

## LECTURE 12: LEARNING DISTRIBUTIONS OVER PERMUTATIONS

PRATIK CHAUDHARI

In this lecture, we will discuss distributions over permutations and derive bounds for sampling complexity of learning such distributions using optimization techniques from the previous few lectures.

### 1. SOME BACKGROUND

Given  $n$  objects, let  $S_n$  be the space of all its permutations. Note that  $|S_n| = n!$ . Let  $\mu : S_n \mapsto [0, 1]$  be some distribution over the permutations; it has  $n! - 1$  degrees of freedom. Our question in this class will “under what conditions can we learn  $\mu$ ”. In particular, we would like a sample complexity that scales as  $O(\text{poly}(n))$ . A number of situations where this problem makes sense are:

- *People’s choice*: Given  $n$  different products and each data point is a set of products and the product chosen by the individual, we would like to learn the ordering of all the  $n$  items. In other words, this model, known as the “rational view model” and proposed by Samuelson in 1937, presumes that an individual has a perfect ordering of the  $n$  items and picks the item with the highest score from the subset shown to her. This model is however full of contradictions because choices of individuals vary based on the current setting, e.g., we may prefer Mexican food over Italian food someday and vice-versa.
- *Discrete choice models*: In this setting, an individual’s choice is a distribution over permutations and these choices are revealed through preferences such as “picking item A over item B”. For example, this is a good model for a sports tournament where team A defeats team B, C defeats D, etc. and we would like to understand the ordering between the teams.
  - “Ranked elections”, which are popular in the Commonwealth countries provide another example. In this election, voters, instead of voting for a particular candidate, provide a list of preferences. Assume every voter  $v$  provides an ordering of  $n$  candidates, i.e,  $\sigma_v \in S_n$ , so given  $\sigma_1, \dots, \sigma_m \in S_n$ , find  $\sigma^* \sim \mu$ .
  - Imagine if there are only 2 candidates, we can easily find the overall winner of this election as the one who has most votes for position 1. This motivates a simple algorithm to decide between  $n$  candidates — at each iteration remove the least voted candidate at position 1 and decide the winner after  $n - 1$  such iterations. It is easy to see that this algorithm however does not work if the electorate is highly polarized and there is a candidate who has 100% votes for the second position.

- See [Diaconis et al. \(1989\)](#) for an empirical analysis using spectral methods for ranked data from elections.

**1.1. Arrow’s impossibility theorem.** Consider the following criteria, they seem like a good set of conditions that we would like to impose upon an algorithm that decides the “best winner” using partial preferences.

- *fairness*: the algorithm weights all votes equally
- *sanity check*: if a certain learnt distribution  $\hat{\mu}$  is such that its corresponding  $\sigma^*(\hat{\mu})$  assigns  $i > j$  for some  $i$  and  $j$ , if we revert a few negative data-points, i.e., ones with  $j > i$  to positive, i.e.,  $i > j$ , and get a new  $\hat{\mu}'$  then the new  $\sigma^*(\hat{\mu}')$  still has  $i > j$ .

Arrow’s impossibility theorem [Arrow \(2012\)](#) states that with three or more candidates, no ranked ordering system can convert the ranked preferences of the voters into a community-wide ranking while also maintaining the above criteria. Put other way, the above conditions lead to a dictatorial election procedure, i.e., one which prefers a certain candidate over others irrespective of the data.

See the article [Arrow’s Theorem Proves No Voting System is Perfect](#) for a nice discussion.

## 2. MULTINOMIAL LOGIT MODEL

The above discussion helps to concretize the problem formulation.

**Problem 1.** Given  $\mu : S_n \mapsto [0, 1]$ , iid samples from  $\mu$ , say  $\sigma_t$  are drawn as follows: at time  $t$ , sample  $\sigma_t \in S_n$  according to  $\mu$ . We however only get to observe the ordering between two items  $i_t, j_t$  that are chosen randomly from  $n$  items, i.e., pick a pair  $i, j$  from the ordered set  $[n] = \{1, 2, \dots, n\}$  and output  $i_t > j_t$  or  $i_t < j_t$  depending upon their index in  $\sigma_t$ . Our objective is to “learn  $\mu$ ” from this data.

The pairs  $i, j$  can be chosen a priori or can be desgined using some scheme. Note that data size is  $O(n^2)$  and hence learning  $\mu$  directly is not possible (since  $\mu$  has  $n! - 1$  parameters), we need to simplify  $\mu$  in some way, which we do as follows.

**2.1. Multinomial Logit Model [Thurstone 1927].** Each item  $i$  has a score

$$S_i = u_i + Z_i$$

associated with it where  $u_i$  is a deterministic factor, e.g., its inherent “goodness” and  $Z_i$  is noise, e.g., a team’s performance on a given day.  $Z_i$  is assumed to independent from everything else, e.g.,  $Z_i \sim N(0, \rho_i)$  where  $\rho_i$  is some parameter based on  $i$ . The descending order of  $S_i$ s provides a permutation of  $[n]$ . [Elo](#) (FIDE’s rating system for chess players) uses this model. Another example is Microsoft’s TrueSkill gaming engine which uses the  $Z_i \sim N(0, \rho_i)$  model.

We will use a different distribution for  $Z_i$  here known as the Gumbel distribution. Thus

$$Z_i \sim \text{Gumbel}(0, 1)$$

where the density of the Gumbel distribution is given by  $f(x) = \exp(-x - \exp(-x))$ . Note that we are assuming  $Z_i$ s to be homogeneous, i.e.,  $S_i \geq S_j$  iff  $u_i \geq u_j$ .  $u_i$ s are the

parameters that we would like to learn from  $O(n \log n)$  samples of the form  $i > j$  or not. Note that this model induces an inherently  $O(n^2)$  distribution.

**Remark 2.** If  $(u_i - u_j)$  is large, we can see that this model performs well. Note that

$$\begin{aligned} P(S_i > S_j) &= P(Z_i - Z_j > u_j - u_i) \\ &= P(N(0, 2\rho^2) > u_j - u_i) \quad \text{if } Z_i \sim N(0, \rho^2) \\ &= 1 - \Phi\left(\frac{u_j - u_i}{\sqrt{2}\rho}\right) \end{aligned}$$

where  $\Phi(x) = P(N(0, 1) \leq x)$  is the CDF of a Gaussian. If  $u_i \gg u_j$ , we have

$$\log P(S_i > S_j) \approx \frac{(u_i - u_j)^2}{4\rho^2} - \frac{1}{2} \log 4\pi\rho^2.$$

Thus  $S_i > S_j$  is very likely.

**2.2. Motivation for Gumbel distribution.** Let us first note a neat fact of this distribution.

**Lemma 3.** *Difference of two Gumbel random variables is a Logistic random variable, i.e.,*

$$X, Y \sim \text{Gumbel}(0, 1) \implies W = X - Y \sim \text{Logistic}(0, 1).$$

*In other words,*

$$P(W < w) = \frac{1}{1 + e^{-w}}.$$

*Proof.* This fact can be proved using convolution, which is messy. We shall use moment generating functions to prove it instead. Note that

$$\begin{aligned} E[\exp(\theta W)] &= E[\exp(\theta X) \exp(-\theta Y)] \\ &= E[\exp(\theta X)] E[\exp(-\theta Y)] \\ \implies M_W(\theta) &= M_X(\theta) M_Y(-\theta). \end{aligned}$$

where  $M_W(\theta)$  is the moment generating function of  $W$ . It is known that

$$E[M_X(\theta)] = \Gamma(1 - \theta) \exp(\theta).$$

where  $\Gamma(n) = (n - 1)!$  if  $n$  is an integer. Thus,

$$\begin{aligned} E[\exp(\theta W)] &= \Gamma(1 - \theta) \exp(\theta) \Gamma(1 + \theta) \exp(-\theta) \\ &= \Gamma(1 - \theta) \Gamma(1 + \theta) \\ &= \frac{\Gamma(1 - \theta) \Gamma(1 + \theta)}{\Gamma(2)} \\ &= \text{Beta}(1 - \theta, 1 + \theta). \end{aligned}$$

where  $\text{Beta}(x, y) = \frac{\Gamma(x)\Gamma(y)}{\Gamma(x+y)}$  is the Beta distribution which is also the MGF of  $\text{Logistic}(0, 1)!$

□

Using Lem. 3, we can see that

$$\begin{aligned}
 P(S_i > S_j) &= P(u_i + Z_i > u_j + Z_j) \\
 &= P(Z_i - Z_j > u_j - u_i) \\
 &= \frac{1}{1 + \exp(u_i - u_j)} \\
 \implies P(i > j) &= \frac{\exp(u_i - u_j)}{1 + \exp(u_i - u_j)} = \frac{e^{u_i}}{e^{u_i} + e^{u_j}}.
 \end{aligned}$$

The probability of  $i$  being chosen over  $j$  is thus proportional to exponentiated intrinsic skill of  $i$ . In a sense, the Gumbel distribution we used above is retro-fitted to achieve this, which is what was prevalent in literature.

However, quite surprisingly, [Luce \(2012\)](#) proved in 1946 that this form for  $P(i > j)$  is the only distribution where we have “irrelevance of external choices”, e.g., given two options for desert, a chocolate cake and coffee, if one chooses coffee, upon being presented a third option, say mint-tea, the individual does not change her choice to chocolate cake. Note that this is not true for the example of elections, if people prefer the marginal risk of choosing candidate A over B, upon being presented a third option C, the electorate might be so polarized that they will conservatively vote for B to avoid C winning at any cost.

Let us give a simulation procedure for the Gumbel distribution. Let  $w_i = e^{u_i}$ . We need to generate  $\sigma$  s.t.  $P(i > j) = \frac{w_i}{w_i + w_j}$ . Start filling up an array of length  $n$  where

$$P(i \text{ is picked for current position}) = \frac{w_i}{\sum_k w_k}.$$

To see that this scheme indeed produces the desired distribution, we use a stopping time argument. Let  $\tau$  be the first time when either item  $i$  or  $j$  is selected. Observe that conditional upon either  $i$  or  $j$  being picked, the probability of  $i$  being picked, i.e., probability of  $i$  being chosen over  $j$  is simply,

$$P(i > j) = \frac{w_i / \sum_k w_k}{w_i / \sum_k w_k + w_j / \sum_k w_k} = \frac{w_i}{w_i + w_j}.$$

### 3. FORMAL SETUP

Our formal setup is as follows.

**Problem 4.** Given  $m$  observations  $(i_k, j_k)$  where  $i_k > j_k$ , i.e., item  $i_k$  is picked over  $j_k$ , reconstruct the exponentiated “skill” parameters  $w_1, \dots, w_n$ .

To solve this problem, we shall maximize the likelihood function. Consider,

$$\begin{aligned} P((i_1, j_1), \dots, (i_m, j_m)) &= \prod_{k=1}^m P((i_k, j_k)) \\ &= \prod_{k=1}^m \frac{w_{i_k}}{w_{i_k} + w_{j_k}} \\ &= \prod_{k=1}^m \frac{\exp(u_{i_k} - u_{j_k})}{1 + \exp(u_{i_k} - u_{j_k})}. \end{aligned}$$

Let  $x_k \in \{0, 1\}^n$  be a vector where  $x_k(i_k) = 1$ ,  $x_k(j_k) = -1$  and all the others are zeros.  $x_k$  thus has a +1 at the  $i_k^{th}$  element and a -1 at the  $j_k^{th}$  element if  $i_k > j_k$ . If we let  $u = [u_1, \dots, u_n]'$  we can see that

$$u'x_k = u_{i_k} - u_{j_k}.$$

Simplify the above equation to get

$$P((i_1, j_1), \dots, (i_m, j_m)) = \prod_{k=1}^m \frac{\exp(u'x_k)}{1 + \exp(u'x_k)}.$$

If we let  $-L(u) = \log P(u; x_1, \dots, x_m)$  be the log-likelihood function, we have

$$-L(u) = \sum_{k=1}^m [u'x_k - \log(1 + \exp(u'x_k))].$$

The ML estimate of  $u$  is then the solution of the following problem

$$\min_{u \in \mathbb{R}^n, \sum u_i = 0} - \sum_{k=1}^m [u'x_k - \log(1 + \exp(u'x_k))]. \quad (1)$$

**Remark 5.** Let us assume that we know some bound  $b$  s.t.  $\|u\|_\infty \leq b$ . Also, note that since  $L(u)$  only depends upon terms of the form  $u_{i_k} - u_{j_k}$ , it is invariant to shifting, so we can set  $\sum u_i = 0$  as another constraint in the problem.

Let us note that Prob. 4 is a convex optimization with nice constraints and hence can be solved efficiently.

**3.1. Sample complexity of learning.** Let  $\hat{u}$  be the solution of Prob. 4 and  $u^*$  be the actual model. By definition, we have

$$\begin{aligned} L(\hat{u}) &\leq L(u^*) \\ \implies L(\hat{u}) - L(u^*) - \nabla L(u^*)' \Delta &\leq -\nabla L(u^*)' \Delta \end{aligned}$$

where  $\Delta = \hat{u} - u^*$ . The idea is to then bound

$$\|\nabla L(u^*)\| \leq \alpha$$

for some  $\alpha$  and get

$$L(\hat{u}) - L(u^*) - \nabla L(u^*)' \Delta \leq \|\nabla L(u^*)\|_2 \|\Delta\|_2.$$

But note that

$$L(\hat{u}) - L(u^*) - \nabla L(u^*)' \Delta \geq \frac{1}{2} \Delta' \nabla^2 L(u^* + \lambda \Delta) \Delta$$

where  $\lambda \in [0, 1]$  and thus  $u^* + \lambda\Delta$  lies on the line joining  $u^*$  and  $\hat{u}$ . (This is an application of the mean value theorem). If we can find some  $\beta$  such that

$$\frac{1}{2} \nabla^2 L(u^* + \lambda\Delta) \geq \beta,$$

we will have

$$\begin{aligned} \beta \|\Delta\|_2^2 &\leq L(\hat{u}) - L(u^*) - \nabla L(u^*)' \Delta \leq \alpha \|\Delta\|_2 \\ \text{which } \implies \|\Delta\|_2 &\leq \frac{\alpha}{\beta} \end{aligned}$$

and we thus get a bound on  $\|\hat{u} - u^*\|$ .

3.1.1. *Getting  $\alpha$  and  $\beta$ .* Let us first compute  $\beta$ . Note that since  $L(\cdot)$  is a sum, we can consider each individual term separately. Also  $u'x_k$  terms in Eqn. (1) vanish while taking the Hessian. Thus if  $z = u^* + \lambda\Delta$  as we used in the previous section, we have

$$\begin{aligned} \nabla^2 L(z) &= \frac{\partial^2}{\partial z_i \partial z_j} \log(1 + \exp(z'x_k)) \\ &= \frac{\partial}{\partial z_j} \sum_{k=1}^m \frac{1}{1 + \exp(z'x_k)} \exp(z'x_k) x_{ki} \\ &= \sum_{k=1}^m x_{ki} \left\{ \frac{\exp(z'x_k) x_{kj}}{1 + \exp(z'x_k)} - \frac{\exp(z'x_k)^2 x_{kj}}{(1 + \exp(z'x_k))^2} \right\} \\ &= \sum_{k=1}^m \frac{x_{ki} x_{kj} \exp(z'x_k)}{1 + \exp(z'x_k)} \left[ 1 - \frac{\exp(z'x_k)}{1 + \exp(z'x_k)} \right] \\ &= \sum_{k=1}^m \frac{x_{ki} x_{kj} \exp(z'x_k)}{(1 + \exp(z'x_k))^2}. \end{aligned}$$

Finally,

$$\Delta' \nabla^2 L(z) \Delta = \sum_k \sum_{i,j} \frac{\Delta_i \Delta_j x_{ki} x_{kj} \exp(z'x_k)}{(1 + \exp(z'x_k))^2}.$$

Now note that  $z = u^* + \lambda\Delta$  and we have assumed  $\|u^*\|_\infty \leq b$  and  $\|\Delta\|_\infty \leq 2b$  (since  $\Delta = \hat{u} - u^*$ ). Thus

$$\begin{aligned} \|z\|_\infty &\leq 3b \quad \text{since } \lambda \in [0, 1] \\ \implies |z'x_k| &\leq 3b. \end{aligned}$$

Let  $\tilde{b} = \exp(3b)$  and see that

$$\begin{aligned} \Delta' \nabla^2 L(z) \Delta &\geq \frac{\tilde{b}}{(1 + \tilde{b})^2} \sum_k \left[ \sum_{i,j} \Delta_i \Delta_j x_{ki} x_{kj} \right] \\ &= \frac{\tilde{b}}{(1 + \tilde{b})^2} \left[ \sum_k (\Delta' x_k)^2 \right]. \end{aligned}$$

It can now be shown using matrix concentration inequalities that

[PC] the notes say that this is  $b$ , it should be  $2b$  instead?

**Claim 6.** For iid samples and  $m > 12 n \log n$ , w.h.p.

$$\frac{1}{m} \sum_{k=1}^m (\Delta' x_k)^2 \geq \frac{1}{3n} \|\Delta\|_2^2.$$

**Claim 7.**

$$\|\nabla L(u^*)\|_2 \leq 2\sqrt{\frac{\log n}{m}} = \alpha.$$

Using the above two claims, we finally have

$$\|\Delta\|_2 \leq \frac{\tilde{b}}{(1 + \tilde{b})^2} \sqrt{\frac{n^2 \log n}{m}}.$$

**Remark 8.** Note that the  $\tilde{b}$  terms in the above bound should not really be there, i.e., the error of our algorithm should not depend upon how well separated different candidates are in terms of their skills. Arguably, larger the value of  $b$ , the easier it is to discover the winner.

write about the intuition Devavrat gave about the  $\frac{n^2 \log n}{m}$  term here.

[PC]

#### 4. RANK CENTRALITY

The material in this section follows [Negahban et al. \(2012\)](#).

#### REFERENCES

- Arrow, K. J. (2012). *Social choice and individual values*, volume 12. Yale university press.
- Diaconis, P. et al. (1989). A generalization of spectral analysis with application to ranked data. *The Annals of Statistics*, 17(3):949–979.
- Luce, R. D. (2012). *Individual choice behavior: A theoretical analysis*. Courier Dover Publications.
- Negahban, S., Oh, S., and Shah, D. (2012). Iterative ranking from pair-wise comparisons. *CoRR*, abs/1209.1688.
- Negahban, S. and Wainwright, M. J. (2012). Restricted strong convexity and weighted matrix completion: Optimal bounds with noise. *The Journal of Machine Learning Research*, 13(1):1665–1697.