Fast Algorithm for Generalized Multinomial Models with Ranking Data

Jiaqi Gu [1] **Guosheng Yin** [1]

Abstract

We develop a framework of generalized multinomial models, which includes both the popular Plackett–Luce model and Bradley–Terry model as special cases. From a theoretical perspective, we prove that the maximum likelihood estimator (MLE) under generalized multinomial models corresponds to the stationary distribution of an inhomogeneous Markov chain uniquely. Based on this property, we propose an iterative algorithm that is easy to implement and interpret, and is guaranteed to converge. Numerical experiments on synthetic data and real data demonstrate the advantages of our Markov chain based algorithm over existing ones. Our algorithm converges to the MLE with fewer iterations and at a faster convergence rate. The new algorithm is readily applicable to problems such as page ranking or sports ranking data.

1. Introduction

Aggregating pairwise comparison data, ranking data, rating data and discrete choice data is an important problem in many fields, including sports ranking (Elo, 1978; Deng et al., 2014), marketing research (McFadden, 1974; Kamishima, 2003; Kamishima & Akaho, 2006), election (Murphy & Martin, 2003; Moors & Vermunt, 2007), classification (Hastie & Tibshirani, 1998) and so on. Various models have been proposed to carry out such tasks, for example, the multinomial model for discrete choices, the Plackett–Luce model for rankings (Luce, 1959; Plackett, 1975) and the Bradley–Terry model for pairwise comparisons (Bradley & Terry, 1952; Huang et al., 2006).

Our work unifies all the aforementioned models in the framework of generalized multinomial models owning to their shared structures in the log-likelihood function. We theoretically show that the maximum likelihood estimator (MLE) under generalized multinomial models corresponds to the probability vector of the stationary distribution of an inhomogeneous Markov chain whose transition matrix is a function of the MLE itself. Motivated by Maystre & Grossglauser (2015), we develop a consistent, computationally efficient and easy-to-implement Markov chain based algorithm for MLE calculation. Our algorithm uniquely converges to the MLE under generalized multinomial models with fewer iterations than existing methods. Extensive experiments on synthetic data and real data reveal that our algorithm outperforms existing ones, such as the minorization–maximization (MM) algorithm (Hunter, 2004; Hunter & Lange, 2004) and the Bayesian weaver algorithm (Dong & Yin, 2018), in terms of the level of robustness, computational efficiency as well as statistical properties.

The rest of this paper is organized as follows. In Section 2, the framework of generalized multinomial models is established and existing methods for inference are reviewed. The details of our Markov chain based algorithm are provided in Section 3. Experiments on synthetic data and real data are conducted in Section 4 to compare our algorithm with existing ones in terms of computational efficiency and statistical properties. Section 5 concludes with discussions.

2. Generalized Multinomial Models

2.1. Framework

Consider d basic cells $c_1, \ldots, c_d$, where c_i is assigned with cell probability p_i ($\sum_{i=1}^d p_i = 1$). Suppose the counts of d basic cells are $a_1, \ldots, a_d$ respectively, then the likelihood function is

$$L(\mathbf{p}) = \prod_{i=1}^{d} p_i^{a_i}, \tag{1}$$

where $\mathbf{p} = (p_1, \ldots, p_d)^T$. Model (1) is known as the complete multinomial model. The completeness of model (1) originates from two conditions:

1. Union of disjoint candidate sets for each multinomial choice is fixed as $\mathscr{C} = \{c_1, \ldots, c_d\}$.

[1]Department of Statistics and Actuarial Science, University of Hong Kong, Hong Kong SAR, China. Correspondence to: Jiaqi Gu <u3005743@hku.hk>, Guosheng Yin <gyin@hku.hk>.

Proceedings of the 36^{th} International Conference on Machine Learning, Long Beach, California, PMLR 97, 2019.

2. Candidate sets for each multinomial choice all consist of only one basic cell, i.e., candidate sets take the shared form of $\{c_i\}$.

In other words, candidate sets for all multinomial choices are fixed as $\{c_1\}, \ldots, \{c_d\}$. Relaxing either of the above conditions would lead to an incomplete multinomial model, under which all multinomial choices share a common probability measure. With examples presented by candidate sets, we introduce three types of incomplete multinomial models in which one or both conditions are relaxed:

A. Candidate sets all consist of only one basic cell but the union of them is not $\mathscr{C}$: $\{c_2\}, \{c_4\}, \ldots, \{c_{2\times\lfloor d/2\rfloor}\}$.

B. Union of candidate sets is $\mathscr{C}$ but candidate sets all consist of more than one basic cell: $\{c_1, c_2\}, \{c_3, \ldots, c_d\}$.

C. Candidate sets all consist of more than one basic cell and the union of them is not $\mathscr{C}$: $\{c_1, c_2\}, \{c_3, c_4\}$ with $d > 4$.

Generalized multinomial models described below unify both the complete and incomplete multinomial models (type A to C). For each multinomial choice with candidate sets $S_1, \ldots, S_l$ where $S_{k_1} \cap S_{k_2} = \emptyset$ for all $k_1 \neq k_2$, the probability of selecting candidate set S_k is

$$P(S_k|S_1, \ldots, S_l) = \frac{\sum_{c_i \in S_k} p_i}{\sum_{k=1}^{l} \sum_{c_i \in S_k} p_i}. \tag{2}$$

Suppose for the j-th observation ($j = 1, \ldots, n$), the candidate sets are S_1^j, …, $S_{l_j}^j$, the union of candidate sets is $\mathcal{C}^j = \cup_{k=1}^{l_j} S_k^j$ and the selected candidate set is $A^j \in \{S_1^j, \ldots, S_{l_j}^j\}$, then the likelihood function under a generalized multinomial model is

$$\begin{aligned} L(\mathbf{p}) &= \prod_{j=1}^{n} P(A^j|S_1^j, \ldots, S_{l_j}^j) \\ &= \prod_{j=1}^{n} \frac{\sum_{c_i \in A^j} p_i}{\sum_{k=1}^{l_j} \sum_{c_i \in S_k^j} p_i} \\ &= \prod_{j=1}^{n} \frac{\sum_{c_i \in A^j} p_i}{\sum_{c_i \in \mathcal{C}^j} p_i}, \end{aligned} \tag{3}$$

and the corresponding log-likelihood function is

$$\ell(\mathbf{p}) = \sum_{j=1}^{n} \Big\{ \log\Big(\sum_{c_i \in A^j} p_i \Big) - \log\Big(\sum_{c_i \in \mathcal{C}^j} p_i \Big) \Big\}. \tag{4}$$

Any model with a log-likelihood function similar to (4) can be classified as a generalized multinomial model. Various generalized multinomial models have been proposed in the literature. One typical case is the Plackett–Luce model for ranking data (Luce, 1959; Plackett, 1975) which assumes that the ranking of cells is generated by sequentially selecting the most preferred cell each time from the remaining ones. Thus, a full ranking of d cells can be decomposed into $d-1$ sequential multinomial selections. Given m rankings $\pi_1, \ldots, \pi_m$ (each π_k here represents a vector of ranks of the cells), the log-likelihood function is

$$\ell(\mathbf{p}) = \sum_{k=1}^{m} \sum_{i=1}^{d-1} \Big\{ \log\big(p_{\pi_k^{-1}(i)}\big) - \log\Big(\sum_{l=i}^{d} p_{\pi_k^{-1}(l)} \Big) \Big\}, \tag{5}$$

where $\pi_k^{-1}(i)$ is the index of the cell with rank i in π_k. Other examples include the Bradley–Terry model for pairwise comparison data (Bradley & Terry, 1952; Huang et al., 2006), certain contingency table models (Chen & Fienberg, 1976) and the general counting experiment on random partitions.

2.2. Existing Methods

The main difficulty to obtain the MLE under generalized multinomial models is to optimize $\ell(\mathbf{p})$ in (4) subject to the constraint $\sum_{i=1}^{d} p_i = 1$, as $\ell(\mathbf{p})$ is not concave due to terms $-\log\big(\sum_{c_i \in \mathcal{C}^j} p_i\big)$ ($j = 1, \ldots, n$). As a result, the MLE equations are analytically and computationally intractable, and thus traditional optimization methods, such as the Newton–Raphson algorithm, cannot guarantee the convergence to the global maximum.

One popular approach to conquering such difficulty uses the MM algorithm (Hunter, 2004; Hunter & Lange, 2004) by constructing a concave surrogate function $Q(\mathbf{p}; \mathbf{p}^{(t)})$ based on the t-th update $\mathbf{p}^{(t)}$, where $Q(\mathbf{p}; \mathbf{p}^{(t)}) \leq \ell(\mathbf{p})$ holds for all $\mathbf{p}$. The optimization of $\ell(\mathbf{p})$ is implemented by continuously updating the surrogate function. The MM algorithm is the standard method to obtain the MLE under generalized multinomial models, although it bears low convergence speed. Another class of methods are fixed point algorithms on the basis of MLE equations, such as the Ford algorithm for pairwise comparisons (Ford, 1957) and the Bayesian weaver algorithm for incomplete multinomial data (Dong & Yin, 2018). Although these algorithms are easy to implement with per-iteration computational cost $O(nd)$, their applications are restricted. The Ford algorithm can only be implemented to pairwise comparison data, the weaver algorithm fails to cope with type B and type C incomplete multinomial models, and the Bayesian weaver algorithm needs a large number of iterations to converge.

Apart from frequentist approaches as mentioned above, Bayesian inference methods have also been developed, see Guiver & Snelson (2009); Caron & Doucet (2012); Caron & Teh (2012) for details. Motivated by the work in Maystre

& Grossglauser (2015), we establish a simple algorithm to obtain the MLE that maximizes the unified log-likelihood function in (4). Although the per-iteration computational cost of our algorithm is $O(nd^2)$, it is still computationally efficient due to its requirement of much fewer iterations to reach convergence.

3. Markov Chain Based Algorithm

To introduce our Markov chain based algorithm for generalized multinomial models, we first define two sets of indexes for each basic cell. Let $W_i = \{j : c_i \in A^j\}$ and $L_i = \{j : c_i \in (\mathcal{C}^j \backslash A^j)\}$ be the indexes of observations where c_i belongs to the selected candidate set and one of the unselected candidate sets, respectively. If we interpret each multinomial choice as a competition among several teams of basic cells, A^j is the team winning the j-th competition, and W_i and L_i correspond to the indexes of competitions where basic cell c_i is one of the winners or losers.

For the j-th observation ($j = 1, \ldots, n$), we define $q_j^+ = \sum_{c_i \in A^j} p_i$ and $q_j^* = \sum_{c_i \in \mathcal{C}^j} p_i$ to be the sums of cell probabilities in the selected candidate set and the union of candidate sets, respectively. As a result, the log-likelihood function (4) can be rewritten as

$$\ell(\mathbf{p}) = \sum_{j=1}^{n} \Big\{ \log(q_j^+) - \log(q_j^*) \Big\}.$$

The MLE equations

$$\frac{\partial \ell(\mathbf{p})}{\partial p_i} = 0 \quad i = 1, \ldots, d, \tag{6}$$

are equivalent to

$$\sum_{j \in W_i} \left(\frac{1}{q_j^+} - \frac{1}{q_j^*} \right) = \sum_{j \in L_i} \frac{1}{q_j^*} \quad i = 1, \ldots, d. \tag{7}$$

Multiplying both sides of (7) with p_i leads to

$$\sum_{i' \neq i} p_{i'} \left(\sum_{j \in W_i \cap L_{i'}} \frac{p_i}{q_j^+ q_j^*} \right) = \sum_{i' \neq i} p_i \left(\sum_{j \in L_i \cap W_{i'}} \frac{p_{i'}}{q_j^+ q_j^*} \right). \tag{8}$$

Assuming a transition matrix $\mathbf{\Sigma}$ parametrized by $\mathbf{p}$ as

$$\mathbf{\Sigma}(\mathbf{p}) = [\sigma_{ii'}(\mathbf{p})]_{d \times d},$$
$$\text{where} \quad \sigma_{ii'}(\mathbf{p}) \propto \sum_{j \in L_i \cap W_{i'}} \frac{p_{i'}}{q_j^+ q_j^*} \quad (i \neq i'),$$

the fact that MLE $\hat{\mathbf{p}}$ is a solution of equation (8) implies that $\hat{\mathbf{p}}$ corresponds to the stationary distribution of a discrete state Markov chain with transition matrix $\mathbf{\Sigma}(\hat{\mathbf{p}})$. This leads to Algorithm 1 and the guideline on its implementation is provided in the Supplementary Material.

Algorithm 1 Markov chain based algorithm

Input: Observations $\{(A^j, \mathcal{C}^j) : j = 1, \ldots, n\}$ and calculate $\{W_i, L_i\}$ for each c_i.
Initialize $\mathbf{p} = (1/d, \ldots, 1/d)^T$.
Initialize $\mathbf{\Sigma}(\mathbf{p}) = \mathbf{0}_{d \times d}$.
repeat
  for $i \in \{1, \ldots, d\}$ **do**
    for $i' \in \{1, \ldots, d\} \backslash \{i\}$ **do**
      Compute

$$\sigma_{ii'}(\mathbf{p}) \leftarrow \sum_{j \in L_i \cap W_{i'}} \frac{p_{i'}}{q_j^+ q_j^*}$$

    end for
  end for
  Compute $\sigma_{ii}(\mathbf{p})$ ($i = 1, \ldots, d$) and then normalize $\mathbf{\Sigma}(\mathbf{p})$ so that $\forall i$, $\sum_{i'=1}^{d} \sigma_{ii'}(\mathbf{p}) = 1$.
  $\mathbf{p} \leftarrow T(\mathbf{p})$ under the transition matrix $\mathbf{\Sigma}(\mathbf{p})$.
until convergence.

The transition matrix $\mathbf{\Sigma}(\mathbf{p})$ has an intuitive interpretation. Specifically, $\sigma_{ii'}(\mathbf{p})$ is the probability of transferring from state i to i' under a Markov chain with the following transition steps:

(i) The state remains unchanged as i with probability $\sigma_{ii}(\mathbf{p})$; otherwise, proceed to step (ii).

(ii) Select one index j from L_i with probability $(1/q_j^*)/(\sum_{j \in L_i} 1/q_j^*)$.

(iii) Select one cell $c_{i'}$ from A^j with probability $p_{i'}/q_j^+$.

Theorem 1 *Define $T(\mathbf{p})$ as the probability vector of the stationary distribution of a discrete state Markov chain with transition matrix $\mathbf{\Sigma}(\mathbf{p})$. The MLE $\hat{\mathbf{p}}$ is the unique solution to equation*

$$T(\mathbf{p}) = \mathbf{p}$$

if the Markov chain with transition matrix $\mathbf{\Sigma}(\mathbf{p})$ is ergodic.

Theorem 2 *For all $\mathbf{p} \in \{\mathbf{p} \geq \mathbf{0} : \sum_{i=1}^{d} p_i = 1\}$, define*

$$T^{k+1}(\mathbf{p}) = T \circ T^k(\mathbf{p}).$$

Under regularized conditions (given in the Supplementary Material),

$$\lim_{k \to \infty} T^k(\mathbf{p}) = \hat{\mathbf{p}}.$$

Theorems 1 and 2 guarantee that Algorithm 1 converges to the MLE $\hat{\mathbf{p}}$ regardless of the starting point. Proofs of both theorems are given in the Supplementary Material.

To make statistical inference under generalized multinomial models, we need to derive the observed information matrix,

$$\mathbf{I}^{\text{obs}}_{d\times d} = -\nabla^2 \ell(\mathbf{p}), \tag{9}$$

where the (i, i')-th element is

$$I^{\text{obs}}_{ii'} = -\frac{\partial^2 \ell(\mathbf{p})}{\partial p_i \partial p_{i'}} = \sum_{j\in W_i \cap W_{i'}} \left(\frac{1}{q_j^+}\right)^2 - \sum_{j\in (W_i\cup L_i)\cap (W_{i'}\cup L_{i'})} \left(\frac{1}{q_j^*}\right)^2.$$

4. Experiments

We conduct experiments on synthetic data: (a) to examine the statistical properties of Algorithm 1; (b) to compare the computational efficiency of our method with existing ones; and (c) to explore the factors that influence the computational efficiency of our method. We also apply our algorithm to real data to investigate its practical performance.

To evaluate the statistical properties of $\hat{\mathbf{p}}$, we consider two criteria: the Kullback–Leibler divergence of $\hat{\mathbf{p}}$ with respect to the true $\mathbf{p}$ for consistency and the Mahalanobis-distance between $\hat{\mathbf{p}}$ and $\mathbf{p}$ for asymptotic normality:

$$\text{KL-divergence: KL} = \sum_{i=1}^{d} \hat{p}_i \log\left(\frac{\hat{p}_i}{p_i}\right)$$
$$\text{Mahalanobis-distance: } D^2 = (\hat{\mathbf{p}} - \mathbf{p})^T \mathbf{I}^{\text{obs}} (\hat{\mathbf{p}} - \mathbf{p})$$

To measure the computational efficiency of the algorithms, we consider the number of iterations to converge, the running time and the convergence rate.

4.1. Statistical Properties

We carry out experiments with $d = 8$ basic cells, 4 cases of experimental setups, and 1,000 samples per case. The setup of each case is described as follows:

I: 280×2^k multinomial choices with candidate sets $\{c_1\}, \ldots, \{c_8\}$.

II: 5×2^k multinomial choices with candidate sets (1) $\{c_1\}, \{c_2\}, \{c_3\}$; (2) $\{c_1\}, \{c_2\}, \{c_4\}$; ...; (56) $\{c_6\}, \{c_7\}, \{c_8\}$.

III: 8×2^k multinomial choices with candidate sets (1) $\{c_1, c_2, c_3, c_4\}, \{c_5, c_6, c_7, c_8\}$; (2) $\{c_1, c_2, c_3, c_5\}, \{c_4, c_6, c_7, c_8\}$; ...; (35) $\{c_1, c_6, c_7, c_8\}, \{c_2, c_3, c_4, c_5\}$.

IV: 1×2^k multinomial choices with candidate sets (1) $\{c_1, c_2, c_3\}, \{c_4, c_5, c_6\}$; (2) $\{c_1, c_2, c_3\}, \{c_4, c_5, c_7\}$; ...; (280) $\{c_3, c_4, c_5\}, \{c_6, c_7, c_8\}$.

Case I corresponds to the complete multinomial model, while cases II to IV represent three different types of incomplete multinomial models discussed in Section 2.1. The sample size is set as $n = 280 \times 2^k$, where k takes values from 0 to 9 to allow n to increase.

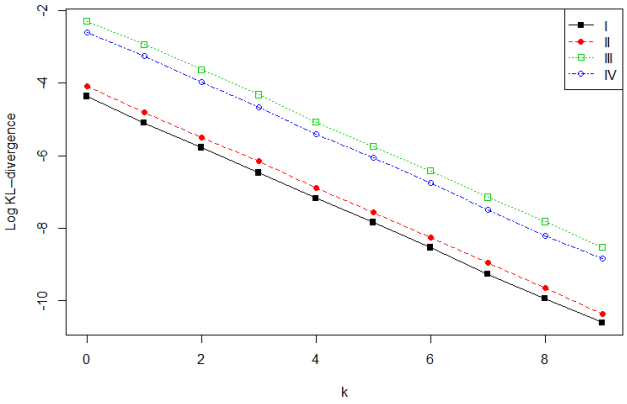

Figure 1. The log KL–divergence averaged over 1000 replications as the sample size increases

Figure 1 shows that the log KL-divergence of $\hat{\mathbf{p}}$ with respect to the true $\mathbf{p}$ decreases as the sample size increases in all cases, indicating that the estimator obtained by our algorithm is consistent.

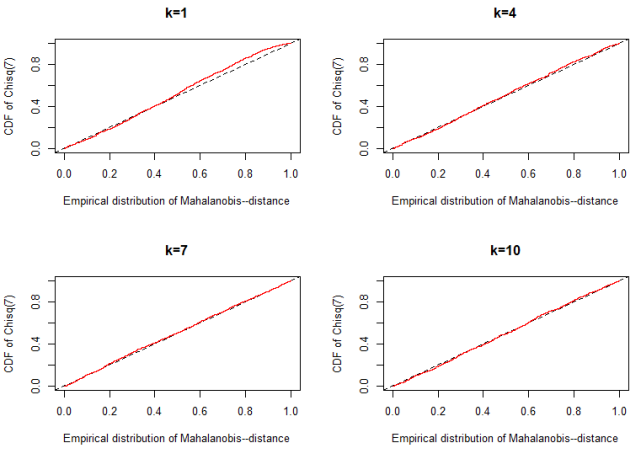

Figure 2. Probability–Probability plots of the Mahalanobis-distance and $\chi^2_{(7)}$ distribution as the sample size increases

Figure 2 shows that the distribution of the Mahalanobis-distance is close to $\chi^2_{(7)}$ as the sample size increases, indicating that our estimator is asymptotically normal.

4.2. Computational Efficiency

We compare the computational efficiency of our algorithm with the MM algorithm and Bayesian weaver algorithm, and investigate the factors that may influence Algorithm 1 in terms of computational efficiency.

4.2.1. Motivation: NASCAR Data

We first study the performance of these algorithms on the NASCAR data (Hunter, 2004). This dataset contains the full rankings of drivers in 36 NASCAR racing rounds in 2002, where 43 out of a total of 87 drivers competed in each round. Table 1 presents the number of iterations needed and the running time for different algorithms when fitting the Plackett–Luce model to the NASCAR data under the convergence condition $\|\mathbf{p}^{(t)} - \mathbf{p}^{(t-1)}\| \leq 10^{-6}$.

Table 1. Comparison of three algorithms (MM, Bayesian weaver, and Markov chain based algorithms) with the NASCAR data

Computational efficiency	MM	Bayesian weaver	Markov chain
No. of iterations	14	2166	7
Running time (s)	0.01	5.08	0.07

Clearly, the number of iterations needed for our Markov chain based algorithm is the smallest among the three algorithms, and its running time ranks the second. The smallest number of iterations needed by our algorithm originates from the fact that its convergence rate is the fastest, as displayed in Figure 3.

4.2.2. Factors Influencing the Convergence Rate

Empirical performance has demonstrated that our algorithm is computationally efficient, for which the underlying mechanisms still need investigation. The factors that may influence the computational efficiency of our algorithm involve the type and level of incompleteness in observed generalized multinomial data, the sample size, the total number of basic cells (d) and the number of different types of components in the log-likelihood function (4). We conduct experiments on synthetic data to examine which factors contribute to the fast convergence of our algorithm compared with others.

4.2.3. Type and Level of Incompleteness

As discussed in Section 2.1, there are two types of incompleteness in the generalized multinomial data:

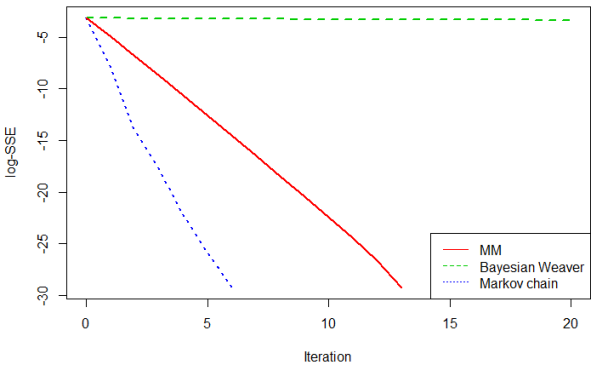

Figure 3. Path of the log-SSE over iterations for three algorithms when fitting the Plackett–Luce model to the NASCAR data (the sum of squared errors: SSE$= \|\mathbf{p}^{(t)} - \hat{\mathbf{p}}\|$)

1. Union of disjoint candidate sets is $\mathcal{C} \subset \mathscr{C}$ (small union incompleteness);

2. All the candidate sets consist of more than one basic cell (composite cell incompleteness).

These two types of incompleteness correspond to relaxation of the two conditions in Section 2.1 where the completeness of model (1) originates.

To study the influence of small union incompleteness in data on the computational efficiency of our algorithm, we implement experiments with $d = 8$ basic cells. The average size of the union of candidate sets $\mathcal{C}$ varies from 2 to 8 with step length 0.06 to create samples with severe incompleteness to full completeness. The level of small union incompleteness is represented by the ratio $|\mathcal{C}|/d$. The smaller the ratio, the stronger the incompleteness. For each level, 100 samples of size 280 are simulated. The average number of iterations needed for different algorithms to converge under different levels of small union incompleteness is exhibited in Figure 4. It shows that the lower the ratio $|\mathcal{C}|/d$, the larger number of iterations other algorithms need to converge, while the number of iterations for our algorithm to converge is stable as the ratio $|\mathcal{C}|/d$ changes. Even in the most severe situation where $|\mathcal{C}|/d$ is far smaller than 1, the number of iterations is less than 10. Thus, our algorithm can resist the negative impact of small union incompleteness in data.

To understand whether composite cell incompleteness in data affect our algorithm in computational efficiency, we include $d = 128$ basic cells in experiments and randomly divide them into several candidate sets so that each candidate set for multinomial choices consists of 2^k basic cells ($k = 1, \ldots, 6$). For each k, 100 samples of size 1000 are

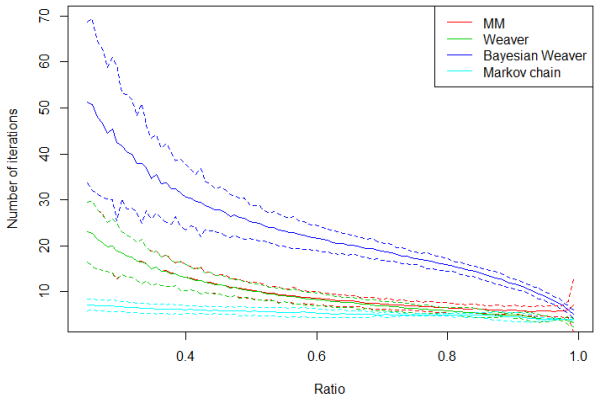

Figure 4. Number of iterations as $|\mathcal{C}|/d$ increases

created. In each sample, different multinomial choices are conducted on different divisions of basic cells while the numbers of basic cells in candidate sets are the same. It is natural that the number of basic cells in candidate sets (2^k) represents the level of composite cell incompleteness. The larger the value of 2^k, the more severe the incompleteness. The average number of iterations needed for different algorithms to converge under different levels of composite cell incompleteness is demonstrated in Figure 5. Although the number of iterations needed for the three algorithms increases as the number of basic cells in candidate sets increases, the advantage of our method over others is magnificent when 2^k is large. However, the advantage is not as notable in the situation where $|\mathcal{C}|/d$ is far smaller than 1.

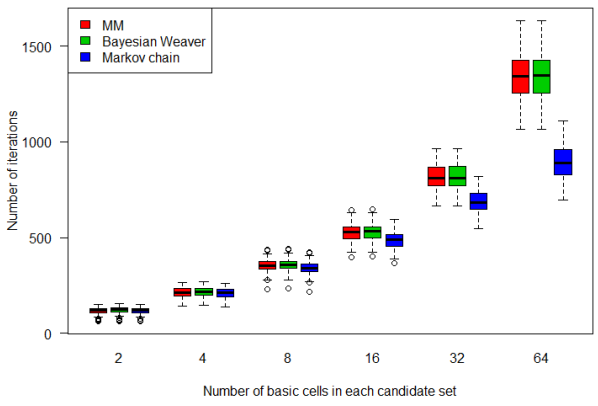

Figure 5. Number of iterations as the number of basic cells in candidate sets increases

The above experiments reveal that our algorithm requires fewer iterations than the MM algorithm and Bayesian weaver algorithm when there are incomplete multinomial observations in the data. The more severe the incompleteness, the more advantage our algorithm possesses in terms of computational efficiency. The advantage is mainly due to small union incompleteness, which has a strong negative impact on other algorithms but little on ours.

4.2.4. Sample Size

We consider cases II and III in Section 4.1 with $k = 0, 1, \ldots, 4$ (sample size $n = 280 \times 2^k$). For each combination of the case and k, 100 samples are generated and three algorithms are applied.

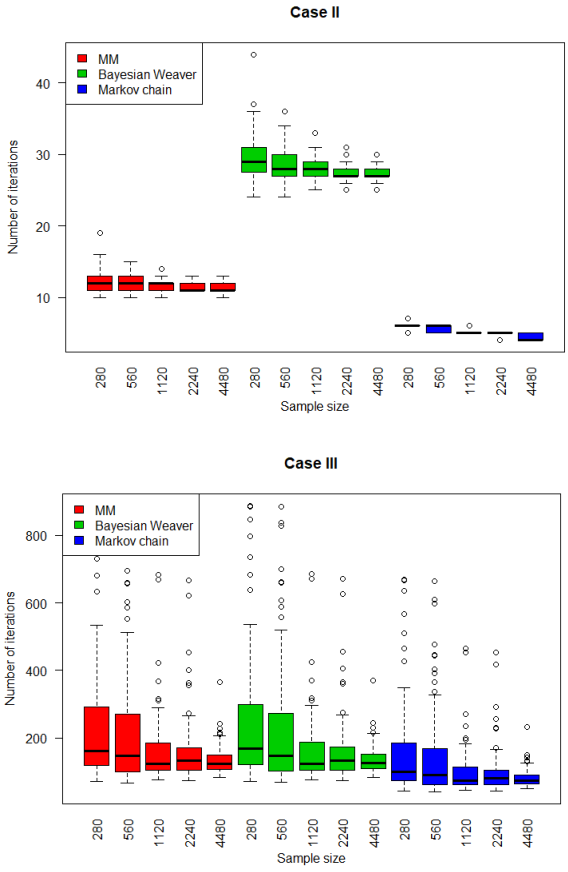

Figure 6. Number of iterations as the sample size increases

Figure 6 illustrates that the computational efficiency of the three algorithms does not change much as the sample size increases, although the variation in the number of iterations decreases. The relative efficiency of Algorithm 1 over others is more obvious in case II compared with case III.

4.2.5. The Number of Basic Cells

To examine the relationship between the computational efficiency and the number of items, the sample size and the type and level of incompleteness in data are kept unchanged as the number of basic cells varies. We execute experiments with a fixed sample size 1000 and two types of incompleteness in Section 4.2.3. The level of small union incompleteness is fixed as $|\mathcal{C}|/d = 1/8$, and for composite cell incompleteness, the number of basic cells in candidate sets is fixed as 8. The number of basic cells d varies from 16 to 128 with step length 8. For each combination of d and a type of incompleteness, 100 samples are created to compare the performance of our algorithm with the MM and Bayesian weaver algorithms.

Table 2. Average number of iterations needed by the three algorithms as the number of basic cells d increases

	Small union incompleteness			Composite cell incompleteness		
		Bayesian	Markov		Bayesian	Markov
d	MM	weaver	chain	MM	weaver	chain
16	18.0	39.7	6.3	436.0	438.3	275.6
24	11.9	27.6	6.0	404.6	407.5	316.0
32	9.8	23.9	6.0	400.4	403.7	337.9
40	8.5	21.8	6.0	383.7	386.9	338.0
48	8.1	20.8	6.0	392.1	395.5	354.5
56	7.4	20.0	6.0	378.9	382.3	348.0
64	7.1	19.5	6.0	371.4	374.8	345.0
72	7.0	19.1	6.0	367.3	370.6	344.2
80	6.7	18.6	6.0	376.8	380.0	356.1
88	6.3	18.4	6.0	366.8	369.9	348.2
96	6.3	18.4	6.0	354.4	357.5	337.9
104	6.1	18.1	6.0	359.2	362.2	343.8
112	6.0	17.9	6.0	357.4	360.5	343.2
120	6.0	17.9	6.0	357.6	360.6	344.4
128	6.0	17.6	6.0	354.4	357.3	342.1

Table 2 shows that when the ratio $|\mathcal{C}|/d$ or the number of basic cells in candidate sets is fixed, the advantage of our algorithm over others diminishes as d increases. Yet, our method still performs the best.

4.2.6. Number of Different Components in Likelihood Function

To study how our algorithm behaves when there are a varying number of different components in the likelihood function, we consider case II in Section 4.1 and split it into three subcases as follows:

IIA: 5 multinomial choices with candidate sets (1) $\{c_1\}$, $\{c_2\}$, $\{c_3\}$; (2) $\{c_1\}$, $\{c_2\}$, $\{c_4\}$; ...; (56) $\{c_6\}$, $\{c_7\}$, $\{c_8\}$.

IIB: Randomly select 28 combinations of candidate sets in subcase IIA, and make 10 multinomial choices for each.

IIC: Randomly select 14 combinations of candidate sets in subcase IIA, and make 20 multinomial choices for each.

Subcase IIA is equivalent to case II in Section 4.1. The numbers of different components in the likelihood function are 64, 36 and 22 for these subcases, respectively. Sample size of these three subcases are fixed at 280. The performance of three algorithms in these three subcases is presented in Figure 7.

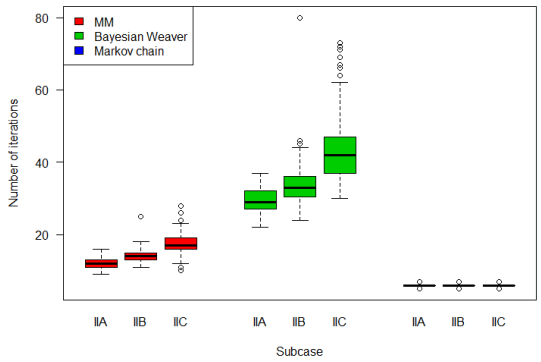

Figure 7. Number of iterations as the number of different components in the likelihood function increases

It can be seen that as the number of different components in the likelihood function decreases, the number of iterations of the MM and Bayesian weaver algorithm increases, but that of our method is more stable. In other words, our algorithm is robust against data sparsity.

4.3. Experiments on Real Datasets

To compare the empirical performance and scalability of Algorithm 1 with existing methods, we apply them to the well-known sushi datasets (Kamishima, 2003) and a Hong Kong Jockey Club (HKJC) horse racing dataset.

4.3.1. Sushi Datasets

The sushi datasets record responses in a questionnaire survey of preferences on sushi. There are two datasets in total. The first one contains 4926 full rankings of 10 specific types of sushi, and the second one records 5000 partial rankings of 100 types of sushi, where only the relative ordering of top 10 preferred is given in each partial ranking.

For each dataset, we apply our method and other algorithms (MM and Bayesian Weaver) to model the data in two ways:

fitting the Plackett–Luce model (PL) with rankings directly, or transforming rankings into pairwise comparisons to fit the Bradley–Terry model (BT). The number of iterations needed for convergence is presented in Table 3.

Table 3. The number of iterations of three algorithms to converge on the sushi datasets in different scenarios

SUSHI DATASET	MODEL	MM	BAYESIAN WEAVER	MARKOV CHAIN
FULL RANKING	PL	9	22	6
PARTIAL RANKING	PL	4	16	4
FULL RANKING	BT	35	84	5
PARTIAL RANKING	BT	40	78	4

Table 3 reveals that our Markov chain based algorithm is computationally more efficient than the other two algorithms, especially when fitting the Bradley–Terry model with pairwise comparisons. This result matches with our conclusion in Section 4.2.3 that the smaller the ratio $|\mathcal{C}|/d$, the larger the number of iterations needed for other algorithms, while our Markov chain based algorithm is stable on it.

We take the third scenario (full ranking with BT) in Table 3 to illustrate the convergence rate of the three algorithms. The result in Figure 8 is consistent with that in Figure 3, i.e., the Markov chain based algorithm has the fastest convergence rate.

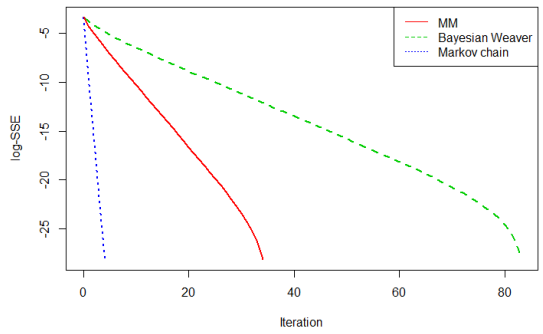

Figure 8. Path of the log-SSE over iterations for three algorithms in the third scenario of Table 3 (full ranking of sushi data with BT)

4.3.2. HKJC HORSE RACING DATASET

The HKJC horse racing dataset records full rankings of 1498 horse races in the Hong Kong Jockey Club throughout the 2008/09 and 2009/10 seasons. During these two seasons, 1677 horses and 92 jockeys participated at least one race. We apply our method and the MM algorithm to fit the Plackett-Luce model in three ways:

- HORSE ONLY: The probability that horse i wins is proportional to p_i ($d = 1677$).
- JOCKEY ONLY: The probability that jockey j wins is proportional to p_j ($d = 92$).
- HORSE + JOCKEY: The probability that the pair of horse i and jockey j wins is proportional to $p_i + p_j$ ($d = 1769$).

Note that small union incompleteness exists in all of three scenarios while composite cell incompleteness exists only in the third one. The number of iterations needed for convergence of different methods is exhibited in Table 4.

Table 4. The number of iterations of two algorithms to converge on HKJC horse racing dataset in different scenarios

SCENARIO	MM	MARKOV CHAIN
HORSE ONLY	178	12
JOCKEY ONLY	36	9
HORSE + JOCKEY	176	77

Table 4 tells that even when the total number of d is large, our Markov chain based algorithm still converges to the MLE with fewer iterations needed than the MM algorithm. This result is consistent with conclusions in Section 4.2.3 which demonstrates the scalability of our method to a large number of basic cells d.

The MLEs corresponding to the real datasets (NASCAR, sushi and HKJC horse racing) are provided in the Supplementary Material.

5. Conclusion

We develop an MLE algorithm for generalized multinomial models which solves the MLE equations by iteratively computing the stationary distribution of an inhomogeneous Markov chain. Our algorithm can be applied to many existing models with similar likelihood function forms, such as the Bradley–Terry model and the Plackett–Luce model. Experiments reveal that our algorithm converges to the MLE with fewer iterations than existing methods, especially when the average size of the union of candidate sets is significantly smaller than the total number of basic cells or the data are sparse. In real data analysis, our algorithm shows robustness in computational efficiency with respect to different types of data and different models, and it yields a faster convergence rate than existing methods.

Acknowledgement

We thank the four anonymous reviewers for insightful suggestions that have significantly improved the paper. This research is supported by the Research Grants Council of Hong Kong (17326316) and TCL Corporate Research (Hong Kong).

References

Bradley, R. A. and Terry, M. E. Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons. *Biometrika*, 39(3/4):324–345, 1952.

Caron, F. and Doucet, A. Efficient Bayesian Inference for Generalized Bradley-Terry Models. *Journal of Computational and Graphical Statistics*, 21(1):174–196, 2012.

Caron, F. and Teh, Y. W. Bayesian Nonparametric Models for Ranked Data. In Pereira, F., Burges, C. J. C., Bottou, L., and Weinberger, K. Q. (eds.), *Advances in Neural Information Processing Systems 25 (NIPS 2012)*, pp. 1520–1528. Curran Associates, Inc., 2012.

Chen, T. and Fienberg, S. E. The Analysis of Contingency Tables with Incompletely Classified Data. *Biometrics*, 32 (1):133, 1976.

Deng, K., Han, S., Li, K. J., and Liu, J. S. Bayesian Aggregation of Order-Based Rank Data. *Journal of the American Statistical Association*, 109(507):1023–1039, 2014.

Dong, F. and Yin, G. Maximum Likelihood Estimation for Incomplete Multinomial Data via the Weaver Algorithm. *Statistics and Computing*, 28(5):1095–1117, 2018.

Elo, A. E. *The Rating of Chessplayers, Past and Present*. Arco Pub., 1978.

Ford, L. R. Solution of a Ranking Problem from Binary Comparisons. *The American Mathematical Monthly*, 64 (8):28–33, 1957.

Guiver, J. and Snelson, E. Bayesian Inference for Plackett-Luce Ranking Models. In *Proceedings of the 26th Annual International Conference on Machine Learning (ICML 09)*, pp. 377–384, New York, NY, USA, 2009. ACM.

Hastie, T. and Tibshirani, R. Classification by Pairwise Coupling. *The Annals of Statistics*, 26(2):451–471, 1998.

Huang, T.-K., Weng, R. C., and Lin, C.-J. Generalized Bradley-Terry Models and Multi-Class Probability Estimates. *Journal of Machine Learning Research*, 7:85–115, 2006.

Hunter, D. R. MM Algorithms for Generalized Bradley-Terry Models. *The Annals of Statistics*, 32(1):384–406, 2004.

Hunter, D. R. and Lange, K. A Tutorial on MM Algorithms. *The American Statistician*, 58(1):30–37, 2004.

Kamishima, T. Nantonac Collaborative Filtering: Recommendation Based on Order Responses. In *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 03)*, pp. 583–588, New York, NY, USA, 2003. ACM.

Kamishima, T. and Akaho, S. Efficient clustering for orders. In *Proceedings of the 2nd International Workshop on Mining Complex Data (ICDMW 06)*, pp. 274–278, 2006.

Luce, R. D. *Individual Choice Behavior: A Theoretical analysis*. Wiley, 1959.

Maystre, L. and Grossglauser, M. Fast and Accurate Inference of Plackett–Luce Models. In Cortes, C., Lawrence, N. D., Lee, D. D., Sugiyama, M., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 28 (NIPS 2015)*, pp. 172–180. Curran Associates, Inc., 2015.

McFadden, D. Conditional Logit Analysis of Qualitative Choice Behavior. In Zarembka, P. (ed.), *Frontiers in Econometrics*, pp. 105–142. New York: Academic Press, 1974.

Moors, G. and Vermunt, J. Heterogeneity in Post-materialists Value Priorities. Evidence from a Latent Class Discrete Choice Approach. *European Sociological Review*, 23(5):631–648, 2007.

Murphy, T. B. and Martin, D. Mixtures of Distance-based Models for Ranking Data. *Computational Statistics & Data Analysis*, 41(3):645–655, 2003.

Plackett, R. L. The Analysis of Permutations. *Applied Statistics*, 24(2):193–202, 1975.