Global and Componentwise Extrapolation for Accelerating Data Mining from Large Incomplete Data Sets with the EM Algorithm

Chun-Nan Hsu Han-Shen Huang Bo-Hou Yang
Institute of Information Science
Academia Sinica
Nankang, Taipei, Taiwan
{chunnan,hanshen,ericyang}@iis.sinica.edu.tw

Abstract

The Expectation-Maximization (EM) algorithm is one of the most popular algorithms for data mining from incomplete data. However, when applied to large data sets with a large proportion of missing data, the EM algorithm may converge slowly. The triple jump extrapolation method can effectively accelerate the EM algorithm by substantially reducing the number of iterations required for EM to converge. There are two options for the triple jump method, global extrapolation (TJEM) and componentwise extrapolation (CTJEM). We tried these two methods for a variety of probabilistic models and found that in general, global extraplolation yields a better performance, but there are cases where componentwise extrapolation yields very high speedup. In this paper, we investigate when componentwise extrapolation should be preferred. We conclude that, when the Jacobian of the EM mapping is diagonal or block diagonal, CTJEM should be preferred. We show how to determine whether a Jacobian is diagonal or block diagonal and experimentally confirm our claim. In particular, we show that CTJEM is especially effective for the semi-supervised Bayesian classifier model given a highly sparse data set.

1. Introduction

The Expectation-Maximization (EM) algorithm [4] is one of the most popular algorithms for data mining from incomplete data. Given an incomplete data set, the EM algorithm iteratively searches for the best parameter vector θ^* that maximizes the log-likelihood of the data. However, when applied to large data sets with a large number of parameters to estimate, the EM algorithm may converge slowly. If the data sets also contain a large proportion of missing data or there are a large number of hidden variables in the model, the convergence of EM can be even slower.

Aitken's acceleration is one of the most commonly used method to speed up fixed-point iteration methods [2]. Since the EM algorithm can be considered as a fixed-point iteration method, we can apply Aitken's acceleration to accelerate the EM algorithm [9, 10].

However, the multivariate version of Aitken's acceleration requires to compute or approximate the Jacobian of the EM mapping matrix, which can be intractable. Many variants of Aitken's acceleration have been proposed to approximate Aitken's acceleration as an extrapolation method. One of the methods is the triple jump extrapolation method (TJEM) [7, 5, 15]. The idea is to estimate the extrapolation rate by considering the previous two estimates of the parameter vectors. The triple jump extrapolation method can effectively accelerate the EM algorithm by substantially reducing the number of iterations required for the EM algorithm to converge. Another benefit of the triple jump method is that it can be easily integrated with existing EM packages for any probabilistic model. We can even integrate the triple jump method with other extrapolation-based acceleration methods, such as the parameterized EM (pEM) [1] and the adaptive overrelaxed EM (aEM) [14], to further accelerate the convergence [6].

The triple jump method can extrapolate the parameter vector with one extrapolation rates for different dimensions. We refer to the former approach as *global extrapolation* and the latter as *componentwise extrapolation*. The componentwise extrapolation of the EM algorithm is referred to as the componentwise triple jump EM algorithm (CTJEM). Hesterberg [5] proposed a global extrapolation method, while Huang et al. [7] described a componentwise extrapolaion method, though in that method many dimensions can be extrapolated together as a *sub-vector*. We tried these two methods for a variety of probabilistic models with synthesized data and found that in general, global extraplolation yields a better performance, but there are cases where componentwise extrapolation yields very high speedup. In some cases, one triple jump can reach the local maximum.

We investigate when componentwise extrapolation should be preferred. We conclude that, when the Jacobian of the EM mapping is diagonal or block diagonal, CTJEM should be preferred. Previously, Schafer [16] also suggested the same, but he did not formally justify this claim and how to determine when the rates are different. In this paper, we demonstrate how to determine whether a Jacobian is diagonal or block diagonal and experimentally confirm our claim.

2. Aitken's Acceleration for EM

Suppose we want to use the EM algorithm to build a probabilistic model with a l-dimensional parameter vector θ from an incomplete data set $\mathcal{D} = (\mathcal{D}_{obs}, \mathcal{D}_{mis})$, where $\mathcal{D}_{obs}$ denoted the observed values and $\mathcal{D}_{mis}$ denotes the missing values. Now let $d = [y_1, y_2, \cdots, y_n]^T$ be the data set with all missing values in $\mathcal{D}$ imputed (i.e., filled in by some estimation method.) and $f(d|\theta)$ be the probability density of d given θ, then

$$L_d(\theta) = \log f(\mathcal{D} = d|\theta) \tag{1}$$

is the log likelihood of d while

$$\tilde{L}_d(\theta) = \log f_{d_{obs}}(\mathcal{D}_{obs} = d_{obs}|\theta) = \tag{2}$$

$$\log \int f(\mathcal{D} = d|\theta) d\mathcal{D}_{mis} \tag{3}$$

is the log likelihood of the observed data d_{obs}. The maximum likelihood principle states that the best parameter vector is the one that maximizes the log likelihood of the observed data. However, it is usually difficult to derive a closed-form solution for the integral for the observed data likelihood with complex probabilistic model. The EM algorithm solves this problem by iteratively imputing the missing data and searching for θ^* that maximizes the expected complete data likelihood.

Let θ be in the space Ω and $\theta^{(t)} \in \Omega$ be the result of the t-th EM iteration, $t = 0, 1, 2, \ldots$. Then the EM algorithm defines a mapping $M : \Omega \to \Omega$ such that $\theta^{(t+1)} = M(\theta^{(t)})$. If M is continuous and $\theta^{(t)}$ converges to a local optimum θ^*, then $\theta^* = M(\theta^*)$. Therefore, the EM algorithm is equivalent to solving θ^* by the *fixed-point iteration* [2]. The multivariate version of Aitken's acceleration can be derived as follows [10]. Suppose that when $t \to \infty$, $\theta^t \to \theta^*$. Then we can express θ^* as

$$\theta^* = \theta^{(t)} + \sum_{h=1}^{\infty} (\theta^{(t+h)} - \theta^{(t+h-1)}). \tag{4}$$

By applying a linear Taylor expansion of $M(\theta^{(t+h-1)})$ around $\theta^{(t+h-2)}$, we have

$$\begin{aligned} & \theta^{(t+h)} - \theta^{(t+h-1)} \\ \approx\ & J(\theta^*)(\theta^{(t+h-1)} - \theta^{(t+h-2)}), \end{aligned} \tag{5}$$

where J is the Jacobian matrix of M. Note that $J(\theta^{(t+h-2)})$ can be approximated by $J(\theta^*)$ near the convergence point. Repeatedly applying (5) in (4) gives

$$\begin{aligned} \theta^* &\approx \theta^{(t)} + \sum_{h=1}^{\infty} J(\theta^*)^h (\theta^{(t)} - \theta^{(t-1)}) \\ &= \theta^{(t)} + (I - J(\theta^*))^{-1}(\theta^{(t)}_{EM} - \theta^{(t)}), \end{aligned} \tag{6}$$

when all eig$(J(\theta^*))$ are between 0 and 1. In Equation 6, we replace $\theta^{(t+1)}$ with $\theta^{(t)}_{EM}$ to emphasize that $\theta^{(t+1)}$ is obtained by applying an EM mapping to $\theta^{(t)}$ here.

The multivariate version of Aitken's acceleration requires to compute or approximate the Jacobian of the EM mapping matrix. From [4], we know that the Jacobian of the EM algorithm is given by

$$J = I - \mathcal{I}_{obs}\mathcal{I}_c^{-1}\ , \tag{7}$$

where I is the $l \times l$ identity matrix,

$$\mathcal{I}_c = E\left[-\frac{\partial^2[\log f(\mathcal{D}|\theta)]}{\partial\theta\partial\theta^T} \middle| \mathcal{D}_{obs}, \theta \right]\Bigg|_{\theta=\theta^*}$$

is the Fisher's information of the expected complete data,

$$\mathcal{I}_{obs} = -\frac{\partial^2 \tilde{L}_d(\theta)}{\partial\theta\partial\theta^T}\Bigg|_{\theta=\theta^*}$$

is the Fisher's information of the observed data. Fisher's information measures how flat the likelihood surface is. Computing Fisher's information can be intractable for complex models with a high dimensional parameter space.

In addition to the complexity of computing the Jacobian matrix, Aitken's acceleration also has the drawbacks including that it may not always converge and may be numerically unstable [8].

3. Triple Jump Extrapolation

The triple jump extrapolation method approximates the largest eigenvalue of the Jacobian matrix. The eigen-decomposition of J is

$$J(\theta^*) = Q\text{diag}(\lambda_1, \ldots, \lambda_n)Q^{-1} = Q\Lambda Q^{-1},$$

where columns of Q are the eigenvectors. Therefore,

$$\begin{aligned} (I - J(\theta^*))^{-1} &= \left[Q\left[I - \Lambda\right]Q^{-1}\right]^{-1} \\ &= Q\text{diag}(\frac{1}{1-\lambda_1}, \ldots, \frac{1}{1-\lambda_n})Q^{-1}. \end{aligned}$$

Since

$$\begin{aligned} \theta^* &\approx \theta^{(t)} + (I - J(\theta^*))^{-1}(\theta^{(t)}_{EM} - \theta^{(t)}) \\ &= Q\left\{Q^{-1}\theta^{(t)} + \left[I - \Lambda\right]^{-1} Q^{-1}(\theta^{(t)}_{EM} - \theta^{(t)})\right\}, \\ Q^{-1}\theta^* &\approx Q^{-1}\theta^{(t)} + \left[I - \Lambda\right]^{-1} Q^{-1}(\theta^{(t)}_{EM} - \theta^{(t)}) \\ \theta^{*e} &\approx \theta^{(t)e} + \left[I - \Lambda\right]^{-1}(\theta^{(t)e}_{EM} - \theta^{(t)e}). \end{aligned}$$

The superscript e in θ^e denotes that it is a transformed parameter vector in the eigenspace. We can derive the Aitken's acceleration along the direction of the i-th dimension as

$$\theta_i^{*e} \approx \theta_i^{(t)e} + \frac{1}{1-\lambda_i}(\theta_{EMi}^{(t)e} - \theta_i^{(t)e}). \tag{8}$$

Let $\varphi^{(t)} = \theta^{(t)} - \theta^*$ denote the difference between current estimated parameter vector to the local maximum. The global rate of convergence of the EM algorithm is defined as the ratio:

$$R = \lim_{t\to\infty} R^{(t)} \equiv \lim_{t\to\infty} \frac{\|\varphi^{(t+1)}\|}{\|\varphi^{(t)}\|} \tag{9}$$

Dempster et al. [4] have shown that $R = \lambda_{max}$, the largest eigenvalue of J. Thus, instead of computing the Jacobian, we can simplify Aitken's acceleration for EM by replacing every eigenvalue λ_i with a single value $\gamma^{(t)}$ such that $\gamma^{(t)}$ is an approximation of λ_{max} at the t-th iteration. That is,

$$\theta^{(t+1)} = \theta^{(t)} + (1-\gamma^{(t)})^{-1}(\theta_{EM}^{(t)} - \theta^{(t)}). \tag{10}$$

We can estimate $\gamma^{(t)}$ as follows. From Equation (5),

$$J(\theta^*)(\theta^{(t)} - \theta^{(t-1)}) \approx \theta_{EM}^t - \theta^{(t)}.$$

Suppose $\gamma^{(t)}$ is an exact approximation of $J(\theta^*)$, then

$$\gamma^{(t)}(\theta^{(t)} - \theta^{(t-1)}) = \theta_{EM}^t - \theta^{(t)}.$$

Since eig$(J(\theta^*))$ is greater than or equal to zero [4], $\theta_{EMi}^{(t)e} - \theta_i^{(t)e}$ has the same direction as $\theta_i^{(t)e} - \theta_i^{(t-1)e}$. To ensure that our extrapolation for each $\theta_i^{(t+1)e}$ is along the same direction as Aitken's acceleration, we need $\gamma^{(t)} \geq 0$ and our estimation of $\gamma^{(t)}$ is thus defined by:

$$\gamma^{(t)} \equiv \frac{\|\theta_{EM}^t - \theta^{(t)}\|}{\|\theta^{(t)} - \theta^{(t-1)}\|}. \tag{11}$$

To obtain $\theta^{(t+1)}$ by (10) and (11), we need to apply the EM algorithm to obtain $\theta^{(t)}$ from $\theta^{(t-1)}$ and apply again to obtain $\theta_{EM}^{(t)}$. Because this is similar to the hop, step and jump phases in triple jump, Huang et al. [7] named this method the *Triple Jump Acceleration.*

In the case that the parameter space has only one dimension, Equation (11) provides an exact approximation of $J(\theta^*) = M'(\theta^*)$. When the convergence is slow, we will have $M' \approx 1$ and $\gamma^{(t)} \approx 1$, too. In that case, $1/(1-\gamma^{(t)})$ will be very large and provide a large acceleration. In a multi-dimensional case, the convergence rate is determined by the largest eig$(J(\theta^*))$. When the eigenvalue is close to one, the convergence will be slow. Aitken's acceleration can provide a large acceleration when we have a good approximation of λ_{max} but may also cause numerical insatiability when $\lambda_{max} \approx 1$. Since $\gamma^{(t)} \leq \lambda_{max}$, the triple jump extrapolation is numerically more stable than directly using the eigenvalues.

The Aitken's acceleration does not guarantee to reach θ^* directly from $\theta^{(t)}$ because it is based on the assumption that $\theta^{(t)}$ is within the neighborhood of θ^*. When $\theta^{(t)}$ is not close enough to θ^*, the extrapolation jumps to a $\theta^{(t+1)}$ that might fail to improve the likelihood. Salakhutdinov et al. [14] showed that with the extrapolation ratio within a certain interval, EM with extrapolation is guaranteed to converge. However, since the ratio in such an interval is too small, the speedup will not be significant. Therefore, they proposed another method called *adaptive overrelaxed EM* (aEM), which switches back to vanilla EM during the search if the new data likelihood is not increased. In this way, the data likelihood will monotonically increase and aEM is guaranteed to converge. We can apply their idea to come up with a variant of EM with the triple jump extrapolation that is guaranteed to converge.

4. Experimental Results with TJEM

This section reports the experimental evaluation of the triple jump accelerated EM algorithm (TJEM) to demonstrate the effectiveness of TJEM. The results reported here are different from those in [7], where we reported the results of applying triple jump to sub-vectors, while here we report the results of applying global triple jump. Previously, Hesterberg [5] also performed experiments for the same purpose, but he only used a quite simple probabilistic model with a two-dimensional parameter vector.

We compared the numbers of iterations required to converge for vanilla EM and TJEM to evaluate their performance. More specifically, the number of iterations is the number of times that an E-step is executed, which is the most costly operation in EM for the probabilistic models used in our experiments and is proportional to the CPU time required to converge.

We synthesized data sets for the following models:

- Hidden Markov Models (HMM): we used five-state, 20-symbol HMMs with randomly initialized parameter vectors to generate training data sets. Each data set contains 500 sequences of an alphabet of 100 symbols.
- Bayesian networks (BN): we used the ALARM model [3]. We randomly synthesized 2,000 examples for each experimental data set.
- Mixture of Gaussians (MoG): we used MoG with Gaussian components that overlapped with one another. We sampled 2,000 cases for each data set using five equal-weight Gaussians with means at $\{(0,0),(0,1),(1,0),(0,-1),(-1,0)\}$ and variances 0.8.

- Semisupervised Bayesian classifier (SB): We used a Bayesian classifier that classifies instances with 100 10-valued discrete features into 5 categories. 3,000 training cases were generated with 90% labels unknown and skewed missing features.

Figure 1 illustrates that TJEM almost always converges faster than EM. For the data sets of each model, we use one scattered plot to show the required convergence iterations. The coordination of each data point is the iterations of TJEM (the X-axis) and EM (the Y-axis) for the same data set. Thus, there are 100 data points in each plot. A data point lays in the upper triangle if TJEM converges faster, and in the lower triangle if EM is faster. We can see that in the 400 learning tasks of all the four models, TJEM converges faster for 392 times, and slower only for eight times.

We also compared the likelihood of the convergent parameter vectors by EM and TJEM. In Figure 1, a circle means that TJEM (the X-axis algorithm) converges at a parameter vector with a higher likelihood, while a box indicates that EM (the Y-axis algorithm) yields a higher likelihood. The size of a data point shows the difference between their likelihoods. A small point means that the difference is less than 10^{-5}, a medium one between 10^{-3} and 10^{-5}, and a large one more than 10^{-3}. We found that TJEM converges with a higher likelihood 60 times for HMM, 90 times for BN, 83 times for MoG, and 60 times in SB. All figures are out of 100 trials for each model. Therefore, TJEM not only accelerates the EM algorithm but also often improves the data likelihood of the learned models.

5. Global and Componentwise Extrapolation

It is also possible to approximate λ_i in each dimension, or divide the parameter space into subspaces and use Equation (11) to obtain an approximation for each subspace, as reported in [7]. In this section, we investigate the conditions when componentwise extrapolation is preferred.

Componentwise extrapolation may accelerate the convergence more effectively than global extrapolation when components of the parameter vector converge at different rates with the EM algorithm [16]. Recall that the global rate of convergence R of EM is defined in Equation (9). Now let $\varphi_i^{(t)} = \theta_i^{(t)} - \theta_i^*$ denote the componentwise difference. The i-th componentwise rate of convergence is defined as

$$R_i = \lim_{t\to\infty} R_i^{(t)} \equiv \lim_{t\to\infty} \frac{\varphi_i^{(t+1)}}{\varphi_i^{(t)}} \ . \tag{12}$$

When $R_i = R$ for all component i, global extrapolation is more appropriate than componentwise extrapolation, and vice versa. The global rate of convergence R is known to be the largest eigenvalue of the Jacobian [4]. R_i is also one of the eigenvalues but due to eigen transformation, R_i is not necessarily the i-th eigenvalue. The following Lemma is helpful for us to understand why R and R_i are eigenvalues and which eigenvalue corresponds to R_i.

Lemma 1 *The $l \times l$ Jacobian matrix J can be decomposed into a linear combination of its eigenvalues*

$$J = \sum_{j=1}^{k} \lambda_j u_j v_j^T = \lambda_1 u_1 v_1^T + \lambda_2 u_2 v_2^T + \cdots + \lambda_k u_k v_k^T \ ,$$

where $1 > \lambda_1 > \cdots > \lambda_k > 0$ are $k(\leq l)$ distinct eigenvalues of J, u_j, v_j $(j = 1, \cdots, k)$ form the bases of the j-th eigenvector spaces for J and J^T, respectively. Moreover,

$$J^t = \sum_{j=1}^{k} \lambda_j^t u_j v_j^T . \tag{13}$$

Proof Since J is a real-valued square matrix and can be decomposed as $J = Q\Lambda Q^{-1}$. Let $Q = [\, u_1 \,, \cdots, \, u_l \,]$ and

$$Q^{-1} = [v_1^T, \, v_2^T, \cdots, \, v_l^T]^T$$

Equation (13) follows immediately from [16], page 293.

With this Lemma, Meng and Rubin [11] showed that the global rate of convergence R for EM is the largest eigenvalue and gave the sufficient and necessary condition of when the componentwise rate $R_i = R$. We restate the proof of their findings less formally here.

A Taylor expansion of $\varphi^{(t)}$ and from Lemma 1, we have

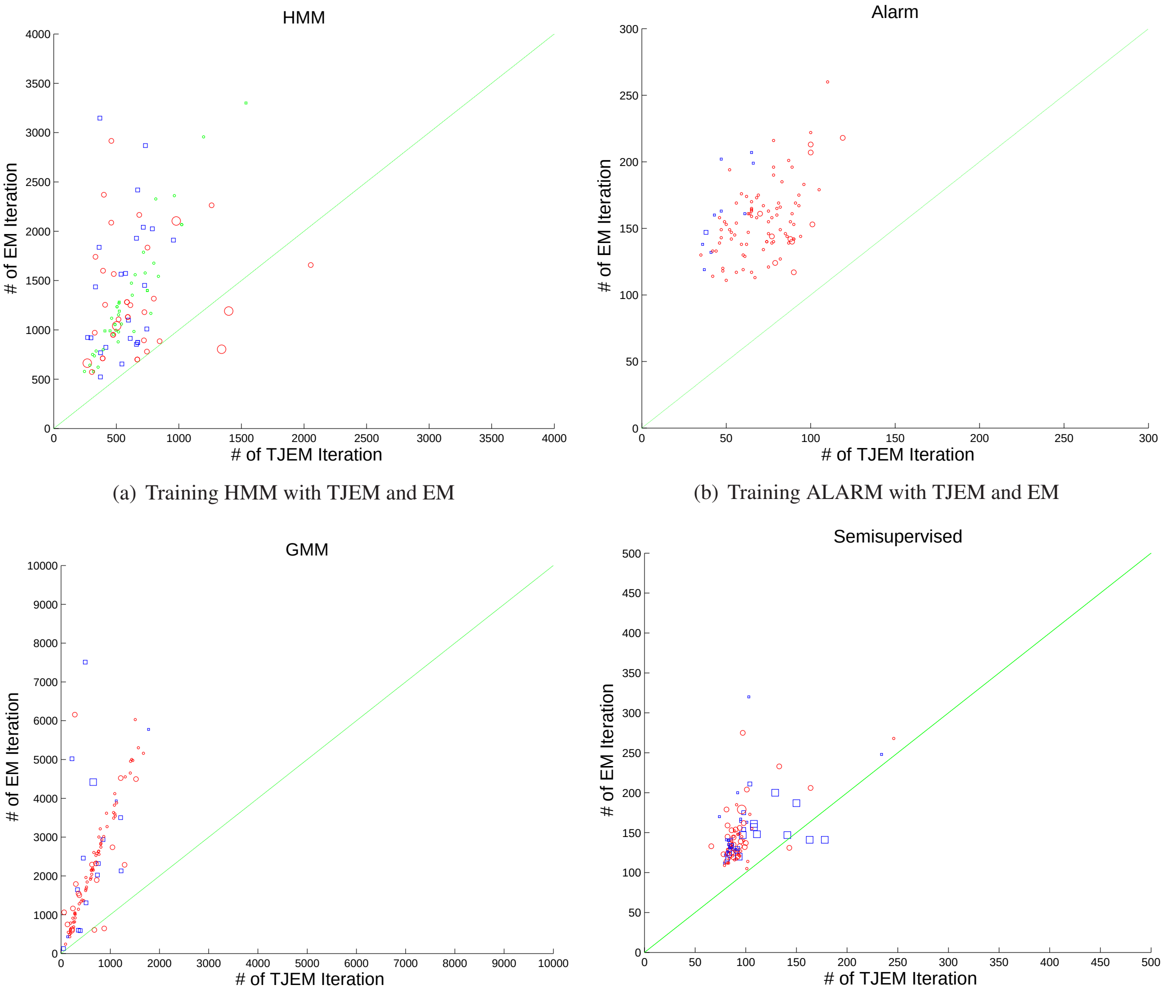
$$\varphi^{(t)} = \sum_{j=1}^{k} \lambda_j^t u_j v_j^T \varphi^{(0)}. \tag{14}$$

That is, the difference between the t-th estimate $\theta^{(t)}$ to the local maximum θ^* is a linear combination of the eigenvalues of J. Now, consider the i-th component θ_i of the parameter vector and the j-th largest eigenvalue λ_j of J. The contribution of λ_j to θ_i is

$$\lambda_j^t \cdot [u_j v_j^T] \cdot \varphi^{(0)} = \lambda_j^t \cdot [u_j v_j^T] \cdot \begin{bmatrix} \vdots \\ \varphi_i^{(0)} \\ \vdots \end{bmatrix} = \lambda_j^t \begin{bmatrix} \vdots \\ w_{ij} \\ \vdots \end{bmatrix} . \tag{15}$$

Note that $u_j v_j^T$ is a matrix defining the eigen transformation of the j-th eigenvalue. $u_j v_j^T$ maps the difference of the i-th component $\varphi_i^{(0)}$ to w_{ij}. If $w_{ij} \neq 0$, then λ_j contributes to the convergence of θ_i and the convergence rate for the i-th component is at least as slow as λ_j.

If for any component i, we have $w_{i1} \neq 0$, that is, the mapping result of the largest eigenvalue λ_1 is nonzero, then



(a) Training HMM with TJEM and EM

(b) Training ALARM with TJEM and EM

(c) Training MoG with TJEM and EM

(d) Training SB with TJEM and EM

Figure 1. Scattered plots that compare the TJEM and EM algorithms. TJEM converges faster and achieves a better likelihood in almost all trials.

the global rate of convergence is at least as slow as the largest eigenvalue λ_1. That is, $R = \lambda_1$. If for a given component i, we have $w_{i1} \neq 0$, then the componentwise rate of convergence for the i-th component is as slow as the global rate of convergence. That is, $R_i = R$. Otherwise, the componentwise rate of convergence is different from the global rate. Meng and Rubin [11] proved this by following the definitions of the componentwise rate and global rate of convergence with Lemma 1.

Corollary 2 *$R_i \neq R$ if $w_{i1} = e_i^T u_1 v_1^T \varphi^{(0)} = 0$, where e_i is the i-th column of the identity matrix I_d.*

An obvious case that makes $w_{i1} = 0$ is when $\varphi^{(0)}$ is a zero vector. That is, our initial value is exactly the local maximum, which is unlikely to happen. Since w_{i1} is the inner product of the i-th row of the matrix $u_1 v_1^T$ and $\varphi^{(0)}$, $w_{i1} = 0$ if they are orthogonal. This is unlikely, too. A more possible case is that the i-th row of $u_1 v_1^T$ is a zero vector.

When $J = Q \Lambda Q^{-1}$ is a diagonal matrix, Q and Q^{-1} will be diagonal, too. As a result, $u_j v_j^T$ will be singular. That is, some of their rows will be zero vectors and thus makes $R_i \neq R$. Therefore, we can conclude that when J is a diagonal matrix, we have $R_i \neq R$, and we should apply componentwise triple jump extrapolation. More precisely, we should also require that J is not only diagonal but also not proportional to the identity matrix so that all eigenvalues are distinct. Similarly, if J is block diagonal, $u_j v_j^T$ will also be singular and lead to the same consequences. We summarize our conclusion with the following claim:

Claim 3 *When the Jacobian J of an EM algorithm application is diagonal, or block diagonal, componentwise triple jump extrapolation may accelerate the convergence faster than global triple jump extrapolation and vice versa.*

6. Case Studies

In this section, we review two simple mixture of Gaussian models whose Jacobians were derived in previous work. One of them is diagonal and the other is not. Then we consider the Bayesian Network models and investigate when their Jacobians are (block) diagonal. We also present experimental results that verify our claims.

6.1. Mixtures of Gaussians

Our first example is from Meng and Rubin [11]. We have a set of one-dimension data $\mathcal{D}_{obs} = X = \{x_i | i = 1, 2, \ldots\}$ from the following distribution:

$$f_{ex1}(X|\mu, \sigma^2) = (1 - \pi)N(\mu, \sigma^2) + \pi N(\mu, \sigma^2/\lambda).$$

That is, the data set comes from a mixture of two Gaussians with the same mean but different variances. Assuming that we know the mixture ratio π and constant λ, then our parameter vector is $\theta = (\mu, \sigma^2)$. We can estimate the parameter vector from data by the EM algorithm by creating a missing, unobservable variable $Q \in \{1, \lambda\}$ that assigns membership of an observed variable X. Therefore, our complete, augmented data set is $\mathcal{D} = \{(x_i, q_i) | i = 1, 2, \ldots\}$.

We can use Equation (7) to compute the Fisher's information of the observed and missing data by differentiating the log-likelihood of the data twice to determine whether the Jacobian of this model is diagonal. If both information matrices, $\mathcal{I}_c$ and $\mathcal{I}_{obs}$, are diagonal, then the Jacobian will be diagonal, too. Though this model is simple, its Jacobian is still quite complex. Nevertheless, Meng and Rubin [11] showed that in this case, the Jacobian is a 2×2 diagonal matrix and empirically show that the componentwise rate of convergence is different.

Interestingly, with a different parameter vector, another one-dimensional Mixtures of Gaussian model from [9] has a Jacobian that is not diagonal. In this case, the parameter vector is $\theta = (\mu_0, \mu_1, \pi)$ with the variance known to be $\sigma^2 = 1$. The distribution for the observed data is

$$f_{ex2}(X|\mu 0, \mu_1, \pi) = (1 - \pi)N(\mu_0, 1) + \pi N(\mu_1, 1)\ .$$

We introduce an additional unobserved membership assignment variable $Q \in \{0, 1\}$ for the augmented complete data set. In this case, Louis [9] showed that $\mathcal{I}_c$ is diagonal, while $\mathcal{I}_{obs}$ is not, though it is symmetric. Thus J is not diagonal.

We then applied the global and componentwise triple jump extrapolation to these simple models. We synthesized a data set with 10,000 data points for both models and ran different EM variants to compare their rate of convergence. We found that for the first model, componentwise triple jump can accelerate the convergence more than global triple jump, while for the second model, global triple jump converges faster. For both models, both triple jump methods converge faster than vanilla EM. Figure 2 plot the curves of convergence of this experiment. The result is consistent with our prediction.

6.2. Bayesian Networks

We now consider a more practical model, the Bayesian Networks, to determine when its Jacobian is diagonal. The EM algorithm is applied to train a Bayesian Network model when we have latent variables whose values are not observable or when some of the values of variables in the training data are missing. The Jacobian of the EM algorithm for the Bayesian network can be obtained from Equation (7). Since our purpose is only to determine if the Jacobian is diagonal, there is no need to obtain the entire Jacobian matrix. In fact, if we can show that the Fisher's information $\mathcal{I}_{obs}$ and $\mathcal{I}_c$ are (block) diagonal, then the Jacobian must be (block) diagonal as well. Therefore, our plan here is to determine if the off-diagonal elements of $\mathcal{I}_{obs}$ and $\mathcal{I}_c$ are zero. For $\mathcal{I}_{obs}$, these off-diagonal elements are the second partial derivatives of the log-likelihood of data with respect to two different parameters. For $\mathcal{I}_c$, these elements are the expectation of the second partial derivatives of the complete data log-likelihood. Since if the second partial derivatives are zero, their expected values must be zero, too, there is no need to obtain the expectation. Thus, it suffices to show just the second partial derivatives with respect to two different parameters to determine if the Jacobian is diagonal.

A Bayesian network consists of a set of variables $X = \{X_i | i = 1, 2, \ldots\}$, the graph structure of the variables, and their conditional probability tables. Suppose we have a variable X_i whose parent nodes include a set of variables denoted by U_i. The conditional probability table for a variable X_i consists of entries of the form

$$w_{ijk} \equiv \Pr(X_i = x_{ik} | U_i = u_{ij})$$

to denote the probability that X_i has its k-th possible value x_{ik} under the condition that its parent U_i has the j-th combination of values, u_{ij}. Since w_{ijk} denotes the probability, to ensure that w_{ijk} is in $[0, 1]$ during the training process, a common technique usually used in practice is applying softmax reparameterization:

$$w_{ijk} = \frac{e^{\theta_{ijk}}}{\sum_{k'} e^{\theta_{ijk'}}}$$

Therefore, the parameters that we want to estimate from data using the EM algorithm are the set $\theta = \{\theta_{ijk} | i, j, k = 1, 2, \ldots\}$.

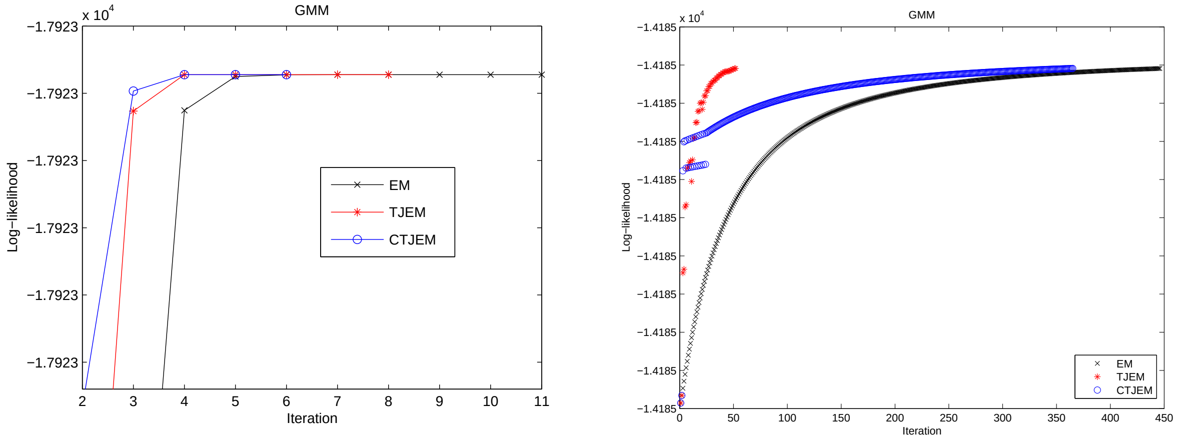


Figure 2. (Left) Convergence rate comparison of EM,TJEM,CTJEM for the example model given in [11], (right) Convergence rate comparison of EM,TJEM,CTJEM for the example model given in [9].

The training data for the Bayesian network is a set $\mathcal{D} = \{\ldots, y, \ldots\}$ where $y = \{\ldots, X_i = x_{ik}, \ldots\}$ is a set of variable-value pairs. Some of the variable's value may be missing either because in that particular case, its value is not available, or because the variable is a latent variable. Many algorithms available for the Bayesian network allow us to efficiently compute the conditional probability

$$P_\theta(y) = P_\theta(y_{mis}|y_{obs})P_\theta(y_{obs}) \ ,$$

the probability of unknown variable values given known variable values and the set of parameters. The Bayesian network allows us to factorize any conditional probability given the values of a subset of variables into an expression of w_{ijk}, the entries in the conditional probability table [13].

To discuss $\mathcal{I}_{obs}$ and $\mathcal{I}_c$, we start by considering the second order partial derivatives of the log-likelihood $L(\theta)$. We suppose that each training example is drawn independently so that $\frac{\partial}{\partial \theta_{ijk}} L(\theta) = \sum_y \frac{\partial}{\partial \theta_{ijk}} \log P_\theta(y)$.

We start from Lemma 4 which summarizes $\frac{\partial}{\partial \theta_{ijk}} w_{ijk}$ that will be frequently used:

Lemma 4 *For Bayesian networks with softmax reparameterization, the derivative of $w_{i'j'k'}$ with respect to θ_{ijk} is:*

$$\frac{\partial w_{i'j'k'}}{\partial \theta_{ijk}} = \begin{cases} 0 & : i \neq i' \text{ or } j \neq j' \\ -w_{ijk}w_{ijk'} & : (i,j) = (i',j') \text{ and } k \neq k' \\ w_{ijk}(1-w_{ijk}) & : (i,j,k) = (i',j',k') \end{cases} \tag{16}$$

Then, Lemma 5 describes the first order derivative of $P(y)$, and when the derivative is zero.

Lemma 5 *For Bayesian networks with softmax reparameterization, the derivative of $P(y)$ with respect to θ_{ijk} is:*

$$\frac{\partial}{\partial \theta_{ijk}} P(y) = P(x_{ik}, u_{ij}, y) - w_{ijk} P(u_{ij}, y), \tag{17}$$

and the derivative must be 0 if u_{ij} d-separates [12] the observations in $y - \{U_i = u_{ij}\}$ and X_i, that is, if $P(X_i|u_{ij}, y) = P(X_i|u_{ij})$.

Lemma 5 implies that X_i must not be initiated in y to satisfy the d-separation condition. Intuitively, if X_i is initiated as x_{ik} in y, y and X_i will never be conditionally independent given u_{ij} because $P(X_i|u_{ij}, y)$ is equal to 1 if $X_i = x_{ik}$ and equal to 0 otherwise. Based on Lemma 5, Theorem 6 shows the conditions in which $\frac{\partial^2 \log P(y)}{\partial \theta_{i'j'k'} \partial \theta_{ijk}} = 0$.

Theorem 6 $\frac{\partial^2 \log P(y)}{\partial \theta_{i'j'k'} \partial \theta_{ijk}}$ *is 0 if any of the following holds:*

1. *Lemma 5 holds, or*
2. *$u_{i'j'}$ d-separates the observations in $y \cup \{u_{ij}, x_{ik}\}$ and $X_{i'}$.*

The first condition is straightforward because the derivative of zero is still zero. We can obtain the second condition by expanding y to $y \cup \{u_{ij}, x_{ik}\}$ and applying Lemma 5 to $\theta_{i'j'k'}$ again.

As for other non-zero second order derivatives, we discuss the situation that $\{u_{ij}, x_{ik}\} \subset y$, and $(i,j) = (i',j')$ in Theorem 7.

Theorem 7 *If u_{ij} and x_{ik} are observed in y, $\frac{\partial^2 \log P(y)}{\partial \theta_{ijk'} \partial \theta_{ijk}}$, the second order derivative of two parameters with the same i, j, is:*

$$\frac{\partial^2 \log P(y)}{\partial \theta_{ijk'} \partial \theta_{ijk}} = \begin{cases} -w_{ijk}(1-w_{ijk}) & : \text{if } k' = k \\ w_{ijk}w_{ijk'} & : \text{if } k' \neq k. \end{cases}$$

Moreover, the derivatives with other (i', j', k')'s are zero.

Now we consider $\mathcal{I}_c$, the Fisher information of a complete data set. From Theorem 7, we can arrange θ_{ijk} so that $\frac{\partial^2 \log P(y)}{\partial \theta^2}$ is a block diagonal matrix, implying that $\mathcal{I}_c = E(\frac{\partial^2 \log P(D)}{\partial \theta^2})$ is also a block diagonal matrix.

Theorem 8 *$\mathcal{I}_c$ is a block diagonal matrix in which each element in each block B_{ij} is:*

$$E\left(\frac{\partial^2 \log P(D)}{\partial \theta_{ijk} \partial \theta_{ijl}}\right)$$

If $\mathcal{I}_{obs}$ is also block diagonal with the same block layout, J will also be a block diagonal matrix. Then, we can apply componentwise triple jump to each block and estimate the maximal eigenvalues for each block.

6.3. Semi-Supervised Bayesian Classifier

We verify our theoretical analysis with experiments on semi-supervised Bayesian classifiers. A Bayesian classifier consists of a cluster random variable C and a set of feature random variables $F_1, \ldots, F_N$. There are N links from C to each F_n. The model assumes that the feature random variables are conditionally independent given C.

Now we discuss $\mathcal{I}_{obs}$ of the model, which is simpler than general Bayesian networks in that every feature nodes share the same parent node. C and F_n might contain missing values. Theorem 9 describes some properties of $\mathcal{I}_{obs}$.

Theorem 9 $\frac{\partial^2}{\partial \theta_{i'j'k'} \partial \theta_{ijk}} \log P(y)$ *for $\mathcal{I}_{obs}$ of Bayesian classifiers is zero if any of the following is satisfied:*

1. *X_i is a feature variable and is not observed, or*

2. *$X_{i'}$ is a feature variable and is not observed.*

Corollary 10 $\frac{\partial^2}{\partial \theta_{i'j'k'} \partial \theta_{ijk}} \log P(y)$ *for $\mathcal{I}_{obs}$ of Bayesian classifiers can be nonzero if any of the following is satisfied:*

1. *$i = i'$ and X_i is a class variable,*

2. *X_i and $X_{i'}$ are feature and class variables and the feature variable is observed, or*

3. *X_i and $X_{i'}$ are feature variables or the same feature variable and are observed.*

From Corollary 10, whether $\mathcal{I}_{obs}$ is block diagonal matrix or not is related to the missing rates. First, we consider the case that the missing rate is low. For example, we suppose that the features are all observed. From Theorem 9, no element is guaranteed to be zero in $\frac{\partial^2 \log P(y)}{\partial \theta^2}$ so that $\sum_{y'} \frac{\partial^2 \log P(y')}{\partial \theta^2}$ is unlikely to be a block diagonal matrix.

However, if the missing rate is high, $\mathcal{I}_{obs}$ is much more close to a block diagonal matrix. An extreme example is that only one feature is observed in every training example. From Theorem 9, most values outside the block diagonal area is zero. Thus, componentwise TJEM is more likely to outperform global TJEM under such circumstances.

We performed experiments to verify the influence of missing rates of training data on the convergence rate of CTJEM and TJEM. We use Bayesian classifiers with 20 feature variables. All the features and class variables have five possible values. We randomly initialized the parameters of Bayesian classifiers and synthesized 10,000 examples with 50% and 90% missing rates. Then, we ran EM, TJEM, and CTJEM to train new classifiers.

Figure 3 and shows an example of the learning task with 50% and 90% missing values in data sets. The EM algorithm took 40 iterations to converge in the less sparse data set, and 837 in highly sparse. The TJEM algorithm accelerated vanilla EM by reducing the number of elapsed iterations to 28 and 351. The CTJEM algorithm took advantage of the block diagonal property described in the previous section, and further accelerated the highly sparse case by converging in only 114 iterations. However, when the missing rate is 50%, $\mathcal{I}_{obs}$ is less close to a block diagonal matrix and CTJEM fails to outperform the TJEM algorithm. Therefore, for a semi-supervised Bayesian classifier model with a large number of missing data, componentwise extrapolation should be preferred.

7. Conclusion

In this paper, we claim that, when the componentwise rate of convergence is different from the global rate of convergence, componentwise extrapolation should be preferred. We show that the componentwise rate and the global rate of convergence are different if the Jacobian of the EM mapping is diagonal or block diagonal. Our results suggest that when considering accelerating the EM algorithm with the triple jump method, we should try TJEM first. If TJEM does not provide satisfactory speedup, we can check the off-diagonal elements of the Jacobian to determine whether CTJEM may produce a better speedup.

Acknowledgements

We thank Dr. Chen-Hsin Chen for his comments. This research is supported in part by the National Science Council, Taiwan, under Grant No. NSC95-3112-B-001-017.

References

[1] E. Bauer, D. Koller, and Y. Singer. Update rules for parameter estimation in Bayesian networks. In *Proc. of the 13th*

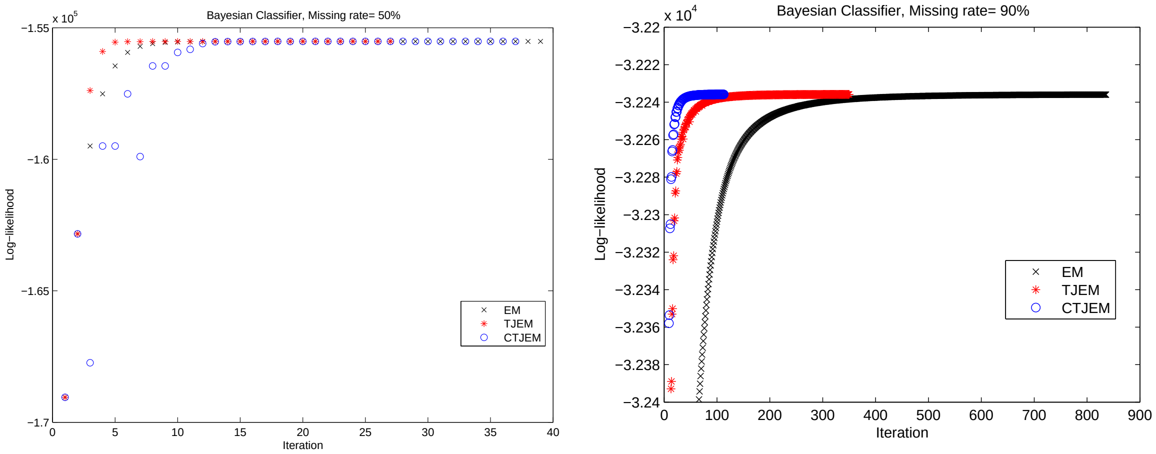


Figure 3. Training Bayesian classifiers with data sets containing 50% (left) and 90% (right) missing values. The data sets reflect our consideration whether $\mathcal{I}_{obs}$ is close to block diagonal (right) or not (left). CTJEM outperformed TJEM when $\mathcal{I}_{obs}$ is close to block diagonal.

Conference on Uncertainty in Artificial Intelligence, pages 3–13, 1997.

[2] R. L. Burden and D. Faires. *Numerical Analysis*. PWS-KENT Pub Co., 1988.

[3] G. F. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347, 1992.

[4] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[5] T. Hesterberg. Staggered aitken acceleration for EM. In *Proceedings of the Statistical Computing Section of the American Statistical Association*, Minneapolis, Minnesota, USA, August 2005.

[6] H.-S. Huang, B.-H. Yang, and C.-N. Hsu. Triple jump aiken accceleration for EM algorithm and its extrapolation-based variants. In preparation.

[7] H.-S. Huang, B.-H. Yang, and C.-N. Hsu. Triple-jump acceleration for the EM algorithm. In *Proceedings of the Fifth IEEE International Conference on Data Mining*, pages 649–652, 2005.

[8] M. Jamshidian and R. I. Jennrich. Acceleration of the EM algorithm by using quasi-newton methods. *Journal of the Royal Statistical Society, , Series B*, 59(3):569–587, 1997.

[9] T. A. Louis. Finding the observed information matrix when using the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 44:226–233, 1982.

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

[11] X.-L. Meng and D. B. Rubin. On the global and componentwise rates of convergence of the EM algorithm. *Linear Algebra and Its Applications*, 199:413–425, 1994.

[12] J. Pearl. *Probabilistic Reasoning in Intelligent Systems : Networks of Plausible Inference*. Morgan Kaufmann, 1988.

[13] S. Russell, J. Binder, D. Koller, and K. Kanazawa. Local learning in probabilistic networks with hidden variables. In *In Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 1146–1152, 1995.

[14] R. Salakhutdinov and S. Roweis. Adaptive overrelaxed bound optimization methods. In *Proceedings of the Twentieth International Conference on Machine Learning*, pages 664–671, 2003.

[15] J. L. Schafer. *Analysis of Incomplete Multivariate Data*. Chapman and Hall, New York, 1997.

[16] S. R. Searle. *Matrix Algebra Useful For Statistics*. Wiley, New York, 1982.

Appendix

Proof of Lemma 4:

$$\frac{\partial w_{ijk}}{\partial \theta_{ijk'}} = -\frac{e^{\theta_{ijk}} e^{\theta_{ijk'}}}{(\sum_{k''} e^{\theta_{ijk''}})^2} = -w_{ijk} w_{ijk'}$$

$$\begin{aligned}\frac{\partial w_{ijk}}{\partial \theta_{ijk}} &= \frac{e^{\theta_{ijk}}}{\sum_{k''} e^{\theta_{ijk''}}} - \left(\frac{e^{\theta_{ijk'}}}{\sum_{k''} e^{\theta_{ijk''}}}\right)^2 \\ &= w_{ijk}(1 - w_{ijk}).\end{aligned}$$

Proof of Lemma 5:

Russell et. al [13] derived

$$\frac{\partial}{\partial \theta_{ijk}} P(y) = \sum_{k'} P(u_{ij}) P(y|u_{ij}, x_{ik'}) \frac{\partial}{\partial \theta_{ijk}} w_{ijk'}. \tag{18}$$

Based on Lemma 4, Equation (18) can be further simplified:

$$\begin{aligned}\frac{\partial}{\partial\theta_{ijk}}P(y) &= P(u_{ij})P(y|u_{ij},x_{ik})\frac{\partial}{\partial\theta_{ijk}}w_{ijk} + \\ &\quad \sum_{k'\neq k} P(u_{ij})P(y|u_{ij},x_{ik'})\frac{\partial}{\partial\theta_{ijk}}w_{ijk'} \\ &= P(u_{ij})P(y|u_{ij},x_{ik})w_{ijk}(1-w_{ijk}) - \\ &\quad \sum_{k'\neq k} P(u_{ij})P(y|u_{ij},x_{ik'})w_{ijk}w_{ijk'} \\ &= P(u_{ij})P(y|u_{ij},x_{ik})w_{ijk} - \\ &\quad w_{ijk}\sum_{k'} P(u_{ij})P(y|u_{ij},x_{ik'})w_{ijk'} \\ &= P(u_{ij},x_{ik},y) - w_{ijk}\sum_{k'}P(u_{ij},x_{ik'},y) \\ &= P(u_{ij},x_{ik},y) - w_{ijk}P(u_{ij},y). \qquad (19)\end{aligned}$$

When u_{ij} d-separates the observations in $y - \{u_{ij}\}$ and X_i, Equation (19) can be rewritten as:

$$\begin{aligned}& P(u_{ij},x_{ik},y) - w_{ijk}P(u_{ij},y) \\ =\ & P(x_{ik}|u_{ij},y)P(u_{ij},y) - w_{ijk}P(u_{ij},y) \\ =\ & w_{ijk}P(u_{ij},y) - w_{ijk}P(u_{ij},y) = 0.\end{aligned}$$

Proof of Theorem 6:

The first condition is straightforward. The second condition can also be proved by Lemma 5. From Equation (17), we have

$$\begin{aligned}& \frac{\partial^2}{\partial\theta_{i'j'k'}\partial\theta_{ijk}}\log P(y) \\ =\ & \frac{1}{P(y)}\frac{\partial}{\partial\theta_{i'j'k'}}\left(P(u_{ij},x_{ik},y) - w_{ijk}P(u_{ij},y)\right) \\ =\ & \frac{1}{P(y)}\frac{\partial}{\partial\theta_{i'j'k'}}P(u_{ij},x_{ik},y) - P(u_{ij},y)\frac{\partial}{\partial\theta_{i'j'k'}}w_{ijk} \\ & -w_{ijk}\frac{\partial}{\partial\theta_{i'j'k'}}P(u_{ij},y). \qquad (20)\end{aligned}$$

We can consider $y' = y \cup \{u_{ij}, x_{ik}\}$ as another observed training example, and $u_{i'j'}$ d-separates y' and $x_{i'k'}$. By Lemma 5, we obtain that $\frac{\partial}{\partial\theta_{i'j'k'}}P(u_{ij},x_{ik},y)$, the first term of the above equation, is 0. Similarly, $\frac{\partial}{\partial\theta_{i'j'k'}}P(u_{ij},y) = 0$. Besides, Lemma 4 describes that $\frac{\partial}{\partial\theta_{i'j'k'}}w_{ijk} = 0$ here. Therefore, Equation (20) is also 0 under the second condition.

Proof of Theorem 7:

We start from

$$\frac{\partial^2 \log P(y)}{\partial\theta_{ijk'}\partial\theta_{ijk}} = \frac{\partial}{\partial\theta_{ijk'}}\left(P(u_{ij},x_{ik}|y) - w_{ijk}P(u_{ij}|y)\right).$$

If u_{ij} and x_{ik} are exactly the observed values in y, $P(u_{ij},x_{ik}|y)$ and $P(u_{ij}|y)$ are 1 and the above equation becomes:

$$\frac{\partial^2 \log P(y)}{\partial\theta_{ijk'}\partial\theta_{ijk}} = \frac{\partial}{\partial\theta_{ijk'}}\left(1 - w_{ijk}\right).$$

From Lemma 4, the second order partial derivative is $-w_{ijk}(1-w_{ijk})$ if $k' = k$, is $w_{ijk}w_{ijk'}$ if $k' \neq k$, and is 0 otherwise.

Proof of Theorem 8:

Let $\tilde{p}(w)$ denote the number of times that the predicate in w occurs in the data set $\mathcal{D}$. For example, suppose $w_{ijk} = \Pr(X_i = x_{ik}|U_i = u_{ij})$, then $\tilde{p}(w_{ijk})$ is the number of times that $X_i = x_{ik}|U_i = u_{ij}$ occurs in $\mathcal{D}$.

$$\begin{aligned}\log f(\mathcal{D}|\theta) &= \sum_{ijk}\tilde{p}(w_{ijk})(\log\frac{e^{\theta_{ijk}}}{\sum_{k'}e^{\theta_{ijk'}}}) \\ &= \sum_{ijk}\tilde{p}(w_{ijk})\theta_{ijk} - \sum_{ij}\tilde{p}(w_{ij})\log\sum_{k'}e^{\theta_{ijk'}}.\end{aligned}$$

Therefore,

$$\begin{aligned}\frac{\partial\log\ f(D|\theta)}{\partial\theta_{ijk}} &= \tilde{p}(w_{ijk}) - \frac{\partial\sum_{ij}\tilde{P}(w_{ij})\log\sum_{k'}e^{\theta_{ijk}}}{\partial\theta_{ijk}} \\ &= \tilde{p}(w_{ijk}) - \sum_{ij}\tilde{p}(w_{ij})\cdot\frac{e^{\theta_{ijk}}}{\sum_{k'}e^{\theta_{ijk'}}}.\end{aligned}$$

For $\theta_{i'j'k'}$, $i' = i, j' = j, k' = k$ or $k' \neq k$, we have $\frac{\partial^2\ \log\ f(y|\theta)}{\partial\theta_{i'j'k'}\partial\theta_{ijk}} = e^{\theta_{ijk}}\cdot\frac{\theta_{ijk}}{\sum_{k'}e^{\theta_{ijk'}}}$; for $\theta_{i'j'k'}$, $i'j'k' \neq ijk$, $\frac{\partial^2\ \log\ f(y|\theta)}{\partial\theta_{i'j'k'}\partial\theta_{ijk}} = 0$.

Proof of Theorem 9:

In the first condition, if X_i is not observed, X_i is d-separated with y by the cluster node. Based on the first condition in Theorem 6, $\frac{\partial^2}{\partial\theta_{i'j'k'}\partial\theta_{ijk}}\log P(y) = 0$.

In the second condition, if $X_{i'}$ is not observed, $X_{i'}$ is d-separated with $\{y, x_{ik}\}$ by the cluster node. Based on the second condition in Theorem 6, $\frac{\partial^2}{\partial\theta_{i'j'k'}\partial\theta_{ijk}}\log P(y) = 0$.

Proof of Corollary 10:

The three conditions are the complement of Theorem 9. The class variable and feature variables are probabilistically dependent because there are direct links between the class and feature variables. Therefore, we know from Lemma 5 that the derivative of $\log P(y)$ with respect to the parameters of the class variable is not guaranteed to be zero because y cannot be d-separated from the class variable. Accordingly, the first is true and the second conditions can be easily verified. The third condition is true because X_i and $X_{i'}$ are not independent of y.