Technical Report

UNIVERSITY OF THE BASQUE COUNTRY
Department of Computer Science and Artificial Intelligence

Sampling and learning the Mallows model under the Ulam distance

Ekhine Irurozki, Josu Ceberio, Borja Calvo, José A. Lozano

January, 2014

San Sebastian, Spain
www.ehu.es/ccia-kzaa
hdl.handle.net/10810/4562

Sampling and learning the Mallows model under the Ulam distance

Ekhine Irurozki, Josu Ceberio, Borja Calvo, Jose A. Lozano

January 17, 2014

Index terms— Permutations Mallows Model Sampling Learning ulam distance Longest Increasing subsequence

Abstract

In this paper we deal with probability distributions over permutation spaces. The Probability model in use is the Mallows model. The distance for permutations that the model uses in the Ulam distance.

1 Ulam distance and Longest Increasing subsequence

The Ulam distance between two permutations σ and π, $D_u(\sigma, \pi)$, is exactly the size of the complement of the longest common subsequence of σ and π or, equivalently, n minus the length of the longest increasing subsequence (LIS) in $\sigma\pi^{-1}$. Therefore, $D_u(\pi)$ is n minus the length of the LIS in π. It can be exactly computed in $O(n\ log\ n)$. For the Ulam distance just the MM case in considered.

2 Generating a permutation at a given Ulam distance

For the generation of permutation at a given Ulam distance will have to make use of more advanced concepts of algebra such as Ferrers diagrams, Standard Young Tableaux or Bratteli diagrams. Also, it is necessary to know how to count Standard Young Tableaux. In order for this manuscript to be self contained, all these concepts are first introduced.

A partition $\lambda = \{\lambda(1), \lambda(2), \ldots, \lambda(k)\} \vdash n$ is a non decreasing sequence of integers such that $\sum_{i=1}^{k} \lambda(i) = n$. A partition can be graphically represented by a Ferrers diagram or Ferrers shape (FS), which is a set of n boxes arranged in table form where there are $\lambda(1)$ boxes in the first row, $\lambda(2)$ in the second one, ... A Young Standard Tableaux (SYT) is FS in which the numbers $1 \ldots n$ are placed each in a box. See Figure 1 for an example of each concept.

(a) (b)

Figure 1: An example of a Ferrers diagram (1a) and a SYT (1b)

These simple graphics are a very useful tool on representation theory or geometry for example. The first question when dealing with FS and SYT is that of counting the number of different SYT that can be built for a given FS. This count is given by the hook length formula and it is one the most celebrated results in combinatorics. Since the hook length formula is not completely necessary for our particular problem and it has not got closed form we will skip its expression. We refer the interested reader to some of the many proves in the literature, including probabilistic and combinatorial ones. [**?**].

Instead of explicitly giving the computation of the hook lengths, we will make use of a Bratteli diagram to compute and store the hook length of every $\lambda \vdash k$ for every $1 \leq k \leq n$. Although one may think that computing the number of SYT for every $1 \leq k \leq n$ is a waste of resources it turns out, as we will later show, that this diagram is extremely useful for an efficient generation of the permutations. In this way, the Bratteli diagram will help us computing the hook lengths in a recursive manner.

The recursion is based on the fact that the generation of a SYT λ of a given shape of n items can be done by placing item n in a valid place and then generating a SYT of $n-1$ items in the shape λ' that results of deleting from λ the box where n lays. Therefore, we will show the relation between different shapes, say $\lambda \vdash n$ and $\lambda' \vdash (n-1)$ with an example. Suppose that we are generating a SYT $\lambda = \{2, 2, 1\} \vdash n = 5$. Recall that the items in a SYT are increasing along rows and down columns. Therefore, item 5 can only be placed at two positions, namely:

Position $(2, 2)$ By choosing this position, as in Figure 2a, the problem is now that of generating a SYT of shape $\lambda'_1 = \{2, 1, 1\} \vdash (n-1) = 4$.

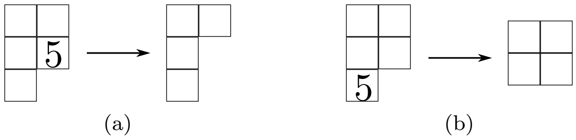

Figure 2: Possible placements for item 5 in $\lambda = \{2,2,1\} \vdash n = 5$ and resulting SYT

Position $(3,1)$ By choosing this position, as in Figure 2b, the problem is now that of generating a SYT of shape $\lambda'_2 = \{2,2\} \vdash (n-1) = 4$.

Therefore, the total number of different SYT of shape $\lambda = \{2,2,1\}$ equals the sum of the hook lengths of λ'_1 and λ'_2. In general, the hook length of $\lambda \vdash n$ equals the sum of the hook lengths of every shape $\lambda' \vdash (n-1)$ that results of deleting a corner box. Note that by a corner box we mean any box that has no box neither to the right nor below.

These relations between every $\lambda \vdash k$ for every $1 \leq k \leq n$ can be depicted using a Bratteli diagram, as the one in Figure 3 shows for $n = 5$. Note that there is a line joining $\lambda \vdash n$ and $\lambda' \vdash (n-1)$ iff λ' results of deleting a corner box in λ. The hook length of each FS is given on top of each one. The computation of the hook lengths on the Bratteli diagram is done as follows.

- The trivial case, situated at the bottom of the figure, has hook length one since there is only one SYT on a FS of one box.
- For any other case, the hook length of λ equals the sum of the hook lengths of every $\lambda' \vdash (n-1)$ such that λ' results of deleting a corner box in λ.

This suggests a recursive algorithm for the computation of the hook lengths.

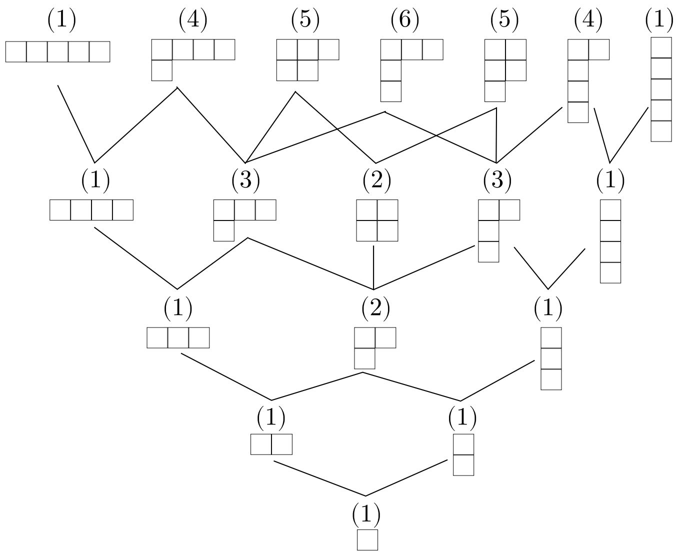

Figure 3: Bratteli diagram for every $\lambda \vdash k$ for every $1 \leq k \leq 5$ where the number in parenthesis is the hook length of each FS

2.1 Random generation of a SYT given a FS

We will now describe how to generate a SYT u.a.r. given a FS λ. We claim that our describe procedure generates every possible SYT with equal probability.

Each SYT is generated from λ in n stages by first inserting item n, then item $(n-1)$, then $n-2$ and so on until the SYT is full. Let us show how item n is placed. Let $\{\lambda_r \vdash (n-1)\}$ be the set of shapes that are related with $\lambda \vdash n$ in the Bratteli diagram. Also, λ_r is obtained from λ by removing the corner box in position (i_r, j_r). Then, the probability of inserting item n in position (i_r, j_r) is proportional to the hook length of λ_r. This process can be carried out iteratively by decreasing n in one unit and settting $\lambda = \lambda_r$ until a full SYT is built. The complete process is shown in Algorithm 1.

2.2 The Robinson-Schensted correspondence

The Robinson-Schensted correspondence (RS) is the key to this paper. It states that there is a bijection between pairs of SYT of the same shape λ, P and Q, and permutations σ. Using the Schensted algorithm [?] it is possible to

Algorithm 1: Random generation of a SYT given a FS

```
Input: FS λ
Output: Random SYT Q of shape λ
for r = n to 1 do
    foreach corner box (i_r, j_r) in λ do
        λ_r ← delete box (i_r, j_r) from λ;
        h_{λ_r} ← hook length of λ_r ;
        P(i_r, j_r) ∝ h_{λ_r}
    end
    Randomly select (i_s, j_s) according to P(i_r, j_r);
    Q(i_s, j_s) = r;
    λ ← λ_s;
end
```

obtain σ given P and Q and vice versa. Algorithm in 2 shows how to generate σ given P and Q in time $O(n^2)$. This algorithm iterates for n steps deleting a box on both P and Q at each step. Also, the items in P are possibly moved during the construction of the permutation, while items in Q are not.

Algorithm 2: RS bijection between pairs of SYT and permutations

```
Input: Q, P: SYT of shape λ
Output: Permutation σ recovered from P and Q
for r = n to 1 do
    Choose i, j such that Q(i, j) = r;
    u ← P(i, j);
    Remove boxes Q(i, j) and P(i, j);
    while i ≠ 1 do
        i ← i − 1;
        Select j such that P(i, j) is the largest entry smaller than u;
        Swap items P(i, j), u;
    end
    σ(r) = u;
end
```

Among the many interesting property of this correspondence, the one that concerns us the most is that which states that the length of the longest increasing subsequence (LIS) of σ equals the number of columns of P and Q.

2.3 Counting permutations at a given Ulam distance

The problem of counting permutations of n items at a given distance d, denoted as $S(n, d)$, is very recurrent in the literature. Although there are not often closed form expressions $S(n, d)$, the sequences for multiple metrics and values of n and d are given in the On-line Encyclopedia of Integer Sequences. The triangle of the number of permutations of n items at each Ulam distance d, denoted as $S_u(n, d)$, is given in `http://oeis.org/A047874`. However, we can also make use of the concepts we have already introduce to count permutations with a given LIS (or equivalently, at a given Ulam distance $d = n - LIS$).

As we have stated, the number of different SYT of a prescribed FS λ is given by the hook length, h_λ. The hook length can be computed by constructing the Bratteli diagram as above or by computing the hook length formula (which we do not explicitly include in this paper).

The RS correspondence states that there is a bijection between every pair of SYT P, Q of the same FS $\lambda \vdash n$ and every permutation $\sigma \in S_n$. The hook length formula states that given a FS $\lambda \vdash n$ there are h_λ^2 different pairs of SYT that can be generated. Since there is a one-to-one correspondence between pairs of SYT of the same FS and permutations, there are h_λ^2 possible permutations that can be possibly generated from λ. Finally, since the LIS of σ equals the number of boxes in the first row of the FS, we conclude that the number of permutations at Ulam distance d, $S_u(n, d)$, is as follows:

$$\sum_{\lambda | \lambda(1) = n-d} h_\lambda^2$$

Due to the lack of close expression, as n increases the exact computation of $S_u(n, d)$ gets impractical. In many situations an approximate bound for $S_u(n, d)$ which can be efficiently computed is therefore required. Such bound is given as follows [1]:

$$\sum_{i=0}^{d} S_u(n,i) \leq d! \binom{n}{d}^2 \tag{1}$$

2.4 Random generation of a permutation at a given Ulam distance

As we have stated in Section 2.2 the RS correspondence claims that there is a bijection between pairs of SYT, P and Q of the same shape λ, and permutations σ. The most remarkable property for our purposes is that the length of the LIS of σ equals the number of columns of P and Q. Therefore, the RS correspondence allows us breaking the problem of randomly generating a permutation at a given Ulam distance.

This process combines all the above theory to generate a permutation at Ulam distance d (with a LIS of length $l = n - d$). This process consists on three smaller stages.

1. Randomly select a FS $\lambda \vdash n$ such that the number of columns of λ equals l. This means that that probability of selecting a particular shape is proportional to the number of permutations with a LIS of length l that can be generated with it, so

$$p(\lambda) \propto \begin{cases} h_\lambda^2 & \text{if the number of columns of } \lambda = l \\ 0 & \text{otherwise} \end{cases}$$

   where h_λ is the hook length of FS λ.

2. U.a.r generate two SYT of shape λ, namely P and Q. We recall that the generation of a given SYT of a prescribed FS is detailed in Section 2.1.

3. Follow Algorithm 2 with SYT P and Q to obtain the permutation σ.

2.5 Improving the efficiency

Note that the whole Bratteli diagram can be stored in disk so there is no need to compute the hook lengths every time the algorithm is run.

There are situations in which one expects to generate permutations that are close to a given one. In this case, the performance can be increased by assuming that the permutations will lay at most at Ulam's distance d'. In this way, one can avoid the generation of the whole Bratteli diagram. This restriction will increase the efficiency of the algorithm.

Note that if the generated permutations of n items are going to be at most at distance d' then, their LIS will be at least $l' = n - d'$. Therefore, the pair of SYT of shape λ that generates the permutation via RS correspondence have at least l' columns. Thus, the insertion to the Bratteli diagram of any FS with less than l' columns can be avoided. Similarly, there are FS in the row of FS of $n - 1$ items that can be omitted. Note that the only FS of $n - 1$ that are related with FS of the n items are those that have at most $l' - 1$ columns in its first row. This idea generates a bounded tree as that in Figure 4.

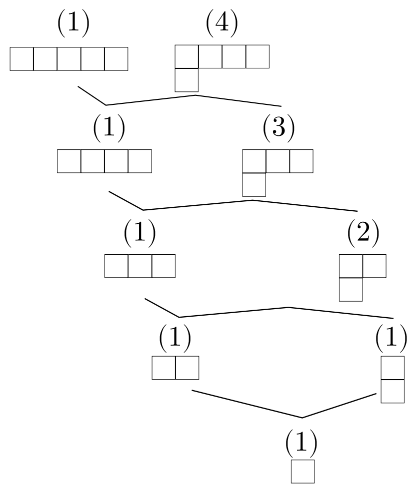

Figure 4: Bounded Bratteli diagram to generate permutations of 5 items with LIS 4 or 5, i.e. at Ulam distance 0 or 1.

3 Probabilistic model, Mallows

In this section we introduce a probability model for permutation spaces. It is the well known Mallows model, very popular in the literature. This model defines the distribution with just two parameters. The first one is the mode of the distribution, σ_0, i.e. the permutation with the largest probability mass. The probability of any permutation decays exponentially as its distance to the central permutation increases. In this paper we consider the Ulam distance. The sharpness of that decreasement is given by the dispersion parameter, θ. Its expression is given by Equation (2).

$$p(\sigma) = \frac{exp(-\theta d_u(\sigma, \sigma_0))}{\psi(\theta)} \tag{2}$$

where $\psi(\theta) = \sum_\sigma exp(-\theta d_u(\sigma, \sigma_0))$. Note that when the dispersion parameter θ is greater than 0, then σ_0 is the mode. On the other hand, with $\theta = 0$ we obtain the uniform distribution and when $\theta < 0$ then σ_0 is the anti mode.

Note that the normalization constant, $\psi(\theta)$, requires a sum over each of the $n!$ permutations of n items, which is a bottleneck when dealing with this distributions. Moreover, there is not a closed expression for it. The good news is that we give an efficient expression for its computation. It is based on the following consideration: every permutation at distance d has the same probability. Moreover, if the number of permutations at Ulam distance $0 \leq d < n$ is given by $S_u(n, d)$, then the normalization constant can be written as follows:

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

In this way, the original expression for the normalization constant consisting of summing $n!$ terms is know transformed into a sum on n terms. Moreover, we showed on Section 2.3 how to explicitly and approximately count the number of permutations at each Ulam distance, $S_u(n, d)$. Therefore, given $S_u(n, d)$ for all d the normalization constant can be computed in $O(n)$.

The most popular extension of the MM is the Generalized Mallows model (GMM). It is aimed to break the constraint that imposes every permutation at the same distance to have the same probability. This is very useful when modeling a situation where the consensus on some positions of the permutation is higher than the consensus in some others. In order to adapt this idea to distributions under the Ulam metric, the expression for the distance must be decomposed in n terms as $d_u(\sigma, \sigma_0) = \sum_{j=1}^n U_j(\sigma\sigma_0^{-1})$ so that the new model could be expressed as

$$p(\sigma) \propto \prod_{j=1}^{n} exp(-\theta_j U_j(\sigma\sigma_0^{-1})) \tag{4}$$

Such decomposition can be as follows:

$$U_j(\sigma\sigma_0^{-1}) = \begin{cases} 0 & \text{if } j \text{ belongs to the LIS in } \sigma\sigma_0^{-1} \\ 1 & \text{otherwise} \end{cases}$$

However, note that any given σ can have possibly many such decompositions, since there are possibly more than one LIS os the maximum length. Thus, under this assumption, a permutation can have different probability values. Therefore, adapting the idea behind GMM to distributions under Ulam distance is not possible.

4 Learning

This section approaches the maximum likelihood estimation for the θ and σ_0 of the distribution given a sample of m i.i.d. permutations $\{\sigma_1, \sigma_2, \ldots, \sigma_m\}$. The log likelihood of the MM is given by

$$\begin{aligned} Ln\, \mathcal{L}(\{\sigma_1, \sigma_2, \ldots, \sigma_m\}|\sigma_0, \theta) &= \sum_{s=1}^{m} Ln\, p(\sigma_s|\sigma_0, \boldsymbol{\theta}) = Ln \prod_{s=1}^{m} \frac{exp(-\theta d(\sigma_s\sigma_0^{-1}))}{\psi(\theta)} \\ &= -\theta \sum_{s=1}^{m} (d(\sigma_s\sigma_0^{-1})) - m Ln\, \psi(\theta) \end{aligned} \tag{5}$$

By looking at Equation (5), we can see that calculating the value of σ_0 that maximizes the equation is independent of θ. Therefore the maximum likelihood estimation problem for the MM can be posed as a two step process in which first the MLE for the central permutation $\hat{\sigma}_0$ is obtained and then the MLE for the dispersion parameter $\hat{\theta}$ for the given $\hat{\sigma}_0$ is calculated.

Problems consisting on finding the permutation that minimizes the sum of the distances to the permutations in the sample are often called median problems and are usually NP-complete. Although the question of the complexity of the median problem is open, it it supposed to be NP-hard. Similar problems such as the Transposition median problem [2] and the Median String problems [3] are NP-complete.

An approximate solution for the String median problem consists on looking for the permutation in the sample that minimizes the sum of the distances, [4]. This approach is called set-median problem. We will use this approximated solution as the MLE for the central permutation $\hat{\sigma}_0$. Note that its computation is done in time $O(m^2 n \, log \, n)$.

The second stage of the MLE problem concerns the estimation of the dispersion parameter for a given central permutation $\hat{\sigma}_0$. Recall that the likelihood is given in Equation (5). Moreover, as shown in Equation (3), the normalization constant $\psi(\theta)$ can be posed as a function of $S_u(n,d)$, the number of permutations of n items at distance d from e. This results on the following expression for the likelihood:

$$-m\theta\bar{d} - mLn \sum_{d=0}^{n-1} S_u(n,d) exp(-\theta d)$$

The MLE for the dispersion parameter θ is obtained by equaling to zero the derivative, which is given by the following equation

$$\frac{\delta \mathcal{L}}{\delta \theta} = -\bar{d} + \frac{\sum_{d=0}^{n-1} S_u(n,d) exp(-\theta d) d}{\sum_{d=0}^{n-1} S_u(n,d) exp(-\theta d)} \tag{6}$$

There is no closed expression for the previous formula. However, we can efficiently solve it with a Newton-Raphson method. Most root finding methods require the derivative of the expression. In our case we require the second derivative of the likelihood, which is as follows.

$$\frac{\delta^2 \mathcal{L}}{\delta \theta^2} = \frac{-(\sum_{d=0}^{n-1} S_u(n,d) exp(-\theta d) d^2)(\sum_{d=0}^{n-1} S_u(n,d) exp(-\theta d)) - (\sum_{d=0}^{n-1} S_u(n,d) exp(-\theta d) d)^2}{(\sum_{d=0}^{n-1} S_u(n,d) exp(-\theta d))^2}$$

4.1 Approximate MLE for the dispersion parameter

As we can see by looking at Equation (6), the MLE for the dispersion parameter relays on the number of permutations of n items at every possible distance, $S_u(n,d)$. Unfortunately, as we have stated in Section 2.3 the computation of $S_u(n,d)$ gets very cumbersome as n increases. Therefore, an approximation on $S_u(n,d)$ can lead to an efficient approximated MLE for the dispersion parameter. We introduce here an approximation on the values $S_u(n,d)$ based on the bounded Bratteli diagram introduced in Section 2.5.

An approximated solution for the generation of permutations of large values of n is given in Section 2.5. This solution offers a method to generate permutations which are at most at distance d', i.e. at any distance $d \leq d'$. It is based on generating a bounded Bratteli diagram in which only the FS with at least $l' = n - d'$ columns are included.

In order to approximate $S_u(n,d)$ we make use of the same bounded Bratteli diagram of Section 2.5. Note that for every FS λ in the bounded Bratteli diagram the hook lengths h_λ are known. Moreover, the number of permutations at each distance $d \leq d'$ is given by Equation (1), i.e., we have $S_u(n,d)$ for each $d \leq d'$. Moreover, since $S_u(n, n-1) = 1$ for every n, we can calculate the number of permutations for which we no dot know their distance.

$$n! = \sum_{d=0}^{d'} S_u(n,d) + \sum_{d=d'+1}^{n-1} S_u(n,d) + 1 \Rightarrow \sum_{d=d'+1}^{n-1} S_u(n,d) = n! - \sum_{d=0}^{d'} S_u(n,d) - 1$$

Our proposed approximation consists on assigning these $\sum_{d=d'+1}^{n-1} S_u(n,d)$ permutations uniformly to each of the groups $S_u(n,d)$ as follows:

$$\forall d' < d < n, \quad S_(n,d) = \frac{\sum_{d=d'+1}^{n-1} S_u(n,d)}{n - d' - 1}$$

As we can see by looking at Equation (6), every term includes the multiplication $S_u(n,d) exp(-\theta d)$. Here, as d increases, $exp(-\theta d)$ decreases. Therefore, the terms $S_u(n,d)$ are multiplied by terms that tend to 0 as d increases. This means that $S_u(n,d)$ for low values of d have more weight in the computation of the likelihood, what justifies our approach.

5 Experiments

Ulam Mallows expectation `http://arxiv.org/pdf/1306.3674.pdf`

References

[1] A. D. Gordon. A measure of agreement between rankings_Gordon1979.pdf. *Biometrika*, 66(1):7–15, 1979.

[2] Martin Bader. The transposition median problem is NP-complete. *Theoretical Computer Science*, 412(12-14):1099–1110, 2011.

[3] C de la Higuera and F Casacuberta. Topology of strings: median string is NP-complete. *Theoretical Computer Science*, 230(1-2):39–48, 2000.

[4] J Olivares-Rodríguez C.; Oncina. A Stochastic Approach to Median String Computation. *Lecture Notes in Computer Science*, 5342:431–440, 2008.