MATHEMATICAL ENGINEERING
TECHNICAL REPORTS

# Iterative proportional scaling via decomposable submodels for contingency tables

Yushi ENDO and Akimichi TAKEMURA

METR 2006–16 March 2006

DEPARTMENT OF MATHEMATICAL INFORMATICS
GRADUATE SCHOOL OF INFORMATION SCIENCE AND TECHNOLOGY
THE UNIVERSITY OF TOKYO
BUNKYO-KU, TOKYO 113-8656, JAPAN

**WWW page: http://www.i.u-tokyo.ac.jp/mi/mi-e.htm**

# Iterative proportional scaling via decomposable submodels for contingency tables

Yushi Endo and Akimichi Takemura
Graduate School of Information Science and Technology
University of Tokyo

March, 2006

**Abstract**

We propose iterative proportional scaling (IPS) via decomposable submodels for maximizing likelihood function of a hierarchical model for contingency tables. In ordinary IPS the proportional scaling is performed by cycling through the elements of the generating class of a hierarchical model. We propose to adjust more marginals at each step. This is accomplished by expressing the generating class as a union of decomposable submodels and cycling through the decomposable models. We prove convergence of our proposed procedure, if the amount of scaling is adjusted properly at each step. We also analyze the proposed algorithms around the maximum likelihood estimate (MLE) in detail. Faster convergence of our proposed procedure is illustrated by numerical examples.

*Keywords and phrases:* chordal graph, hierarchical model, $I$-projection, iterative proportional fitting, Kullback-Leibler divergence.

## 1 Introduction

Iterative proportional scaling algorithm for contingency tables, first proposed by Deming and Stephan [8], has been well studied and generalized by many authors. Ireland and Kullback [13] proved convergence of IPS and Fienberg [10] gave a simpler proof of convergence from geometric consideration. Darroch and Ratcliff [6] made a generalization to IPS and its geometrical property was studied by Csiszár [5]. Csiszár [4] also gave a more general proof of convergence and justified IPS in a general framework. Extension of IPS to continuous case was studied in Kullback [16] and Ruschendorf [21]. Effective algorithms and implementations of IPS have been also studied by many authors, including [9], [14], [15], [20].

In this paper, we propose another generalization of IPS based on decomposable submodels. Decomposable models or graph decompositions have been already considered by

Jiroušek [14], Jiroušek and Přeučil [15] and Malvestuto [20]. However they used decomposable models for efficient implementation of conventional IPS. Here we use decomposable submodels for generalizing IPS itself. In our algorithm we adjust a larger set of marginals than the conventional IPS. The set of marginals form the generating class of a decomposable submodel. By adjusting more marginals, our proposed algorithm achieves a faster convergence to the maximum likelihood estimate than the conventional IPS, although at present it seems difficult to theoretically prove that our procedure is always faster. We prove convergence of our proposed procedure, if we adjust the amount of scaling at each step. We also analyze in detail the behavior of the proposed algorithms around the maximum likelihood estimate. As shown in Section 4 our procedure works well in practice without adjusting the amount of scaling at each step.

The organization of this paper is as follows. In Section 2 we summarize notations and basic facts on hierarchical models and decomposable models for multiway contingency tables. In Section 3 we propose a generalized IPS via decomposable submodels, prove its convergence and clarify its behavior close to the maximum likelihood estimate. In Section 4 we perform some numerical experiments to illustrate the effectiveness of the proposed procedure. Some discussions are given in Section 5.

# 2 Preliminaries

In this section we summarize notations and preliminary materials on decomposable models and conventional IPS.

We follow the notation of Lauritzen [19]. Let $\Delta$ denote the set of variables of a multiway contingency table. For each $\delta \in \Delta$, $\mathcal{I}_\delta = \{1, 2, \ldots, I_\delta\}$ denotes the set of levels of $\delta$. The set of cells is denoted by $\mathcal{I} = \times_{\delta\in\Delta}\mathcal{I}_\delta$. Let $n(i)$ denote the frequency of a cell $i \in \mathcal{I}$ and let $n = \sum_{i\in\mathcal{I}} n(i)$ denote the total sample size. Throughout the paper we denote the relative frequency (empirical distribution) by $r(i) = n(i)/n$. For a cell $i$ and a subset of variables $a \subset \Delta$, the marginal cell of $i$ for $a$ is denoted by $i_a \in \mathcal{I}_a = \times_{\delta\in a}\mathcal{I}_\delta$ and the marginal relative frequency of $i_a$ is denoted by $r(i_a)$.

A graph $G$ is a pair $(V, E)$, where $V$ is a finite set of vertices and $E$ is the set of edges. A *clique* of graph $G$ is a maximal complete subgraph. A path of length $m$ from a vertex $\alpha$ to a vertex $\beta$ is a sequence $\alpha = \alpha_0, \ldots, \alpha_m = \beta$ of distinct vertices such that $(\alpha_{i-1}, \alpha_i) \in E$ for all $i = 1, \ldots, m$. A cycle is a path such that $\alpha = \beta$. A chord is an edge between two non-consecutive vertices on a path. A graph $G$ is *chordal* if every cycle of length $n \geq 4$ has a chord. A set $S \subset V$ is a $(\alpha, \beta)$-separator if all paths from $\alpha$ to $\beta$ pass through $S$, and $S$ is minimal $(\alpha, \beta)$-separator if no proper subset of $S$ is a $(\alpha, \beta)$-separator. A minimal vertex separator is a minimal $(\alpha, \beta)$-separator for some $\alpha$ and $\beta$. Properties of a chordal graph and its minimal vertex separators are well studied ([2], [17], [18]).

A generating class $\mathcal{C} = \{C_1, \ldots, C_m\}$ of a hierarchical model is the family of the maximal interaction terms in the hierarchical model. We denote a hierarchical model simply by its generating class $\mathcal{C}$. A hierarchical model is a *decomposable model* if its generating class coincides with the set of cliques of a chordal graph. It is well known

([11], [12], [19]) that the elements of $\mathcal{C}$ of a decomposable model can be ordered to satisfy the running intersection property:

$$\text{(RIP)} \quad \text{For each } 2 \le j \le m, \text{ there exists } 1 \le k \le j-1, \text{ such that} \\ C_j \cap (C_1 \cup C_2 \cup \cdots \cup C_{j-1}) \subset C_k.$$

An ordering $(C_1, \ldots, C_m)$ satisfying RIP is called a *perfect sequence*. For a perfect sequence of a decomposable model let

$$S_j = C_j \cap (C_1 \cup C_2 \cup \cdots \cup C_{j-1}), \qquad 2 \le j \le m.$$

Then $S_2, \ldots, S_m$ are minimal vertex separators of the corresponding chordal graph. The number of times a minimal vertex separator $S$ appears in any perfect sequence is the same and called the *multiplicity* of $S$. We denote the multiplicity of $S$ by $\nu(S)$. In this paper

$$\mathcal{S} = \{S_2, \ldots, S_m\},$$

denotes the multiset of minimal vertex separators, where each minimal vertex separator $S$ appears $\nu(S)$ times in $\mathcal{S}$. Using a perfect sequence, the indices of cliques and minimal vertex separators can be made to satisfy the condition,

$$S_j \subset C_j, \quad 2 \le j \le m. \tag{1}$$

In this paper we index cliques and separators such that (1) is satisfied.

The MLE of the cell probabilities $\{p(i)\}$ of a hierarchical model $\mathcal{C}$ is obtained as the unique solution (within the model) of the likelihood equation

$$p(i_C) = r(i_C), \qquad \forall C \in \mathcal{C}, \ \forall i_C. \tag{2}$$

In the following we denote the cell probabilities of the MLE by $\{p^*(i)\}$. The MLE of a decomposable model is explicitly written as

$$p^*(i) = \begin{cases} \dfrac{\prod_{C \in \mathcal{C}} r(i_C)}{\prod_{S \in \mathcal{S}} r(i_S)}, & \text{if } \ r(i_C) > 0, \ \forall C \in \mathcal{C}, \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

For obtaining MLE for other graphical or hierarchical models we need some iterative procedure. The following conventional IPS, cycling through the elements of the generating class, is commonly used for this purpose. In the following let $p^{(t)}(i)$ denote the estimate of the probability of the cell $i$ at the $t$-th step of iteration.

**Algorithm 0** (Conventional IPS)
Let $p^{(0)}(i) \equiv 1/n$. The updating formula is given as

$$p^{(t+1)}(i) = p^{(t)}(i) \times \frac{r(i_C)}{p^{(t)}(i_C)}, \tag{4}$$

where $C = C_j,\ \ j = (t \mod m) + 1$.

The Kullback-Leibler divergence (KL-divergence) from a distribution $\{p(i)\}$ to another distribution $\{q(i)\}$ is denoted by

$$I(p : q) = \sum_{i \in \mathcal{I}} p(i) \log \frac{p(i)}{q(i)}.$$

The log sum inequality (Chapter 2 of [3]) for non-negative numbers $a_1, \ldots, a_N$ and $b_1, \ldots, b_N$ is

$$\sum_{i=1}^{N} a_i \log \frac{a_i}{b_i} \geq a \log \frac{a}{b}, \qquad a_i \geq 0,\ b_i \geq 0, \quad a = \sum_{i=1}^{N} a_i,\ b = \sum_{i=1}^{N} b_i,$$

where $a \log \frac{a}{0} = \infty$ if $a > 0$, and $0 \log 0 = 0$. The equality holds if and only if $a_i / b_i = \text{const}$.

# 3 Iterative proportional scaling via decomposable submodels

In this section we propose a generalization of conventional IPS and study its properties. At each step of our procedure we update a larger set of marginals, which form a decomposable submodel. We prove convergence of our proposed procedure, if the amount of scaling is adjusted properly at each step. We also give a detailed analysis of our procedure when the current estimate is close to MLE.

## 3.1 Proposed algorithms

As in the previous section let $\mathcal{C} = \{C_1, \ldots, C_m\}$ denote the generating class of a hierarchical model. Another generating class $\mathcal{C}' = \{C'_1, \ldots C'_v\}$ is a submodel of $\mathcal{C}$ if for each $C'_j$ there exists $C_i$ such that $C'_j \subset C_i$. A submodel $\mathcal{C}'$ is decomposable if $\mathcal{C}'$ is the set of cliques of a chordal graph.

Let $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_u$ be a family of submodels of $\mathcal{C}$. We say that $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_u$ *properly span* $\mathcal{C}$ if the following two conditions are satisfied.

$$\begin{aligned} &1. \quad \forall i\ \exists j \ \text{ s.t. } \ C_i \in \mathcal{C}_j, \\ &2. \quad \forall j\ \exists C \in \mathcal{C}_j \ \text{ s.t. } \ C \in \mathcal{C}. \end{aligned}$$

The first condition means that each element of $\mathcal{C} = \{C_1, \ldots, C_m\}$ appears as an element of some submodel. The second condition means that each submodel $\mathcal{C}_j$ contains at least one element of the original $\mathcal{C}$. Furthermore from now on we consider the case that each $\mathcal{C}_j$ is decomposable. Let $G_j$ denote the chordal graph associated with $\mathcal{C}_j$ and let $\mathcal{S}_j$ denote the multiset of the minimal vertex separators of $G_j$, $j = 1, \ldots, u$.

We now describe our proposed procedure. Because of the problem with normalization discussed later, we denote the estimated cell probability at the $t$-th step by $q^{(t)}(i)$.

**Algorithm 1** Let $q^{(0)} \equiv 1/n$. We cycle through $\mathcal{C}_1, \mathcal{C}_2, \dots, \mathcal{C}_u$ and for the $t$-th step we update the estimated cell probabilities as follows

$$q^{(t+1)}(i) = q^{(t)}(i) \times \frac{\prod_{C \in \mathcal{C}_j} r(i_C)}{\prod_{S \in \mathcal{S}_j} r(i_S)} \times \frac{\prod_{S \in \mathcal{S}_j} q^{(t)}(i_S)}{\prod_{C \in \mathcal{C}_j} q^{(t)}(i_C)}, \qquad j = (t \mod u) + 1, \tag{5}$$

and the normalized cell probabilities as $p^{(t+1)}(i) = q^{(t+1)}(i) / \sum_{k \in \mathcal{I}} q^{(t+1)}(k)$.

**Example 3.1.** Consider a 4-way contingency table $H \times J \times K \times L$ and the following hierarchical model ("4-cycle model")

$$p_{hjkl} = \exp(a_{hj} + b_{jk} + c_{kl} + d_{hl}).$$

By slight abuse of notation write $\Delta = \{H, J, K, L\}$. In this case, the family of submodels that properly span $\mathcal{C}$ is, for example, as follows.

$$\begin{aligned} \mathcal{C}_1 &= \{\{H, J\}, \{J, K\}, \{K, L\}\} \\ \mathcal{C}_2 &= \{\{H, J\}, \{K, L\}, \{H, L\}\} \end{aligned}$$

For each submodel, the updating procedure is performed as follows.

$$\begin{aligned} q^{(t+1)}(i) &= q^{(t)}(i) \times \frac{r(i_{hj}) \times r(i_{jk}) \times r(i_{kl})}{r(i_j) \times r(i_k)} \times \frac{q^{(t)}(i_j) \times q^{(t)}(i_k)}{q^{(t)}(i_{hj}) \times q^{(t)}(i_{jk}) \times q^{(t)}(i_{kl})}, \\ q^{(t+2)}(i) &= q^{(t+1)}(i) \times \frac{r(i_{hj}) \times r(i_{kl}) \times r(i_{hl})}{r(i_h) \times r(i_l)} \times \frac{q^{(t+1)}(i_h) \times q^{(t+1)}(i_l)}{q^{(t+1)}(i_{hj}) \times q^{(t+1)}(i_{kl}) \times q^{(t+1)}(i_{hl})}. \end{aligned}$$

Because the 4-cycle model is graphical, we can express the model and submodels as in Figure 1.

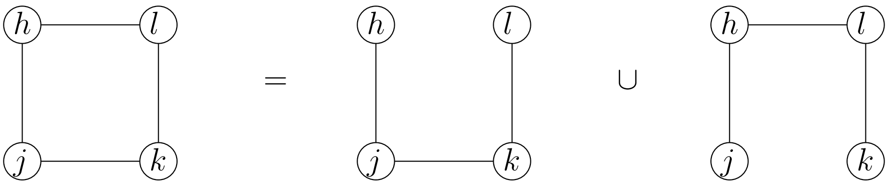

Figure 1: A decomposable submodels of Example 3.1

Algorithm 1 is a natural generalization of conventional IPS. However unfortunately it is difficult to prove convergence of Algorithm 1, although in practice it works well and has converged to MLE in all of our experiments. The difficulty lies in the fact that the sum $\sum_{i\in\mathcal{I}} q^{(t+1)}(i)$ after updating might exceed 1 (i.e. $\sum_{i\in\mathcal{I}} q^{(t+1)}(i) > 1$) in Algorithm 1 even if $q^{(t)}$ is normalized as $\sum_i q^{(t)}(i) = 1$. On the other hand, it should be noted that the normalization is irrelevant in Algorithm 1 because the normalizing constant is canceled on the right-hand side of (5). In Algorithm 1 we can simply ignore normalization and update $\{q^{(t)}\}$ as (5).

In order to deal with the theoretical difficulty concerning the normalization of $\{q^{(t+1)}\}$ we consider adjusting the amount of updating as follows.

**Algorithm 2** Let $\alpha = \alpha^{(t)} \geq 0$. We cycle through $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_u$ and for the $t$-th step we update the unnormalized estimated cell probabilities as

$$q^{(t+1)}(i) = q^{(t)}(i) \times \left( \frac{\prod_{C\in\mathcal{C}_j} r(i_C)}{\prod_{S\in\mathcal{S}_j} r(i_S)} \times \frac{\prod_{S\in\mathcal{S}_j} q^{(t)}(i_S)}{\prod_{C\in\mathcal{C}_j} q^{(t)}(i_C)} \right)^{\alpha}, \qquad j = (t \mod u) + 1, \tag{6}$$

and the normalized cell probabilities as $p^{(t+1)}(i) = q^{(t+1)}(i) / \sum_{k\in\mathcal{I}} q^{(t+1)}(k)$.

Note that in Algorithm 2 we do not have to normalize at each step. Normalization can be performed any time independent of the updating of the unnormalized cell probabilities. In the following we write

$$g(\alpha) = \sum_{i\in\mathcal{I}} q^{(t+1)}(i) = \sum_{i\in\mathcal{I}} q^{(t)}(i) \left( \frac{\prod_{C\in\mathcal{C}_j} r(i_C)}{\prod_{S\in\mathcal{S}_j} r(i_S)} \times \frac{\prod_{S\in\mathcal{S}_j} q^{(t)}(i_S)}{\prod_{C\in\mathcal{C}_j} q^{(t)}(i_C)} \right)^{\alpha}. \tag{7}$$

**Lemma 3.1.** *Consider the $t$-th step of* Algorithm 2. *Suppose that $\sum_i q^{(t)}(i) = 1$ and there exists a $C \in \mathcal{C}_j$ and $i_C$ such that $q^{(t)}(i_C) \neq r(i_C)$. Then there exists a unique $\alpha = \alpha_0 = \alpha_0(q^{(t)}) > 0$ such that $\sum_i q^{(t+1)}(i) = 1$.*

*Proof.* In view of (3) we have

$$1 = \sum_{i\in\mathcal{I}} \frac{\prod_{C\in\mathcal{C}_j} r(i_C)}{\prod_{S\in\mathcal{S}_j} r(i_S)} = \sum_{i\in\mathcal{I}} \frac{\prod_{C\in\mathcal{C}_j} q^{(t)}(i_C)}{\prod_{S\in\mathcal{S}_j} q^{(t)}(i_S)}.$$

Therefore if

$$\frac{\prod_{C\in\mathcal{C}_j} r(i_C)}{\prod_{S\in\mathcal{S}_j} r(i_S)} \times \frac{\prod_{S\in\mathcal{S}_j} q^{(t)}(i_S)}{\prod_{C\in\mathcal{C}_j} q^{(t)}(i_C)} \leq 1 \tag{8}$$

for all $i$, then the equality in (8) holds for all $i$ with $q^{(t)}(i) > 0$. Therefore under the condition of the lemma there exists at least one cell $i \in \mathcal{I}$ such that

$$\frac{\prod_{C\in\mathcal{C}_j} r(i_C)}{\prod_{S\in\mathcal{S}_j} r(i_S)} \times \frac{\prod_{S\in\mathcal{S}_j} q^{(t)}(i_S)}{\prod_{C\in\mathcal{C}_j} q^{(t)}(i_C)} > 1, \qquad q^{(t)}(i) > 0.$$

Then the right-hand side of (6) for this $i$ is strictly convex in $\alpha$ and diverges to $+\infty$ as $\alpha \to \infty$. Then the sum $g(\alpha)$ in (7) is also strictly convex in $\alpha$ and diverges to $+\infty$ as $\alpha \to \infty$.

Consider the differential of $g(\alpha)$ at $\alpha = 0$,

$$\begin{aligned}
g'(0) &= \sum_i q^{(t)}(i) \log \left( \frac{\prod_{C \in \mathcal{C}_j} r(i_C)}{\prod_{S \in \mathcal{S}_j} r(i_S)} \times \frac{\prod_{S \in \mathcal{S}_j} q^{(t)}(i_S)}{\prod_{C \in \mathcal{C}_j} q^{(t)}(i_C)} \right) \\
&= \sum_{C \in \mathcal{C}_j} \sum_i q^{(t)}(i) \log \frac{r(i_C)}{q^{(t)}(i_C)} - \sum_{S \in \mathcal{S}_j} \sum_i q^{(t)}(i) \log \frac{r(i_S)}{q^{(t)}(i_S)} \\
&= \sum_{C \in \mathcal{C}_j} \sum_{i_C} q^{(t)}(i_C) \log \frac{r(i_C)}{q^{(t)}(i_C)} - \sum_{S \in \mathcal{S}_j} \sum_{i_S} q^{(t)}(i_S) \log \frac{r(i_S)}{q^{(t)}(i_S)}.
\end{aligned}$$

Let $\mathcal{C}_j = \{C_1, \ldots, C_v\}$ and $\mathcal{S}_j = \{S_2, \ldots, S_v\}$, where the cliques and the minimal vertex separators of $\mathcal{C}_j$ are indexed to satisfy (1). Then,

$$\sum_{i_{C_1}} q^{(t)}(i_{C_1}) \log \frac{r(i_{C_1})}{q^{(t)}(i_{C_1})}$$

is the negative of KL-divergence and nonpositive. By the log sum inequality,

$$\sum_{i_{C_k}} q^{(t)}(i_{C_k}) \log \frac{r(i_{C_k})}{q^{(t)}(i_{C_k})} - \sum_{i_{S_k}} q^{(t)}(i_{S_k}) \log \frac{r(i_{S_k})}{q^{(t)}(i_{S_k})}$$

is also nonpositive for $2 \leq k \leq v$. Equality holds if and only if

$$r(i_C) = q^{(t)}(i_C), \quad \forall C \in \mathcal{C}_j.$$

Then, except for such a case, $g(0) = 1$, $g'(0) < 0$, $g(\infty) = \infty$, and $g(\alpha)$ is strictly convex in $\alpha$. Therefore there exists a unique $\alpha_0 > 0$ such that $g(\alpha_0) = 1$. □

Applying Lemma 3.1, we define the following algorithm.

**Algorithm 3** We cycle through $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_u$ and for the $t$-th step we update the estimated cell probabilities as follows

$$p^{(t+1)}(i) = p^{(t)}(i) \times \left( \frac{\prod_{C \in \mathcal{C}_j} r(i_C)}{\prod_{S \in \mathcal{S}_j} r(i_S)} \times \frac{\prod_{S \in \mathcal{S}_j} p^{(t)}(i_S)}{\prod_{C \in \mathcal{C}_j} p^{(t)}(i_C)} \right)^{\alpha_0}, \qquad j = (t \mod u) + 1, \quad (9)$$

where $\alpha_0 = \alpha_0(p^{(t)}) \geq 0$.

## 3.2 Convergence of the proposed algorithms

In this section, we prove the convergence of proposed algorithms. As before let $\{r(i)\}$ denote the empirical distribution and let $\{p^*(i)\}$ denote the MLE. Because we consider hierarchical models, the following equation holds ([4], [5]).

$$I(r : q) = I(r : p^*) + I(p^* : q).$$

$I(r : q)$ corresponds to the log likelihood. Therefore we can prove the convergence of our algorithms by proving $I(p^* : q^{(t)}) \to 0$ as $t \to \infty$.

**Theorem 3.1.** Algorithm 3 *converges to MLE.*

*Proof.* Consider KL-divergence after updating,

$$\begin{aligned} I(p^*; p^{(t+1)}) &= \sum_i p^*(i) \log \frac{p^*(i)}{p^{(t+1)}(i)} \\ &= \sum_i p^*(i) \log \frac{p^*(i)}{p^{(t)}(i) \times \left( \frac{\prod_{C \in \mathcal{C}_j} r(i_C)}{\prod_{S \in \mathcal{S}_j} r(i_S)} \times \frac{\prod_{S \in \mathcal{S}_j} p^{(t)}(i_S)}{\prod_{C \in \mathcal{C}_j} p^{(t)}(i_C)} \right)^{\alpha_0}} \\ &= \sum_i p^*(i) \log \frac{p^*(i)}{p^{(t)}(i)} - \alpha_0 \sum_i p^*(i) \log \frac{\prod_{C \in \mathcal{C}_j} r(i_C)}{\prod_{S \in \mathcal{S}_j} r(i_S)} \times \frac{\prod_{S \in \mathcal{S}_j} p^{(t)}(i_S)}{\prod_{C \in \mathcal{C}_j} p^{(t)}(i_C)}. \end{aligned}$$

Write $\mathcal{C}_j = \{C_1, \ldots, C_v\}$ and $\mathcal{S}_j = \{S_2, \ldots, S_v\}$ as in the proof of Lemma 3.1. Then,

$$\sum_i p^*(i) \log \frac{r(i_{C_1})}{p^{(t)}(i_{C_1})} = \sum_{i_{C_1}} r(i_{C_1}) \log \frac{r(i_{C_1})}{p^{(t)}(i_{C_1})}$$

is a KL-divergence, and nonnegative. By the log sum inequality,

$$\sum_{i_{C_k}} r(i_{C_k}) \log \frac{r(i_{C_k})}{p^{(t)}(i_{C_k})} - \sum_{i_{S_k}} r(i_{S_k}) \log \frac{r(i_{S_k})}{p^{(t)}(i_{S_k})}$$

is also nonnegative for $2 \leq k \leq v$. Therefore,

$$I(p^*; p^{(t+1)}) \leq I(p^*; p^{(t)})$$

holds. Equality holds if and only if $r(i_C) = q^{(t)}(i_C)$, $\forall C \in \mathcal{C}_j$. We see that $I(p^*; p^{(t)})$ always decreases after updating. The rest of the proof is the same as the classical one ([16]). □

**Corollary 3.1.** *Using* $0 < \alpha(q^{(t)}) \leq \alpha_0(q^{(t)})$*,* Algorithm 2 *converges to MLE.*

*Proof.* Consider KL-divergence after updating,

$$I(p^*;p^{(t+1)}) = \sum_i p_i^* \log \frac{p^*(i)}{q^{(t)}(i)} - \alpha \sum_i p^*(i) \log \frac{\prod_{C\in\mathcal{C}_j} r(i_C)}{\prod_{S\in\mathcal{S}_j} r(i_S)} \times \frac{\prod_{S\in\mathcal{S}_j} q^{(t)}(i_S)}{\prod_{C\in\mathcal{C}_j} q^{(t)}(i_C)} + \log g(\alpha).$$

Because $\alpha \le \alpha_0$, $\log g(\alpha)$ is nonpositive and $I(p^*;p^{(t)})$ always decreases after updating. The rest of the proof is the same as Theorem 3.1. □

At this point we discuss Algorithm 3 from a geometric viewpoint of $I$-projection in the sense of Csiszár ([4], [5]). In our procedure we adjust a larger set of marginals than the conventional IPS and in practice KL-divergence decreases more in our proposed algorithms than the conventional IPS for each step. However it is difficult to guarantee this theoretically. The difficulty lies in the fact that the updating rule (9) is not a projection. In fact, if we repeat (9) twice with the same $\mathcal{C}_j$ then the cell probabilities change, whereas in the conventional IPS repeating the same updating step twice does not change the cell probabilities after the first update. We can understand the situation as follows. Starting from the current estimate $\{p^{(t)}(i)\}$ suppose that we repeat the step (9) with the same $\mathcal{C}_j$ until the cell probabilities converge to $\{p^\star(i)\}$. Then the limit $\{p^\star(i)\}$ maximizes the likelihood function among $\{p(i)\}$ of the form

$$p(i) = p^{(t)}(i) \prod_{C\in\mathcal{C}_j} \mu(i_C). \tag{10}$$

The right-hand side of (10) forms a log-affine model through $\{p^{(t)}(i)\}$ (Section 4.2.3 of [19]). Since updating a single $C \in \mathcal{C}_j$ in the conventional IPS is a special case of (10), it follows that

$$I(p^* : p^\star) \le I(p^* : p^{(t+1)'}), \tag{11}$$

where $\{p^{(t+1)'}(i)\}$ is the updated estimate by the conventional IPS for some $C \in \mathcal{C}_j$. Therefore a larger decrease of KL-divergence of our procedure compared to conventional IPS is only guaranteed in the sense of (11). The situation will become more clear when we analyze the behavior of Algorithm 3 close to MLE in the next section.

## 3.3 Analysis of behavior close to the maximum likelihood estimate

In this section, we study the behavior of our algorithms when the current estimate is already close to MLE. We assume that MLE is in the interior of the parameter space and $p^*(i) > 0$ for all $i \in \mathcal{I}$. We analyze the behavior of $\alpha_0$. We also consider the value of $\alpha = \alpha_1$ which reduces the KL-divergence most and the value of $\alpha = \alpha_2$ such that KL-divergence decreases in Algorithm 2 for $0 \le \alpha \le \alpha_2$.

We repeatedly use the following expansion,

$$\log(1+x) = x - \frac{x^2}{2} + O(x^3), \quad x \to 0. \tag{12}$$

Assume that the current estimate $\{p^{(t)}(i)\}$ is close to MLE in the following sense. For sufficiently small $\varepsilon > 0$ and for all $C \in \mathcal{C}$, $S \in \mathcal{S}$, $i_C$, $i_S$ we have

$$1 - \varepsilon < \frac{r(i_C)}{p^{(t)}(i_C)}, \frac{r(i_S)}{p^{(t)}(i_S)} < 1 + \varepsilon. \tag{13}$$

The following proposition describes the behavior of $\alpha_0$ in Algorithm 3.

**Proposition 3.1.** *Assume $\{p^{(t)}(i)\}$ is close to MLE in the sense of (13). Then*

$$\alpha_0 = \frac{\sum_i p^{(t)}(i) \left\{ \sum_{C \in \mathcal{C}_j} \left( \frac{r(i_C)}{p^{(t)}(i_C)} - 1 \right)^2 - \sum_{S \in \mathcal{S}_j} \left( \frac{r(i_S)}{p^{(t)}(i_S)} - 1 \right)^2 \right\}}{\sum_i p^{(t)}(i) \left\{ \sum_{C \in \mathcal{C}_j} \left( \frac{r(i_C)}{p^{(t)}(i_C)} - 1 \right) - \sum_{S \in \mathcal{S}_j} \left( \frac{r(i_S)}{p^{(t)}(i_S)} - 1 \right) \right\}^2} + O(\varepsilon). \tag{14}$$

Before giving a proof of this Proposition we rewrite the numerator of the right-hand side of (14). Let $\mathcal{C}_j = \{C_1, \ldots, C_v\}$ and $\mathcal{S}_j = \{S_2, \ldots, S_v\}$, where the cliques and the minimal vertex separators of $\mathcal{C}_j$ are indexed to satisfy (1). Then

$$\begin{aligned}
\sum_i p^{(t)}(i) &\left\{ \sum_{C \in \mathcal{C}_j} \left( \frac{r(i_C)}{p^{(t)}(i_C)} - 1 \right)^2 - \sum_{S \in \mathcal{S}_j} \left( \frac{r(i_S)}{p^{(t)}(i_S)} - 1 \right)^2 \right\} \\
&= \sum_{i_{C_1}} p^{(t)}(i_{C_1}) \left( \frac{r(i_{C_1})}{p^{(t)}(i_{C_1})} - 1 \right)^2 \\
&\qquad + \sum_{k=2}^{v} \sum_{i_{C_k}} p^{(t)}(i_{C_k}) \left( \frac{r(i_{C_k})}{p^{(t)}(i_{C_k})} - \frac{r(i_{S_k})}{p^{(t)}(i_{S_k})} \right)^2 .
\end{aligned} \tag{15}$$

Therefore the numerator is nonnegative. Also note that the denominator of the right-hand side of (14) can be written as

$$\begin{aligned}
\sum_{C \in \mathcal{C}_j} &\left( \frac{r(i_C)}{p^{(t)}(i_C)} - 1 \right) - \sum_{S \in \mathcal{S}_j} \left( \frac{r(i_S)}{p^{(t)}(i_S)} - 1 \right) \\
&= \left( \frac{r(i_{C_1})}{p^{(t)}(i_{C_1})} - 1 \right) + \sum_{k=2}^{v} \left( \frac{r(i_{C_k})}{p^{(t)}(i_{C_k})} - \frac{r(i_{S_k})}{p^{(t)}(i_{S_k})} \right) .
\end{aligned} \tag{16}$$

We see that the numerator of $\alpha_0$ consists of the diagonal square terms when we expand the square of denominator in the form of (16). We now give a proof of Proposition 3.1.

*Proof.* Consider the following expansion,

$$\log \frac{\prod_{C \in \mathcal{C}_j} r(i_C)}{\prod_{S \in \mathcal{S}_j} r(i_S)} \times \frac{\prod_{S \in \mathcal{S}_j} p^{(t)}(i_S)}{\prod_{C \in \mathcal{C}_j} p^{(t)}(i_C)} = \sum_{C \in \mathcal{C}_j} \log \frac{r(i_C)}{p^{(t)}(i_C)} - \sum_{S \in \mathcal{S}_j} \log \frac{r(i_S)}{p^{(t)}(i_S)}$$

$$
\begin{aligned}
&= \sum_{C \in \mathcal{C}_j} \left( \frac{r(i_{C_j})}{p^{(t)}(i_C)} - 1 \right) - \sum_{S \in \mathcal{S}_j} \left( \frac{r(i_S)}{p^{(t)}(i_S)} - 1 \right) + O(\varepsilon^2) \\
&= O(\varepsilon).
\end{aligned}
$$

Then the $s$-th derivative of $g(\alpha)$ at 0 is

$$
\begin{aligned}
g^{(s)}(0) =& \sum_i p^{(t)}(i) \left( \log \frac{\prod_{C \in \mathcal{C}_j} r(i_C)}{\prod_{S \in \mathcal{S}_j} r(i_S)} \times \frac{\prod_{S \in \mathcal{S}_j} p^{(t)}(i_S)}{\prod_{C \in \mathcal{C}_j} p^{(t)}(i_C)} \right)^s \\
=& O(\varepsilon^s).
\end{aligned}
$$

The first and the second order derivatives of $g(\alpha)$ at 0 are,

$$
\begin{aligned}
g^{(1)}(0) =& \sum_i p^{(t)}(i) \left( \log \frac{\prod_{C \in \mathcal{C}_j} r(i_C)}{\prod_{S \in \mathcal{S}_j} r(i_S)} \times \frac{\prod_{S \in \mathcal{S}_j} p^{(t)}(i_S)}{\prod_{C \in \mathcal{C}_j} p^{(t)}(i_C)} \right) \\
=& \sum_{C \in \mathcal{C}_j} \sum_{i_C} p^{(t)}(i_C) \log \frac{r(i_C)}{p^{(t)}(i_C)} - \sum_{S \in \mathcal{S}_j} \sum_{i_S} p^{(t)}(i_S) \log \frac{r(i_S)}{p^{(t)}(i_S)} \\
=& \sum_{C \in \mathcal{C}_j} \sum_{i_C} p^{(t)}(i_C) \left\{ \frac{r(i_C)}{p^{(t)}(i_C)} - 1 - \frac{1}{2} \left( \frac{r(i_C)}{p^{(t)}(i_C)} - 1 \right)^2 \right\} \\
& - \sum_{S \in \mathcal{S}_j} \sum_{i_S} p^{(t)}(i_S) \left\{ \frac{r(i_S)}{p^{(t)}(i_S)} - 1 - \frac{1}{2} \left( \frac{r(i_S)}{p^{(t)}(i_S)} - 1 \right)^2 \right\} + O(\varepsilon^3) \\
=& \sum_{C \in \mathcal{C}_j} \sum_{i_C} \left\{ (r(i_C) - p^{(t)}(i_C)) - \frac{p^{(t)}(i_C)}{2} \left( \frac{r(i_C)}{p^{(t)}(i_C)} - 1 \right)^2 \right\} \\
& - \sum_{S \in \mathcal{S}_j} \sum_{i_S} \left\{ (r(i_S) - p^{(t)}(i_S)) - \frac{p^{(t)}(i_S)}{2} \left( \frac{r(i_S)}{p^{(t)}(i_S)} - 1 \right)^2 \right\} + O(\varepsilon^3) \\
=& \frac{1}{2} \sum_i p^{(t)}(i) \left\{ \sum_{S \in \mathcal{S}_j} \left( \frac{r(i_S)}{p^{(t)}(i_S)} - 1 \right)^2 - \sum_{C \in \mathcal{C}_j} \left( \frac{r(i_C)}{p^{(t)}(i_C)} - 1 \right)^2 \right\} + O(\varepsilon^3),
\end{aligned}
$$

and

$$
\begin{aligned}
g^{(2)}(0) =& \sum_i p^{(t)}(i) \left( \log \frac{\prod_{C \in \mathcal{C}_j} r(i_C)}{\prod_{S \in \mathcal{S}_j} r(i_S)} \times \frac{\prod_{S \in \mathcal{S}_j} p^{(t)}(i_S)}{\prod_{C \in \mathcal{C}_j} p^{(t)}(i_C)} \right)^2 \\
=& \sum_i p^{(t)}(i) \left\{ \sum_{C \in \mathcal{C}_j} \left( \frac{r(i_C)}{p^{(t)}(i_C)} - 1 \right) - \sum_{S \in \mathcal{S}_j} \left( \frac{r(i_S)}{p^{(t)}(i_S)} - 1 \right) \right\}^2 + O(\varepsilon^3).
\end{aligned}
$$

Then, we expand $g(\alpha)$ at 0,

$$
g(\alpha) = g(0) + \alpha g^{(1)}(0) + \frac{\alpha^2}{2} g^{(2)}(0) + O(\varepsilon^3).
$$

Assuming normalization at each step of the algorithm, we have $g(0) = 1$ and substituting $\alpha_0$ for $\alpha$, we obtain

$$\begin{aligned}\alpha_0 &= \frac{-2g^{(1)}(0)}{g^{(2)}(0)} + O(\varepsilon)\\ &= \frac{\sum_i p^{(t)}(i)\left\{\sum_{C\in\mathcal{C}_j}\left(\frac{r(i_C)}{p^{(t)}(i_C)}-1\right)^2 - \sum_{S\in\mathcal{S}_j}\left(\frac{r(i_S)}{p^{(t)}(i_S)}-1\right)^2\right\}}{\sum_i p^{(t)}(i)\left\{\sum_{C\in\mathcal{C}_j}\left(\frac{r(i_C)}{p^{(t)}(i_C)}-1\right) - \sum_{S\in\mathcal{S}_j}\left(\frac{r(i_S)}{p^{(t)}(i_S)}-1\right)\right\}^2} + O(\varepsilon).\end{aligned}$$

□

Consider (15) and (16). If the signs of the terms on the right hand side of (16) are "random" then we can expect that $\alpha_0$ is close to 1. We can imagine that $\{p^{(t)}(i)\}$ converges to MLE from various directions. Then $\alpha_0$ is close to 1 "on the average". Furthermore as shown in the following proposition $\alpha_0$ is the optimum value of the adjustment close to MLE. We believe that this is the reason that Algorithm 1 works very well in practice.

**Proposition 3.2.** *Assume $\{p^{(t)}(i)\}$ is close to MLE in the sense of (13). Then*

$$\alpha_1 = \alpha_0 + O(\varepsilon), \tag{17}$$

*where $\alpha_1$ is the value of $\alpha$ which reduces the KL-divergence most.*

*Proof.* Define $F(\alpha)$ by

$$F(\alpha) = \alpha \sum_i p^*(i) \log \frac{\prod_{C\in\mathcal{C}_j} r(i_C)}{\prod_{S\in\mathcal{S}_j} r(i_S)} \times \frac{\prod_{S\in\mathcal{S}_j} p^{(t)}(i_S)}{\prod_{C\in\mathcal{C}_j} p^{(t)}(i_C)}, \tag{18}$$

which corresponds to the decrease of KL-divergence before normalization. Consider the derivative of $F(\alpha)$,

$$\begin{aligned}F^{(1)}(\alpha) =& \sum_i p^*(i)\log \frac{\prod_{C\in\mathcal{C}_j} r(i_C)}{\prod_{S\in\mathcal{S}_j} r(i_S)} \times \frac{\prod_{S\in\mathcal{S}_j} p^{(t)}(i_S)}{\prod_{C\in\mathcal{C}_j} p^{(t)}(i_C)}\\ =& g^{(1)}(0) + \sum_i p^{(t)}(i)\left(\frac{p^*(i)}{p^{(t)}(i)}-1\right)\log \frac{\prod_{C\in\mathcal{C}_j} r(i_C)}{\prod_{S\in\mathcal{S}_j} r(i_S)} \times \frac{\prod_{S\in\mathcal{S}_j} p^{(t)}(i_S)}{\prod_{C\in\mathcal{C}_j} p^{(t)}(i_C)}\\ =& g^{(1)}(0) + \sum_i p^{(t)}(i)\left(\frac{p^*(i)}{p^{(t)}(i)}-1\right)\left\{\sum_{C\in\mathcal{C}_j}\left(\frac{r(i_C)}{p^{(t)}(i_C)}-1\right) - \sum_{S\in\mathcal{S}_j}\left(\frac{r(i_S)}{p^{(t)}(i_S)}-1\right)\right\}\\ &+ O(\varepsilon^3)\\ =& g^{(1)}(0) + \sum_i p^{(t)}(i)\left\{\sum_{C\in\mathcal{C}_j}\left(\frac{r(i_C)}{p^{(t)}(i_C)}-1\right)^2 - \sum_{S\in\mathcal{S}_j}\left(\frac{r(i_S)}{p^{(t)}(i_S)}-1\right)^2\right\} + O(\varepsilon^3)\end{aligned}$$

$$= - g^{(1)}(0) + O(\varepsilon^3).$$

Consider the derivative of $F(\alpha) - \log g(\alpha)$ and equating 0, we obtain,

$$F^{(1)}(\alpha_1) - \frac{g^{(1)}(\alpha_1)}{g(\alpha_1)} = 0.$$

Then

$$\begin{aligned} g^{(1)}(\alpha_1) &= g^{(1)}(0) + \alpha_1 g^{(2)}(0) + O(\varepsilon^3), \\ g(\alpha_1) &= g(0) + \alpha_1 g^{(1)}(0) + \frac{\alpha_1^2}{2} g^{(2)}(0) + O(\varepsilon^3) \\ &= 1 + O(\varepsilon^2) \end{aligned}$$

and

$$F^{(1)}(\alpha_1) - g^{(1)}(\alpha_1) + O(\varepsilon^3) = -g^{(1)}(0) - g^{(1)}(0) - \alpha_1 g^{(2)}(0) + O(\varepsilon^3) = 0.$$

Therefore we have

$$\alpha_1 = \frac{-2g^{(1)}(0)}{g^{(2)}(0)} + O(\varepsilon) = \alpha_0 + O(\varepsilon).$$

□

Finally we show that KL-divergence decreases in the range $0 < \alpha < 2\alpha_0$. This result indicates that in Algorithm 2, $\alpha > \alpha_0$ often decreases KL-divergence in practice.

**Proposition 3.3.** *Assume $\{p^{(t)}(i)\}$ is close to MLE in the sense of (13). Then*

$$\alpha_2 = 2\alpha_0 + O(\varepsilon). \tag{19}$$

*where $\alpha_2$ is the value of $\alpha$ such that $I(p^* : p^{(t+1)}) = I(p^* : p^{(t)})$ in* Algorithm 2.

*Proof.*

$$0 = F(\alpha_2) - \log g(\alpha_2) = -\alpha_2 g^{(1)}(0) - \alpha_2 g^{(1)}(0) + \frac{\alpha_2^2}{2} g^{(2)}(0) + O(\varepsilon^3)$$

and

$$\alpha_2 = \frac{-4g^{(1)}(0)}{g^{(2)}(0)} + O(\varepsilon) = 2\alpha_0 + O(\varepsilon).$$

□

We show the behavior of $\log g(\alpha)$ and $F(\alpha) - \log g(\alpha)$ in Figure 2. Proposition 3.1, Proposition 3.2 and Proposition 3.3 indicate that in many cases we can decrease KL-divergence by using $\alpha = 1$. In the next section we illustrate this by numerical experiments.

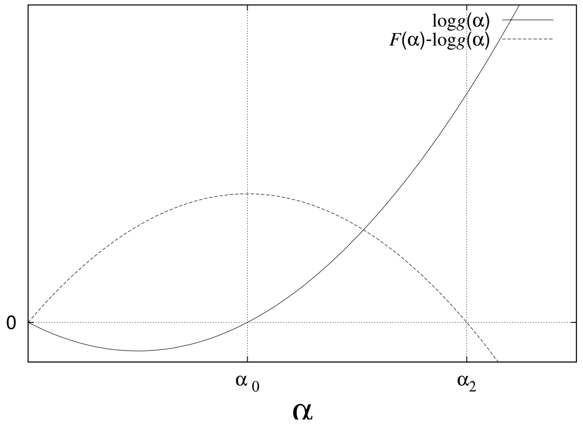

Figure 2: Behavior of $\log g(\alpha)$ and $F(\alpha) - \log g(\alpha)$

# 4 Numerical experiments

In this section, we compare our Algorithm 1 with the conventional IPS by numerical experiments. In a case of three-way contingency table, we consider the hierarchical model that contains all two-dimensional interaction terms (the model without the 3-factor interactions). Similarly for $J \geq 4$ we consider $J$-way cycle model with the generating class $\{\{1,2\},\{2,3\},\ldots,\{J-1,J\},\{J,1\}\}$. As a family of decomposable submodels which properly span the model we use the set of two decomposable submodels obtained by deleting one element of generating class of the hierarchical model.

Table 1: The submodels in numerical experiments

| Dim | Hierarchical model | Decomposable submodels |
|---|---|---|
| 3 | $M_3 = \{12, 23, 13\}$ | $M_3 \setminus \{13\}, M_3 \setminus \{12\}$ |
| 4 | $M_4 = \{12, 23, 34, 14\}$ | $M_4 \setminus \{14\}, M_4 \setminus \{23\}$ |
| 5 | $M_5 = \{12, 23, 34, 45, 15\}$ | $M_5 \setminus \{15\}, M_5 \setminus \{23\}$ |
| 6 | $M_6 = \{12, 23, 34, 45, 56, 16\}$ | $M_6 \setminus \{16\}, M_6 \setminus \{34\}$ |
| 7 | $M_7 = \{12, 23, 34, 45, 56, 67, 17\}$ | $M_7 \setminus \{17\}, M_7 \setminus \{34\}$ |
| 8 | $M_8 = \{12, 23, 34, 45, 56, 67, 78, 18\}$ | $M_8 \setminus \{18\}, M_8 \setminus \{45\}$ |

We show the considered model and its submodels in Table 1, where $\{1, 2\}$ is abbreviated as 12. For example in the 5-way case we span $M_5 = \{12, 23, 34, 45, 15\}$ by $M_5 \setminus \{15\}$ and $M_5 \setminus \{23\}$ as illustrated in Figure 3.

Each variable takes two levels. We generated random contingency tables by filling each cell by uniform random integers from 1 to $10^6$ and we obtained MLE by Algorithm 1 and standard IPS for each contingency table. As the convergence criterion we used

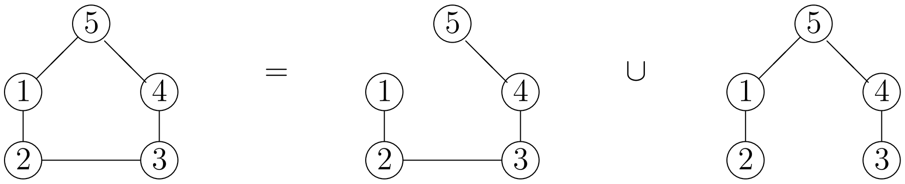

Figure 3: A decomposable submodels in a 5-way case

$\sum_i |p^{(t+1)}(i) - p^{(t)}(i)| \leq 10^{-6}$. For each dimension we generated 1000 contingency tables and took the average of the number of steps to convergence.

The results are shown in Table 2. In all of our runs Algorithm 1 converged to MLE. We show the appearance of convergence in Figure 4 to Figure 9. The vertical axis is the average of the logarithm of $D(t) = \sum_i |p^{(t+1)}(i) - p^{(t)}(i)|$ and the horizontal axis is the number of steps. The experiments shows that Algorithm 1 always converges faster than conventional IPS. It is interesting to note that in the case of conventional IPS, $\log D(t)$ decreases by a large amount periodically. The period is $J - 1$, which is one less than the complete cycling of $\{\{1,2\},\{2,3\},\ldots,\{J-1,J\},\{J,1\}\}$. On the other hand for Algorithm 1 $\log D(t)$ decreases steadily because we adjust a larger set of marginals at each step.

Table 2: The number of steps to convergence

| Dim | Conventional IPS | Algorithm 1 |
|---|---|---|
| 3 | 54.228 | 39.918 |
| 4 | 23.02 | 12.744 |
| 5 | 18.188 | 7.789 |
| 6 | 17.226 | 6.199 |
| 7 | 13.979 | 4.063 |
| 8 | 15.272 | 3.987 |

## 5 Some discussions

For using the proposed algorithms, we have to find a family of decomposable submodels that properly span a generating class of a hierarchical model. We recommend spanning the generating class by a small number of large decomposable submodels. Here large decomposable submodels might mean maximal submodels in the sense of model inclusion or submodels with largest degrees of freedom. In the literature some methods for finding a maximal chordal subgraph of a given graph are studied ([1], [7], [22]). In the case of

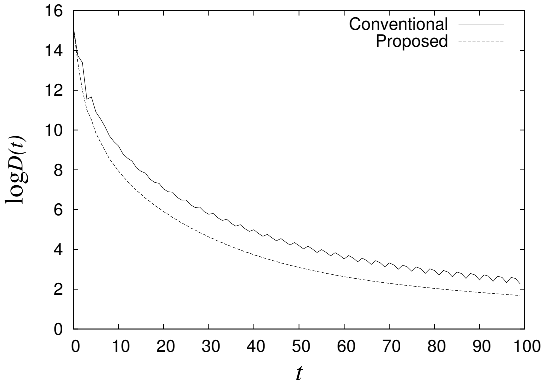

Figure 4: Convergence in the 3-way case

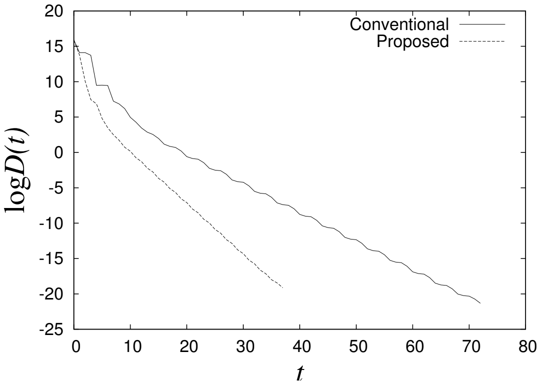

Figure 5: Convergence in the 4-way case

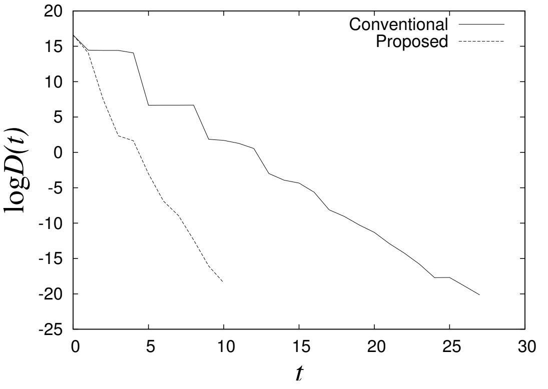

Figure 6: Convergence in the 5-way case

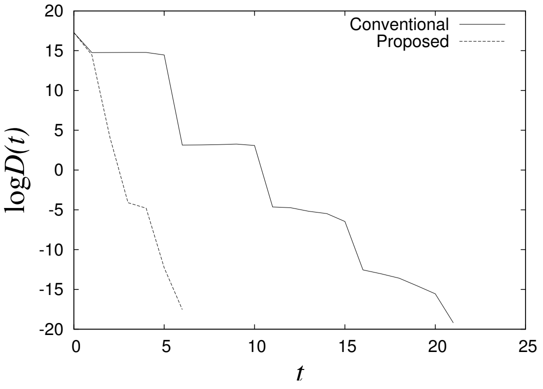

Figure 7: Convergence in the 6-way case

graphical models, this might give a solution to our problem. However we have to satisfy the condition that each element of a generating class is contained in at least one decomposable submodel. Therefore we need a method to find a maximal chordal subgraph under the restriction that specific cliques are contained. For general hierarchical models we are not aware of any algorithm in existing literature for obtaining a family decomposable submodels properly spanning the generating class of the hierarchical model.

In this paper we compared various algorithms of IPS in terms of the number of steps to convergence. We showed that the proposed algorithm converges faster than conventional IPS by numerical experiments. However we should mention that the computational complexity for each step depends on the algorithms and comparison of computational time may be different from comparison of the number of steps. Note that the complexity of each step of IPS depends also on actual implementation of IPS (see e.g. [9], [14], [15], [20]). In this paper we have not investigated efficient implementation of the steps of our algorithms. Efficient implementation of our procedure is left to further investigations.

**Acknowledgment.** The authors are grateful to Satoshi Kuriki for very useful comments.

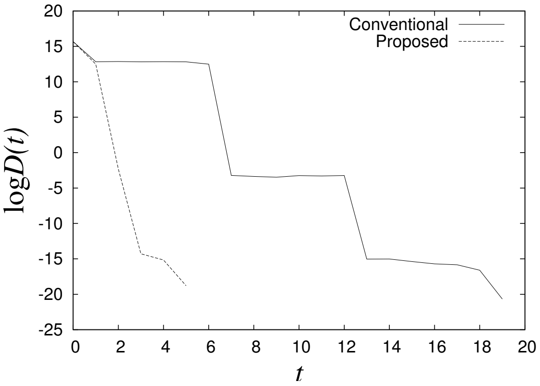

Figure 8: Convergence in the 7-way case

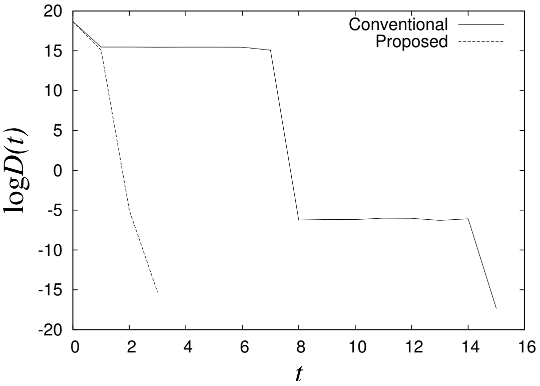

Figure 9: Convergence in the 8-way case

# References

[1] Berry, A.,Heggernes, P. and Villanger, Y. (2003), A vertex incremental approach for dynamically maintaining chordal graphs. *Lecture Notes in Computer Science*, Vol.2906, pp.47–57.

[2] Blair, J. R. S. and Peyton, B. W. (1994), An introduction to chordal graphs and clique trees. *Graph theory and sparse matrix computation*, IMA Vol. Math. Appl., Vol.56, pp.1–29, Springer, New York.

[3] Cover, T. M. and Thomas, J. A. (1991). *Elements of Information Theory.* Wiley, New York.

[4] Csiszár, I. (1975), $I$-divergence geometry of probability distributions and minimization problems. *Ann. Probab.*, Vol.3, pp.146–158.

[5] Csiszár, I. (1989), A geometric interpretation of Darroch and Ratcliff's generalized iterative scaling. *Ann. Stat.*, Vol.17, pp.1409–1413.

[6] Darroch, J. N. and Ratcliff, D. (1972), Generalized iterative scaling for log-linear models. *Ann. Math. Statist.*, Vol.43, pp.1470–1480.

[7] Dearing, P. M., Shier, D. R. and Warner, D. D. (1988), Maximal chordal subgraphs. *Disc. Appl. Math.*, Vol.20, pp.181–190.

[8] Deming, W. E. and Stephan, F. F. (1940), On a least squares adjustment of a sampled frequency table when the expected marginal totals are known. *Ann. Math. Statist.*, Vol.11, pp.427–444.

[9] Denteneer, D. and Verbeek, A. (1986), A fast algorithm for iterative proportional fitting in log-linear models. *Computational Statistics and Data Analysis*, Vol.3, pp.251–264.

[10] Fienberg, S. E. (1970), An iterative procedure for estimation in contingency tables. *Ann. Math. Statist.*, Vol.41, pp.907–917.

[11] Hara, H. and Takemura, A. (2005a). Improving on the maximum likelihood estimators of the means in Poisson decomposable graphical models. Technical Report METR 05-08, University of Tokyo. Submitted for publication.

[12] Hara, H. and Takemura, A. (2005b). Bayes admissible estimation of the means in Poisson decomposable graphical models. Technical Report METR 05-22, University of Tokyo. Submitted for publication.

[13] Ireland, C. T. and Kullback, S. (1968), Contingency tables with given marginals. *Biometrika*, Vol.55, pp.179–188.

[14] Jiroušek, R. (1991), Solution of the marginal problem and decomposable distributions. *Kybernetika*, Vol.27, pp.403–412.

[15] Jiroušek, R. and Přeučil, S. (1995), On the effective implementation of the iterative proportional fitting procedure. *Computational Statistics and Data Analysis*, Vol.19, pp.177–189.

[16] Kullback, S. (1968), Probability densities with given marginals. *Ann. Math. Statist.*, Vol.39, pp.1236–1243.

[17] Kumar, P. S. and Madhavan, C. E. V. (1998), Minimal vertex separators of chordal graphs. *Discrete Applied Mathematics*, Vol.89, pp.155–168.

[18] Kumar, P. S. and Madhavan, C. E. V. (2002), Clique tree generalization and new subclass of chordal graphs. *Discrete Applied Mathematics*, Vol.117, pp.109–131.

[19] Lauritzen, S. L. (1997), *Graphical Models.* Clarendon Press, Oxford.

[20] Malvestuto, F. M. (1989), Computing the maximum-entropy extension of given discrete probability distributions. *Computational Statistics and Data Analysis*, Vol.8, pp.299–311.

[21] Rüschendorf, L. (1995), Convergence of the iterative proportional fitting procedure. *Ann. Stat.*, Vol.23, pp.1160–1174.

[22] Xue, J. (1994), Edge-maximal triangulated subgraphs and heuristics for the maximum clique problem. *Networks*, Vol.24, pp.109–120.