrieval and image analysis [2][5][6][13][14]. We can also apply an LDA-like Bayesian multi-topic analysis to microarray data, where we regard samples as documents and genes as words. However, microarray data are given as a real matrix, not as a non-negative integer matrix. Therefore, researchers apply LDA after introducing Gaussian distributions in place of word multinomial distributions and provide an efficient probabilistic model, *Latent Process Decomposition (LPD)* [9], where topics in LDA are called *processes*.

As we can find Dirichlet prior distributions for word multinomials in LDA, we can find prior distributions for Gaussian distributions in LPD. To be precise, Gaussian priors are prepared for mean parameters, and Gamma priors are for precision parameters. However, as far as we know, there are still no reports on how we can reestimate *hyperparameters*, i.e., parameters of these prior distributions, and there are also no reports on whether we can improve microarray analysis by using hyperparameter reestimation. Therefore, in this paper, we provide a hyperparameter reestimation technique for LPD and show the results of experiments using microarray data of realistically large size.

Our method is based on a marginalized variational Bayesian inference (MVB) proposed by Ying et al. [15]. Marginalized variational Bayesian inference, alternatively called *collapsed* variational Bayesian inference [12], theoretically achieves better lower bounds than conventional variational Bayesian inferences

[4][9]. In this paper, we propose a method for maximizing lower bounds further by reestimating hyperparameters, where we use empirical Bayes point estimates [4] as new values for hyperparameters. We denote our method by *MVB+*.

The experiments presented in [15] are only based on microarray data of small size. Therefore, we use microarray data of realistically large size, where the number of genes (features) ranges from 3,000 to 18,000. Our experiments using large microarray data will show that hyperparameter reestimation can achieve better results in lower bound maximization and also in sample clustering.

The rest of the paper is organized as follows. Section 2 describes MVB for LPD. In Section 3, we provide the details of MVB+. Section 4 presents the results of our experiments. Section 5 concludes the paper with future works.

2 Latent Process Decomposition (LPD)

Latent Process Decomposition (LPD) [9] can be regarded as a latent Dirichlet allocation (LDA) [4] re-designed for microarray data. LPD and LDA share a special feature, *topic multiplicity*. That is, both in LDA and in LPD, each document (sample) is modeled as a mixture of multiple topics (processes). With respect to this point, LPD and LDA are completely the same.

However, LPD is different from LDA with respect to observed data generation, because microarray data are always given as a matrix of real values. Therefore, LPD uses Gaussian distributions in place of word multinomial distributions in LDA. A generative description of LPD can be given as follows:

- Draw a Gaussian distribution $N(x;\mu_{gk},\lambda_{gk})$ for each pair of gene g and process k from prior distributions. To be precise, a mean parameter μ_{gk} is drawn from a Gaussian prior $N(\mu;\mu_0,\lambda_0)$ and a precision parameter λ_{gk} is drawn from a Gamma prior $\mathrm{Gam}(\lambda;a_0,b_0)$.
  In LDA, this part corresponds to a determination of word probability with respect to a specific topic by drawing a topic-wise word multinomial distribution from a corpus-wide Dirichlet prior.
- Draw a multinomial distribution $\mathrm{Mult}(z;\theta_d)$ for each sample d from a symmetric Dirichlet prior distribution $\mathrm{Dir}(\theta;\alpha)$.
  This part is completely the same with LDA by identifying samples in LPD with documents in LDA, and processes in LPD with topics in LDA.
- For an occurrence of gene g in sample d, draw a process z_{dg} from $\mathrm{Mult}(z;\theta_d)$, and then draw an observed real value x_{dg} from $N(x;\mu_{gz_{dg}},\lambda_{gz_{dg}})$.
  This part is similar to a topic drawing from $\mathrm{Mult}(z;\theta_d)$ followed by a word drawing from the word multinomial corresponding to the drawn topic in LDA. However, in case of LPD, observed data are real, and each gene occurs exactly once in each sample.

While LPD is described as a generative model for microarray data, it can be easily applied to other real matrix data.

The generative description shown above leads to the full joint distribution:

$$
\begin{aligned}
&p(\mathbf{x},\mathbf{z},\theta,\mu,\lambda|\alpha,\mu_0,\lambda_0,a_0,b_0)\\
&=\prod_d p(\theta_d|\alpha)\prod_{g,k} p(\mu_{gk}|\mu_0,\lambda_0)\prod_{g,k} p(\lambda_{gk}|a_0,b_0)\prod_{d,g} p(z_{dg}|\theta_d)p(x_{dg}|\mu_{gz_{dg}},\lambda_{gz_{dg}})\\
&=\prod_d \frac{\Gamma(K\alpha)}{\Gamma(\alpha)^K}\prod_k \theta_{dk}{}^{n_{dk}+\alpha-1}\cdot\prod_{g,k}\sqrt{\frac{\lambda_0}{2\pi}}\exp\left\{-\frac{\lambda_0(\mu_{gk}-\mu_0)^2}{2}\right\}\\
&\cdot\prod_{g,k}\frac{b_0{}^{a_0}}{\Gamma(a_0)}\lambda_{gk}{}^{a_0-1}e^{-b_0\lambda_{gk}}\cdot\prod_{d,g}\sqrt{\frac{\lambda_{gz_{dg}}}{2\pi}}\exp\left\{-\frac{\lambda_{gz_{dg}}(x_{dg}-\mu_{gz_{dg}})^2}{2}\right\}
\end{aligned}
\tag{1}
$$

In this paper, we adopt a *marginalized variational Bayesian inference (MVB)* proposed in [15] for LPD and introduce hyperparameter reestimation to MVB, because MVB achieves better inference results than without marginalization as shown in [15]. An outline of MVB for LPD is given below.

We first marginalize process multinomial parameters θ_{dk} in Eq. (1) as follows:

$$
\begin{aligned}
&p(\mathbf{x},\mathbf{z},\mu,\lambda|\alpha,\mu_0,\lambda_0,a_0,b_0)=\int p(\mathbf{x},\mathbf{z},\theta,\mu,\lambda|\alpha,\mu_0,\lambda_0,a_0,b_0)d\theta\\
&=\prod_d \frac{\Gamma(K\alpha)}{\Gamma(\alpha)^K}\frac{\prod_k\Gamma(n_{dk}+\alpha)}{\Gamma(\sum_k n_{dk}+K\alpha)}\cdot\prod_{g,k}\sqrt{\frac{\lambda_0}{2\pi}}\exp\left\{-\frac{\lambda_0(\mu_{gk}-\mu_0)^2}{2}\right\}\\
&\cdot\prod_{g,k}\frac{b_0{}^{a_0}}{\Gamma(a_0)}\lambda_{gk}{}^{a_0-1}e^{-b_0\lambda_{gk}}\cdot\prod_{d,g}\sqrt{\frac{\lambda_{gz_{dg}}}{2\pi}}\exp\left\{-\frac{\lambda_{gz_{dg}}(x_{dg}-\mu_{gz_{dg}})^2}{2}\right\},
\end{aligned}
\tag{2}
$$

where n_{dk} denotes the number of genes whose observed data in sample d are generated from a Gaussian distribution corresponding to process k. Our aim is to obtain a parameter values maximizing the log likelihood:

$$
\log p(\mathbf{x}|\alpha,\mu_0,\lambda_0,a_0,b_0)=\log\int\sum_{\mathbf{z}} p(\mathbf{x},\mathbf{z},\mu,\lambda|\alpha,\mu_0,\lambda_0,a_0,b_0)d\mu d\lambda \tag{3}
$$

The maximization of this log likelihood is intractable. Therefore, we use a variational method and introduce an approximated posterior $q(\mathbf{z},\mu,\lambda)$ to obtain a lower bound of the log likelihood via Jensen's inequality:

$$
\log p(\mathbf{x}|\alpha,\mu_0,\lambda_0,a_0,b_0)\geq\log\int\sum_{\mathbf{z}} q(\mathbf{z},\mu,\lambda)\log\frac{p(\mathbf{x},\mathbf{z},\mu,\lambda|\alpha,\mu_0,\lambda_0,a_0,b_0)}{q(\mathbf{z},\mu,\lambda)}d\mu d\lambda \tag{4}
$$

Further, we assume that $q(\mathbf{z},\mu,\lambda)$ can be factorized as $q(\mathbf{z})q(\mu)q(\lambda)$. For each of $q(\mathbf{z})$, $q(\mu)$, and $q(\lambda)$, we choose a multinomial distribution defined over processes $\mathrm{Mult}(z_{dg};\gamma_{dg})$, a Gaussian distribution $N(\mu_{gk};m_{gk},l_{gk})$, and a Gamma distribution $\mathrm{Gam}(\lambda_{gk};a_{gk},b_{gk})$, respectively. Among the parameters of the approximated posterior, multinomial parameters γ_{dgk} of $\mathrm{Mult}(z_{dg};\gamma_{dg})$ can be interpreted as showing how closely gene g in sample d relates to process k. This interpretation is important in applications.

After some calculations, we obtain a lower bound L of the log likelihood:

$$\begin{aligned}
L = &\sum_{\mathbf{z}} \prod_{d,g,k} \gamma_{dgk}{}^{\delta_{dgk}} \sum_{d,k} \log\Gamma(n_{dk}+\beta) - \frac{\lambda_0}{2}\sum_{g,k}\left\{\frac{1}{l_{gk}} + (m_{gk}-\mu_0)^2\right\} \\
&+ (a_0 - a_{gk})\sum_{g,k}\psi(a_{gk}) - b_0\sum_{g,k}\frac{a_{gk}}{b_{gk}} + \frac{1}{2}\sum_{d,g,k}\gamma_{dgk}\{\psi(a_{gk}) - \log b_{gk}\} \\
&- \frac{1}{2}\sum_{d,g,k}\gamma_{dgk}\frac{a_{gk}}{b_{gk}}\left\{\frac{1}{l_{gk}} + (x_{dg}-m_{gk})^2\right\} - \sum_{d,g,k}\gamma_{dgk}\log\gamma_{dgk} - \frac{1}{2}\sum_{g,k}\log l_{gk} \\
&+ \sum_{g,k}\log\Gamma(a_{gk}) - a_0\sum_{g,k}\log b_{gk} + \sum_{g,k}a_{gk} - \frac{DG\log 2\pi}{2} - DK\log\Gamma(a) \\
&+ D\{\log\Gamma(K\alpha) - \log\Gamma(G+K\alpha)\} + GK\left\{\frac{\log\lambda_0 + 1}{2} + a_0\log b_0 - \log\Gamma(\alpha)\right\} \quad (5)
\end{aligned}$$

where δ_{dgk} is 1 when $z_{dg} = k$ and 0 otherwise, and $\psi(\cdot)$ means digamma function. Our aim is now to maximize L. The details of update formula derivation are referred to [15]. Here we only include the resulting formulas.

$$\begin{aligned}
\gamma_{dgk} \leftarrow &\log\left(\alpha + \sum_{d',g',k'}\gamma_{d'g'k'} - \gamma_{dgk}\right) - \frac{\sum_{d',g',k'}\gamma_{d'g'k'}(1-\gamma_{d'g'k'}) - \gamma_{dgk}(1-\gamma_{dgk})}{2(\alpha + \sum_{d',g',k'}\gamma_{d'g'k'} - \gamma_{dgk})^2} \\
&+ \frac{1}{2}\{\psi(a_{gk}) - \log b_{gk}\} - \frac{a_{gk}}{2b_{gk}}\left\{\frac{1}{l_{gk}} + (x_{dg}-m_{gk})^2\right\} \quad (6)
\end{aligned}$$

$$l_{gk} \leftarrow \frac{a_{gk}\sum_d \gamma_{dgk}}{b_{gk}} + \lambda_0 \quad (7)$$

$$m_{gk} \leftarrow \frac{1}{l_{gk}}\left\{\frac{a_{gk}\sum_d \gamma_{dgk}x_{dg}}{b_{gk}} + \mu_0\lambda_0\right\} \quad (8)$$

$$a_{gk} \leftarrow a_0 + \frac{1}{2}\sum_d \gamma_{dgk} \quad (9)$$

$$b_{gk} \leftarrow \frac{1}{2}\sum_d \gamma_{dgk}\left\{\frac{1}{l_{gk}} + (x_{dg}-m_{gk})^2\right\} + b_0 \quad (10)$$

The lower bound L in Eq. (5) will also be utilized to estimate inference quality in the experiments presented in Section 4. However, the first term in Eq. (5) is intractable. [15] gives the following approximation:

$$\begin{aligned}
&\sum_{\mathbf{z}} \prod_{d,g,k} \gamma_{dgk}{}^{\delta_{dgk}} \sum_{d,k} \log\Gamma(n_{dk}+\alpha) \\
&\approx DK\log\Gamma(\alpha) + \sum_{d,g,k}\gamma_{dgk}\left\{\log\left(\alpha + \sum_{g'>g}\gamma_{dg'k}\right) - \frac{\sum_{g'>g}\gamma_{dg'k}(1-\gamma_{dg'k})}{2(\alpha + \sum_{g'>g}\gamma_{dg'k})^2}\right\} \quad (11)
\end{aligned}$$

3 MVB+

As shown in Eq. (5), , the lower bound L includes the following hyperparameters: α, μ_0, λ_0, a_0, b_0. We can maximize L by taking derivatives of L with respect to these hyperparameters. This idea is not pursued in previous researches [9][15]. Therefore, we show how to reestimate hyperparameters in this section. We denote MVB for LPD accompanied with hyperparameter reestimation by *MVB+*.

First, we consider the reestimation of α. However, the derivative of L with respect to α suggests a difficulty in obtaining an update efficient in execution time even after introducing an approximation in Eq. (11). Further, a marginalized variational Bayesian inference for LDA [12] uses a fixed value for α. Therefore, we do not update α in MVB+. We leave as an open problem deriving an efficient update for α.

Second, by taking the derivative of L with respect to μ_0, we can obtain a simple update formula:

$$\mu_0 \leftarrow \frac{\sum_{g,k} m_{gk}}{GK} \tag{12}$$

Third, the derivative of L with respect to λ_0 is

$$\frac{\partial L}{\partial \lambda_0} = -\frac{1}{2}\sum_{g,k}\left\{\frac{1}{l_{gk}} + \left(m_{gk} - \mu_0\right)^2\right\} + \frac{GK}{2\lambda_0} \tag{13}$$

While we can obtain an update $\lambda_0 \leftarrow GK \big/ \sum_{g,k}\{1/l_{gk} + (m_{gk} - \mu_0)^2\}$, preliminary experiments reveal that this update often results in unstable numerical computations. Therefore, we do not reestimate λ_0.

Finally, we take the derivatives of L with respect to the rest two hyperparameters, a_0 and b_0. For b_0, a simple update formula

$$b_0 \leftarrow \frac{GKa_0}{\sum_g \sum_k a_{gk} / b_{gk}} \tag{14}$$

follows. However, some trick is required for a_0, because the derivative leads to:

$$\psi(a_0) \leftarrow \frac{\sum_g \sum_k \left\{\psi(a_{gk}) - \log b_{gk}\right\}}{GK} + \log b_0 \tag{15}$$

and digamma function should be inverted. We use a method in [8] for digamma function inversion. We reproduce the method for evaluating $\psi^{-1}(y)$ here.

1. If $y \geq -2.22$ then $x \leftarrow \exp(y) + 0.5$, else $x \leftarrow -1/(y - \psi(1))$.
2. Repeat the following until convergence: $x \leftarrow x - (\psi(x) - y)/\psi'(x)$.

We denote trigamma function by $\psi'(\cdot)$.

It should be noted that we can use the same formula Eq. (5) for computing L even when we reestimate hyperparameters. Therefore, we can compare inference quality of MVB+ with that of MVB by using L.

4 Experiments

4.1 Comparison strategy

In this section, we present the results of our experiments to reveal how our hyperparameter reestimation works. We compare MVB+ with MVB by (i) lower bound L (see Eq. (5)) and also by (ii) quality of sample clusters. In comparing MVB+ with MVB by L, larger values are better, because our task is to maximize L as far as possible. On the other hand, when we compare MVB+ with MVB by sample clustering, we take the following strategy.

A clustering of samples is induced by determining a process k satisfying

$$k = \arg\max_{k'} \sum_g \gamma_{dgk'} \tag{16}$$

for each sample d, where γ_{dgk} for all d, g, k are obtained as a result of a sufficient number of update iterations in MVB+ or in MVB. We evaluate the quality of a clustering of samples by, first, computing precision and recall for each cluster. Then, for each pair of precision *P* and recall *R*, we compute an *F*-score as their harmonic mean, i.e., *F* = 1 / (1/*P* + 1/*R*). We use the precision, recall, and *F*-score averaged over all clusters as an evaluation measure for each clustering results.

4.2 Datasets

Previous research [15] only use datasets of small size, where the number of genes ranges from 500 to 1,000. Therefore, in this paper, we use datasets of realistically large size, available at [1] and [16], whose specification is given in Table 1.

“Leukemia” dataset from [1], referred as LK in this paper, provides three labels ALL/MLL/AML as a prefix of each sample name. When we use these three labels as true cluster labels, both MVB+ and MVB give quite poor performance. Therefore, we compare MVB+ with MVB based on the binary clustering task (ALL,MLL)/AML after identifying ALL with MLL.[1]

For “Five types of breast cancer” dataset from [16], referred as D1 in this paper, we find a description telling that meaningfull classifier should try to distinguish labels A from B in [16]. However, LPD seems quite week in this task, because both MVB+ and MVB rarely give a cluster where the number of B samples is larger than that of A samples. In other words, almost all resulting clusters are dominated by A samples. Therefore, D1 dataset is used only for comparison based on lower bounds.

For “Three types of bladder cancer” dataset from [16], referred as D2, the prepared three labels T1/T2+/Ta are used as is.

“Healthy tissues” dataset from [16], referred as D3, has too many true cluster labels (35 labels) for only 103 samples. We faced difficulty in obtaining clustering results worthy to evaluate by precision, recall, and *F*-score. Therefore, D3 dataset is also used only for comparison based on lower bounds.
Implementation

[1] ALL: acute lymphoblastic leukemias, AML: acute myelogenous leukemias, MLL: lymphoblastic leukemias with mixed-lineage leukemia gene translocations [1].

Table 1. Dataset specification.

Dataset name (abbreviation)	# of samples	# of genes
Leukemia (LK) [1]	72	12582
Five types of breast cancer (D1) [16]	286	17816
Three types of bladder cancer (D2) [16]	40	3036
Healthy tissues (D3) [16]	103	10383

4.3 Implementation

We implemented MVB+ and MVB in C language and compile by `gcc -O3`. Inference computations are executed on a PC equipped with Intel Core2 Quad CPU Q9550 @ 2.83 GHz and 8GBytes memory. Digamma and trigamma functions are estimated from asymptotic series [10]. We used the wine dataset [3] to prove the soundness of our implementation by comparing lower bounds with Figure 1 of [15].

As an initialization for both MVB+ and MVB, we choose real random numbers for γ_{dgk} satisfying $\sum_k \gamma_{dgk} = 1$ and then initialize parameters, m_{gk} , l_{gk} , a_{gk} , and b_{gk} , based on randomly initialized γ_{dgk} . Hyperparameter values are set as follows: $\alpha = 1$, $\mu_0 = 0$, $\lambda_0 = 1$, $a_0 = 20$, and $b_0 = 20$. These values are also regarded as initial values for hyperparameter reestimation in MVB+. Observed real values x_{dg} in each microarray data are normalized in the same manner with [15].

The running time of MVB+ and MVB is proportional to the product of the following four numbers: the number of samples, the number of genes, the number of processes, and the number of iterations. For example, in case of dataset D1, MVB+ requires about 174 minutes for 500 iterations when the number of processes is 10. We found that MVB+ increases running time by at most 10% when compared with MVB.

4.4 Results

Figure 1 presents lower bounds obtained for LK (top left panel), D1 (top right), D2 (bottom left), and D3 (bottom right). The horizontal axis shows the number of iterations, and the vertical axis shows the lower bounds. We tested the integers from 2 to 10 as the number of processes. 500 iterations of updates are enough for most cases. However, lower bounds for 1,000 iterations are shown only for LK dataset, because convergence is slow. As Figure 1 shows, lower bounds obtained by MVB+ (solid lines) are larger than that by MVB (dashed lines). We can conclude that lower bound improvement is achieved.

Previous research [15] discusses that we can select an appropriate number of processes by comparing lower bounds obtained for different numbers of processes. Figure 2 shows the average and the standard deviation of 10 lower bounds obtained by MVB+ (solid line) and MVB (dashed line) staring from 10 different initializations for each number of processes ranging from 2 to 10. The horizontal axis shows the number of processes. Lower bounds are recorded when a change no more than 1.0^{-6} can be observed. It seems difficult to find a unique significant peak in all cases. We may need other methods, e.g. a nonparametric Bayesian approach [11], to choose an appropriate number of processes for large datasets.

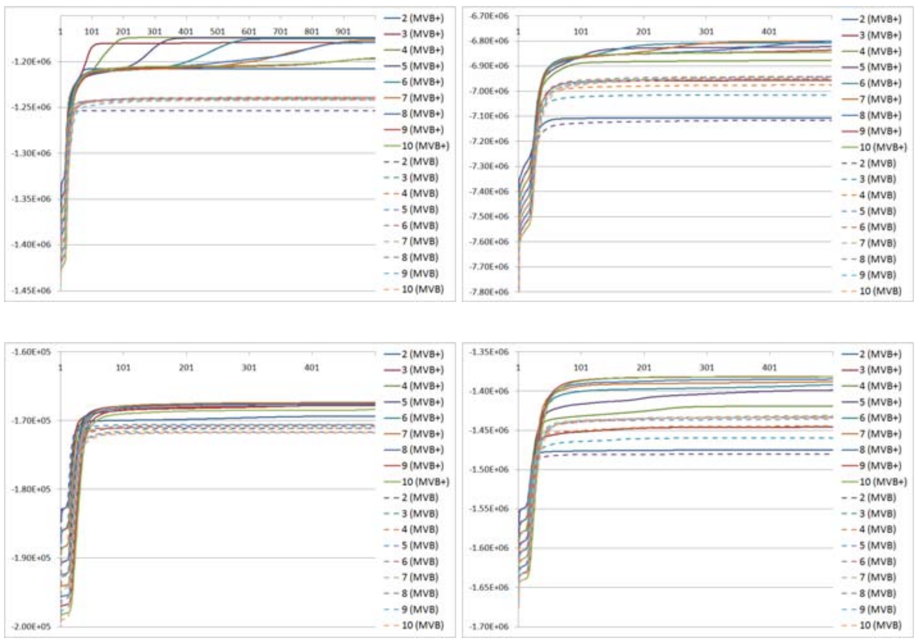

Fig. 1. Lower bounds obtained by MVB+ (solid lines) and MVB (dashed lines). Each graph gives the average of lower bounds obtained by inferences starting from 10 different random initializations. The number of processes ranges from 2 to 10. The datasets are LK (top left panel), D1 (top right), D2 (bottom left), and D3 (bottom left). MVB+ provides better results.

Table 2 provides precisions, recalls, and F-scores averaged over inferences starting from 100 different initializations for LK and D2 datasets. Actually, we discard 10 least frequently occurred clustering results among 100, because extraordinarily good or bad clusters are occasionally obtained for both MVB+ and MVB. We assume that an appropriate number of clusters is chosen beforehand in accordance with the true number of clusters. Namely, we set the number of processes to two and three for LK and D2 datasets, respectively. As Table 2 shows, MVB+ realizes better clustering than MVB. Since 100 clustering results given by MVB for LK dataset are the same, the standard deviation is zero.

Table 2. Precisions, recalls, and F-scores averaged over inferences starting from 100 different initializations. Standard deviations are also presented.

dataset	method	precision	recall	*F*-score
LK	MVB+	0.934±0.007	0.931±0.010	0.932±0.009
	MVB	0.930±0.000	0.924±0.000	0.927±0.000
D2	MVB+	0.837±0.038	0.822±0.032	0.829±0.033
	MVB	0.779±0.084	0.751±0.069	0.763±0.071

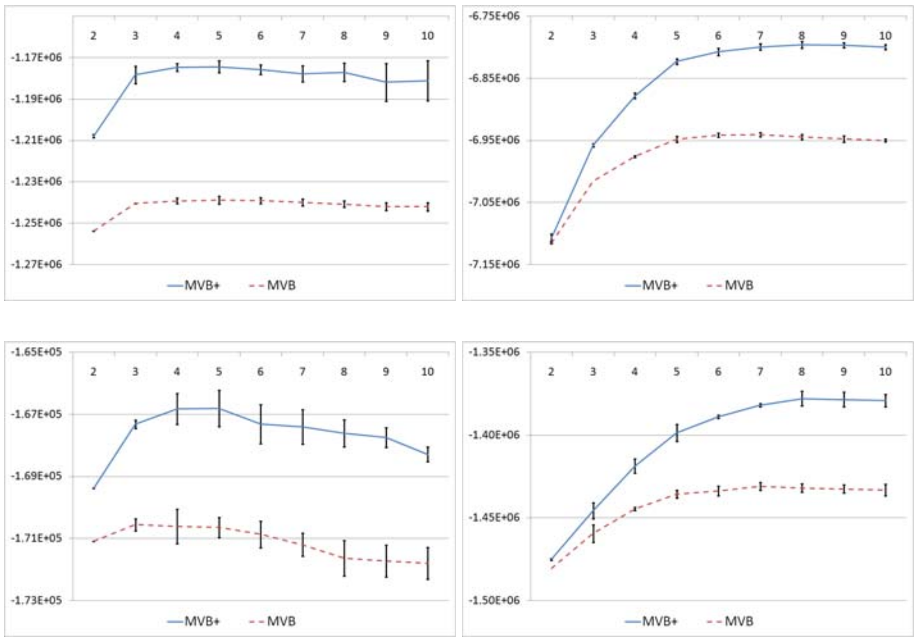

Fig. 2. Averages and standard deviations of lower bounds obtained by MVB+ (solid line) and MVB (dashed line) starting from 10 different random initializations. The datasets are LK (top left panel), D1 (top right), D2 (bottom left), and D3 (bottom left). MVB+ always provides better lower bounds than MVB for all numbers of processes. However, it is difficult to find a clear peak among different numbers of processes.

Our experiments also revealed an important difference between MVB+ and MVB. Figure 3 includes two images visualizing posterior parameters γ_{dgk} for LK dataset when the number of processes is two. These images are constructed as follows. We first select an inference result arbitrarily among a lot of results for each of MVB+ and MVB. Then, we choose the larger one among γ_{dg1} and γ_{dg2} for each pair of gene g and sample d. That is, we assign one among the two processes to each gene/sample pair. The assignments of different processes are shown by different colors. Rows and columns of images in Figure 3 correspond to genes and samples, respectively. Namely, each pixel in these images corresponds to a gene/sample pair.

When we compare the image for MVB+ (left) with that for MVB (right), the former shows row-wise diversity in process assignments. Intuitively speaking, the visualization for MVB+ is more "noisy" than MVB. Therefore, it can be concluded that MVB+ preserves diversity among genes as diversity among process assignments. In contrast, MVB is likely to give almost the same process assignments for all genes. In fact, the rows in the visualization for MVB looks quite similar to each other. While we arbitrarily select an inference result for each of MVB+ and MVB, other results also lead to the same conclusion.

Since we can use the set of γ_{dgk} for a fixed g as a feature vector of gene g in clustering or classifying genes, this difference between MVB+ and MVB will show importance in such applications.

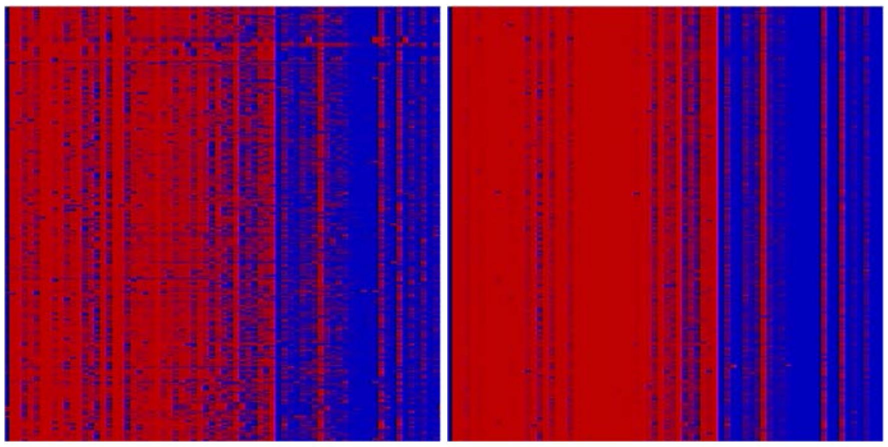

Fig. 3. Visualization of γ_{dgk} for the first 288 genes in LK dataset. The width of images is scaled to 400% for visibility. Each row corresponds to a gene, and each column to a sample. For each gene/sample pair, we choose one process based on γ_{dgk}. Different processes are shown by different colors. MVB+ (left panel) preserves diversity among genes better than MVB (right).

5 Conclusion

In this paper, we provide a hyperparamter reestimation technique *MVB+* for marginalized variational Bayesian inference of LPD. MVB+ achieves further maximization of lower bounds. Also for sample clustering, MVB+ gives better results. Further, MVB+ can preserve diversity among genes as diversity among process assignments.

Our experiments also show that it is difficult to guess an appropriate number of processes based on lower bounds for microarray data of realistically large size. It is a future work to devise a method for determining an appropriate number of processes. Further, with respect to computational efficiency, MVB+ consist of complicated numerical operations. Especially, evaluating digamma and trigamma functions from asymptotic series is time-consuming. Therefore, it is also an important future work to accelerate inferences, e.g. by using GPGPU [7].

Reference

1. Armstrong, S.A., Staunton, J.E., Silverman, L.B., Pieters, R., den Boer, M.L., Minden, M.D., Sallan, S.E., Lander, E.S., Golub, T.R., Korsemeyer, S.J.: MLL Translocations Specify a Distinct Gene Expression Profile that Distinguishes a Unique Leukemia. Nature Genetics 30, 41--47 (2002)
2. Arora, R., Ravindran, B.: Latent Dirichlet Allocation and Singular Value Decomposition Based Multi-document Summarization. In: 25th IEEE International Conference on Data Mining, 713--718 (2008)

3. Blake, C.L., Newman D.J., Hettich S., Merz C.J.: UCI Repository of Machine Learning Databases, http://archive.ics.uci.edu/ml/ (1998)
4. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent Dirichlet Allocation. Journal of Machine Learning Research 3, 993--1022 (2003)
5. Larlus, D., Jurie, F.: Latent Mixture Vocabularies for Object Categorization and Segmentation. Image and Vision Computing 27**,** Issue 5, 523--534 (2009)
6. Lienou, M., Maitre, H., Datcu, M.: Semantic Annotation of Large Satellite Images Using Latent Dirichlet Allocation. In: ESA-EUSC Workshop (2008)
7. Masada, T., Hamada, T., Shibata, Y., Oguri, K.: Accelerating Collapsed Variational Bayesian Inference for Latent Dirichlet Allocation with Nvidia CUDA Compatible Devices. In: 22nd International Conference on Industrial, Engineering & Other Applications of Applied Intelligent Systems, to appear (2009)
8. Minka, T.: Estimating a Dirichlet Distribution. http://research.microsoft.com/en-us/um/people/minka/ (2000)
9. Rogers, S., Girolami, M., Campbell, C., Breitling, R.: The Latent Process Decomposition of cDNA Microarray Datasets. IEEE/ACM Trans. on Computational Biology and Bioinformatics 2, 143--156 (2005)
10. Smith, D.M.: Multiple-Precision Gamma Function and Related Functions. Transactions on Mathematical Software 27, 377--387 (2001)
11. Teh, Y.W., Jordan, M.I., Beal, M.J., Blei, D.M.: Hierarchical Dirichlet Processes. Journal of the American Statistical Association 101:476, 1566--1581 (2006)
12. Teh, Y.W., Newman, D., Welling, M.: A Collapsed Variational Bayesian Inference Algorithm for Latent Dirichlet Allocation. Advances in Neural Information Processing Systems 19, 1378--1385 (2006)
13. Wang, X.G., Grimson, E.: Spatial Latent Dirichlet Allocation. Advances in Neural Information Processing Systems 20, 1577--1584 (2008)
14. Xing, D.S., Girolami, M.: Employing Latent Dirichlet Allocation for Fraud Detection in Telecommunications. Pattern Recognition Letters 28, 1727--1734 (2007)
15. Ying, Y.M., Li, P., Campbell, C.: A Marginalized Variational Bayesian Approach to the Analysis of Array Data. BMC Proceedings 2 (Suppl. 4):S7 (2008)
16. Zinovyev, A., http://www.ihes.fr/~zinovyev/princmanif2006/