Bernoulli **25**(2), 2019, 1160–1188
https://doi.org/10.3150/17-BEJ1017

Mallows and generalized Mallows model for matchings

EKHINE IRUROZKI[1,2,*] BORJA CALVO[2] and JOSE A. LOZANO[1,2,**]

[1]*Basque Center for Applied Mathematics (BCAM), Mazarredo 14, 48009 Bilbao, Spain. E-mail:* [*]*eirurozki@bcamath.org;* [**]*jlozano@bcamath.org*
[2]*Intelligent Systems Group, University of the Basque Country, Manuel Lardizabal, 1, 20018 Donostia/San Sebastián, Spain. E-mail: borja.calvo@ehu.eus*

The Mallows and Generalized Mallows Models are two of the most popular probability models for distributions on permutations. In this paper, we consider both models under the Hamming distance. This models can be seen as models for matchings instead of models for rankings. These models cannot be factorized, which contrasts with the popular MM and GMM under Kendall's-τ and Cayley distances. In order to overcome the computational issues that the models involve, we introduce a novel method for computing the partition function. By adapting this method we can compute the expectation, joint and conditional probabilities. All these methods are the basis for three sampling algorithms, which we propose and analyze. Moreover, we also propose a learning algorithm. All the algorithms are analyzed both theoretically and empirically, using synthetic and real data from the context of e-learning and Massive Open Online Courses (MOOC).

Keywords: Generalized Mallows Model; hamming; learning; Mallows Model; matching; sampling

1. Introduction

Permutations appear naturally in a wide variety of domains, from Social Sciences [48] to Machine Learning [8]. Probability models over permutations are an active research topic in areas such as preference elicitation [7], information retrieval [17], classification [9], etc. Some prominent examples of these are models based on pairwise preferences [34], Placket–Luce [35,43], and Mallows Models [36]. One can find in the recent literature on distributions on permutations both theoretical discussions and practical applications, as well as extensions of all the aforementioned models.

In this paper, we focus on the Mallows Model (MM), an exponential model which depends on the definition of a distance for permutations. It is specified by the location parameter, σ_0, and a dispersion parameter, θ. There are multiple extensions of the MM. Some of the most popular extensions are non-parametric models [38], infinite permutations [21,39] and mixture models [14,40,42]. However, the Generalized Mallows Model (GMM) [19] is the most popular among all these extensions. It is also an exponential model, which instead of one single spread parameter, requires the definition of k spread parameters θ_j, each affecting a particular position of the permutation. In this way, it is possible to model a distribution with more emphasis on the consensus of certain positions of the permutation while having more uncertainty in some others.

The original definition of the MM included two different distances for permutations [36], namely Kendall's-τ and Spearman's-ρ. However, in [16] this definition was extended to six different distances, giving rise to the family of the distance-based probability models. The new distances considered were Hamming, Ulam, Cayley and Spearman's-footrule. Kendall's-τ has become the most recurrent in both theoretical and applied studies due to its nice theoretical properties. This attention of the community has led to several estimating and sampling processes of the MM and GMM under the Kendall's-τ distance [19,37].

Kendall's-τ is a natural measure of discrepancy between permutations when they are interpreted as rankings and therefore, MM and GMM under the Kendall's-τ distance are usually found in the preference domain. Nevertheless, Kendall's-τ is not adequate for measuring differences between matchings, despite being possible to represent both matchings and rankings with permutations. Moreover, other application fields, such as computer vision [30] or optimization [5], have also encouraged the search for new efficient probabilistic models for non-ranking permutation data. Recently, efficient algorithms for managing MM and GMM under the Cayley distance have been proposed [25]. The MM and GMM under the Cayley distance have already shown their utility in application problems such as the quadratic assignment problem (QAP) and the Permutation Flowshop Scheduling Problem (PFSP) [6]

The Hamming distance is one of the most popular metrics for permutations [15,29,45]. Clearly, it is not a natural measure of disagreement between rankings, but in the case when permutations represent matchings of bipartite graphs, Hamming is the most reasonable choice.

In this paper, we use for the first time the GMM under the Hamming distance. For both MM and GMM under the Hamming distance we propose efficient sampling and learning methods. In order to reasoning over permutations, we derive expressions for the computation of the partition function of the models, the expectation, the marginal and conditional probabilities.

This paper tries to bring MM and GMM to other domains apart from ranking so they can be as useful for non-ranking permutation data as they are in the ranking domain. In particular, we have focused on matchings. The differences between rankings and matchings are discussed throughout the manuscript, including decomposition of the distance, the sufficient statistics, the estimates, factorability, ... In this way, the notation of permutations and matchings will be used interchangeably. Special attention has been placed in the computational tractability of the operations, which are also empirically tested.

A good example of the applicability of the considered models is the multi-object tracking, problem which considers (1) a set of objects moving around and (2) a vision system consisting on some noisy sensors which are used to identify each of the objects and track them along time. Typical domains are sport players, people in an airport or in the street or animals in a controlled environment. In the particular context of a football match, suppose that there is a set of cameras continuously monitoring the players. This imperfect system distinguishes the players and tries to follow the track which each player is following along the match. In this context, a matching (represented by a permutation) $\sigma = \sigma(1), \ldots, \sigma(n)$ means that track 1 follows player $\sigma(1)$. When the players are not close to each others the system has no problem assigning a track to each player. However, when two players are close to each other, the noisy system can get confused. If there are several tracking systems, it can happen that different systems have different assignments. Under this context, it is reasonable to assume that the sample of matchings of the different sensors follow a MM of GMM for matchings. In other words, the sample is unimodal (there is one only assignment with maximum probability) and the probability of any other assignment different from the mode decays with its distance to the true assignment. Therefore, it is possible to aggregate the information of all the systems by fitting a MM or GMM under the Hamming distance. Moreover, the dispersion parameters of the model can be interpreted as a measure of confidence in the consensus, that is, the higher the dispersion parameter, the less probable that the system has been confused for two players.

The full-class set problem [32] is an example of classification problem where matchings arise. In the motivating application, a teacher who wants to automatically register the attendance at their lessons uses a set of photographs of the students to learn a classifier. Then, for the faces detected in a photograph of the whole classroom, the classifier individually predicts the identity of each face knowing that no identity can be predicted twice. In other words, the classifier predicts a matching between faces and photographs.

Another well-known example is the protein structure alignment problem [51]. For aligning two proteins, one has to find a transformation of the features of the first protein to the features of the second one. We can also think about it as a matching between the atoms in one protein and atoms in the other. This problem can be posed as a weighted bipartite matching problem and a distribution over the set of possible matchings is a way of capturing the uncertainty inherent to the problem.

In this paper, we consider the probability distributions for matchings as those describe above. Moreover, every domain in which permutations arise and the Hamming distance is the natural measure of discrepancies between them, MM and GMM under the Hamming distance will be more likely suitable than distributions designed for rankings. Throughout the paper, we include discussion comparing the distance-based models under the Hamming and the Kendall's-τ distances. The goal of the discussion is to support one of the main thesis of this paper, highlighting that the differences between models are bigger than their similarities. It turns out that there is no general method for the efficient computation of most expressions or factorization of general MM. This means that the derivation of the computationally efficient expressions for sampling and learning must be carefully carried out for each different distance. The algorithms to deal with MM and GMM under Kendall's-τ and Cayley distances are based on the possibility of factorizing such models. Since MM and GMM cannot be factorized, we make use of different machinery to obtain efficient sampling and learning algorithms.

The rest of the paper is organized as follows. Section 2 introduces the Hamming distance and gives the necessary basic definitions and operations for permutations. Section 3 introduces the probabilistic models considered. Section 3.1 gives efficient expressions for the normalization constants of both Mallows and Generalized Mallows models under the Hamming distance, which are the base for the sampling and learning procedures introduced in this paper. Section 4 introduces three sampling algorithms for the Mallows Model and two for the Generalized Mallows. Section 5 deals with the estimation of the parameters of the distribution given a dataset of permutations. In Section 7, the experimental evaluation is performed and Section 8 concludes the paper.

2. Permutations: Basic definitions and preliminary results

Permutations are bijections of the set of integers $\{1, \dots, n\}$ onto itself. They are usually denoted with the Greek letters σ or π. From now on, we will use a notation where $\sigma(i) = j$ means that item j is in position i and represents the permutation σ as $\sigma = \sigma(1)\sigma(2)\cdots\sigma(n)$. A special permutation which is worth mentioning is the identity permutation, $e = 123 \cdots n$ which maps each item j to position j.

By composing two permutations σ and π of n elements, we obtain a new permutation $\sigma \circ \pi$ such that $\sigma \circ \pi(i) = \sigma(\pi(i))$, which will be denoted as $\sigma\pi$. In general, the composition is not commutative. Some exceptions to this general rule include the composition of a permutation σ and its inverse σ^{-1}, which results in the identity, $\sigma\sigma^{-1} = \sigma^{-1}\sigma = e$, and the composition with the identity, $\sigma e = e\sigma = \sigma$.

The Hamming distance between two permutations $d(\sigma, \pi)$ counts the number of point-wise disagreements they have, $d(\sigma, \pi) = \sum_{j=1}^{n} \mathbb{1}_{\sigma(j) \neq \pi(j)}$ where $\mathbb{1}_A$ denotes the indicator function of a subset A. Its invariance property asserts that $d(\sigma, \pi) = d(\sigma\gamma, \pi\gamma)$ for every permutation γ. Particularly taking $\gamma = \pi^{-1}$ and since $\pi\pi^{-1} = e$ one can write, w.l.o.g., $d(\sigma, \pi) = d(\sigma\pi^{-1}, e)$. The distance from any permutation to the identity is denoted as an univariate function $d(\sigma\pi^{-1}, e) = d(\sigma\pi^{-1})$ that will simplify the notation. The implications of the invariance property are relevant since, as we will later explain, from now on we can w.l.o.g. assume that the reference permutation is the identity.

A recurrent concept in the permutation literature is that of *fixed point*, namely a position of the permutation in which $\sigma(j) = j$. In the same way, iff $\sigma(j) \neq j$ then j is an *unfixed point*. Therefore, the Hamming distance between a permutation σ and the identity, $d(\sigma)$, counts the number of unfixed points of σ, $d(\sigma) = \sum_{j=1}^{n} \mathbb{1}_{\sigma(j) \neq j}$.

Rather than the distance $d(\sigma)$, we will sometimes be interested in the sets of fixed and unfixed points of σ. This information is encoded on $\boldsymbol{H}(\sigma) = (H_1(\sigma), \dots, H_n(\sigma))$, a binary vector in which $H_j(\sigma) = 0$ iff $\sigma(j) = j$.

The $\boldsymbol{H}(\sigma)$ vector is referred to as the distance decomposition since $d(\sigma) = \sum_{j=1}^{n} H_j(\sigma)$. We will use the expressions "σ has a fixed point at j", $\sigma(j) = j$ and $H_j(\sigma) = 0$ interchangeably throughout the paper. The concept of derangement is also relevant for the understanding of this manuscript. A derangement is a permutation in which there are no fixed points.

Rankings and matchings. It is usually the case that both concepts are represented by permutations. However, the ranking involves the idea of an ordering among the items while the matching is a sequence of assignments. Along this paper, we will notice the differences between both concepts since the probability models for then behave very differently.

Let $G = (V, E)$ be a bipartite graph with $2 \times n$ vertices. A matching is a bijection between the items of the first set to the items of the second. The Hamming distance between the matching σ and the identity matching counts the number of edges which differ in σ and e. Note that for the edge from node j there is one such "correct" assignment and $n - 1$ "incorrect" assignments. This binary behaviour is captured by the terms of the distance decomposition vector.

In order to compare the Hamming and the Kendall's-τ distance decomposition vectors, let us introduce a classical process for ranking a set of n items. This method is carried out as a sequence of n stages. Let w_i be the probability of item i of being selected as the favourite. The process begins by randomly (proportionally to w_i) selecting the most preferred item for rank 1 in a first stage, then randomly (proportionally to w_i of the items not already selected) selecting the best among the remaining items for rank 2 in the second stage and so on. We can define a vector $\mathbf{V}(\sigma) = (V_1(\sigma), \dots, V_{n-1}(\sigma))$ where $V_j(\sigma)$ can be understood as a measure of how accurate the decision at stage j of the ranking process was [11]. In this way, $V_j(\sigma)$ ranges from 0 (correct decision, the most preferred item among the remaining ones was selected) to $n - j$ (the worst decision possible). The vector $\mathbf{V}(\sigma)$ is also known as inversion vector.

It is worth highlighting the following property for both distance decomposition vectors.

Property 1. While the terms of $H_j(\sigma)$ are not independent for a uniformly random permutation σ, the terms of the inversion vector $V_j(\sigma)$ are.

2.1. Results on enumerative combinatorics

Most results in this paper are based on combinatorial arguments. For example, it is recurrent the case of counting the number of different derangements of n items, $S(n)$. This expression [46,47] is

$$S(d) = (d-1) * S(d-1) + (d-1) * S(d-2)$$

with $S(0) = 1$ and $S(1) = 0$. It follows that the number of permutations of n items at Hamming distance d (in which d items are deranged) is

$$S(n,d) = \binom{n}{d} S(d).$$

The sequence $S(n,d)$ for every d can be computed in time $O(n)$.

Some results of the paper rely on the random generation of permutations at a given distance. The fact that recursive descriptions of combinatorial objects can be translated into algorithms of random generation is classical [18]. Several generators can also be found in the literature [26]. We refer the reader interested on more details to [25].

Note that the number of permutations that have fixed points at positions $1 \le i_1 < i_2 < \cdots < i_k \le n$ is the same as the number of permutations that have fixed points at positions $1, 2, \dots, k$. This count is denoted as $f(n,k)$ and it is easy to see that $f(n,k) = (n-k)!$.

The same situation happens when we are counting the permutations with unfixed positions, that is, the number of permutations such that have at least k unfixed points at positions $1 \le i_1 < i_2 < \cdots < i_k \le n$ is the same as the number of permutations that have unfixed points at positions $1, 2, \dots, k$ (being positions $k+1, \dots, n$ either fixed or unfixed). This count is denoted as $g(n,k)$ and can be computed using an inclusion-exclusion approach [47].

$$g(n,k) = n! + \sum_{i=1}^{k} (-1)^i \binom{k}{i} f(n,i) = n! + \sum_{i=1}^{k} (-1)^i \frac{k!(n-i)!}{i!(k-i)!}. \tag{2.1}$$

Equation (2.1) counts the number of permutations with at least k unfixed points in time $O(n)$. A detailed derivation of this expression can be found in [25].

2.2. Frequency matrix

The frequency matrix, F, is a summary of a sample of permutations $\{\sigma_1, \dots, \sigma_m\}$. This matrix is defined as an $n \times n$-dimensional matrix where $F_{i,j}$ counts the number of permutations in the sample such that have item j at position i, that is, $F_{i,j} = \sum_{s=1}^{m} \mathbb{1}_{\sigma_s(i)=j}$. Matrix F contains the sufficient of the MM and GMM as we will show later. Moreover, it can be used to compute $d(\sigma_s, \sigma_0)$ for every σ_s in the sample efficiently as follows.

Lemma 1. *The sum of the distances from σ_0 to each of the permutations in the sample $\{\sigma_1, \dots, \sigma_m\}$ can be computed by means of the frequency matrix F as follows.*

$$\sum_{s=1}^{m} d(\sigma_s, \sigma_0) = \sum_{s=1}^{m}\sum_{j=1}^{n} H_j\left(\sigma_s\sigma_0^{-1}\right) = \sum_{j=1}^{n}(m - F_{\sigma_0^{-1}(j),j}).$$

The proof of Lemma 1 is given in the supplemental article [27].

2.3. Algebraic results

The efficient computation of several statistical quantities considered in this paper are based on the calculation of the Elementary Symmetric Polynomials (ESP) on the parameters of the distribution, $\boldsymbol{\theta} = (\theta_1, \dots, \theta_n)$. In this section, we introduce the notion of ESP and show how to compute them efficiently. Moreover, we will give an expression for their derivatives.

First of all, let us define the set $R_k = \{(i_1, \dots, i_k) | 1 \le i_1 < i_2 < \cdots < i_k \le n\}$ as the set of all the different groups of k ordered indices out of the n total indices $\{1, \dots, n\}$. The ESP of degree k on a set of n variables, $\gamma_k(X_1, \dots, X_n)$, is defined as follows:

$$\gamma_k(X_1, \dots, X_n) = \sum_{r \in R_k}\prod_{j \in r} X_j.$$

By abusing notation, γ_k will be used to denote the ESP of degree k, $\gamma_k = \gamma_k(X_1, \dots, X_n)$. The ESP γ_k can be efficiently computed with the following recursion [2]:

$$\gamma_k(X_1, \dots, X_n) = \begin{cases} 1 & \text{if } k = 0, \\ \sum_{j=1}^{n} X_j & \text{if } k = 1, \\ \gamma_k(X_1, \dots, X_{n-1}) + \gamma_{k-1}(X_1, \dots, X_{n-1})X_n & \text{otherwise.} \end{cases} \tag{2.2}$$

By using the above formula, the computational complexity for computing the ESP $\gamma_k(X_1, \dots, X_n)$ is $O(n^2)$. Note that a naive computation will require time $O(2^n)$.

Splitting the ESP. In both learning and sampling processes, it is necessary to compute the ESP on all the subgroups of $n-1$ variables. The naive approach is to take each of the n possible subgroups of $n-1$ variables and compute the ESP of every subgroup as shown in Equation (2.2). This will require $O(n^3)$ time. We introduce a method for computing the ESP on all the subgroups of $n-1$ variables in $O(n^2)$.

In the following lines, there is an example of the elementary symmetric polynomial on 4 variables. Each γ_k is computed by adding up every product inside the braces.

$$\gamma_1 = \begin{cases} \gamma_1^1 = \begin{cases} X_1 \end{cases} \\ \bar{\gamma}_1^1 = \begin{cases} X_2 \\ X_3 \\ X_4 \end{cases} \end{cases} \qquad \gamma_2 = \begin{cases} \gamma_2^1 = \begin{cases} X_1X_2 \\ X_1X_3 \\ X_1X_4 \end{cases} \\ \bar{\gamma}_2^1 = \begin{cases} X_2X_3 \\ X_2X_4 \\ X_3X_4 \end{cases} \end{cases}$$

$$\gamma_3 = \begin{cases} \gamma_3^1 = \begin{cases} X_1X_2X_3 \\ X_1X_2X_4 \\ X_1X_3X_4 \end{cases} \\ \bar{\gamma}_3^1 = \begin{cases} X_2X_3X_4 \end{cases} \end{cases} \qquad \gamma_4 = \begin{cases} \gamma_4^1 = \begin{cases} X_1X_2X_3X_4 \end{cases} \\ \bar{\gamma}_4^1 = \begin{cases} 0. \end{cases} \end{cases}$$

As we can see, the terms are divided into two groups. By γ_k^i we denote the subset of the terms in γ_k that include the term X_i, and by $\bar{\gamma}_k^i$ the terms in γ_k that do not include X_i. Recall that $R_k = \{(i_1, \dots, i_k) | 1 \le i_1 < i_2 < \dots < i_k \le n\}$. We can now state that:

$$\gamma_k^i = \sum_{r \in A} \prod_{j \in r} X_j \qquad \text{where } A = \{R \subseteq R_k | i \in R\},$$

$$\bar{\gamma}_k^i = \sum_{r \in A} \prod_{j \in r} X_j \qquad \text{where } A = \{R \subseteq R_k | i \notin R\}.$$

We introduce a recursion for computing γ_k^i and $\bar{\gamma}_k^i$ for every $1 \le i, k \le n$ given γ_k in time $O(n^2)$. It is based on the following two relations:

$$\gamma_k^i = \bar{\gamma}_{k-1}^i X_i \qquad \forall i \in \{1, \dots, n\}, \tag{2.3}$$

$$\bar{\gamma}_k^i = \gamma_k - \gamma_k^i \qquad \forall i \in \{1, \dots, n\}. \tag{2.4}$$

The recursive algorithm for computing γ_k^i and $\bar{\gamma}_k^i$ for every $1 \le i, k \le n$ given γ_k is as follows. Let the base cases be $\gamma_0^i = \bar{\gamma}_0^i = 1$ and let γ_k for $1 \le k \le n$ be computed as shown in Equation (2.2). Equations (2.3) and (2.4) define a recursive procedure to compute γ_k^i and $\bar{\gamma}_k^i$ for all $1 \le i, k \le n$ in $O(n^2)$.

Derivatives. In the case of our probabilistic models, the variables in the ESP are exponential functions of the form $X_i = (\exp(\theta_i) - 1)$. Thus, the techniques for computing the derivatives explained in [2] are not valid here. We give here an efficient expression for the first derivatives with respect to θ_i which can be obtained by the chain rule:

$$\frac{\partial \gamma_k}{\partial \theta_i} = \frac{\partial \gamma_k}{\partial X_i} \cdot \frac{\partial X_i}{\partial \theta_i} = \bar{\gamma}_{k-1}^i \exp(\theta_i). \tag{2.5}$$

3. Mallows and generalized Mallows model

The Mallows Model (MM) is an exponential model for permutations based on distances. The MM can be expressed as follows:

$$p(\sigma) = \frac{\exp(-\theta d(\sigma, \sigma_0))}{\psi(\theta)} \qquad \text{where } \psi(\theta) = \sum_{\sigma} \exp\big(-\theta d(\sigma, \sigma_0)\big). \tag{3.1}$$

Here $\theta \in \mathbb{R}$ is a spread parameter, σ_0 is the central permutation, $d(\sigma, \sigma_0)$ represents a distance between σ and σ_0 and $\psi(\theta)$ is the partition function. For a probability model on matchings, the central permutation can be denoted as central matching. Note that when the dispersion parameter θ is greater than 0, then σ_0 is the mode, and the closer a permutation σ is to σ_0, the larger $p(\sigma)$. On the other hand, with $\theta = 0$, we obtain the uniform distribution and when $\theta < 0$, then σ_0 is the anti-mode.

One can easily find situations in which the disagreement between permutations depend, not only in the number of discrepancies, but also in the positions of those discrepancies. Consider the example in the Introduction in which the matchings provided by several sensor systems is to be aggregated into a single, consensus matching. Suppose that player 1 has not crossed with other player along the match and therefore, most systems agree in the track associated to player 1. However, players 2 and 3 have been very close to each other most of the time, so there is a larger uncertainty regarding their tracks. Such situation can be modelled under the Generalized Mallows Model (GMM), an extension of the MM. This is done by defining different spread parameter to the track of each player. As far as the authors know, this is the first time that the GMM has been considered under the Hamming distance.

In the same way as MM, the GMM is an exponential model and relies on a distance for permutations. The main difference between both is that, while the MM uses a single spread parameter, the GMM uses k spread parameters, each affecting a particular position of the permutation. Moreover, in order to base the GMM on a particular distance, this distance must be decomposed as a sum on k terms. In the case of the Hamming distance, the distance decomposition vector, defined in Section 2.1, has dimension n. It follows that we can use the GMM under the Hamming distance by defining an n-dimensional dispersion parameter, $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_n)$. It is defines as follows:

$$p(\sigma) = \frac{\exp(-\sum_{j=1}^{n} \theta_j H_j(\sigma\sigma_0^{-1}))}{\psi(\boldsymbol{\theta})} \qquad \text{where } \psi(\boldsymbol{\theta}) = \sum_{\sigma} \exp\left(-\sum_{j=1}^{n} \theta_j H_j(\sigma)\right).$$

The efficient management of the MM and GMM under Cayley and Kendall's-τ distances rely in the factorization of the distributions. Unfortunately, neither MM nor GMM under the Hamming distance cannot be factorized since the Hamming distance cannot be posed as a sum of n independent terms, as shown in Property 1 (page 1164) [19]. Moreover, even the naive computation of the partitioning function of MM and GMM is intractable for $n > 10$ in a reasonable time regardless of the distance considered. This means that, in order to handle distributions of permutations of medium-big size, it is necessary to rely in other alternatives.

Recall that the frequency matrix F defined in Section 2.2 contains the sufficient statistics of the MM and GMM.

Rankings and matchings. The GMM has been used so far under the Kendall's-τ and Cayley distances. In both cases, due to Property 1, the distributions are factorable and have tractable expressions. Moreover, all the machinery employed for obtaining the expressions of the former models are not valid for the MM and GMM under the Hamming distance.

It is also worth noticing how each θ_j can be interpreted. If a matching σ comes from a distribution with central matching σ_0 where $\sigma_0(i) = j$, the larger θ_j the more likely that $\sigma(i) = j$. This contrasts with the intuition behind the GMM under the Kendall's-τ distance, with parameters $\boldsymbol{\theta}$ and central ranking σ_0, where $\sigma_0(j) = i$. In this context, the larger θ_j, the more likely that $\sigma(j) \le i$ (or, in other words, the larger the probability that item j is ranked in the first i positions in σ).

In the ranking domain, the matrix F is generally denoted as *rank marginal matrix*. Curiously, for the MM under the Kendall's-τ distance, the classical model for the ranking domain, the rank marginal matrix does not contain the sufficient statistics. It turns out that in that context it is the *precedence matrix* which contains the sufficient statistic, that is, a square matrix M such that M_{ij} counts the number of permutations σ in which $\sigma(i) < \sigma(j)$ (meaning that item i is ranked before j).

3.1. Partition function for the Mallows model

The naive computation of the partition functions in the MM and GMM sums over $n!$ permutations. Clearly, this sum is an important bottle-neck. Fortunately, a closed form for the partition function for the MM under the Hamming distance is given in [19] which is as follows:

$$\psi(\theta) = n! \exp(-\theta n) \sum_{k=0}^{n} \frac{(\exp(\theta) - 1)^k}{k!}. \tag{3.2}$$

The computational complexity of Equation (3.2) is $O(n)$ which highly improves the naive complexity of $O(n!)$.

3.2. Normalization constant for the generalized Mallows model

An efficient expression for the normalization constant for the GMM can be found by relating the normalization constant to the moment generating function of the distance decomposition vector.

In particular, the process of finding the efficient expression for the normalization starts by expressing it as a function of the moment generating function (MGF) of $\mathbf{H}(\sigma) = (H_1(\sigma), \ldots, H_n(\sigma))$ under the uniform distribution. Then, we will find an computationally cheap expression for the probability generating function (PGF) by making use of a Taylor expansion. Finally, we will relate both PGF and MGF.

The notation used is as follows. Let $\boldsymbol{\varepsilon}(\sigma) = (\varepsilon_1(\sigma), \ldots, \varepsilon_n(\sigma))$ be a binary vector such that $\varepsilon_j(\sigma) = 1 - H_j(\sigma)$. Assuming that σ comes from the uniform distribution, then both $\mathbf{H}(\sigma)$ and $\boldsymbol{\varepsilon}(\sigma)$ are binary random vectors. We denote by $P_0(\boldsymbol{\varepsilon}(\sigma) = \boldsymbol{\varepsilon})$ the probability under the uniform distribution of a permutation σ of having the distance decomposition $\mathbf{1}-\boldsymbol{\varepsilon}$. Being the multivariate (joint) MGF of the random vector $\mathbf{X}$ defined as $M_{\mathbf{X}}(\mathbf{t}) = E[\prod_{j=1}^{n}\exp(t_j X_j)]$, the normalization constant, $\psi(\boldsymbol{\theta})$, can be posed as a function of the MGF of $\boldsymbol{\varepsilon}(\sigma)$ coming from the uniform distribution:

$$\begin{aligned}
\psi(\boldsymbol{\theta}) &= n! \sum_{\mathbf{H}\in\{0,1\}^n} P_0\big(\mathbf{H}(\sigma)=\mathbf{H}\big)\exp\Big(-\sum_j \theta_j H_j(\sigma)\Big)\\
&= n! \sum_{\boldsymbol{\varepsilon}\in\{0,1\}^n} P_0\big(\boldsymbol{\varepsilon}(\sigma)=(\varepsilon_1,\dots,\varepsilon_n)\big)\exp\Big(-\sum_j \theta_j (1-\varepsilon_j)\Big)\\
&= n! \sum_{\boldsymbol{\varepsilon}\in\{0,1\}^n} P_0\big(\boldsymbol{\varepsilon}(\sigma)=(\varepsilon_1,\dots,\varepsilon_n)\big)\exp\Big(-\sum_j \theta_j\Big)\exp\Big(\sum_j \theta_j\varepsilon_j\Big)\\
&= n!\exp\Big(-\sum_j \theta_j\Big) \sum_{\boldsymbol{\varepsilon}\in\{0,1\}^n} P_0\big(\boldsymbol{\varepsilon}(\sigma)=(\varepsilon_1,\dots,\varepsilon_n)\big)\exp\Big(\sum_j \theta_j\varepsilon_j\Big)\\
&= n!\exp\Big(-\sum_j \theta_j\Big) M_{\boldsymbol{\varepsilon}}(\boldsymbol{\theta}).
\end{aligned}$$

Note that there cannot be any $\mathbf{H}(\sigma)$ such that $\sum_{j=1}^{n} H_j(\sigma)=1$. Therefore, the probability of such a vector is zero.

We consider the multivariate case of the PGF of $\boldsymbol{\varepsilon}(\sigma)$ under the uniform distribution, which is defined as follows:

$$f_{\boldsymbol{\varepsilon}}(\mathbf{t}) = f_{\boldsymbol{\varepsilon}}(t_1,\dots,t_n) = E\big[t_1^{\varepsilon_1}\cdots t_n^{\varepsilon_n}\big] = \sum_{(\varepsilon_1,\dots,\varepsilon_n)} P_0\big(\boldsymbol{\varepsilon}(\sigma)=(\varepsilon_1,\dots,\varepsilon_n)\big)t_1^{\varepsilon_1}\cdots t_n^{\varepsilon_n}.$$

The Taylor expansion of a multivariate function $f(\mathbf{t})$ at $\mathbf{t}=\mathbf{1}$ is:

$$f_{\boldsymbol{\varepsilon}}(\mathbf{t}) = \sum_{k=0}^{\infty}\frac{1}{k!}\sum_{x_1+\cdots+x_n=k}\binom{k}{x_1\cdots x_n}\frac{\partial^k f_{\boldsymbol{\varepsilon}}}{\partial t_1^{x_1}\cdots\partial t_n^{x_n}}\bigg|_{\mathbf{t}=\mathbf{1}}(t_1-1)^{x_1}\cdots(t_n-1)^{x_n}.$$

In order to give the Taylor expansion for $f_{\boldsymbol{\varepsilon}}$, we need to know its derivative for variable t_i:

$$\begin{aligned}
\frac{\partial f_{\boldsymbol{\varepsilon}}}{\partial t_i} &= \sum_{(\varepsilon_1,\dots,\varepsilon_n)} P_0\big(\boldsymbol{\varepsilon}(\sigma)=(\varepsilon_1,\dots,\varepsilon_n)\big)t_1^{\varepsilon_1}\cdots\varepsilon_i t_i^{\varepsilon_i-1}\cdots t_n^{\varepsilon_n}\\
&= \sum_{(\varepsilon_1,\dots,\varepsilon_n)|\varepsilon_i=0} P_0\big(\boldsymbol{\varepsilon}(\sigma)=(\varepsilon_1,\dots,\varepsilon_n)\big)t_1^{\varepsilon_1}\cdots 0 t_i^{0-1}\cdots t_n^{\varepsilon_n}\\
&\quad + \sum_{(\varepsilon_1,\dots,\varepsilon_n)|\varepsilon_i=1} P_0\big(\boldsymbol{\varepsilon}(\sigma)=(\varepsilon_1,\dots,\varepsilon_n)\big)t_1^{\varepsilon_1}\cdots 1 t_i^{1-1}\cdots t_n^{\varepsilon_n}\\
&= 0 + \sum_{(\varepsilon_1,\dots,\varepsilon_n)|\varepsilon_i=1} P_0\big(\boldsymbol{\varepsilon}(\sigma)=(\varepsilon_1,\dots,\varepsilon_n)\big)\prod_{j\neq i} t_j^{\varepsilon_j}.
\end{aligned}$$

Its evaluation around $\mathbf{t} = (1, \dots, 1)$ is:

$$\frac{\partial f_{\boldsymbol{\varepsilon}}}{\partial t_i}\Big|_{\mathbf{t=1}} = \sum_{(\varepsilon_1,\dots,\varepsilon_n)|\varepsilon_i=1} P_0\big(\boldsymbol{\varepsilon}(\sigma) = (\varepsilon_1, \dots, \varepsilon_n)\big) 1^{\varepsilon_1} \cdots 1 \cdot 1^{1-1} \cdots 1^{\varepsilon_n}$$
$$= \sum_{(\varepsilon_1,\dots,\varepsilon_n)|\varepsilon_i=1} P_0\big(\boldsymbol{\varepsilon}(\sigma) = (\varepsilon_1, \dots, \varepsilon_n)\big) 1.$$

Note that this is equivalent to the probability under the uniform distribution of a permutation with a fixed point at position i, that is, the number of permutations of $n-1$ items divided by $n!$, $(n-1)!/n!$. The second order derivative with respect to t_i equals 1. The second order cross partial derivatives equal the probability under the uniform distribution of a permutation σ in which i_1 and i_2 are fixed points, that is, the number of permutations of n items with fixed points in i_1 and i_2 divided by $n!$

$$\frac{\partial^2 f_{\boldsymbol{\varepsilon}}}{\partial t_{i_1} \partial t_{i_2}}\Big|_{\mathbf{t=1}} = \sum_{(\varepsilon_1,\dots,\varepsilon_n)|\varepsilon_{i_1}=1,\varepsilon_{i_2}=1} P_0\big(\boldsymbol{\varepsilon}(\sigma) = (\varepsilon_1, \dots, \varepsilon_n)\big) 1^{\varepsilon_1} \cdots 1 \cdot 1^{1-1} \cdots 1^{\varepsilon_n}.$$

Then, in general, the kth order cross partial derivatives equal:

$$\frac{\partial^k f_{\boldsymbol{\varepsilon}}}{\partial t_{i_1} \cdots \partial t_{i_k}}\Big|_{\mathbf{t=1}} = \frac{(n-k)!}{n!}.$$

Since $(\varepsilon_1, \dots, \varepsilon_n) \in \{0,1\}^n$, then $\binom{k}{\varepsilon_1 \cdots \varepsilon_n} = k!$ and since $\partial^k f_{\boldsymbol{\varepsilon}} / \partial t_i^k = 0$ for $k > 1$, the Taylor series cannot be expanded more than $n+1$ terms. Therefore, the Taylor expansion around $\mathbf{1}$ can be equivalently written as (recall that $R_k = \{(i_1, \dots, i_k) | 1 \leq i_1 < i_2 < \cdots < i_k \leq n\}$):

$$f_{\boldsymbol{\varepsilon}}(\mathbf{t}) = \sum_{k=0}^{n} \frac{1}{k!} \sum_{r \in R_k} \binom{k}{\varepsilon_{i_1} \cdots \varepsilon_{i_n}} \frac{\partial^k f_{\boldsymbol{\varepsilon}}}{\partial t_{i_1} \cdots \partial t_{i_k}}\Big|_{\mathbf{t=1}} \prod_{i \in r} (t_i - 1)$$
$$= \sum_{k=0}^{n} \frac{1}{k!} \sum_{r \in R_k} \frac{k!(n-k)!}{n!} \prod_{i \in r} (t_i - 1)$$
$$= \sum_{k=0}^{n} \frac{(n-k)!}{n!} \sum_{r \in R_k} \prod_{i \in r} (t_i - 1).$$

If γ_k denotes the Elementary Symmetric Polynomial (ESP) of degree k, then $\gamma_k((t_1 - 1), \dots, (t_n - 1)) = \sum_{r \in R_k} \prod_{i \in r} (t_i - 1)$ and therefore:

$$f_{\boldsymbol{\varepsilon}}(\mathbf{t}) = \sum_{k=0}^{n} \frac{(n-k)!}{n!} \gamma_k\big((t_1 - 1), \dots, (t_n - 1)\big).$$

An efficient formulation for the computation of ESP, $\gamma_k((t_1-1),\dots,(t_n-1))$ is given in Section 2.3. Note that $M_{\boldsymbol{\varepsilon}}(\mathbf{t}) = E[\prod_{j=1}^n \exp(t_j\varepsilon_j)] = E[\prod_{j=1}^n \exp(t_j)^{\varepsilon_j}]$ and $f_{\boldsymbol{\varepsilon}}(\mathbf{t}) = E[\prod_{j=1}^n t_j{}^{\varepsilon_j}]$. It follows that $M_{\boldsymbol{\varepsilon}}(\mathbf{t}) = f_{\boldsymbol{\varepsilon}}(\exp(\mathbf{t}))$.

Thus, the probability generating function can be given as follows.

$$M_{\boldsymbol{\varepsilon}}(\mathbf{t}) = \sum_{k=0}^{n} \frac{(n-k)!}{n!}\gamma_k\big(\big(\exp(t_1)-1\big),\dots,\big(\exp(t_n)-1\big)\big). \tag{3.3}$$

Finally, the normalization constant can thus be given as follows:

$$\begin{aligned}\psi(\boldsymbol{\theta}) &= n!\exp\Big(-\sum_j \theta_j\Big)M_{\boldsymbol{\varepsilon}}(\boldsymbol{\theta}) \\ &= \exp\Big(-\sum_j \theta_j\Big)\sum_{k=0}^{n}(n-k)!\gamma_k\big(\big(\exp(\theta_1)-1\big),\dots,\big(\exp(\theta_n)-1\big)\big).\end{aligned} \tag{3.4}$$

The computational complexity of Equation (3.5) is $O(n^3)$, n times the complexity of the computation of the ESP.

3.3. Expected value, marginal and conditional probabilities

In this section, we deal with the expressions for the expected value of the distance in MM and the expected value of the distance decomposition vector in GMM. We also consider the marginal and conditional probabilities for the GMM under the Hamming distance based on Equation (3.3). Since the MM is a particular case of the GMM in which every θ_j has equal value, the results of the GMM model can be applied for both.

The following two theorems deal with the expected value of the distance and distance decomposition vector.

Theorem 1. *The expected value of the distance under the MM with the Hamming distance and dispersion parameter θ is as follows*:

$$E_\theta[d] = \frac{n\sum_{k=0}^{n}\frac{(\exp(\theta)-1)^k}{k!} - \exp(\theta)\sum_{k=0}^{n-1}\frac{(\exp(\theta)-1)^k}{k!}}{\sum_{k=0}^{n}\frac{(\exp(\theta)-1)^k}{k!}}.$$

Theorem 2. *The expected value of the distance decomposition vector $\mathbf{H}(\sigma)$ under the GMM with the Hamming distance and dispersion parameter $\boldsymbol{\theta}$ is as follows*:

$$E_{\boldsymbol{\theta}}[\mathbf{H}] = \left(1 - \frac{\sum_{k=1}^{n}(n-k)!\exp(\theta_1)\bar{\gamma}_{k-1}^{1}}{\sum_{k=0}^{n}(n-k)!\gamma_k},\dots,1 - \frac{\sum_{k=1}^{n}(n-k)!\exp(\theta_n)\bar{\gamma}_{k-1}^{n}}{\sum_{k=0}^{n}(n-k)!\gamma_k}\right),$$

where $\gamma_k = \gamma_k((\exp(\theta_1)-1),\dots,(\exp(\theta_n)-1))$ and $\bar{\gamma}_k^i = \bar{\gamma}_k^i((\exp(\theta_1)-1),\dots,(\exp(\theta_n)-1))$ denotes the ESP of degree k of the previous set of variables except for $(\exp(\theta_i)-1)$.

Proof. The expected distance under the an exponential model can be derived from the MGF in the following way [19].

$$E_\theta[D] = \frac{\partial \operatorname{Ln} M_D(t)}{\partial t}\Big|_{t=-\theta}.$$

The expected distance decomposition vector is then as follows.

$$\begin{aligned} E_{\boldsymbol{\theta}}[\mathbf{H}] &= \sum_{\boldsymbol{b}} P(\mathbf{H} = \boldsymbol{b})\boldsymbol{b} = E_{\boldsymbol{\theta}}[\mathbf{1} - \boldsymbol{\varepsilon}] = \mathbf{1} - E_{\boldsymbol{\theta}}[\boldsymbol{\varepsilon}] \\ &= \mathbf{1} - \left(\frac{\partial \operatorname{Ln} M_{\boldsymbol{\varepsilon}}(\mathbf{t})}{\partial t_1}\Big|_{\mathbf{t}=\boldsymbol{\theta}}, \dots, \frac{\partial \operatorname{Ln} M_{\boldsymbol{\varepsilon}}(\mathbf{t})}{\partial t_n}\Big|_{\mathbf{t}=\boldsymbol{\theta}} \right). \end{aligned}$$ □

Due to the factorial size of permutation spaces, the naive computation of the marginal distribution of a GMM under the Hamming distance is infeasible. We introduce a method for computing such marginal distributions by adapting the reasoning used in Equation (3.5) to sum over the subset of permutations of interest.

We consider two disjoint sets of items, A and B, and two sets of permutations, $\operatorname{fix}(A)$ and $\operatorname{unfix}(B)$. The set $\operatorname{fix}(A)$ includes every permutation in which j is a fixed point for every $j \in A$. We define the set $\operatorname{unfix}(B)$ in the same way, as the set of permutations that have an unfixed point at position $j \in B$. Recall that by $g(n,k)$ we denote the number of permutations of n items in which there are at least k unfixed points and a recursion to compute it is given in Equation (2.1).

Theorem 3. *Let $a = |A|$ and $b = |B|$. The marginal distribution of the set of permutations in which every $i \in A$ is a fixed point and every $j \in B$ is an unfixed point – the permutations in the intersection* $\operatorname{fix}(A) \cap \operatorname{unfix}(B)$ *– is as follows*:

$$\begin{aligned} \sum_{\sigma \in \operatorname{fix}(A) \cap \operatorname{unfix}(B)} p(\sigma) &= \frac{\sum_{\sigma \in \operatorname{fix}(A) \cap \operatorname{unfix}(B)} \exp(\sum_{j=1}^n -\theta_j H_j(\sigma))}{\psi(\boldsymbol{\theta})} \\ &= \frac{\exp(-\sum_{j \notin A} \theta_j) \sum_{k=0}^{n-a-b} g(n-a-k, b) \bar{\gamma}_k^{AB}(T_1, \dots, T_n)}{\exp(-\sum_j \theta_j) \sum_{k=0}^n (n-k)! \gamma_k(T_1, \dots, T_n)}, \end{aligned} \tag{3.5}$$

where $T_i = (\exp(\theta_i) - 1)$. *Note that by* $\bar{\gamma}_k^{AB}((\exp(\theta_1) - 1), \dots, (\exp(\theta_n) - 1))$ *we denote the ESP of degree k of the set of variables* $\{(\exp(\theta_j) - 1) | j \notin A \cup B\}$.

Proof in the supplemental article [27].

Regarding the computational complexity of this calculation, two different aspects must be considered. On the one hand, the complexity of computing $g(n-a, k+b)$ is $O(n-a)$. On the other hand, the complexity of computing $\bar{\gamma}_k^{AB}$ is $O((n-a-b)^2)$. Therefore, the complexity of the calculation of the marginal $\sum_{\sigma \in \operatorname{fix}(A) \cap \operatorname{unfix}(B)} p(\sigma)$ is $O(\max\{(n-a), (n-a-b)^2\})$.

It is well known that the conditional distribution can posed in terms of the joint distribution as follows:

$$p(X|Y) = \frac{p(X \cap Y)}{p(Y)}.$$

This definition can be easily translated to the GMM under the Hamming distance as stated in the following lemma.

Corollary 1. *Let A, A', B and B' be four disjoint sets of items. The probability of the permutations having fixed points at positions $j \in A'$ and unfixed points at positions $j \in B'$, given that the items $j \in A$ are fixed points and the items $j \in B$ are unfixed points, is as follows.*

$$\begin{aligned} &p\big(\mathrm{fix}\big(A'\big) \cap \mathrm{unfix}\big(B'\big) | \mathrm{fix}(A) \cap \mathrm{unfix}(B)\big) \\ &\quad = \frac{\sum_{\sigma \in \mathrm{fix}(A \cup A') \cap \mathrm{unfix}(B \cup B')} p(\sigma)}{\sum_{\sigma \in \mathrm{fix}(A) \cap \mathrm{unfix}(B)} p(\sigma)}. \end{aligned} \tag{3.6}$$

The computational complexity in this case is thus the same as the complexity of the marginal computation.

4. Sampling

In this section, we show how to generate permutations from both MM and GMM. We introduce here three sampling algorithms: The first one generates samples from an approximate distribution of MM and GMM, the second one generates samples from both MM and GMM while the last one generates only from the MM.

The three algorithms generate samples assuming that the mode is the identity, $\sigma_0 = e$. In case $\sigma_0 \neq e$, one can move a sample centered around e to be centered around σ_0 as follows. Let $\{\pi_1, \dots, \pi_m\}$ be a sample of permutations centered around the identity. A sample $\{\sigma_1, \dots, \sigma_m\}$ centered around σ_0 can be obtained from the previous one by composing each permutation with σ_0, that is, $\sigma_i = \pi_i \sigma_0$ for every $1 \leq i \leq m$.

4.1. Gibbs sampling algorithm

We have adapted the Gibbs sampler to generate samples from the MM and GMM. This algorithms samples a distribution which gets closer to the original one at each step, so in the limit the target distribution is sampled. The Gibbs algorithm proceeds as follows:

1. Generate a permutation σ u.a.r.
2. Build a new permutation σ' equal to σ in all but two positions chosen u.a.r. These two positions are swapped.
3. Let $\beta = \min\{1, p(\sigma')/p(\sigma)\}$. With probability β the algorithm accepts the candidate permutation moving the chain to σ', $\sigma = \sigma'$, and goes back to step (2). Otherwise, it discards σ' and goes back to step (2).

The initial samples are discarded (burn-in period) until the Markov chain approaches its stationary distribution and so samples from the chain are samples from the distribution of interest. Then, the above process in repeated until the algorithm generates a given number of permutations.

Algorithm 1: Random generation of $\boldsymbol{H}(\sigma)$ from the MM or GMM

```
Input: $\theta$ (resp. $\boldsymbol{\theta}$) dispersion parameters in MM (resp. GMM)
Output: $\boldsymbol{H}(\sigma)$ random distance decomposition vector
$A = B = \varnothing$;
for $i \leftarrow 1$ to $n$ do
    prob $= p(\text{fix}(i)|\text{fix}(A) \cap \text{unfix}(B))$ as shown in Corollary 1;
    with probability prob /* $j$ is fixed */
        $h_i(\sigma) = 0$;
        $A = A \cup \{i\}$

    otherwise /* $i$ is unfixed */
        $h_i(\sigma) = 1$;
        $B = B \cup \{i\}$;
    end
end
```

The computation of each permutation has complexity $O(n)$. It is thus a quick algorithm. However, we should remark that this is an approximate algorithm.

4.2. Chain sampling algorithm

We propose a method for generating permutations from a MM or a GMM based on the chain rule. The process of generating each permutation can be divided in two stages. In the first stage, it randomly generates a distance decomposition vector, $\boldsymbol{H}(\sigma)$. The sampling finishes by uniformly at random generating a permutation σ consistent with the given $\boldsymbol{H}(\sigma)$.

The generation of the random distance decomposition vector $\boldsymbol{H}(\sigma)$ is carried out with the well known chain rule for probability distributions. The chain rule expresses the joint distribution as a product of conditional distributions.

In this way, the Chain sampling algorithm uses the conditional probabilities of the GMM given in Equation (3.6) to sample each of the terms in the decomposition vector $\boldsymbol{H}(\sigma)$.

The complete process for the generation of $\boldsymbol{H}(\sigma)$ can be found in Algorithm 1. The process of randomly generating a permutation finishes by generating uniformly at random a permutation σ consistent with $\boldsymbol{H}(\sigma)$. Some discussion on the topic can be found in Section 2.1. A careful implementation of the Chain algorithm leads to a computational complexity of $O(n^2)$. Details can be found in the supplemental article [27].

4.3. Distances sampling algorithm

The Distances sampler can be used for generating from the MM under the Hamming distance but not for generating from the GMM.

The probability under the MM of a permutation at distance d is as follows:

$$p(d) = \sum_{\sigma | d_h(\sigma, \sigma_0^{-1}) = d} p(\sigma) = S_h(n,d) \frac{\exp(-\theta d)}{\psi(\theta)}, \tag{4.1}$$

where $S_h(n,d)$ denotes the number of permutations at distance d. Note that the normalization constant $\psi(\theta) = \sum_\sigma \exp(-\theta d_h(\sigma))$ can be expressed as the sum of n terms in the following way:

$$\psi(\theta) = \sum_{d=0}^{n} S_h(n,d) \exp(-\theta d). \tag{4.2}$$

Therefore, the Distances sampling process divides the process of generating a permutation from a given MM in two different stages as follows:

1. Randomly select the distance d considering the probabilities of Equation (4.1).
2. Uniformly at random generate one permutation among those at distance d from the identity. For a discussion on the random generation of permutations at a prescribed Hamming distance see Section 2.1.

Therefore, the time complexity of the generation of each permutation using this method is $O(n)$ given the sequence $S_h(n,d)$. Summarizing, this is a fast algorithm for the generation of exact samples from the MM model. Unfortunately, this algorithm cannot generate samples from the GMM model.

5. Learning

The parameters of a probability distribution are traditionally fitted via maximum likelihood estimation. In [13], it is stated that the maximum likelihood estimate of the parameters of a L-decomposable distribution, such as the MM, can be done by iterative scaling. Unfortunately, this only includes the dispersion parameters.

For a sample of m i.i.d. permutations $\{\sigma_1, \ldots, \sigma_m\}$ the MLE for the parameters of the distribution are those which maximize the likelihood function. Even though the MM is a particular case of the GMM, the MLE for the parameters of the distribution are different for each model. Therefore, we focus on each model separately.

5.1. Mallows model

The log-likelihood expression for the MM model is as follows:

$$\operatorname{Ln} \mathcal{L}\big(\{\sigma_1, \sigma_2, \ldots, \sigma_m\} | \sigma_0, \theta\big) = -m\theta\bar{d} - m \operatorname{Ln} \psi(\theta), \tag{5.1}$$

where $\bar{d} = \sum_{i=1}^{m} d(\sigma_i \sigma_0^{-1})/m$. By looking at Equation (5.1), we can see that calculating the value of σ_0 that maximizes the equation is independent of θ. Therefore the MLE estimation problem can be posed as a two step process in which first the central matching is obtained and then the dispersion parameter for the given $\hat{\sigma}_0$ is computed.

Central matching. The MLE for the central matching is given by the permutation that minimizes the sum of the Hamming distances to the sample. Let us pose the problem in a different way. Let F be the frequency matrix of the sample – the sufficient statistics – and consider the bipartite graph interpretation of the sample, both of which are defined in Section 2.2.

The problem of finding the permutation that minimizes the Hamming distance to the sample is equivalent to finding the maximum weighted bipartite matching. Finding that matching is known as the linear assignment problem (LAP) [3], and it is solved by selecting one entry of F per row and column in such a way that their sum is maximum. We denote this MLE for the central matching as σ_{LAP}. The Hungarian algorithm [31] solves this problem in $O(n^3)$.

Dispersion parameter. Once the central matching is known, the MLE for the dispersion parameter is obtained by taking the derivative in Equation (5.1) respect θ and equaling to zero.

$$\frac{d(-m\theta\bar{d} - m \operatorname{Ln}\psi(\theta))}{d\theta} = 0 \quad \rightarrow \quad -m\bar{d} - m\frac{d\psi(\theta)/d\theta}{\psi(\theta)} = 0.$$

The MLE for θ is then given by the θ that satisfies the next equation:

$$\frac{\exp(\theta)\sum_{k=0}^{n-1}\frac{(\exp(\theta)-1)^k}{k!} - n\sum_{k=0}^{n}\frac{(\exp(\theta)-1)^k}{k!}}{\sum_{k=0}^{n}\frac{(\exp(\theta)-1)^k}{k!}} + \bar{d} = 0. \tag{5.2}$$

Although no closed form for θ in Equation (5.2) exists, root finding algorithms such as Newton–Raphson can efficiently recover θ.

To sum up, the maximum likelihood parameters of a MM model can be done in polynomial time.

5.2. Generalized Mallows model

In this section, we describe the maximum likelihood estimation process of a given sample coming from a GMM model. The log-likelihood can be expressed as follows:

$$\begin{aligned}
\operatorname{Ln}\mathcal{L}\bigl(\{\sigma_1,\sigma_2,\dots,\sigma_m\}|\sigma_0,\boldsymbol{\theta}\bigr) &= \sum_{i=1}^{m}\operatorname{Ln} p(\sigma_i|\sigma_0,\boldsymbol{\theta}) \\
&= \sum_{i=1}^{m}\left(\sum_{j=1}^{n} -\theta_j H_j\bigl(\sigma_i\sigma_0^{-1}\bigr) - \operatorname{Ln}\psi(\boldsymbol{\theta})\right) \\
&= m\sum_{j=1}^{n} -\theta_j\bar{H}_j - m\operatorname{Ln}\psi(\boldsymbol{\theta}),
\end{aligned} \tag{5.3}$$

where $\bar{H}_j = \sum_{i=1}^{m} H_j(\sigma_i\sigma_0^{-1})/m$. Note that the permutation that maximizes the likelihood does need to be the same as the permutation that minimizes the sum of the distances to the permutations in the sample, that is, the central permutation for the MM may not be the same for the GMM.

In contrast to the MM estimation process, learning the GMM cannot be divided into different stages for the estimation of the different parameters θ_j and σ_0. Therefore, an exact algorithm needs to search for every parameter at the same time. However, it is interesting to show how to compute the dispersion parameters, θ_j, given the central matching, σ_0.

Let us bear in mind that the frequency matrix F is a sufficient statistic for σ_0. Moreover, the vector $(\bar{H}_1, \dots, \bar{H}_n)$ is the sufficient statistic for $\boldsymbol{\theta}$ for a given σ_0, whose expression is given by equaling to zero the derivative of the likelihood with respect to θ_i. In other words, given a permutation σ_0, the MLE for $\boldsymbol{\theta}$ is that which satisfies the following system of equations:

$$\frac{\sum_{k=1}^{n}(n-k)!\exp(\theta_i)\bar{\gamma}_{k-1}^{i}}{\sum_{k=0}^{n}(n-k)!\gamma_k} + \bar{H}_i = 0, \qquad 1 \le i \le n. \tag{5.4}$$

Recall that γ_k denotes the ESP $\gamma_k((\exp(\theta_1)-1), \dots, (\exp(\theta_n)-1))$, and $\bar{\gamma}_{k-1}^{i}$ denotes the ESP of all variables except for $(\exp(\theta_i)-1)$, which is the ESP of degree $k-1$ of the set of variables $\{(\exp(\theta_1)-1), \dots, (\exp(\theta_n)-1)\} \setminus \{(\exp(\theta_i)-1)\}$. The efficient computation of γ_k and a fast algorithm for obtaining $\bar{\gamma}_k^i$ given γ_k^i can be found in Section 2.3. Note that there is no closed-form expression for the dispersion parameters in Equation (5.4), the system of equations must be solved with numerical methods such as the multidimensional Newton–Raphson.

Summarizing, by expressing $\boldsymbol{\theta}$ as a function of σ_0 with Equation (5.4), we have posed the learning of a GMM as a combinatorial problem of finding the permutation σ_0 that maximizes Equation (5.3).

Although the computational complexity of estimating the parameters of a GMM under the Hamming distance is not known, the authors conjecture that it is NP-hard. We propose an approximate algorithm for fitting a GMM which is based on its asymptotic properties.

Rankings and matchings. Opposite to the case of the GMM under Kendall's-τ or Cayley distances, there likelihood cannot be factorized (as a consequence of Property 1), page 1164.

5.3. Approximate MLE for the GMM

We propose an approximate algorithm for obtaining the MLE for the parameters from a sample of permutations generated from a GMM under the Hamming distance, $\{\sigma_1, \dots, \sigma_m\}$. This algorithm is based on the asymptotic properties of the model. The problem statement is as follows. Let F be the frequency matrix on a sample $\{\sigma_1, \dots, \sigma_m\}$ and let σ_{LAP}^m be the solution to the LAP in F. The rest of the section is based on the following theorem.

Theorem 4. *The permutation σ_{LAP}^m is an asymptotically unbiased estimator for the central matching of* $\{\sigma_1, \dots, \sigma_m\}$.

The proof can be found in the supplemental article [27].

Based on Theorem 4 we propose using σ_{LAP}^m as an approximate MLE for the consensus permutation. As we have stated, LAP can be solved in polynomial time [31]. There also exist several implementations for the LAP such as [28].

The MLE for the dispersion parameters are obtained by solving the system of equations in (5.4). A multidimensional Newton–Raphson implementation is provided in [44].

Rankings and matchings. Learning the central ranking is known as the Kemeny problem [1], which happens to be NP-hard (in contrast to the polynomial complexity of learning the maximum weighted bipartite matching). There exists an approximation to the Kemeny ranking by the name of Borda which offers a factor 5 approximation of the optimal ranking [10] and also an asymptotically optimal estimator of the real central ranking [19]. It can be computed with the preference matrix and it is based on averaging the rankings of each item and ordering the items according to those averages. This contrasts with the idea behind the Hungarian algorithm, which iterates looking for the largest entry in the columns and rows of F.

The fact that the MLE for the central matching is equivalent to the maximum weighted bipartite matching (to a linear assignment) confirms the intuition that the MM and GMM under the Hamming distance is indeed a model for matchings.

6. Conditional independence, partial permutations and Bayesian interpretation

Independence for distributions on permutations has been stated in several ways [12,13,23,24]. We focus on the conditional independence that naturally arise from the L-decomposability property which was shown in [12]. For any σ resulting from the ranking process introduced in page 2, the L-decomposability [11] is satisfied iff the probability of selecting an item at stage k depends solely on the items remaining at that stage, not on the order of the already selected items.

$$p(\sigma) = \prod_{k=1}^{n-1} p_{\{i_k,\dots,i_n\}}\big(\sigma(k)\big).$$

The L-decomposability property induces the following conditional independence relation among items: For all k, given the set of items receiving the first k ranks, the ordering of these items and the ordering of the remaining items are independent.

$$p\big(\sigma^{-1}(k) = i_k | \sigma^{-1}(1) = i_1, \dots, \sigma_{k-1}^{-1}(k-1) = i_{k-1}\big) = p_{\{i_k,\dots,i_n\}}(i_k).$$

Let σ be a random permutation. The probability model p is TL-decomposability [12] if the following holds:

$$p(\sigma\pi) \text{ is L-decomposable for every } \pi.$$

Both MM and GMM under the Hamming distance are TL-decomposable.

Partial permutations are among the most referred extensions of general permutations, particularly when n rockets. Among the sampling algorithms presented in this paper, the Multistage sampler can generate partial permutations. Fitting a model consisting on partial permutation is also possible with the algorithms presented in this paper.

The most natural way of introducing prior information about the parameters of the model is the Bayesian framework. The MM and GMM under the Bayesian perspective have already been considered [20,22,39–41]. In [50], the Bayesian approach is analyzed in the case of the Kendall's-τ distance, although most of their study is applicable for any right invariant distance such as Hamming. They define two non-uniform prior distributions, one for the central permutation and other for the dispersion parameter. Then, assuming independence among both distributions, they calculate the posterior joint distribution. They also propose computationally tractable methods for Bayesian inference and their approach could be adapted for the Hamming distance.

7. Experiments

This section is devoted to show the efficiency of the proposed algorithms in terms of computational time and accuracy. We will first deal with the sampling algorithms and then with the learning algorithms. The code used to run the experiments has been made public in the CRAN repository by the name of PerMallows. A manuscript introducing it can be found in [26]. Moreover, the supplemental article [27] shows how easy it is to fit and sample distributions a GMM with it.

7.1. Sampling

In this paper, we propose three different learning algorithms. The Distances algorithm generates samples from the MM, the Chain algorithm from the MM and GMM while the Gibbs algorithm, on the other hand, generates samples from approximations of both MM and GMM. The sampling experiments are designed to compare the performance of the algorithms with three different criteria.

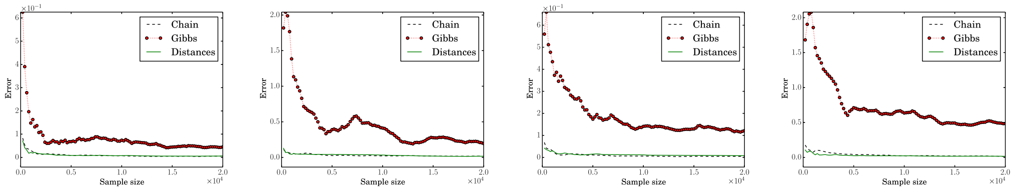
1. The evolution of the error as the sample size, m, increases.
2. The evolution of the error as the computational time increases.

The error in the MM is measured as the sum of the differences between the expected distance, $E[D]$, and the actual distances of the permutations in the samples. In the GMM the error is measured as the sum of the differences between the expected H_j, $E[H_j]$, and the actual $\bar{H}_j$. The expression of the expectations are given in Theorems 1 and 2.

For each of the different evaluation criteria, the following procedure has been carried out.

1. For each particular setting of n and θ (resp. $\boldsymbol{\theta}$) generate several samples of size $m = 200, 400, 600, \ldots, 19\,800, 20\,000$ with each of the sampling algorithms. Measure the error of each sample and plot the results.
2. For each particular setting of n and θ (resp. $\boldsymbol{\theta}$) generate several samples for $t = 1, 2, 3, \ldots, 14, 15$ seconds with each of the sampling algorithms. Measure the error of each sample and plot the results.

The parameter setting is as follows. The number of items in the permutations considered are $n \in \{5, 50, 100, 150\}$. The dispersion parameters in the MM case are $\theta \in \{0.1, 0.5, 1, 2, 3\}$. In the GMM case, the first of the dispersion parameters is $\theta_1 \in \{0.1, 0.5, 1, 2, 3\}$ while the rest are set such that $\theta_j = \theta_1 - (j - 1)(\theta_1/2(n - 1))$ for $j > 1$, that is, θ_1 is the largest parameter while the



(a) $n = 50, \theta = 0.1$ (b) $n = 50, \theta = 2$ (c) $n = 150, \theta = 0.1$ (d) $n = 150, \theta = 2$

Figure 1. Error (as the sum of the differences between the expected distance, $E[D]$, and the actual distance of the sample, in the Y-axis) of each algorithm as the sample size (X-axis) grows for different θ and n in MM. Average of 10 repetitions.

value of the rest decrease linearly to $\theta_n = \theta_1/2$. For each parameter configuration 10 experiments are run and the average results of them are given. The central permutation is the identity. The Gibbs algorithm discards the first n^2 permutations as part of the burning-period.

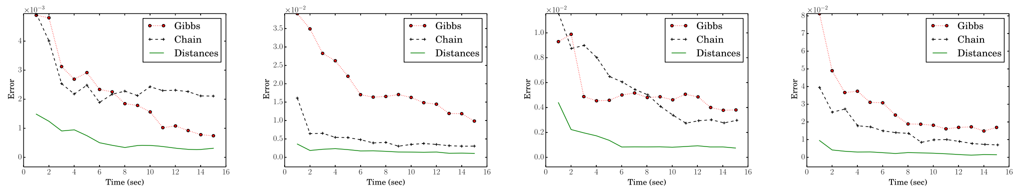
Due to the lack of space we have only introduced in this paper a representative selection of the experiments. However, the complete results can be found in https://github.com/isg-ehu/ekhine.irurozki/blob/master/hamming_full_results.pdf. In particular, we include in this paper the results of the values of $n \in \{50, 150\}$ and the dispersion parameters $\theta \in \{0.1, 2\}$ in MM ($\theta_1 \in \{0.1, 2\}$ in GMM).

Results for MM. By looking at Figure 1, we can see the evolution of the error as the size of the generated sample grows. Note that the error of the Distances and Chain algorithms are similar and almost overlap. The error of the Gibbs is always larger than the error of the other two algorithms, and these differences between the errors increase with θ. However, the Gibbs is a very fast algorithm. The computational time required for the generation of the samples increases linearly with m, but the time required by the Gibbs is insignificant with respect to the time required by the other two. Gibbs required less than 50 milliseconds for each of the instances considered here, the Distances required around 100 milliseconds and Multistage around 600 ms for the instances of $n = 50$ and around 5 seconds for the instances of $n = 150$. Therefore, the question that naturally arises is – what happens if all the algorithms are run for the same time?

By looking at Figure 2, we can see the evolution of the error as the computational time given grows. The distances sampling algorithm has the best trade-off between error and computational time. The Chain and Gibbs sampling algorithms have a similar performance when the distribution to sample is almost uniform. However, as θ grows and the sample becomes more peaked, the Chain algorithm is clearly more accurate than the Gibbs.

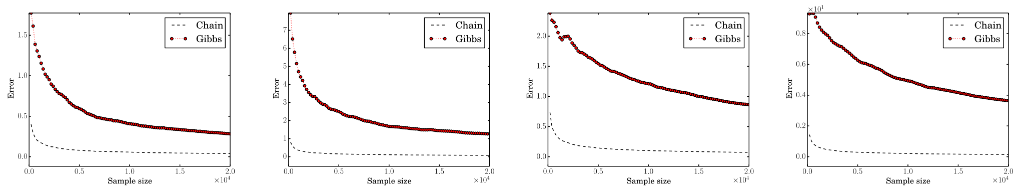
Consequently, we can state that the method with the best trade-off between computational time and accuracy is the Distances sampling algorithm. On the other hand, the Chain is as accurate as the Distances method but slower. The reason to keep it in consideration is that the Distances sampling algorithm cannot generate samples from the GMM, while the Chain can.

Results for GMM. The results relative to the evolution of the error of the generated sample (from a GMM) as m grows are given in Figure 3. Recall that for the GMM only Chain and Gibbs algorithms can be applied. Similar conclusions can be drawn from the results of the MM and GMM. As the sample size grows, the error slowly decreases for the Gibbs while being stable and close to zero for the Chain sampler. Again, the Chain sampling algorithm is much slower than



(a) $n = 50, \theta = 0.1$ (b) $n = 50, \theta = 2$ (c) $n = 150, \theta = 0.1$ (d) $n = 150, \theta = 2$

Figure 2. Error (as the sum of the differences between the expected H_j, $E[H_j]$, and the actual $\bar{H}_j$, in the Y-axis) of each algorithm as the computational time (X-axis) grows for different θ and n in MM. Average of 10 repetitions.

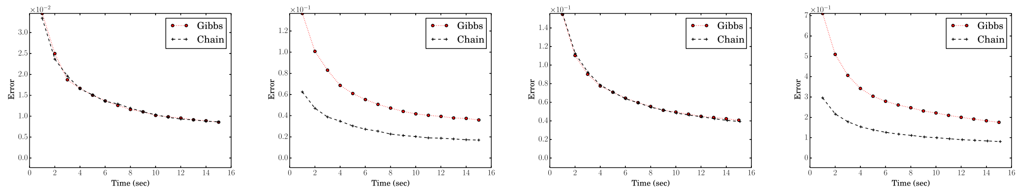


(a) $n = 50, \theta_1 = 0.1$ (b) $n = 50, \theta_1 = 2$ (c) $n = 150, \theta_1 = 0.1$ (d) $n = 150, \theta_1 = 2$

Figure 3. Error (as the sum of the differences between the expected distance, $E[D]$, and the actual distances, in the Y-axis) of each algorithm as the computational time (X-axis) grows for different θ and n in GMM. Average of 10 repetitions.

the Gibbs but, at the same time, much more accurate. The running times are similar to the MM distribution.

Figure 4 shows the result of running both algorithms for the same computational time. Again, when the GMM is close to the uniform distribution the error results are similar, but the Chain algorithm outperforms the Gibbs for non-uniform distributions. Moreover, this difference increases as $\boldsymbol{\theta}$ increases.



(a) $n = 50, \theta_1 = 0.1$ (b) $n = 50, \theta_1 = 2$ (c) $n = 150, \theta_1 = 0.1$ (d) $n = 150, \theta_1 = 2$

Figure 4. Error (as the sum of the differences between the expected distance, $E[D]$, and the actual distances, in the Y-axis) of each algorithm as the computational time (X-axis) grows for different θ and n in GMM. Average of 10 repetitions.

7.2. Learning

Recall that the MLE for the central permutation of the MM can be obtained in polynomial time with the well-known Hungarian algorithm and the dispersion parameter, θ, for a given σ_0 can be computed with a Newton–Raphson algorithm. Therefore, we omit the MM from the learning experiments and focus thus on the GMM.

We have previously shown that the σ_{LAP} is an asymptotically unbiased estimator for the consensus permutation of a sample from a GMM. Therefore, this experimental section is designed to show how the quality of the estimated parameters evolve as the sample size, m, increases. Given that no efficient exact method for the MLE for the central permutation is known, the evaluation process will consist of generating a sample from the model centered around σ_0, learning the parameters and evaluating them w.r.t. σ_0. W.l.o.g. the consensus permutation is the identity, $\sigma_0 = e$. Among the proposed sampling algorithms the Chain is the most accurate and thus the one we will use. The evaluation criteria can be defined in different ways, we have chosen the following:

1. Compare the estimated central permutation and that which generated the sample
2. Compare the likelihood of the sample given the parameters that generated the sample with the likelihood of the sample with those estimated.

The evaluation procedure for each particular setting of n and $\boldsymbol{\theta}$ is as follows: Generate several samples of size $m = 1000, 2000, 3000, \dots, 9000, 10\,000$ with the Chain sampling algorithm. Estimate σ_{LAP} and then:

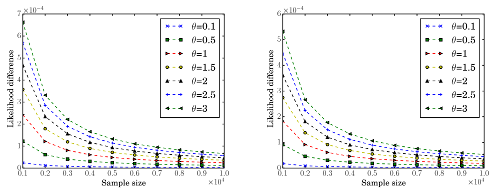
1. Measure the Hamming distance between the actual and the estimated central permutations, $d(\sigma_{\text{LAP}}, \sigma_0)$.
2. Obtain the associated dispersion parameters for σ_{LAP} and compute the likelihood of the sample given σ_{LAP}, denoted as $\mathcal{L}_{\text{LAP}}$. Compute the likelihood of the sample given the parameters that generated the distribution, $\mathcal{L}_0$. The relative error of $\mathcal{L}_{\text{LAP}}$ is given by

$$\frac{\mathcal{L}_{\text{LAP}} - \mathcal{L}_0}{m\mathcal{L}_{\text{LAP}}}. \tag{7.1}$$

   The sample size m in the denominator is aimed to put in the same scale the likelihood for different samples sizes.

The parameter setting is as follows. The number of items in the permutations considered are $n \in \{5, 50, 100, 150\}$. The first of the dispersion parameters is $\theta_1 \in \{0.1, 0.5, 1, 2, 3\}$ while the rest are set such that $\theta_j = \theta_1 - (j-1)(\theta_1/2(n-1))$ for $j > 1$, that is, θ_1 is the largest parameter while the value of the rest decrease linearly to $\theta_n = \theta_1/2$. For each parameter configuration 10 experiments are run and the average results of them are given. W.l.o.g. the central permutation is the identity.

Results for GMM. Figure 5 shows the evolution of the likelihood as the sample size grows, where the error is measured as shown in Equation (7.1). Each plot includes the results of every dispersion parameter of a given n. The plots clearly show that as m increases the error in the likelihood decreases. Even in the cases where the estimated central permutation is correct, the estimated dispersion parameters are better estimated with larger sample sizes, resulting in a better likelihood.



(a) $n = 50$ (b) $n = 150$

Figure 5. Evolution of the error of the estimated parameters for GMM as m grows.

It is important to note that the results are only comparable for the instances of the same n and $\boldsymbol{\theta}$. This means that even though the results of an instance of parameters n and m have smaller error for small values of θ than for large values of $\boldsymbol{\theta}$, we cannot state that the former results are better than the latter.

7.3. Real data experiments

In order to demonstrate the applicability of the GMM under the Hamming distance, we propose an example of the context of e-learning and Massive Open Online Courses (MOOC), which have become very popular lately.

The evaluation of the student is part of every course, including those referred to. One of the types of exercises that a student can be presented with is a *matching question*, in which the student is given two columns of items and the objective is to match each item in the left column with an item in the right columns. We illustrate the use of the GMM with the results of a matching question in which a group of people is given a set of six countries in South-East Asia and six cities (or provinces) of the same area and they are asked to match each city with the country in which it is situated in less than one minute. The question is detailed in Table 1. An answer for this test can be written as a permutation σ where $\sigma(i) = j$ means that city j is matched to country i. This question has been answered by 37 people of similar age, geographic location and education level and the results can be accessed in https://github.com/isg-ehu/ekhine.irurozki.

The analysis of the results starts by counting how many people match each country with each city. This summary is given by the frequency matrix, Table 2, in which entry $M_{i,j}$ represents how many people matched country i with city j.

Let us now fit a GMM under Hamming to the given sample with the `PerMallows` package, which is freely available online, see the supplemental article [27]. The resulting model has parameters $\sigma_0 = 365124$ and $\boldsymbol{\theta} = (1.3898382, 0.1666533, 2.6922244, 0.7526361, 0.3282186, 2.3546746)$.

Note that the consensus matching of the distribution, σ_0, is also the most frequent matching in the sample with 9 repetitions. Moreover, σ_0 is also the correct answer. The dispersion parameters can be interpreted as a measure of the agreement of the population in matching each item. In particular, for a consensus permutation such that $\sigma_0(i) = j$, the higher θ_j, the more people situated city j in country i. Note that the largest dispersion parameter is θ_3, meaning that the

Table 1. Matching question: assign every city in the left column with the country of the right column in which it is situated

Countries	Cities and regions
A: Indonesia	1: Ho Chin Min
B: Thailand	2: Vientian
C: Cambodia	3: Bali
D: Vietnam	4: Penang
E: Laos	5: Phnom Penh
F: Malaysia	6: Phuket

assignment of Indonesia–Bali (A-3) is the most frequent in the sample among those assignments in the central matching.

A more detailed analysis of the dispersion parameters consists of checking whether there is any relation between the amount the people in the population that correctly matched the pair city-country with the importance of the country as a touristic destination. For this last measure we have considered the number of tourist arrivals in 2015. This data is provided by the World Tourism Organization (UNWTO), the United Nations agency responsible for tourism [49], in its yearly publication UNWTO Tourism Highlights 2016 Edition.

The parameters θ_3 and θ_6 are the highest in $\boldsymbol{\theta}$, which means that there is a strong belief in the population that Bali and Phuket belong to Indonesia and Thailand respectively. This is supported by the fact that Indonesia and Thailand are the third and first destinations among the six considered. On the other hand, θ_5 and θ_2 are the smallest in $\boldsymbol{\theta}$, which means that the pairs Cambodia–Phnom Penh and Laos–Vientian are those which a smallest fraction of people guessed. Again, this is supported by the fact that Cambodia and Laos are also the two least visited countries among the six considered.

Having a distribution over the results of the matching question allows us to calculate, for example, the probability that a respondent does not guess the matching of any item. Also, we could be interested in knowing if a respondent has answered uniformly at random, that is, if the respondent's matching is more likely to come from the uniform distribution than from the distribution of the population.

Table 2. Frequency matrix, $M_{i,j}$ counts the number of permutations in the sample σ in which $\sigma(i) = j$

0.02	0.00	**0.81**	0.08	0.08	0.00
0.00	0.08	0.08	0.02	0.05	**0.75**
0.16	0.16	0.00	0.24	**0.35**	0.08
0.56	0.21	0.05	0.13	0.02	0.00
0.10	**0.32**	0.02	0.08	0.35	0.10
0.13	0.21	0.02	**0.43**	0.13	0.05

We have stated that the population that answered this matching question is homogeneous in the sense that they have similar age, education level, etc. However, the distributions introduced in this paper could be the basis for developing methods to find the different clusters in the population, similarly to [4], where a clustering method for rank data – using the MM and GMM under the Kendall's-τ distance – is proposed. Other examples can be found in the literature, among which we highlight [33] and [42].

8. Conclusions

The MM under the Hamming distance can be thus thought as a probability model on matchings. In this situation, there is a “correct” matching and any other matching is less probable as its similarities with the correct one decrease. The GMM models situations in which the correct matching of certain items is more important than the matching of others.

The computation of the partition function is usually the bottleneck of the MM and GMM. The first result presented in this paper is an efficient expression of the partition function for the GMM under the Hamming distance. This expression is the key to efficiently compute the expected distance, the expected number of fixed points and the marginal and conditional probabilities.

We propose three different sampling algorithms. The Gibbs sampler is an adaptation of the well known Monte Carlo algorithm. It is very fast but it generates samples from approximations to the MM and GMM of interest. The Chain sampling algorithm, which can generate samples from MM and GMM, makes use of the well known chain rule for probability. The chain rule computes the joint distribution using conditional probabilities. Therefore, our proposed method to compute conditional probabilities is crucial for this sampling algorithm. The last proposed sampling algorithm is the Distances, which is based on counting and u.a.r. generating permutations at a given distance. Thus, it has a very strong combinatorial basis. It is a fast and accurate algorithm which can only generate samples from a MM. Summarizing, the experimental evaluation we strongly recommend the Distances sampler for the generation from the MM and the Chain sampler for the generation from the GMM.

Regarding the learning process, we show how to estimate the parameters of a sample from a MM in polynomial time. Despite the complexity of the estimation of GMM not being known, we conjecture that it is NP-complete. We propose a very efficient learning algorithm and we show that the estimated parameters are asymptotically unbiased estimators of the real ones.

Given that the popularity of the MM and GMM is due to its use in the ranking domain – and under the Kendall's-τ distance, we have discussed the similarities and differences of rankings and matchings or Kendall's-τ and Hamming throughout the paper.

We have tried to take a step towards the popularization of the Hamming distance-based probability models for permutations by providing diverse, efficient and accurate algorithms for the most critical operations for distributions: calculating the probability, marginalizing, conditioning, computing the expectation and, probably most important, sampling and learning.

Acknowledgements

We are very grateful to Usue Mori for her crucial help. We would like to thank both reviewers for their insightful comments on the paper, as these comments led us to an improvement of the work.

This work has been partially supported by the Basque Government (IT609-13), the Spanish Ministry of Economy and Competitiveness (TIN2016-78365-R), the Elkartek program, the BERC program 2018-2021 (Basque Government) and Severo Ochoa SEV-2017-0718 (Spanish Ministry of Science, Innovation and Universities).

Supplementary Material

Supplement to "Mallows and generalized mallows model for matchings" (DOI: 10.3150/17-BEJ1017SUPP; .pdf). We provide additional proofs and code for reproducing the experiments in the paper.

References

[1] Ali, A. and Meila, M. (2012). Experiments with Kemeny ranking: What works when? *Math. Social Sci.* **64** 28–40.

[2] Baker, F.B. and Harwell, M.R. (1996). Computing elementary symmetric functions and their derivatives: A didactic. *Appl. Psychol. Meas.* **20** 169–192.

[3] Belongie, S., Malik, J. and Puzicha, J. (2002). Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.* **24** 509–522.

[4] Busse, L.M. and Buhmann, J.M. (2007). Cluster analysis of heterogeneous rank data. In *International Conference on Machine Learning* 113–120.

[5] Ceberio, J., Irurozki, E., Mendiburu, A. and Lozano, J.A. (2014). Extending distance-based ranking models in estimation of distribution algorithms. In *IEEE Congress on Evolutionary Computation* 2459–2466.

[6] Ceberio, J., Mendiburu, A. and Lozano, J.A. (2015). Kernels of Mallows models for solving permutation-based problems. In *Genetic and Evolutionary Computation Conference* (*GECCO*-2015) 505–512. Madrid.

[7] Chen, L. and Pu, P. (2004). Survey of preference elicitation methods. Technical report.

[8] Cheng, W. and Hüllermeier, E. (2009). A new instance-based label ranking approach using the Mallows model. In *Advances in Neural Networks* (*ISNN*). *Lecture Notes in Computer Science* **5551** 707–716. Springer.

[9] Cheng, W. and Hullermeier, E. (2009). A simple instance-based approach to multilabel classification using the Mallows model. In *Workshop Proceedings of Learning from Multi-Label Data* 28–38. Bled, Slovenia.

[10] Coppersmith, D., Fleischer, L.K. and Rurda, A. (2010). Ordering by weighted number of wins gives a good ranking for weighted tournaments. *ACM Trans. Algorithms* **6** Art. 55, 13. MR2682624

[11] Critchlow, D.E., Fligner, M.A. and Verducci, J.S. (1991). Probability models on rankings. *J. Math. Psych.* **35** 294–318.

[12] Csiszár, V. (2008). Conditional independence relations and log-linear models for random matchings. *Acta Math. Hungar.* **122** 131–152.

[13] Csiszár, V. (2009). On L-decomposability of random orderings. *J. Math. Psych.* **53** 294–297.

[14] D'Elia, A. and Piccolo, D. (2005). A mixture model for preferences data analysis. *Comput. Statist. Data Anal.* **49** 917–934. MR2141426

[15] Deza, M. and Huang, T. (1998). Metrics on permutations, a survey. *J. Comb. Inf. Syst. Sci.* **23** 173–185. MR1737796

[16] Diaconis, P. (1988). *Group Representations in Probability and Statistics*. IMS.

[17] Farah, M. and Vanderpooten, D. (2007). An outranking approach for rank aggregation in information retrieval. In *Conference on Research and Development in Information Retrieval* (*ACM SIGIR*), *SIGIR* '07 591–598. New York, NY, USA: ACM.

[18] Flajolet, P., Zimmermann, P. and Van Cutsem, B. (1994). A calculus for the random generation of labelled combinatorial structures. *Theoret. Comput. Sci.* **132** 1–35.

[19] Fligner, M.A. and Verducci, J.S. (1986). Distance based ranking models. *J. R. Stat. Soc. Ser. B. Stat. Methodol.* **48** 359–369.

[20] Fligner, M.A. and Verducci, J.S. (1990). Posterior probabilities for a consensus ordering. *Psychometrika* **55** 53–63. MR1060264

[21] Gnedin, A. and Olshanski, G. (2012). The two-sided infinite extension of the Mallows model for random permutations. *Adv. in Appl. Math.* **48** 615–639.

[22] Gupta, J. and Damien, P. (2002). Conjugacy class prior distributions on metric-based ranking models. *J. Roy. Statist. Soc. Ser. B* **64** 433–445.

[23] Huang, J. and Guestrin, C. (2012). Uncovering the riffled independence structure of rankings. *Electron. J. Stat.* **6** 199–230.

[24] Huang, J., Guestrin, C., Jiang, X. and Guibas, L. (2009). Exploiting probabilistic independence for permutations. In *Artificial Intelligence and Statistics* (*AISTATS*).

[25] Irurozki, E. (2014). Sampling and Learning Distance-Based Probability Models for Permutation Spaces. Ph.D. thesis, University of the Basque Country.

[26] Irurozki, E., Calvo, B. and Lozano, J.A. (2016). PerMallows: An R package for Mallows and generalized Mallows models. *J. Stat. Softw.* **71** 1–30.

[27] Irurozki, E., Calvo, B. and Lozano, J.A. (2019). Supplement to "Mallows and generalized Mallows model for matchings." DOI:10.3150/17-BEJ1017SUPP.

[28] Jonker, R. and Volgenant, A. (1987). A shortest augmenting path algorithm for dense and sparse linear assignment problems. *The Netherlands Computing* **38** 325–340.

[29] Kammerdiner, A., Krokhmal, P.A. and Pardalos, P.M. (2009). On the Hamming distance in combinatorial optimization problems on hypergraph matchings. *Optim. Lett.* **4** 609–617.

[30] Kondor, R., Howard, A. and Jebara, T. (2007). Multi-object tracking with representations of the symmetric group. In *International Conference on Artificial Intelligence and Statistics* 211–218.

[31] Kuhn, H.W. (1955). The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly* **2** 83–97.

[32] Kuncheva, L.I. (2010). Full-class set classification using the Hungarian algorithm. *Int. J. Mach. Learn. Cybern.* **1** 53–61.

[33] Lee, P.H. and Yu, P.L.H. (2012). Mixtures of weighted distance-based models for ranking data with applications in political studies. *Comput. Statist. Data Anal.* **56** 2486–2500.

[34] Lu, T. and Boutilier, C. (2011). Learning Mallows models with pairwise preferences. In *International Conference on Machine Learning* (*ICML*) 145–152.

[35] Luce, R.D. (1959). *Individual Choice Behavior*: *A Theoretical Analysis*. New York: Wiley. MR0108411

[36] Mallows, C.L. (1957). Non-null ranking models. *Biometrika* **44** 114–130.

[37] Mandhani, B. and Meila, M. (2009). Tractable search for learning exponential models of rankings. *J. Mach. Learn. Res.* **5** 392–399.

[38] Mao, Y. and Lebanon, G. (2008). Non-parametric modeling of partially ranked data. *J. Mach. Learn. Res.* **9** 2401–2429.

[39] Meila, M. and Bao, L. (2008). Estimation and clustering with infinite rankings. In *Uncertainty in Artificial Intelligence* (*UAI*) 393–402. Corvallis, Oregon: AUAI Press.

[40] Meila, M. and Chen, H. (2010). Dirichlet process mixtures of generalized Mallows models. In *Uncertainty in Artificial Intelligence* (*UAI*) 285–294.

[41] Meila, M. and Chen, H. (2016). Bayesian non-parametric clustering of ranking data. *IEEE Trans. Pattern Anal. Mach. Intell.* **38** 2156–2169.

[42] Murphy, T.B. and Martin, D. (2003). Mixtures of distance-based models for ranking data. *Comput. Statist. Data Anal.* **41** 645–655. MR1973732

[43] Plackett, R.L. (1975). The analysis of permutations. *J. R. Stat. Soc., A* **24** 193–202.

[44] Press, W.H., Teukolsky, S.A., Vetterling, W.T. and Flannery, B.P. (1992). *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed. New York, NY: Cambridge Univ. Press.

[45] Schiavinotto, T. and Stützle, T. (2007). A review of metrics on permutations for search landscape analysis. *Comput. Oper. Res.* **34** 3143–3153.

[46] Sloane, N.J.A. (2009). Subfactorial or rencontres numbers, or derangements. Available at https://oeis.org/A000166.

[47] Stanley, R.P. (1986). *Enumerative Combinatorics*. Belmont, CA, USA: Wadsworth Publishing Company.

[48] Thurstone, L.L. (1927). A law of comparative judgment. *Psychological Review* **24** 273–286.

[49] UNWTO (2016). UNWTO Tourism Highlights, 2016 Edition.

[50] Vitelli, V., Sørensen, Ø., Crispino, M., Frigessi, A. and Arjas, E. (2014). Probabilistic preference learning with the Mallows rank model. Preprint. Available at arXiv:1405.7945.

[51] Wang, Y., Makedon, F., Ford, J. and Huang, H. (2004). A bipartite graph matching framework for finding correspondences between structural elements in two proteins. In *Engineering in Medicine and Biology Society*, 2004. *IEMBS'*04. 26*th Annual International Conference of the IEEE* **2** 2972–2975. IEEE.

Received September 2016