Variational Bayes Co-clustering with Auxiliary Information

Motoki Shiga
Dept. of Computer Science and Engineering,
Toyohashi University of Technology
shiga@cs.tut.ac.jp

Hiroshi Mamitsuka
Bioinformatics Center
Kyoto University
mami@kuicr.kyoto-u.ac.jp

ABSTRACT

Clustering is a fundamental technique in data mining to identify essential group structures in a given data matrix. Traditional clustering methods are one-way clustering, which has however limitations for high-dimensional matrices or matrices with missing values. One possible solution is co-clustering, which does clustering both columns and rows simultaneously. Also auxiliary information over columns or rows is helpful to stabilize/improve the performance of clustering. We propose a new co-clustering approach, which can incorporate auxiliary information on both columns and rows. Our approach is based on a probabilistic model, for which we present an efficient method for estimating parameters, based on variational Bayesian learning. Our problem setting can be semi-supervised, by which our approach can be applied to various data mining applications. We evaluated the performance of the proposed approach by using both synthetic and real datasets, confirming the clear advantage of incorporating auxiliary information as well as of our method over two competing methods.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications—*Data mining*; I.5.3 [**Pattern Recognition**]: Clustering—*Algorithms*

General Terms

Algorithms and Experimentation

1. INTRODUCTION

Clustering is one of the most fundamental techniques in data mining to identify essential group structures from a given data matrix. Typical clustering methods are k-means, spectral clustering, and mixture model-based probabilistic learning, all being one-way clustering, which partitions only either of rows (examples) or columns (features) of a given matrix into groups. One-way clustering however does not perform so well if a given matrix is high-dimensional (i.e. the matrix size being biased) or very sparse (i.e. a matrix with many missing values). One possible solution for this problem is two-way clustering or co-clustering, which does clustering both columns and rows of a given matrix simultaneously, typically by matrix factorization [3] or probabilistic model learning [5]. In fact, co-clustering is useful for a lot of applications, including bioinformatics, recommender systems or text mining. A typical example is gene expression analysis, where a given data matrix has expression values of genes (rows) under some conditions (columns) like cancer types or normal. In this case, clustering both genes and conditions (or samples) is helpful to identify genes, which are significantly related with certain cancer types. Another example is recommendation by using a matrix over customers (rows) and items (columns) with purchase records, meaning a binary matrix, in which the element with one indicates that the corresponding customer bought the corresponding item. In this case, many elements in the given matrix are missing, while co-clustering captures groups of both customers and items relatively well even under this setting, which is useful for estimating missing values.

A data matrix is the main input, while in most cases, we can have additional auxiliary information over columns and/or rows, by which clustering performance can be improved. For example, in gene expression analysis, we can have group information over samples. Similarly gene similarity can be measured by gene networks, such as metabolic pathways, protein-protein interactions or gene regulatory networks. Using gene networks or group information over samples is useful for improving the performance of gene clustering. This situation is often the case with recommendation. We can use similarity or group information on customers and/or items. This type of information works for improving clustering customers or items and estimating missing values in the matrix used by recommendation systems. Overall, clustering with this type of auxiliary information is useful and generally so-called semi-supervised clustering, which is now very common, by which a variety of methods have been developed for this setting in the past decade [2].

We propose a new co-clustering approach, which uses the main data matrix as well as auxiliary information for both columns and rows. Fig. 1 is a schematic picture of these inputs of our problem setting. Our approach is based on a probabilistic model for our co-clustering setting with three different inputs, i.e. the main matrix and auxiliary information for columns and rows. We then present an efficient and robust algorithm for estimating probability parameters

MultiClust '13, August 11, 2013, Chicago, Illinois, USA.

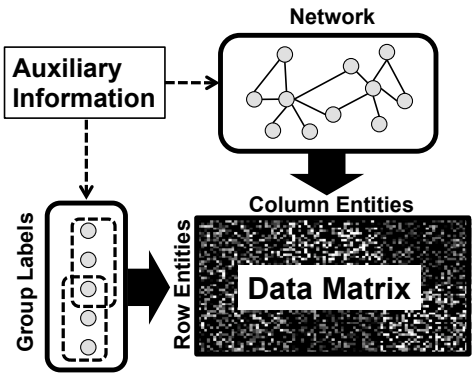

Figure 1: Problem setting: data matrix with auxiliary information

of our model based on variational Bayesian learning.

We have evaluated the performance of the proposed method, by using both synthetic and real datasets, demonstrating the advantage of our approach over two competing methods, Bayesian co-clustering and k-means.

2. METHOD

2.1 Preliminaries and Notations

Let $\boldsymbol{X} \in \mathcal{R}^{M \times N}$ be a data matrix. We assume that auxiliary information is given by a graph (or adjacency matrix) for both columns and rows. So if group labels are given (instead of a graph), we can generate a graph by connecting two items (nodes) in the same group by one edge. We denote an auxiliary graph of columns by $\boldsymbol{W}^{(z)}$ and that of rows by $\boldsymbol{W}^{(h)}$. Let K and L be the number of clusters for columns and rows, respectively, where K and L are assumed to be given. Let z_n be a latent variable (or a cluster label) of column n and h_m be that of row m, where $\boldsymbol{z} = (z_1, \ldots, z_N)$ and $\boldsymbol{h} = (h_1, \ldots, h_M)$. We define a Gaussian distribution as $\mathcal{N}(x|\mu, \sigma^2) := (2\pi\sigma^2)^{-\frac{1}{2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right)$, a Gamma distribution as $\mathcal{G}(x|\alpha, \beta) := \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} e^{-\beta x}$, and a student's t-distribution as $\mathcal{T}(x|\mu, \lambda, \nu) := \frac{\Gamma(\nu/2+1/2)}{\Gamma(\nu/2)} \left(\frac{\lambda}{\pi\nu}\right)^{\frac{1}{2}} \left\{1 + \frac{\lambda(x-\mu)^2}{\nu}\right\}^{-\nu/2-1/2}$, where $\Gamma(\cdot)$ is the gamma function given by $\Gamma(x) := \int_0^\infty t^{x-1} e^{-t} dt$. Let $\psi(\cdot) := \frac{d}{dx} \log \Gamma(x)$ be the digamma function and $\delta(x, y)$ be the delta function, which gives one if $x = y$; otherwise zero.

2.2 Probabilistic Model for Data Matrix with Auxiliary Information

We first assume that the joint probability of data matrix $\boldsymbol{X}$, latent variables $\boldsymbol{z}$ and $\boldsymbol{h}$, and parameter $\boldsymbol{\theta}$, can be written as follows:

$$\begin{aligned} &p(\boldsymbol{X}, \boldsymbol{z}, \boldsymbol{h}, \boldsymbol{\theta}|\boldsymbol{\theta}_0) \\ &\quad := p(\boldsymbol{X}|\boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{h})\ p(\boldsymbol{z}|\boldsymbol{w}^{(z)})\ p(\boldsymbol{h}|\boldsymbol{w}^{(h)})\ p(\boldsymbol{\theta}|\boldsymbol{\theta}_0), \end{aligned}$$

where $\boldsymbol{\theta}_0$ is a vector of hyper-parameters. We assume that the conditional probability of data matrix $\boldsymbol{X}$, given parameter $\boldsymbol{\theta}$, latent values $\boldsymbol{z}$ and $\boldsymbol{h}$ is given as a Gaussian distribution, as follows:

$$\begin{aligned} p(\boldsymbol{X}|\boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{h}) &:= \prod_{m=1}^{M} \prod_{n=1}^{N} p\Big(X_{mn}|\boldsymbol{\theta}_{h_m z_n}\Big) \\ p\Big(X|\boldsymbol{\theta}_{lk}\Big) &:= \mathcal{N}\Big(X\Big|\mu_{lk}, (s_{lk})^{-1}\Big) \end{aligned}$$

where $\theta_{(l,k)} := (\mu_{lk}, s_{lk})$. The idea behind this model is to make items in a co-cluster have the same value.

We further assume that the prior distributions of latent variables for columns and rows are given, with auxiliary information, as follows:

$$\begin{aligned} p(\boldsymbol{z}|\boldsymbol{w}^{(z)}) := \frac{1}{C_z} \exp\Bigg\{ &\sum_{i=1}^{N} \sum_{j=1}^{N} w_A^{(z)} \cdot W_{ij}^{(z)} \delta(z_i, z_j) \\ &+ \sum_{k=1}^{K} w_k^{(z)} \sum_{n=1}^{N} \delta(z_n, k) \Bigg\} \\ p(\boldsymbol{h}|\boldsymbol{w}^{(h)}) := \frac{1}{C_h} \exp\Bigg\{ &\sum_{i=1}^{M} \sum_{j=1}^{M} w_A^{(h)} \cdot W_{ij}^{(h)} \delta(h_i, h_j) \\ &+ \sum_{l=1}^{L} w_l^{(h)} \sum_{m=1}^{M} \delta(h_m, l) \Bigg\} \end{aligned}$$

where C_z and C_h are coefficients to normalize so that the value of the integral is one, and $\boldsymbol{w}^{(z)}$ and $\boldsymbol{w}^{(h)}$ are hyper-parameters. The $p(\boldsymbol{z}|\boldsymbol{w}^{(z)})$ and $p(\boldsymbol{h}|\boldsymbol{w}^{(h)})$ are introduced by Markov random fields or Boltzmann machine. That is, these probabilities represent graph structures. These probabilities can be larger if latent labels $\boldsymbol{z}$ or $\boldsymbol{h}$ are shared by connected neighborhoods more in auxiliary graphs $\boldsymbol{W}^{(z)}$ or $\boldsymbol{W}^{(h)}$, respectively. This means that auxiliary information are incorporated into co-clustering through these probabilistic distributions. The weights of auxiliary graphs can be adjusted by hyperparameters $w_A^{(z)}$ and $w_A^{(h)}$.

We assume the prior distribution of parameter $\boldsymbol{\theta}$ as conjugate prior distributions, as follows:

$$\begin{aligned} p(\boldsymbol{\theta}|\boldsymbol{\theta}_0) &:= \prod_{l=1}^{L} \prod_{k=1}^{K} p\Big(\boldsymbol{\theta}_{lk}\Big|\boldsymbol{\theta}_0\Big) \\ p\big(\boldsymbol{\theta}_{lk}\big|\boldsymbol{\theta}_0\big) &:= \mathcal{N}\Big(\mu_{lk}|\mu_0, (\xi_0 s_{lk})^{-1}\Big)\ \mathcal{G}(s_{lk}|\alpha_0, \beta_0) \end{aligned}$$

2.3 Co-clustering based on Variational Bayes Learning

The true posterior distribution of parameters $p(\boldsymbol{z}, \boldsymbol{h}, \boldsymbol{\theta}|\boldsymbol{X}, \boldsymbol{\theta}_0)$ is computationally intractable by Bayesian learning in our setting. We thus apply a variational Bayesian (VB) approach [1], i.e. an approximation of Bayesian learning. In the VB approach, we first assume that the posterior distributions of parameters are independent of each other as follows:

$$q(\boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{h}) := \left\{\prod_{l=1}^{L} \prod_{k=1}^{K} q(\boldsymbol{\theta}_{lk})\right\} \left\{\prod_{n=1}^{N} q(z_n)\right\} \left\{\prod_{m=1}^{M} q(h_m)\right\}$$

where $q(\cdot)$ means a VB posterior distribution. The VB posterior distribution is then optimized from given data by minimizing Kullback-Leibler (KL) divergence between the true posterior distribution and the VB posterior distribution. This minimization is equivalent to minimizing the upper bound of the Bayesian free energy given by Jensen's

inequality as follows:

$$\begin{aligned} F_t(\boldsymbol{X}|\boldsymbol{\theta}_0) &:= -\log \sum_{\boldsymbol{z}} \sum_{\boldsymbol{h}} \int_{\boldsymbol{\theta}} p(\boldsymbol{X}, \boldsymbol{z}, \boldsymbol{h}, \boldsymbol{\theta}|, \boldsymbol{\theta}_0)\ d\boldsymbol{\theta} \\ &\leq -\sum_{\boldsymbol{z}} \sum_{\boldsymbol{h}} \int_{\boldsymbol{\theta}} q(\boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{h}) \log \frac{p(\boldsymbol{X}, \boldsymbol{z}, \boldsymbol{h}, \boldsymbol{\theta}|\boldsymbol{\theta}_0)}{q(\boldsymbol{\theta}, \boldsymbol{z}, \boldsymbol{h})} d\boldsymbol{\theta} \\ &:= F[q] \end{aligned}$$

$F[q]$ is called variational Bayes free energy.

Here, for simplicity, we denote the expectations over latent variables $\boldsymbol{z}$ and $\boldsymbol{h}$ by: $\overline{z_n^k} := q(z_n = k)$, $\overline{h_m^l} := q(h_m = l)$, $\overline{H_l} := \sum_m \overline{h_m^l}$, $\overline{Z_k} := \sum_n \overline{z_n^k}$, $\overline{X_{lk}} := \sum_m \sum_n \overline{h_m^l z_n^k} X_{mn}$, and $\overline{X_{lk}^2} := \sum_m \sum_n \overline{h_m^l z_n^k} X_{mn}^2$. We can then derive the VB posterior distribution of μ_{lk} and s_{lk} with the variation of $F[q]$ with respect to $q(\mu_{lk}, s_{lk})$, as follows:

$$\begin{aligned} q\big(\mu_{lk}\big) &= \mathcal{T}\Big(\mu_{lk}\Big|\bar{\mu}_{lk}, \frac{\alpha_{lk}}{\beta_{lk}}\xi_{lk}, \alpha_{lk}\Big) \\ q\big(s_{lk}\big) &= \mathcal{G}\Big(s_{lk}\Big|\alpha_{lk}, \beta_{lk}\Big), \end{aligned}$$

where

$$\begin{aligned} \alpha_{lk} &= \alpha_0 + \frac{1}{2}\overline{H_l Z_k} \\ \beta_{lk} &= \beta_0 + \frac{1}{2}\Big\{\xi_0\mu_0^2 + \overline{X_{lk}^2} - \xi_{lk}\bar{\mu}_{lk}^2\Big\} \\ \bar{\mu}_{lk} &= \frac{\xi_0\mu_0 + \overline{X_{lk}}}{\xi_{lk}} \\ \xi_{lk} &= \xi_0 + \overline{H_l Z_k} \end{aligned}$$

On the other hand, the posterior distribution of z_n can be given by using the variation of $F[q]$ with respect to $q(z_n)$, as follows:

$$q\big(z_n = k\big) = \frac{\exp\big(\gamma_{nk}\big)}{\sum_{k=1}^K \exp\big(\gamma_{nk}\big)},$$

where

$$\begin{aligned} \gamma_{nk} &= \sum_{j=1}^N w_A^{(z)} W_{nj}^{(z)} \overline{z_j^k} + w_k^{(z)} - \frac{1}{2}\sum_{l=1}^L \overline{H_l}\frac{\alpha_{lk}}{\xi_{lk}(\alpha_{lk}-1)} \\ &\quad -\frac{1}{2}\sum_{l=1}^L \frac{\alpha_{lk}}{\beta_{lk}} \sum_{m=1}^M \overline{h_m^l}\big(\bar{\mu}_{lk} - X_{mn}\big)^2 \\ &\quad +\frac{1}{2}\sum_{l=1}^L \overline{H_l}\Big\{\psi(\alpha_{lk}) - \log\beta_{lk}\Big\}. \end{aligned}$$

Similarly, the posterior distribution of h_m is given by

$$q\big(h_m = l\big) = \frac{\exp\big(\eta_{ml}\big)}{\sum_{k=1}^L \exp\big(\eta_{ml}\big)},$$

where

$$\begin{aligned} \eta_{ml} &= \sum_{j=1}^M w_A^{(h)} W_{mj}^{(h)} \overline{h_j^l} + w_l^{(h)} - \frac{1}{2}\sum_{k=1}^K \overline{Z_k}\frac{\alpha_{lk}}{\xi_{lk}(\alpha_{lk}-1)} \\ &\quad -\frac{1}{2}\sum_{k=1}^K \frac{\alpha_{lk}}{\beta_{lk}} \sum_{n=1}^N \overline{z_n^k}\big(\bar{\mu}_{lk} - X_{mn}\big)^2 \\ &\quad +\frac{1}{2}\sum_{k=1}^K \overline{Z_k}\Big\{\psi(\alpha_{lk}) - \log\beta_{lk}\Big\}. \end{aligned}$$

Fig. 2 shows a pseudocode of our co-clustering algorithm, which we call *VBCA*, standing for Variational Bayes learning for Co-clustering with Auxiliary information. Finally in prediction, for column n and row m, we can assign cluster labels $\hat{z}_n$ and $\hat{h}_m$ by using the posterior distributions, which are optimized by VBCA, as follows: $\hat{z}_n \leftarrow \arg\max_k q(z_n = k)$ and $\hat{h}_m \leftarrow \arg\max_l q(h_m = l)$.

Input : $\boldsymbol{X}$, $\boldsymbol{W}^{(z)}$, $\boldsymbol{W}^{(h)}$, K, L, $\boldsymbol{\theta}_0$
Output : $q(\boldsymbol{z})$, $q(\boldsymbol{h})$, $q(\boldsymbol{\theta})$

VBCA$(\boldsymbol{X}, \boldsymbol{W}^{(z)}, \boldsymbol{W}^{(h)}, K, L, \boldsymbol{\theta}_0)$
1: Initialize $q(\boldsymbol{z})$ and $q(\boldsymbol{h})$
2: **while** $F[q]$ is not converged **do**
3: VB-M step: Update $q(\boldsymbol{\mu})$ and $q(\boldsymbol{s})$
4: VB-E step: Update $q(\boldsymbol{z})$ and $q(\boldsymbol{h})$
5: Calculate $F[q]$
6: **end while**

Figure 2: Pseudocode of VBCA

3. EXPERIMENTS

We used the following parameter setting throughout our experiments: $w_k^{(z)} = w_l^{(h)} = 1$, $\alpha_0 = 2$, $\beta_0 = 1$, $\xi_0 = 1$, $\mu_0 = 0$ and $w_A^{(z)} = w_A^{(h)}$. We used the true number of clusters as an input. We evaluated the performance of VBCA, comparing with Bayesian Co-Clustering (BCC) [5] and k-means, by using normalized mutual information (NMI) [6] between estimated clusters $\mathcal{C}_E$ and true clusters $\mathcal{C}_T$, as follows: NMI $:= \frac{\mathrm{MI}(\mathcal{C}_E, \mathcal{C}_T)}{\sqrt{\mathrm{H}(\mathcal{C}_E)}\sqrt{\mathrm{H}(\mathcal{C}_T)}}$, where $\mathrm{MI}(\mathcal{C}_E, \mathcal{C}_T)$ is mutual information, $\mathrm{H}(\mathcal{C})$ is entropy, and $\mathrm{H}(\mathcal{C}_E, \mathcal{C}_T)$ is joint entropy.

3.1 Synthetic Data

Data Setting: we fixed $K = 3$, $L = 2$ and the number of columns and rows per cluster at 50 and 20, respectively. That is, we set $z_n^* = k$ $(50(k-1)+1 \leq n \leq 50k)$ for column cluster k, and $h_m^* = k$ $(20(l-1)+1 \leq m \leq 20l)$ for row cluster l. Under these true clusters, we generated element (m, n) of data matrix X_{mn} according to Gaussian distribution $\mathcal{N}(\mu_{lk}, \sigma^2)$, where $l = h_m^*$, $k = z_n^*$ and $\boldsymbol{\mu}$ were fixed at $\mu_{11} = 1$, $\mu_{12} = 0$, $\mu_{13} = 1$, $\mu_{21} = 0$, $\mu_{22} = 0$ and $\mu_{23} = 0.5$, while different values were examined for σ^2. We generated auxiliary networks by using three parameters, R_{in}: the ratio of intra-cluster edges in a cluster, R_{out}: the ratio of inter-cluster edges in a cluster, and R_s: the ratio of labeled nodes to all nodes. First, for columns, i.e. $W^{(z)}$, we randomly generated $3 \times 50 \times 49 \times R_{in}$ intra-cluster edges and $2 \times 50^2 \times R_{out}$ inter-cluster edges. Similarly for rows, i.e. $W^{(h)}$, $2 \times 20 \times 19 \times R_{in}$ intra-cluster edges and $20^2 \times R_{out}$ inter-cluster edges were generated. If R_{out} is large (comparing with R_{in}), clusters cannot be separated well, meaning that the auxiliary network can be like noise. So we examined different values of R_{out}, fixing R_{in} at 0.3. To realize a semi-supervised setting, once after we generated auxiliary networks, we randomly discarded all edges connecting to $50 \times (1 - Rs)$ nodes in $W^{(z)}$ and $20 \times (1 - Rs)$ nodes in $W^{(h)}$. We used three types of parameter settings of (R_{out}, R_s, σ^2): 1) $(0.1, 1.0, 4)$, 2) $(0.15, 1.0, 3)$ and 3) $(0.1, 0.75, 4)$, where twenty datasets were generated randomly for each set.

Results: Fig. 3 shows the average NMI of the competing methods over 20 datasets for each parameter setting.

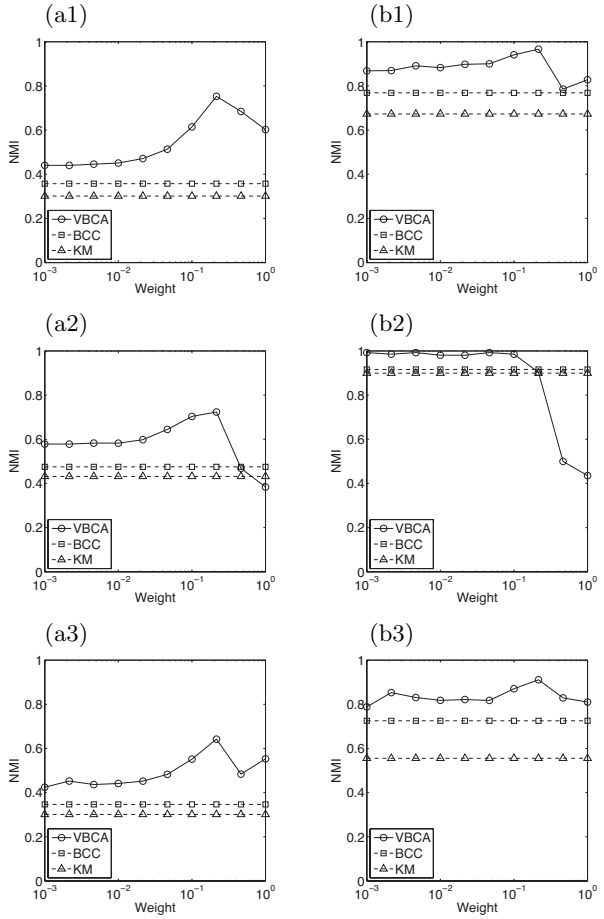

Figure 3: Average NMI on synthetic data: (a1), (a2) and (a3) for columns and (b1), (b2) and (b3) for rows, under the parameter settings of 1), 2) and 3), respectively.

The horizontal axis shows the weight for auxiliary networks. From this figure, we can see that for almost all values of the weight for auxiliary networks, the average NMI of VBCA was higher than those of BCC and k-means for both columns and rows, particularly the highest NMI of VBCA being clearly higher than competing methods. This demonstrates that using auxiliary information was advantageous as well as that VBCA could incorporate this information well enough. (We note that larger weight values make $q(\boldsymbol{z})$ and $q(\boldsymbol{h})$ focus on single clusters only, by which the NMI of VBCA becomes worse for larger weight values.)

3.2 Real Data

Data Setting: We used a gene expression dataset in [4] which has 42 human (tumor tissue) samples, consisting of 10 medulloblastomas, 10 rhabdoid, 10 malignant glioma, 8 supratentorial PNETS and 4 normal cerebella, resulting in $L = 4$. We first selected 2,128 genes, by 1) first filtering out genes with small absolute values and small variances over all samples and 2) then discarding genes that were differentially expressed between diseased tissue samples and normal samples, by using t-test with p-value of 0.01. We then further chose genes to be used in our experiments, with generating gold standard clusters by using Gene Ontology (GO), which is standard gene categories. That is, we randomly chose K GO terms, where each term has 50 to 100 genes, and used genes in these K true clusters. We examined three values (3, 5 and 10) for K. We then generated auxiliary networks (twenty times for each value of K) by randomly choosing pairs in the same cluster as edges for samples and genes, limiting to only $0.75 \times M$ samples and $0.75 \times N$ genes, because of implementing a semi-supervised setting.

Table 1: Average NMI for real gene expression data

K		VBCA	BCC	k-means
3	Gene	**0.14** $\pm$ 0.17	0.01 $\pm$ 0.01	0.01 $\pm$ 0.01
	Sample	**0.69** $\pm$ 0.08	0.28 $\pm$ 0.20	0.55 $\pm$ 0.08
5	Gene	**0.21** $\pm$ 0.15	0.02 $\pm$ 0.01	0.02 $\pm$ 0.01
	Sample	**0.62** $\pm$ 0.15	0.36 $\pm$ 0.19	0.56 $\pm$ 0.07
10	Gene	**0.29** $\pm$ 0.11	0.04 $\pm$ 0.01	0.04 $\pm$ 0.01
	Sample	**0.58** $\pm$ 0.15	0.45 $\pm$ 0.07	0.56 $\pm$ 0.06

Results: Table 1 shows the average NMI over twenty runs of three competing methods (VBCA is with $w_A^{(z)} = w_A^{(h)} = 10^2$). This table shows that VBCA clearly outperformed the other two methods for all settings, indicating that VBCA could incorporate auxiliary networks well enough to improve the performance.

4. CONCLUDING REMARKS

We have proposed a new and efficient method, VBCA, for co-clustering with auxiliary information. VBCA is based on a probabilistic model, for which probability parameters are estimated by using variation Bayes learning. Our experimental results with synthetic and real datasets confirmed the clear performance advantage of using auxiliary information and of VBCA against two competing methods. Possible future work is to extend our approach to be applied to a high-dimensional data array, such as a tensor.

5. ACKNOWLEDGMENTS

This work is partially supported by Okawa Foundation, JSPS KAKENHI 25870322 and 24300054, and Collaborative Research Program of Institute for Chemical Research, Kyoto University (grant #2012-24 and #2013-19).

6. REFERENCES

[1] C. M. Bishop. Approximate inference. In *Pattern Recognition and Machine Learning*, chapter 10, pages 461–522. Springer, 2006.

[2] O. Chapelle, B. Schölkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, Cambridge, MA, 2006.

[3] C. Ding, T. Li, W. Peng, and H. Park. Orthogonal nonnegative matrix t-factorizations for clustering. In *KDD*, pages 126–135, New York, NY, USA, 2006.

[4] S. Pomeroy et al. Prediction of central nervous system embryonal tumour outcome based on gene expression. *Nature*, 415(6870):436–42, January 2002.

[5] H. Shan and A. Banerjee. Bayesian co-clustering. In *ICDM*, pages 530–539, Washington, DC, USA, 2008.

[6] A. Strehl and J. Ghosh. Relationship-based clustering and visualization for high-dimensional data mining. *INFORMS Journal on Computing*, 15(2):208–230, 2003.