Mallows model under the Ulam distance: a feasible combinatorial approach

Ekhine Irurozki
Intelligent Systems Group
Basque Country University, Spain
ekhine.irurozqui@ehu.es

Josu Ceberio
Intelligent Systems Group
Basque Country University, Spain
josu.ceberio@ehu.es

Borja Calvo
Intelligent Systems Group
Basque Country University, Spain
borja.calvo@ehu.es

Jose A. Lozano
Intelligent Systems Group
Basque Country University, Spain
ja.lozano@ehu.es

Abstract

The Mallows model is one of the best-known probability models for permutation spaces. It is traditionally used under the Kendall's-τ distance although it has been proposed with different distances for permutations. In this paper, we deal with the Mallows model under the Ulam distance. We show how to efficiently compute the normalization constant and the expected distance. Moreover, we propose two different sampling algorithms.

1 Introduction

The distance-based or Mallows model (MM) is an exponential family model based on a definition of distance for permutations [5]. This distance is usually the Kendall's-τ, in part because its theoretical properties simplify the computation of some of the most common operations for distributions, such as generating samples from the model.

In this paper, we focus on the MM under the Ulam distance, $d_u(\sigma, \pi)$. The Ulam distance counts the length of the complement of the longest common subsequence in σ and π [1], [2], [7]. It follows that the Ulam distance between a permutation σ and the identity, $d_u(\sigma)$, equals n minus the length of the Longest Increasing Subsequence (LIS) of σ. The classical example to illustrate the Ulam distance [4] considers a shelf of books in the order specified by σ. The objective is to order the books as specified by π with the minimum possible number of movements, where a movement consists of taking a book and inserting it in another position (delete-insert); the minimum number of movements is exactly $d_u(\sigma, \pi)$. The complexity of computing it is $O(n\,log\,l)$, where l is the length of the LCS of σ and π.

Despite the fact that Ulam is a popular distance for permutations, the MM under this distance has not been extensively considered. This is mainly due to the fact that there is no general method to work with a MM. For example, a naive approach to compute the normalization constant, is computationally intractable ($O(n!)$) for medium size permutations, say $n > 10$. In this paper, we try to take a step forward to increase the popularity of the MM under the Ulam distance by providing efficient expressions for the normalization constant and the expected value of the distance. Moreover, we introduce two sampling algorithms, an approximate one and an exact one.

2 Preliminary background

The techniques used in this paper come mainly from the combinatorial arena. Due to the lack of space, the manuscript can not be self contained. However, we give the basic concepts and recommend [3] for an excellent reference in combinatorics of permutations.

Let n is a positive integer. A partition of n, $\lambda = \{\lambda(1), \ldots, \lambda(k)\} \vdash n$, is a non-increasing sequence of positive integers whose sum is n. A partition can be graphically depicted as a Ferrer's diagram (FD). A Standard Young Tableaux (SYT) is a FD in which the boxes have been labeled with the integer from 1 to n in such a way that the numbers are increasing in each row and down each column. The Hook length formula shows how to compute the number of different SYT of shape λ that can be generated, which is denoted as h_λ. The link between permutations and SYT is the Robinson-Schensted (RS) correspondence, which provides a bijection between pairs of SYT of the same shape and permutations. This paper relies on the property of the RS which states that the length of the LIS of σ generated from a FD of shape λ equals the number of columns in λ, $\lambda(1)$.

Generating a permutation at distance d uniformly at random The sequence of the number of permutations of n items at each possible Ulam distance d is denoted as $S_u(n, d)$ and can be found in [8], along with some recurrences to compute it. Now we show how the RS correspondence allows us to break the problem of randomly generating a permutation at a given Ulam distance into three stages:

1. Randomly select a FD of shape $\lambda \vdash n$. The probability of selecting each shape λ is proportional to the number of permutations at distance d that can be generated with it. Let us detail this last point.

   Let h_λ be the Hook length of shape λ. It follows that there are h_λ^2 different possible pairs of SYT of the given shape λ. The RS correspondence defines a bijection between pairs of SYT of the same shape λ and permutations, so the number of possible permutations that are generated from a FD of shape λ is h_λ^2.

   Moreover, the length of the LIS of σ generated from a FD of shape λ equals $\lambda(1)$. Since the length of the LIS of σ equals $n - d(\sigma)$, the probability of shape λ is as follows.

   $$p(\lambda) \propto h_\lambda^2 \quad \forall\, \lambda \text{ such that } \lambda(1) = n - d \quad ; \quad p(\lambda) = 0 \quad \text{otherwise}$$

2. Uniformly at random generate two SYT of shape λ, namely P and Q. The random generation of SYT of shape λ has been addressed in [6]. In particular, the authors give a probabilistic proof of the Hook length which can be adapted to randomly generate a SYT.

3. Generate σ given P and Q with the inverse Schensted algorithm [3].

3 MM under the Ulam distance

The MM under the Ulam distance can be written as follows:

$$p(\sigma) = \frac{exp(-\theta d_u(\sigma\sigma_0^{-1}))}{\psi(\theta)} \tag{1}$$

where the central permutation is denoted σ_0 and the spread parameter θ. This distribution is not factorizable as far as the authors know. However, we can take advantage of two facts to efficiently compute the normalization constant, namely (1) the Ulam distance is right invariant and (2) every two permutations at the same distance from σ_0 have equal probability. Considering these facts the normalization constant and the expected distance can be written as shown in Equations (2) and (3) respectively.

$$\psi(\theta) = \sum_{d=0}^{n-1} S_u(n, d) exp(-\theta d) \tag{2}$$

$$E_\theta[D] = \frac{\sum_{d=0}^{n-1} S_u(n, d) exp(-\theta d) d}{\psi(\theta)} \tag{3}$$

4 Sampling

The random generation of samples from a MM under the Ulam distance is addressed by means of the Gibbs algorithm, an approximate algorithm, and a novel exact algorithm by the name of Distances sampler. The proposed algorithms generate a sample centered around the identity. To set a different σ_0, each permutation in the sample should be composed with σ_0.

Gibbs sampling algorithm The Gibbs sampler is a popular algorithm in the statistical community. It can be adapted to generate a chain of samples from an approximate distribution of the MM under the Ulam distance as follows:

1. Generate uniformly at random a permutation σ.
2. Generate uniformly at random two integers, i, j in the range $[1, n]$.
3. Build a new permutation σ' as follows:

$$\sigma' = \begin{cases} \sigma(1), \ldots, \sigma(i-1), \sigma(i+1), \ldots, \sigma(j), \sigma(i), \sigma(j+1), \ldots, \sigma(n) & \text{if } i < j \\ \sigma(1), \ldots, \sigma(j-1)\sigma(i), \sigma(j), \ldots, \sigma(i-1), \sigma(j+1), \ldots, \sigma(n) & \text{if } i > j \end{cases}$$

4. Let $\beta = min\{1, p(\sigma')/p(\sigma)\}$. With probability β the algorithm accepts the candidate permutation moving the chain to σ', so $\sigma = \sigma'$, and goes back to step 2. Otherwise, it discards σ' and goes back to step 2.

The above process is repeated until the algorithm generates a given number of permutations. The complexity of generating each permutation is $O(n \, log \, n)$.

Distances sampling algorithm The probability of obtaining a permutation at Ulam distance d from the identity permutation is as follows.

$$p(d) = \sum_{\sigma | d_u(\sigma) = d} p(\sigma) = S_u(n, d) \frac{exp(-\theta d)}{\psi(\theta)} \tag{4}$$

Recall that the normalization constant is given in Equation (2). In this way, the process of sampling a permutation from a MM under the Ulam distance can be done in two stages.

1. Randomly select a distance d taking into account Equation (4).
2. Uniformly at random generate a permutation at distance d from e as shown in Section 2.

5 Sampling experiments

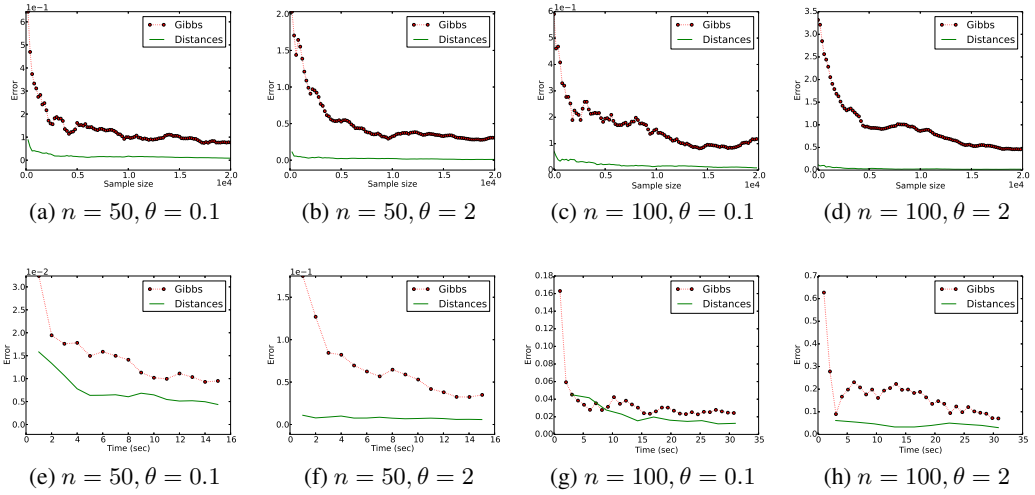
The code of the proposed algorithms is freely available on-line. They are included in the `PerMallows` package, an `R` interface to a `c++` code for reasoning on permutation spaces. `PerMallows` can be found in the CRAN repository.

The sampling algorithms are evaluated regarding two different criteria: (1) the comparison of the error of the samples of each algorithm as the number of permutations generated, m, grows and (2) -since the Gibbs sampler is much faster than the Distances algorithm- the evolution of the error as the computational time increases.

The error of sample $\{\sigma_1, \ldots, \sigma_m\}$ is measured as the difference between the average distance to the sample $\bar{d} = \sum_{i=1}^{m} d(\sigma_i, \sigma_0)/m$, and the expected distance, $E_\theta[d]$, which is given in Equation (3). W.l.o.g. the central permutation is the identity.

The Gibbs algorithm discards the first n^2 permutations as part of the burning-period. The parameter setting used is $n \in \{5, 50, 100\}$ and $\theta = \{0.1, 0.5, 1, 2, 3\}$. We include in this section the results of the values of $n = \{50, 100\}$ and $\theta = \{0.1, 2\}$ and the complete results can be found on-line[1].

[1] `http://www.sc.ehu.es/ccwbayes/members/ekhine/full_results_ulam.pdf` for the complete results.



(a) $n = 50, \theta = 0.1$ (b) $n = 50, \theta = 2$ (c) $n = 100, \theta = 0.1$ (d) $n = 100, \theta = 2$

(e) $n = 50, \theta = 0.1$ (f) $n = 50, \theta = 2$ (g) $n = 100, \theta = 0.1$ (h) $n = 100, \theta = 2$

Figure 1: Error of each sampling algorithm as the computational time grows for different θ and n in the MM.

We have considered two computational approaches: the first one consists of storing both the sequence $S_u(n, d)$ for every d and the collection of FD in the RAM memory. The second one is to save those structures to disk. Experiments of $n = \{5, 50\}$ follow the former approach. However, the amount of memory required for the experiments of $n = 100$ was unaffordable as the memory required is proportional to the number of partitions of 100, which is about 2×10^8. Therefore, for the experiment of $n = 100$ the FD were first stored into files. In this case, the process of storing the data in the disk took about half an hour.

Results The top row in Figure 1 shows the evolution of the error of the sample as m grows. Clearly, the error of the sample decreases as m increases for both sampling algorithms. However, the error of the Distances sampler is always smaller than the error of Gibbs, being notorious the fact that, for every n and θ, the error of a sample of $m = 20000$ generated with Gibbs is larger than the error of a sample of $m = 200$ permutations of Distances.

Our next concern is to evaluate the algorithms in the case when both are allowed to run for the same time, bottom row in Figure 1. The Gibbs sampler is much faster than the Distances and thus its samples are vastly bigger given the same computational time. However, the error of the Distances is always smaller than error of the Gibbs sampler, with the exception on the generation of small samples form an uniform distribution of $n = 100$. In every case, the difference between both algorithms becomes more evident as θ increases.

6 Conclusions

In this paper we describe computationally tractable algorithms for the Mallows model under the Ulam distance, including the computation of the normalization constant and the expected distance. We also show to efficiently generate samples from the models introducing two algorithms, one of which is approximate and the other exact. The latter makes use of a novel uniformly at random generator of permutations at a given Ulam distance.

Acknowledgments

This work has been partially supported by the Basque Government's program Saiotek and Research Groups 2013-2018 under Grant IT-609-13; Ministerio de Educacion y Ciencia under Grant TIN2013-41272P; COM-BIOMED network in computational biomedicine (Carlos III Health Institute). Ekhine Irurozki holds a grant BES-2009-029143 from the Spanish Ministry of Science and Innovation.

References

[1] D. Aldous and P. Diaconis. Longest increasing subsequences: from patience sorting to the Baik-Deift-Johansson Theorem. *Society*, 36(4):413–432, 1999.

[2] J. Baik, P. Deift, and K. Johansson. On the distribution of the length of the longest increasing subsequence of random permutations. *Journal of the American Mathematical Society*, 12(4):1119–1178, 1999.

[3] M. Bóna. *Combinatorics of permutations*. Discrete mathematics and its applications. Chapman & Hall/CRC Press, Boca Raton, FL, London, 2004.

[4] P. Diaconis. *Group representations in probability and statistics*. Institute of Mathematical Statistics, 1988.

[5] M. A. Fligner and J. S. Verducci. Distance based ranking models. *Journal of the Royal Statistical Society*, 48(3):359–369, 1986.

[6] C. Greene, A. Nijenhuis, and H. S. Wilf. A Probabilistic Proof of a Formula for the Number of Young Tableaux of a Given Shape. *Advances in Mathematics*, 31:104–109, 1979.

[7] M. Saks and C. Seshadhri. Estimating the Longest Increasing Sequence in Polylogarithmic Time. In *Foundations of Computer Science, FOCS*, pages 458–467. IEEE Computer Society, 2010.

[8] N. J. A. Sloane. Triangle of numbers by length of the longest increasing subsequence, https://oeis.org/A126065, 2009.