How to Order Sushi

A Nonparametric Approach to Modeling Rank Data

William Chen

1 April, 2014

Abstract

Rank data is often encountered in our daily lives (e.g. sports team rankings, horse races, voting). The data is deceptively simple, yet learning from the data is far from straightforward. Traditional random utility models (RUMs), such as the Plackett-Luce RUM and Normal RUM, seek to capture the structure of rank data via distributional assumptions on latent utilities. This can make inference tractable, but leaves the models inexpressive and unable to fully capture features of data. I propose a new class of nonparametric random utility models (NPRUMs) for rank data, and present an estimation algorithm based on variational Monte Carlo expectation-maximization and kernel density methods. I show that NPRUMs provide better insights into random utilities in different settings, such as elections and sushi preferences. In particular, the model outperforms existing models in terms of out-of-sample likelihood, rank smoothing, and rank completion.

Acknowledgments

Firstly, I would like to thank my thesis advisor David Parkes for his encouragement, support, and time during my work on this thesis. Your advice, instruction, and high expectations have been invaluable in providing direction and guidance, and your multiple rounds of feedback in the past year have helped this research and my research interest grow and develop. I wanted to thank my mentor Hossein Azari Soufiani for your mentorship and patience, working closely with me for over a year to develop these ideas. The many meetings have been productive, and I hope helpful for you as well as you near turning in your own dissertation. I wanted to thank both David Parkes and Hossein Azari Soufiani for getting me up to speed with the material, working closely with me to get a similar paper submitted to ICML, and for introducing me to the promising world of research.

I want to thank Joe Blitzstein for being a fantastic mentor, teacher, and friend over the past four years – you've helped me grow and flourish as a statistics student and teaching fellow and I owe much of my passion in statistics, storytelling, and teaching statistics to you. I would like to thank Ryan Adams for his feedback and encouragement for when this thesis started off as a final project for CS 281 – it was very helpful in the early stages of this thesis. I want to thank Mark Glickman for getting me interested in looking at pairwise comparison algorithms to predict the March Madness tournament.

I also wanted to extend thanks to my main statistics study group, composing of Sebastian Chiu, Jessy Hwang, and Raj Bhuptani. We've gone through a lot of hard statistics classes together and the support I received from all of you were vital for my survival in those classes. Sebastian Chiu, I'd especially like to acknowledge you for your support and all those late nights of ping pong and pool, sometimes well-deserved after hard work. I especially appreciate our insightful discussions about life, data science, and the future. Your help has gotten me through many classes - thank you. I wanted to thank my very good friend Raj Bhuptani for letting his thesis, work ethic, and commitment to quality be an inspiration for me. You've provided very helpful guidance on formatting and making this look like a thesis. You're always a great person to bounce ideas off of, ask for advice, and a friend and mentor that I look up to.

I wanted to thank my girlfriend Yuan Jiang for her endless support that kept me motivated and empowered to finish, especially during long nights working on this thesis. Your presence and support is always appreciated. Finally, I'd like to thank my family, for whom I hope this thesis makes proud.

Contents

List of Figures **xi**

List of Tables **xv**

1 Introduction **1**

1.1 Contributions 4

1.2 Related Work 4

1.3 Outline 5

2 Preliminaries **7**

2.1 Notation 7

2.2 Types of Rank Data 8

2.3 Traditional Random Utility Models 10

2.3.1 Plackett-Luce RUM 10

2.3.2 Normal RUM and its extensions 11

3 Nonparametric Random Utility Models **13**

3.1 Kernel Density Estimation 13

3.2 Rank Position Distribution 16

3.3 Motivation for NPRUM 17

4 An Estimation Algorithm **19**

4.1 E-step 20

4.2 M-step 22

4.3 Algorithm 23

5 Datasets **25**

5.1 Sushi dataset 25

5.2 Sushi Rank Positions 26

5.3 Sushi Pairwise Preferences 28

5.4 Sushi Recommendations 30

5.5 Election Dataset 32

5.6 Nascar Dataset 34

6 Experimental Results **37**

6.1 Capturing Heterogeneity and Correlation of Utilities 38

6.1.1 Heterogeneity 38

6.1.2 Utility Correlation 40

6.2 RPD Prediction via Smoothing 41

6.3 RUM Comparison Results 44

6.4 Rank Completion 47

6.5 Summary 49

7 Discussion **51**

7.1 Distributional assumptions 51

7.2 Estimation 52

7.2.1 Time Complexity 52

7.2.2 Space Complexity 52

7.3 Inference 53

8 Conclusions 55

8.1 Extensions 56

References 59

List of Figures

1.1 Random utility model. 2

1.2 A set of draws are observed from the random utility distributions of items A, B, and C. 3

1.3 A rank order is formed between items A, B, C according to realized utilities. 3

2.1 Different types of rank data: full, top, and sub rankings. 8

3.1 Two parametric density estimates (Gamma and Normal) and one non-parametric density estimate (Kernel density estimate). 13

3.2 The Kernel density estimate of the utility distribution of alternative A is formed by summing up Gaussian kernels placed on the data points. 14

3.3 Kernel desntiy estimates with various bandwidths. 15

3.4 As example RPD. The space has $2^2 = 4$ squares, but only $2! = 2$ are associated with actual permutations. Those permutations are associated with densities. 16

4.1 In this example, the observed rank order $\{A \succ B\}$ and the latent utility distributions from the M-step are used to sample utilities u_A and u_B. The steps alternate between making draws for A and B, always preserving the relation $u_A > u_B$. This continues until convergence. The resulting utilities are passed to the M-step. 20

4.2 In this example, five values of u_a are returned from the E-step. No distributional assumptions are made in NPRUM, and so the random utility distribution for alternative A is estimated using Gaussian kernels with bandwidth h. The resulting kernel density estimate is passed back to the E-step. (copy of Figure 3.2) 22

5.1 The ten sushi variants in the Kamishima sushi dataset. 26

5.2 Agents' favorite, second-favorite, third-favorite, and least favorite sushi variants. . . 27

5.3 Marginal rank position distributions of all 10 types of sushi. 28

5.4 The pairwise preference matrix for the sushi dataset (only the upper right triangle is shown for visualization clarity). Each entry is the probability that the sushi variant in the row is preferred over the sushi variant in the column. 29

5.5 Each graph compares the distribution of second-preference sushi between two types of agents: those who listed the first sushi of a pair as first-preference, and those who listed the second sushi of a pair. The p-values are from a (simulated) Fisher Exact Test with 100,000 simulated draws. Error bands show 95% confidence intervals for proportions. 31

5.6 The pairwise preference matrix for the election dataset. 32

5.7 Pairwise win matrix of Nascar dataset. Drivers are ordered by final ranking in the 2002 Nascar Tournament, which was decided by a modified positional voting rule. Drivers ranked in the top half have all participated in 20 or more races, while almost all drivers in the bottom half have participated in 10 races or fewer. Gray squares indicate that the pair of racers have never faced each other in the same match. . . . 36

6.1 Estimated Gumbel densities on utilities in the Plackett-Luce RUM. 38

6.2 Estimated densities on utilities in the Normal RUM. 39

6.3 Estimated densities on utilities in the Multitype Normal RUM. 39

6.4 Estimated densities on utilities in the Nonparametric RUM ($h = 0.11$). Bandwidth chosen through cross-validation (see Section 6.3). 40

6.5 Marginal rank position distributions of the sushi dataset (copy of Figure 5.3) 40

6.6 Joint distribution for two sets of positively correlated (salmon roe and sea urchin) and negatively correlated (cucumber roll and fatty tuna) sushi. The orange region represents the preference salmon roe over sea urchin or cucumber roll over fatty tuna. 41

6.7 Pairwise correlation matrix for observed utilities of all ten sushi variants. 42

6.8 The empirical RPD of first 50 sushi agents 43

6.9 The NPRUM fit on first 50 sushi agents. In this graph $h = 0.12$, the best performing bandwidth from the experimental smoothing results from Figure 6.11. 43

6.10 The regenerated RPD. This is the RPD on 5000 *simulated* agents drawn from NPRUM fit on first 50 sushi agents. 43

6.11 Out-of-sample RPD prediction performance. x-axis is bandwidth (h). y-axis is TVD. 75 repetitions are done for each data point. Error bars represent 95% confidence intervals. n represents the number of agents for which RPD was smoothed. 45

6.12 The choice of h affects the underlying nonparametric random utility distribution. If h is too low, there are spurious artifacts. If h is too high, it drowns out the features of the distribution. 46

6.13 Out-of-sample rank completion performance. x-axis is bandwidth (h). y-axis is weighted mean TVD. 100 repetitions are done for each data point. Error bars represent 95% confidence intervals. n represents the number of agents used as training for rank completion. 48

List of Tables

5.1 The thirteen most common ballots (all ballots that at least 4 voters submitted). . . 34

6.1 Datasets used for evaluation. † denotes a subset of the full data 37

6.2 (top) Average log likelihood. (bottom) Total variation distance between pairwise matrices. Numbers in bold are significantly better than other methods. * means that the method does not converge 46

6.3 Runtime (seconds). Numbers in bold are significantly better than other methods. * means the method does not converge. 46

6.4 Performance summary of five implemented solutions on the Sushi dataset. Standard errors are included in parentheses. Smoothing is evaluated with $n = 50$ and completion is evaluated with $n = 100$. Bandwidth is set to the $h = 0.11$ used in Section 6.3. 49

Chapter 1

Introduction

Rank data appears in many places. Examples include user preferences, search engine rankings, chess match results, and votes in elections. Given this data, example ranking problems include assessing preferences between electric and gasoline cars (Beggs et al., 1981), aggregating search rankings into meta-search results (Dwork et al., 2001), determining winners of tournaments (Hunter, 2004), and declaring the winner of an election (Gormley and Murphy, 2006).

The need to analyze or aggregate rank data presents an interesting and challenging machine learning problem, especially due to the factorial size of the rank order space. For example, in inference, it may be intractable to enumerate over all rank orders to find the maximum a posteriori probability ranking.

One approach to learning from rank data is to assume that the rankings come from a probabilistic model. A specific case of probabilistic models are random utility models (RUMs). RUMs are statistical methods for ranking problems adopted from economics (Thurstone, 1927; McFadden, 1974) to infer preferences or importance between alternatives (Xia et al., 2008; Azari Soufiani et al., 2012).

In rank data problems, there are n ranks, denoted $\vec{\pi}_1, \vec{\pi}_2, \ldots, \vec{\pi}_n$, over the m alternatives, denoted $\mathcal{C}$. For example, on the three alternatives $\mathcal{C} = \{c_1, c_2, c_3\}$, the first rank order $\vec{\pi}_1$ can be $\{c_1 \succ c_2 \succ c_2\}$,

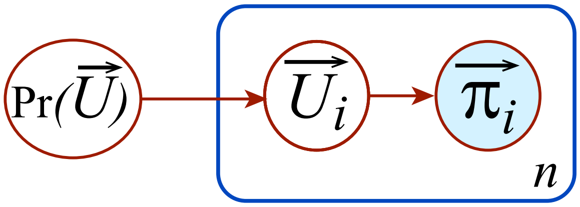


Figure 1.1: Random utility model.

and the second rank order $\vec{\pi}_2$ can be $\{c_2 \succ c_1 \succ c_3\}$. Here, $n = 2$ and $m = 3$. The symbol $\succ$ denotes a preference relationship. The subscript i in $\vec{\pi}_i$ lets us index over all of the ranks.

In RUMs, each rank ordering $\vec{\pi}_i$ arises from a sample of random utilities $\vec{u}_i$ for the alternatives $\mathcal{C}$ according to the joint distribution $\Pr(\vec{u})$. Then, the alternatives are ordered by their utilities, and ranks are reported as $\vec{\pi}_i$ (Figure 1.1). Using a random utility model, the goal of learning from rank data includes performing inference on $\Pr(\vec{u})$ in order to predict, model, and complete ranks. Figures 1.2 and 1.3 illustrates concretely how ranks are formed in a random utility model. Throughout this thesis, Pr is used as shorthand for $\Pr(\vec{u})$.

A rank $\vec{\pi}_i$ may also be a partial rank ordering instead of a full rank ordering. A full rank ordering is given when all available alternatives are ranked. A partial rank ordering is given when only a subset of alternatives are ranked (e.g. games, competitions, races) or the orders provide only top preferences out of a set of alternatives (e.g. candidates in elections).

Since a ranking is determined by the order of utility values, the probability of observing a specific preference rank is simply the probability of the corresponding ordering of utility draws. Let $\pi(j)$ denotes the jth ranked alternative in rank order π. Let $U_{\pi(1)}$ denote a random draw from the random utility distribution of alternative $\pi(1)$. For an example with three alternatives:

$$Pr(\pi(1) \succ \pi(2) \succ \pi(3)) = Pr(U_{\pi(1)} > U_{\pi(2)} > U_{\pi(3)}) \tag{1.1}$$

In existing RUMs, the joint probability is a product distribution. Two popular RUM methods are the Plackett-Luce (PL) model (Luce, 1959; Block and Marschak, 1960; Plackett, 1975; Marden,

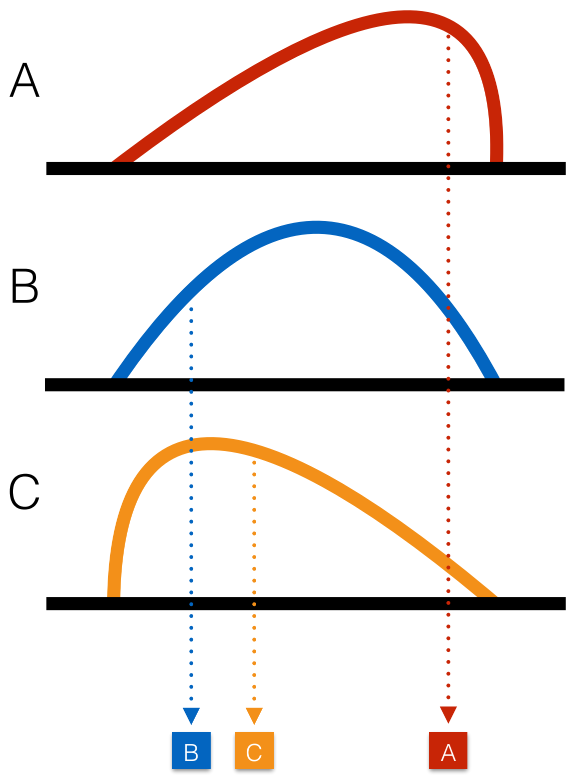


Figure 1.2: A set of draws are observed from the random utility distributions of items A, B, and C.

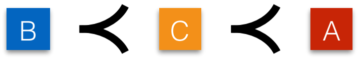


Figure 1.3: A rank order is formed between items A, B, C according to realized utilities.

1995) and the Thurstone model (Thurstone, 1927). The Thurstone model was defined for pairwise preferences but has been generalized to full rank data by Azari et al. (2012). Plackett-Luce and Thurstone assume that the latent random utility on each alternative is are drawn from independent Gumbel and Normal distributions, respectively.

The parametric distributions restrict the space of possible random utility functions. In addition, these models preclude correlation structure between utilities. In this thesis, I present the nonparametric random utility model (NPRUM), which is designed to address these issues.

1.1 Contributions

I forgo the aforementioned parametric and independence assumptions in existing RUMs, and instead fit a nonparametric model that allows both flexible densities and correlation between the utility on each alternative. I present a Monte Carlo expectation-maximization algorithm to perform this fit, and provide an implementation in the `StatRank` package in R (Chen and Azari Soufiani, 2013). Additionally, I present a framework to perform inference on the random utility model via sampling from the resulting joint utility distribution. I motivate NPRUM by providing a connection to the rank position distribution.

My empirical results show that NPRUM is a better out-of-sample fit to Kamishima's sushi dataset (2003), Tideman's election dataset (2006), and the 2002 Nascar Winston Cup Series dataset (Hunter, 2004). It outperforms existing RUMs in multiple predictive tasks, including estimating the pairwise preference matrix, smoothing ranks given noisy data, and predicting a user's second preference given their first. It also demonstrates superior predictive average log-likelihood. I show that the good predictive performance is due to NPRUM's flexibility, and its ability to model correlation.

NPRUMs also unlock new understandings via post-processing and visualization. I illustrate this in Section 6.

1.2 Related Work

Performing inference on rank data is a well-studied problem (Dwork et al., 2001; Conitzer, 2006; Ailon, 2007; Truchon, 2008; Xia et al., 2008). Famous ranking algorithms include PageRank (Page et al., 1999), the Elo method for chess ratings (Elo, 1978), and the Bowl Championship Series / Ratings Percentage Index for NCAA college football and basketball (Langville and Meyer, 2012). However, these studies and applications all adapt parametric models.

There are other nonparametric methods in the literature, such as that used by Ammar & Shah (2011) on pairwise data. This thesis focuses on nonparametric models for rank data.

The most relevant related work involves other random utility models. The most common example from literature is the Plackett-Luce RUM (PLRUM), which has been applied to rank documents (Cao et al., 2007), model the effect of party politics (Gormley and Murphy, 2007), and model dietary preferences in lactating cows (Nombekela et al., 1994). Other papers explore new algorithms to fit PLRUM via message-passing (Guiver and Snelson, 2009) and minorize-maximization (Hunter, 2004). Others explore extensions of PLRUM that increase the expressiveness of the model with mixture modeling (Gormley and Murphy, 2008; Azari Soufiani et al., 2013).

Another prominent RUM from literature is the Thurstone RUM, which introduces an uncertainty parameter for each alternative. Thurstonian models have been adopted for massive online game systems with Glicko (Glickman, 1999) for chess games and TrueSkill (Herbrich et al., 2006) for XBox Live games. Azari et al. (2012) provides a Monte Carlo expectation-maximization algorithm to extend Thurstone RUM from pairwise comparisons to rank data. I use Monte Carlo expectation-maximization and add certain procedures (e.g. slice sampling and kernel density estimation) to fit nonparametric distributions.

See Section 2.3 for more discussion on PLRUM, Normal RUM, and extensions of Normal RUM.

1.3 Outline

In the next section (Section 2), I introduce the notation used throughout this thesis, the three types of rank data that I study, and give a brief summary of existing RUMs. Section 3 introduces nonparametric random utility models, kernel density estimation, and rank position distributions. I motivate nonparametric RUMs by describing their relationship to the distribution of positions of alternatives in rank data. I describe an estimation procedure and algorithm in Section 4. In Section 5, I describe the datasets used in this thesis and motivate applications of NPRUM.

In Section 6, I present the main experimental results and provide performance comparisons between NPRUM and existing methods. In addition, I present a novel application of the nonparametric

RUM, showing that partial data about a ranking can be used to predict the full ranking.

Section 7 discusses the tradeoffs of using a nonparametric RUM versus existing parametric models. Finally, I conclude and present ideas for future work in Section 8.

Chapter 2

Preliminaries

In this chapter, I provide a summary of the notation used in this thesis, the three types of rank data I used in evaluating the models, and a brief background on existing random utility models.

2.1 Notation

There are n rankings (sometimes attributed to n agents) of m alternatives indexed by i and j respectively. Let $\mathcal{C} = \{c_1, \ldots, c_m\}$ denote the set of m alternatives. Let $\Pi = \{\vec{\pi}_1, \ldots, \vec{\pi}_n\}$ denote the data, where each $\vec{\pi}_i$ is a ranking (full or partial) over $\mathcal{C}$. The count n is the total number of rankings provided. The rankings may come from n agents or may be the result of search engine results, chess games, and so forth.

The alternatives $\mathcal{C}$ are associated with utilities $u_{c_1}, \ldots, u_{c_m}$ from the interval $[0, 1]$ according to a nonparametric joint density $Pr : [0, 1]^m \to \mathbb{R}$. The preference ordering of the alternatives $\mathcal{C}$ is determined by the the relative magnitudes of utilities $u_{c_1}, \ldots, u_{c_m}$ – alternatives with higher utilities are ranked higher.

2.2 Types of Rank Data

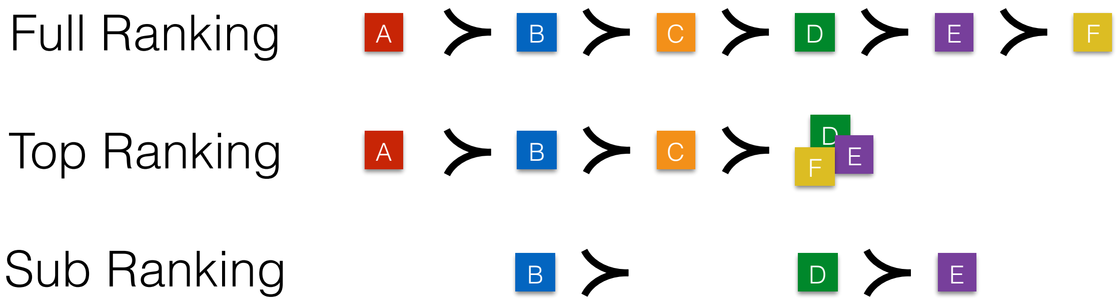


Figure 2.1: Different types of rank data: full, top, and sub rankings.

There are different types of rank data. In this thesis, I consider the full ranking, sub ranking, and top ranking types. See Figure 2.1.

Definition 1 Full Ranking: *A full ranking has all alternatives $\mathcal{C}$ ranked. One observes the ranking $\vec{\pi} = [\pi(1) \succ \pi(2) \succ \cdots \succ \pi(m)]$, containing all m alternatives.*

Given Pr, the probability for a ranking $\vec{\pi} = [\pi(1) \succ \pi(2) \succ \cdots \succ \pi(m)]$ (which is equivalent to $[u_{\pi(1)} > u_{\pi(2)} > \cdots > u_{\pi(m)}]$) is defined as follows:

$$\Pr(\vec{\pi}) = \int_{u_{\pi(m)} < \cdots < u_{\pi(1)}} \Pr(\vec{u}_\pi) d\vec{u}_\pi \tag{2.1}$$

Definition 2 Top Ranking: *A top ranking provides full rankings on a proper subset $\mathcal{C}' \subsetneq \mathcal{C}$ with at least two alternatives. All elements of $\mathcal{C}'$ are preferred over the elements in $\mathcal{C}'^c$, defined $\mathcal{C}'^c = \{c \in \mathcal{C} | c \notin \mathcal{C}'\}$. No information is gained of the preference relationship within the set $\mathcal{C}'^c$.*

One observes the ranking $\vec{\pi} = [\pi(1) \succ \pi(2) \succ \cdots \succ \pi(m') \succ \{\pi_c\}_{c \in \mathcal{C}'^c}]$, where the set of m' (where $m' < m$) alternatives in $\mathcal{C}'$ are fully ranked and preferred over the other alternatives in $\mathcal{C}'^c$. This ranking π implies $[u_{\pi(1)} > \cdots > u_{\pi(m')} > \max(\{u_c\}_{c \in \mathcal{C}'^c})]$. The probability of observing such a ranking is:

$$\Pr(\vec{\pi}) = \int_{\max\left(\{u_c\}_{c\in\mathcal{C}'^c}\right)<u_{\pi(m')}<\cdots<u_{\pi(1)}} \Pr(\vec{u}_\pi)d\vec{u}_\pi \tag{2.2}$$

This data type occurs for example in elections with many candidates. Voters fill out their top positions with their preferred candidates, and may leave their less desired candidates unranked.

Definition 3 Sub Ranking: *A sub ranking provides full rankings on a proper subset $\mathcal{C}' \subsetneq \mathcal{C}$ with at least two alternatives. No information is learned about the alternatives in the set $\mathcal{C}'^c$, or about the relationship between the sets $\mathcal{C}'^c$ and $\mathcal{C}'$.*

One observes the ranking $\pi = [\pi(1) \succ \pi(2) \succ \cdots \succ \pi(m')]$ on the set of m' (where $m' < m$) alternatives $\mathcal{C}$. This ranking π implies $[u_{\pi(1)} > \cdots > u_{\pi(m')}]$. The probability of observing such a ranking is:

$$\Pr(\vec{\pi}) = \int_{u_{\pi(m')}<\cdots<u_{\pi(1)}} \Pr(\vec{u}'_\pi)d\vec{u}'_\pi \tag{2.3}$$

where $\vec{u}'$ is the vector of all $u \in \mathcal{C}'$.

This commonly occurs in race or competition data, where only a subset of the racers and competitors are compared in each ranking.

Integrals 2.1, 2.2, and 2.3 are computationally difficult to compute without distributional assumptions. Yet understanding them is vital in order to perform inference. I will use Monte Carlo methods to estimate probabilities of rank orders and the likelihoods of observed data.

2.3 Traditional Random Utility Models

2.3.1 Plackett-Luce RUM

The Plackett-Luce RUM (PLRUM) is named in honor of independent work by Plackett (1975) and Luce (1959). It has found many applications, including to horse-racing (Plackett, 1975), the analysis of Irish election data (Gormley and Murphy, 2007), and permutation-based optimization problems (Ceberio et al., 2013),

The key feature of the PLRUM is that it satisfies Luce's Choice Axiom, which states that pairwise preferences between two items are not affected by the presence or absence of other items being compared (1959). This implies that the probability of choosing any alternative c_i over the other alternatives in $\mathcal{C}$ is

$$P(c_i \succ \text{the rest}) = \frac{\gamma_{c_i}}{\sum_{c_j \in \mathcal{C}} \gamma_{c_j}}, \tag{2.4}$$

where the (non-negative) γ_{c_j} represents the sole parameter for each alternative. In PLRUM, the γ parameters are manifested as the parameters in the latent Gumbel distribution. This simple representation allows easy calculation of probabilities of permutations:

$$P(c_1 \succ c_2 \succ c_3 \succ c_4) = \frac{\gamma_{c_1}}{\gamma_{c_1} + \gamma_{c_2} + \gamma_{c_3} + \gamma_{c_4}} \frac{\gamma_{c_2}}{\gamma_{c_2} + \gamma_{c_3} + \gamma_{c_4}} \frac{\gamma_{c_3}}{\gamma_{c_3} + \gamma_{c_4}} \tag{2.5}$$

While Luce's Choice Axiom simplifies tasks such as identifying maximal posterior probability rankings and distributions over rankings, the same axiom makes the model less useful for other tasks. For example, one application of rank completion is to predict an agent's second preference given the agent's first preference. A model that features Luce's Choice Axiom could not adequately perform this task since observing a first choice gives no information about the rest. Indeed, we see in Section 6.4 that PLRUM performs poorly in rank completion compared to other methods.

2.3.2 Normal RUM and its extensions

The Normal RUM extends Thurstone's model for pairwise data, which assumes that the random utility distributions are independent, normally distributed, and have the same variance (Thurstone, 1927; Luce, 1977). The Normal RUM allows for both rank data (instead of pairwise data) and differing variances between the random utility distributions (Azari Soufiani et al., 2012). A further extension called Normal Multitype RUM (Azari Soufiani et al., 2013), classifies rank data into multiple clusters, each of which has its own Normal RUM. This allows for heterogeneous data and a more expressive latent utility model.

Chapter 3

Nonparametric Random Utility Models

As discussed in the introduction, previous RUMs impose distributional assumptions on $\Pr(\vec{u})$ and restrict the joint distribution as a product distribution. I propose nonparametric random utility models (NPRUMs) with a nonparametric and correlated joint utility distribution.

In Section 3.1, I describe and motivate the kernel density estimation procedure that estimates the marginal utility distributions. In Section 3.2, I define the rank position distribution, connect it with NPRUMs, and then use it in Section 3.3 to motivate the theoretical need for NPRUMs.

3.1 Kernel Density Estimation

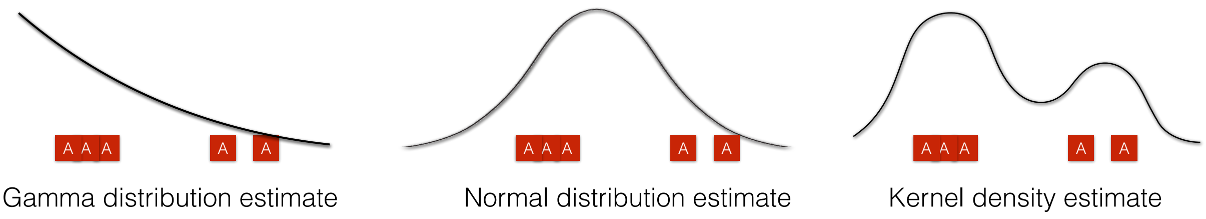


Figure 3.1: Two parametric density estimates (Gamma and Normal) and one non-parametric density estimate (Kernel density estimate).

The fundamental random utility model assumption is that ranks are formed from utilities drawn from a joint distribution. Changing the joint distribution gives rise to different RUMs, such as PLRUMs which assume Gumbel utility distributions, and Normal RUMs which assume Normal utility distributions. A good choice of joint distribution will explain the utilities well, while a poor choice will ignore key features of the utilities. For example, Figure 3.1 illustrates three density estimates fit on five utilities of alternative A. There appears to be evidence of bimodality in the utilities and a strong peak on the left, yet the two parametric densities fail to capture this pattern. Parametric assumptions impose restrictions that may ignore key features of the dataset. Nonparametric density estimation methods are not as susceptible to the same issues.

The NPRUM method uses kernel density estimation (KDE) to estimate the latent utility distribution that drives the rank model (Rosenblatt, 1956; Parzen, 1962). KDE smooths out the observed data by first placing kernels on each of the data points, then summing up the kernels to form the kernel density estimate. This is illustrated in Figure 3.2.

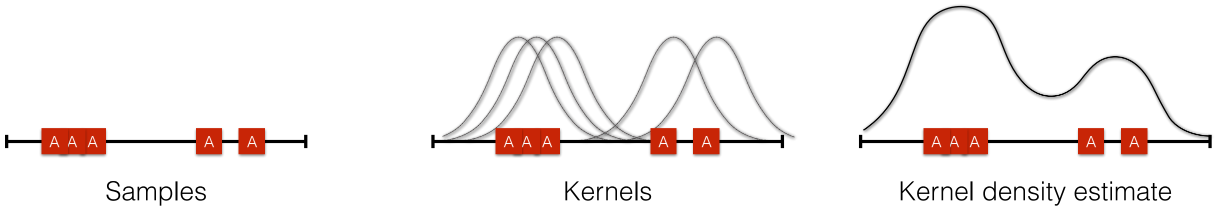


Figure 3.2: The Kernel density estimate of the utility distribution of alternative A is formed by summing up Gaussian kernels placed on the data points.

NPRUM uses Gaussian kernels with a bandwidth parameter $h > 0$. The Gaussian kernels assure that the KDE will be continuous, with smoothness imposed by the bandwidth parameter. The bandwidth controls for overfitting, as a bandwidth high enough will remove spurious data artifacts. A bandwidth that is too high however, will drown out features of the dataset. See Figure 3.3.

Specifically, given a set of sample utilities u_{ij} for a specific alternative j, the marginal utility

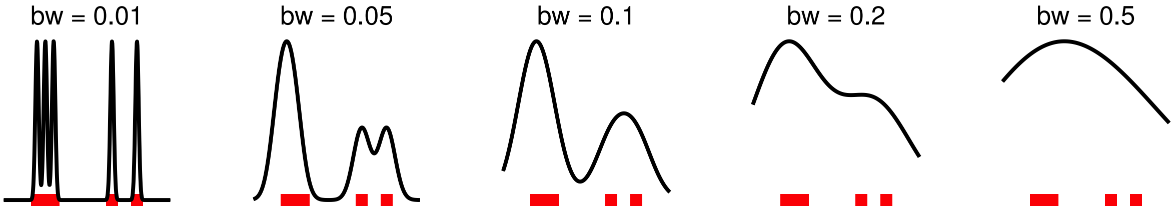


Figure 3.3: Kernel desntiy estimates with various bandwidths.

distribution of alternative j ($\Pr_j$) is estimated as:

$$\Pr_j(x) \propto \begin{cases} 0 & \text{if } x \notin (0,1) \\ \sum_i \phi_h(x - u_{ij}) & \text{if } x \in (0,1) \end{cases} \tag{3.1}$$

where $\phi_h(x) \propto \exp\{-\frac{x^2}{2h^2}\}$, the density function of the kernel $\mathcal{N}(0, h^2)$. I adopt a bounded range for the random utilities. This fixes the effect of h, and prevents the need to consider positive affine transformations of the RUMs. Picking a set of evaluation points for KDEs is simpler when the support is finite. The specific bounded interval $[0, 1]$ is chosen for simplicity.

Since unrestricted KDEs will allocate density outside of the bounded interval, I only consider density within the bounds. I rescale the resulting $\Pr_j(x)$ such that it integrates to 1. To store the function, I evaluate the $\Pr_j(x)$ on a set of evenly-spaced evaluation points $x \in \{0, 1/d, 2/d, .., 1\}$ for a d which indicates the resolution of the nonparametric densities.

For the NPRUM method, KDE is only performed on one-dimensional data. While the output of the algorithm is a m-dimensional nonparametric density, density calculation is not needed here. This is because inference only requires sampling from the m-dimensional joint density, which can be accomplished by selecting a utility vector and adding a perturbation (from the kernel). This helps avoid the need to evaluate the joint density on an m-dimensional grid of d^m evaluation points.

3.2 Rank Position Distribution

Random utility models are useful because they allow us to represent discrete rank data in continuous space. The rank position distribution (RPD) is one way to represent this data in continuous m-dimensional space. It is a density on an m-dimensional hypercube, where each dimension represents an alternative and takes on values between $[0, m]$. If a draw from this hypercube has a value of u_j in the jth dimension, this implies that the rank order of the alternative in the draw is $\lceil u_j \rceil$, where $\lceil \cdot \rceil$ denotes the ceiling function.

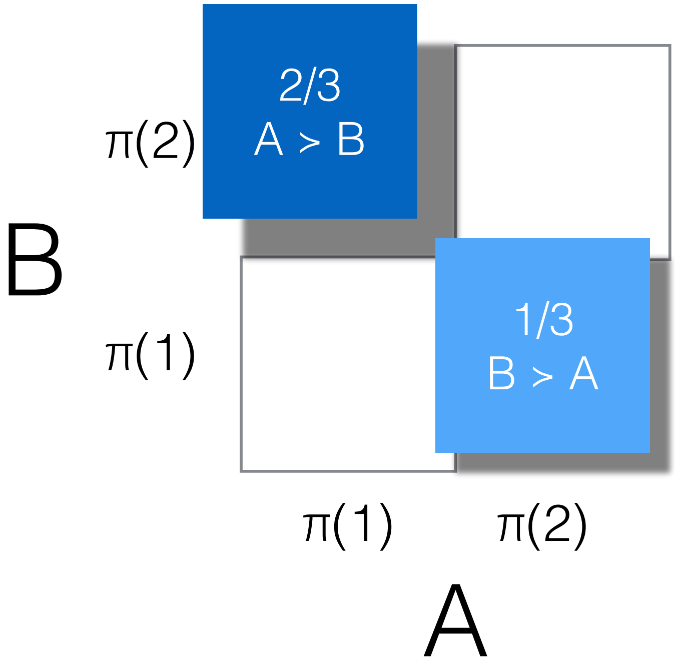


Figure 3.4: As example RPD. The space has $2^2 = 4$ squares, but only $2! = 2$ are associated with actual permutations. Those permutations are associated with densities.

Draws from the RPD can imply m^m possible rank orderings, but only $m!$ of them are valid rank orderings – permutations of the ranks $1, 2, \ldots, m$. Only the positions within the RPD that imply valid permutations have non-negative densities. In those positions, the densities are exactly the empirical probability of observing that permutation. I present an example RPD on two alternatives in Figure 3.4, representing the toy dataset $\{A \succ B\}, \{A \succ B\}, \{B \succ A\}$.

Definition 4 Rank Position Distribution (RPD): *For a distribution on permutations* $\Pr(\vec{\pi})$*, the rank position distribution is defined as the following on an* m*-dimensional hypercube* $\vec{r} \in [0, m]^m$*:*

$$\Pr(\vec{r}) = \begin{cases} \Pr(\vec{\pi}) & \lceil r_j \rceil = \pi(j) \quad \forall j \in \{1, 2, \ldots, m\} \\ 0 & \textit{otherwise} \end{cases}$$

For a numeric example, consider observing the ranks $\{c_1 \succ c_2 \succ c_3\}$, $\{c_1 \succ c_2 \succ c_3\}$, and $\{c_2 \succ c_3 \succ c_1\}$. The rank position distribution is a distribution over the m-dimensional hypercube $\vec{r} \in [0, m]^m$. The region where $\lceil r_1 \rceil = 1, \lceil r_2 \rceil = 2, \lceil r_3 \rceil = 3$ corresponds to the permutation $\{c_1 \succ c_2 \succ c_3\}$. That region has a density of 2/3, the empirical probability of observing the permutation. The region where $\lceil r_2 \rceil = 1, \lceil r_3 \rceil = 2, \lceil r_1 \rceil = 3$ corresponds to the permutation $\{c_2 \succ c_3 \succ c_1\}$. That region has a density of 1/3, the empirical probability of observing the permutation. The remainder of the hypercube is allocated no density. We confirm that the density in the hypercube integrates to 1.

In this way, the RPD is the m-dimensional histogram density that encodes the distribution for all permutations π given in the dataset.

3.3 Motivation for NPRUM

Draws from the RPD imply rankings. The RPD has the property that each of these implied rankings occurs with a probability exactly equal to the empirical probability of observing that ranking in the original dataset.

For small n, $\Pr(\vec{r})$ is an approximation for the true RPD. The greater the n, the better the estimation of the true RPD. However, since $\Pr(\vec{r})$ has on the order of $O(m!)$ probabilities to estimate, $n >> m!$ data points would be needed in order to get a good statistical approximation. This is not feasible for modest values of m (e.g. $m > 10$).

NPRUM can be used to approximate $\Pr(\vec{r})$ instead of relying on more samples. NPRUM also estimates a joint utility density on a hypercube (NPRUM uses $[0,1]^m$ instead of RPD's $[0,m]^m$), but imposes a continuous, smooth density within the hypercube unlike the multidimensional "checkerboard" seen in RPD.

The smoothness constraints on NPRUM achieved through KDE allow permutations in the data to "borrow" information from other permutations nearby in the hypercube. An MLE achieved via Monte Carlo EM is used to estimate this nonparametric random utility distribution.

Chapter 4

An Estimation Algorithm

As described in Section 2, computation of the likelihood function involves a multidimensional integral, which means that direct optimization of the likelihood function is intractable. Thus, I adopt a Monte Carlo expectation-maximization (EM) algorithm similar to the one presented by Azari et al. (2012). The EM algorithm is particularly well-suited for applications involving a latent variable space (Dempster et al., 1977). The algorithm iteratively determines the maximum likelihood estimation of the joint distribution $\Pr^*(\vec{u})$.

The algorithm is composed of iterations of an E-step and an M-step. Given $\Pr^t(\vec{u})$ from the previous iteration t, the following are performed on each iteration $t+1$:

$$\text{E-step}: \quad Q(\Pr, \Pr^t) = E_{\vec{u}} \left\{ \log \prod_{i=1}^{n} \Pr(\vec{u}_i, \pi_i) \mid D, \Pr^t \right\} \tag{4.1}$$

$$\text{M-step}: \quad \Pr^{t+1} \in \arg\max_{\Pr} Q(\Pr, \Pr^t) \tag{4.2}$$

The algorithm starts with an initialization of the joint density to $\Pr^0(\vec{u})$ to a uniform distribution. The algorithm alternates between the E-step and the M-step. If successful, it converges to the maximum likelihood estimate $\Pr^*(\vec{u})$.

4.1 E-step

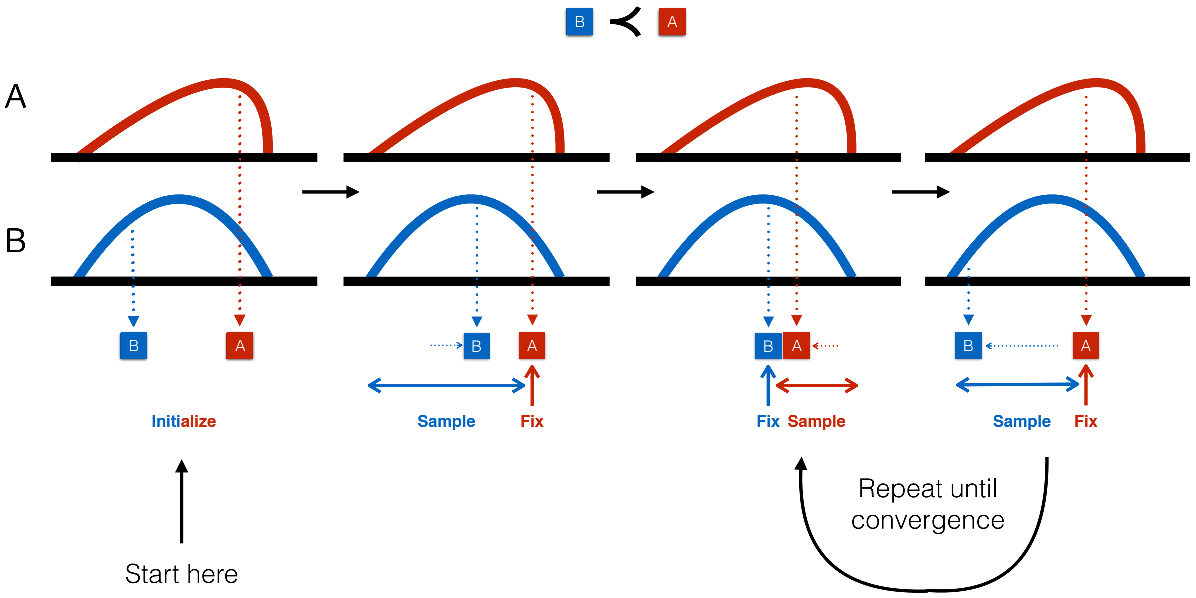


Figure 4.1: In this example, the observed rank order $\{A \succ B\}$ and the latent utility distributions from the M-step are used to sample utilities u_A and u_B. The steps alternate between making draws for A and B, always preserving the relation $u_A > u_B$. This continues until convergence. The resulting utilities are passed to the M-step.

For ease of presentation, I will assume in this chapter that every data point is contributed by an agent. The E-step samples a vector of utilities for each agent that is conditional on their observed rank preference. This process is illustrated in Figure 4.1.

Since drawing directly from the joint density is intractable, I rely on variational Monte-Carlo methods. Specifically, I adopt a Gibbs sampler to sample each utility sequentially.

Within the Gibbs sampler, slice sampling (Neal, 2003) is used to sample latent utilities. Slice sampling is a Monte Carlo algorithm for sampling draws from any distribution. It starts by selecting a point on a random vertical slice of the distribution, then samples a point from the corresponding horizontal slice of the distribution. This repeats until a sufficient number of samples are made (Neal, 2003). Tarlow et al. (2012) argues slice sampling is well suited for sampling latent variables in

Monte Carlo EM. This appropriateness is partly due to its ability to sample from a wide range of distributions while not requiring any tuning parameters. The method utilizes Neal's implementation of his slice sampler (2008). I leave a more detailed explanation of this algorithm and implementation to Neal (2003).

To sample a value for $u_{\pi(j)}$, I use the slice sampler to draw from the truncated marginal utility distribution of that alternative. The truncation depends on the type of rank data. Let $\pi(j)$ denote the alternative in the jth position. $u_{\pi(j)}$ denotes its utility. The full ranking type of data imposes the following restriction on utilities:

$$u_{\pi(j)} \in \begin{cases} \big(u_{\pi(2)}, 1\big) & j = 1 \\ \big(u_{\pi(j+1)}, u_{\pi(j-1)}\big) & 1 < j < m \\ \big(0, u_{\pi(m-1)}\big) & j = m \end{cases} \tag{4.3}$$

In the top ranking type of data, only alternatives in $\mathcal{C}'$ are ranked (let $|\mathcal{C}'| = m' < m$). These alternatives are preferred over those left unranked. The top ranking type imposes the following restriction on utilities:

$$u_{\pi(j)} \in \begin{cases} \big(u_{\pi(2)}, 1\big) & j = 1 \\ \big(u_{\pi(j+1)}, u_{\pi(j-1)}\big) & 1 < j < m' \\ \big(\max_{c_j \in \mathcal{C}'^c} u_{c_j}, u_{\pi(m'-1)}\big) & j = m' \\ \big(0, u_{\pi(m')}\big) & m' < j \leq m \end{cases} \tag{4.4}$$

In the sub ranking type of data, only alternatives in $\mathcal{C}'$ are ranked (let $|\mathcal{C}'| = m' < m$). Unlike the top ranking type, the data observation provides no information is available about alternatives that are not in $\mathcal{C}'$. Because of this, the E-step does not sample utilities for the unranked alternatives.

The sub ranking type imposes the following restriction on utilities:

$$u_{\pi(j)} \in \begin{cases} \big(u_{\pi(2)}, 1\big) & j = 1 \\ \big(u_{\pi(j+1)}, u_{\pi(j-1)}\big) & 1 < j < m' \\ \big(0, u_{\pi(m'-1)}\big) & j = m' \\ \text{NA} & m' < j \leq m \end{cases} \tag{4.5}$$

For all types of datasets, this sampling is repeated until convergence for each row of rank data. Convergence can be determined by a convergence diagnostic such as the Gelman and Rubin diagnostic (1992), which uses multiple chains of samples to determines whether the chains are sufficiently similar. It accomplishes this by comparing within-chain variance and between-chain variance.

The result of the E-step is a set of utilities that is passed to the M-step.

4.2 M-step

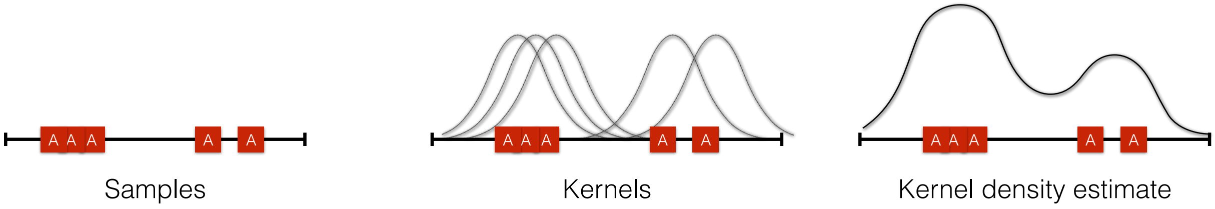


Figure 4.2: In this example, five values of u_a are returned from the E-step. No distributional assumptions are made in NPRUM, and so the random utility distribution for alternative A is estimated using Gaussian kernels with bandwidth h. The resulting kernel density estimate is passed back to the E-step. (copy of Figure 3.2)

In the M-step (illustrated in Figure 4.2), the joint utility distribution is inferred from the utility samples from the E-step. However, KDE on many dimensions is intractable because the number of evaluation points grows exponentially with the m. Therefore, I adopt a variational method and estimate the joint distribution as a product distribution $\Pr(\vec{u}) = \prod_j \Pr_j(\vec{u})$.

The algorithm estimates each marginal density via kernel density estimation. More information about the kernel density estimation is presented in Section 3.1. The result is a set of utility distributions that is passed back to the E-step.

Even though the M-step only uses only the marginal distributions for inference, the final output of the Monte Carlo EM algorithm retains the correlation structure, because it consists of samples of utilities returned from the E-step.

4.3 Algorithm

The output of the Monte Carlo EM algorithm provides the means to construct the joint distribution over utilities using KDE. This joint distribution is easy to sample from, because the method can draw a random $\vec{u}_i$ and a corresponding value from the kernel associated with the point. See Algorithm 1 for a summary.

Algorithm 1 Monte Carlo EM algorithm for NPRUM

```
set $t \leftarrow 0$
set $\Pr^0 \leftarrow$ Uniform
repeat
   (Variational MCMC E-step)
   for all rank data $i$ do
      repeat
         for all alternatives $j$ do
            $u_{ij}^{t+1} \leftarrow$ slice sample from $\Pr_j^t(u_{ij}|u_{i(-j)}, \pi_i)$
         end for
      until Gibbs convergence
   end for
   (Variational M-step)
   for all alternatives j do
      (KDE estimation of $\Pr_j$)
      $\Pr'_j(x) \leftarrow \mathbb{I}_{x\in(0,1)} \sum_i \exp\left\{\frac{-(x-u_{ij}^{t+1})^2}{2h^2}\right\}$
      $\Pr_j^{t+1}(x) \leftarrow \Pr'_j(x) / \int_0^1 \Pr'_j(x)dx$
   end for
   $t \leftarrow t+1$
until Convergence of all $\Pr_j^t$
return Joint KDE on the $n \times m$ matrix of latent $u_{ij}$
```

Chapter 5

Datasets

The purpose of this chapter is to introduce the three datasets used in this thesis, and to delineate the features that I want the rank aggregation model to express. While reviewing properties of the datasets, I will introduce the metrics used to evaluate the various rank data methods. The Sushi dataset receives more attention in this chapter because it is the focus of all four metrics in this thesis.

The Election and Nascar datasets are used to demonstrate NPRUM's effectiveness on the two types of partial rank data: top and sub. Election datasets are common examples of top rank data, since agents may choose and rank their top few preferences out of a list of candidates. Competition datasets (such as Nascar) are common examples of sub rank data, where only a subset of alternatives are compared at a time.

5.1 Sushi dataset

In this section, I introduce and perform exploratory analysis on one of Toshihiri Kamishima's Sushi Preference datasets (Kamishima, 2003). The 10 types of sushi in the dataset are shrimp, sea eel, tuna, squid, sea urchin, salmoe roe, egg, fatty tuna, tuna roll, and cucumber roll.

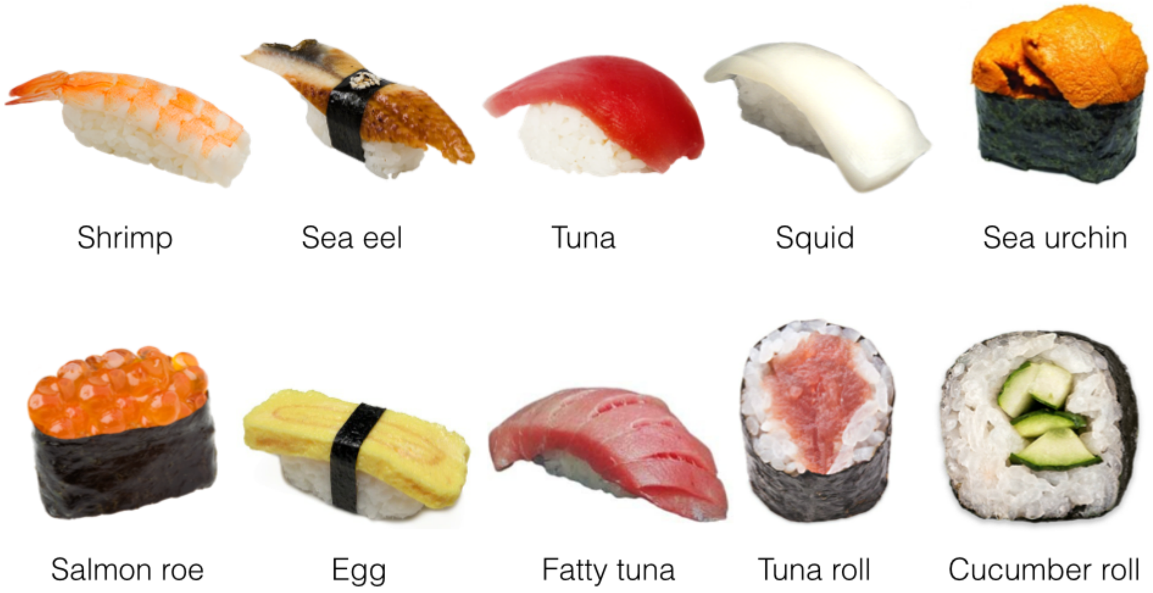


Figure 5.1: The ten sushi variants in the Kamishima sushi dataset.

The data was collected by Toshihiro Kamishima and his colleagues at the National Institute of Advanced Industrial Science and Technology in Japan. Kamishima et al. surveyed 5000 individuals living in Japan about their preferences between the ten sushi variants and asked the individuals to rank the sushi in the order of their preference. The dataset and the corresponding paper (Kamishima, 2003) have been influential in ranking research and have been used to evaluate new methods in collaborative filtering (Lee et al., 2010; Chen and Cheng, 2008) and rank aggregation problems (Lu and Boutilier, 2011; Bonilla et al., 2010).

5.2 Sushi Rank Positions

One natural query we can explore is the distribution of the agents' first, second, third, and last sushi preferences on sushi.

Fatty tuna is a common favorite and cucumber roll is a common least-favorite. Interestingly, there is an element of controversiality about sea urchin: 15% rank sea urchin their favorite, while 20%

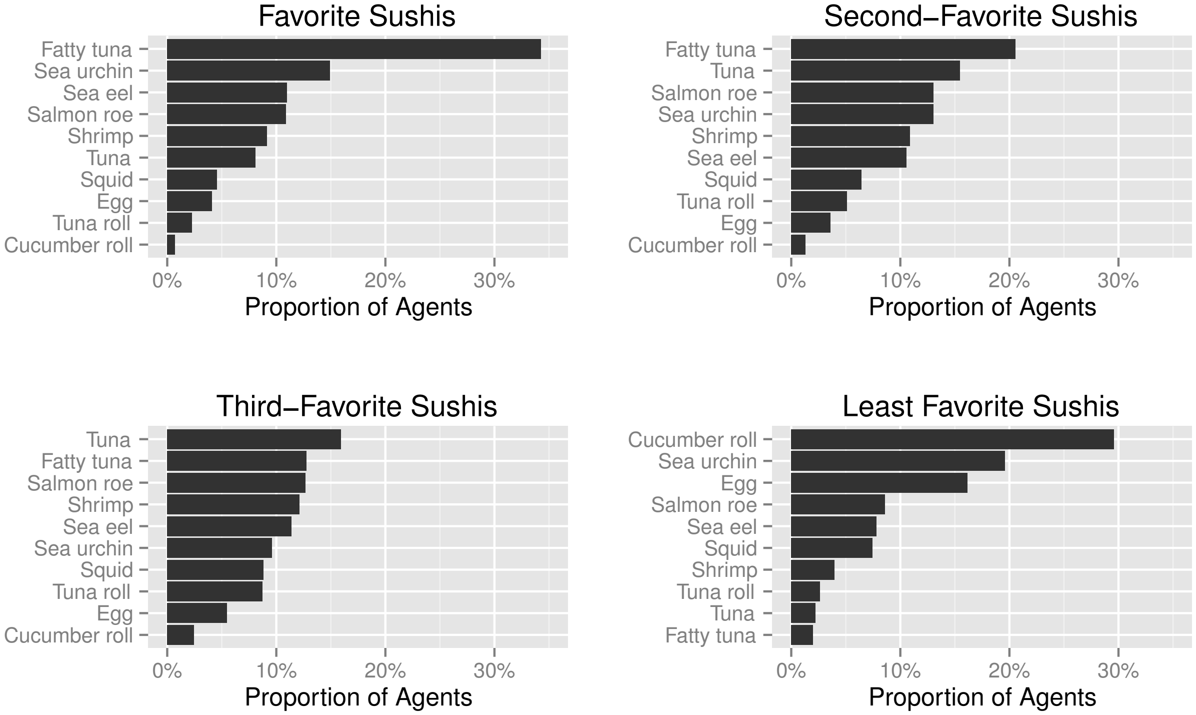


Figure 5.2: Agents' favorite, second-favorite, third-favorite, and least favorite sushi variants.

rank sea urchin their least favorite. A good model will capture this disagreement about sea urchin (i.e. this diversity in data).

Figure 5.3 plots the rank distributions for each sushi variant – these are the marginals of the rank position distribution described in Definition 4. We can verify that fatty tuna tends to appear early in agents' ranks, cucumber roll tends to appear later, and sea urchin is indeed controversial, with a bimodal marginal rank position distribution.

Later on in Section 6.2, I will study different methods for modeling rank data. The total variation distance (TVD), denoted $\delta(P, Q)$, will be used to measure the quality of the estimation, where

$$\delta(P, Q) = \frac{1}{2}||P - Q||_1, \tag{5.1}$$

and P is the actual and Q is the estimated marginal rank position distributions. The evaluation

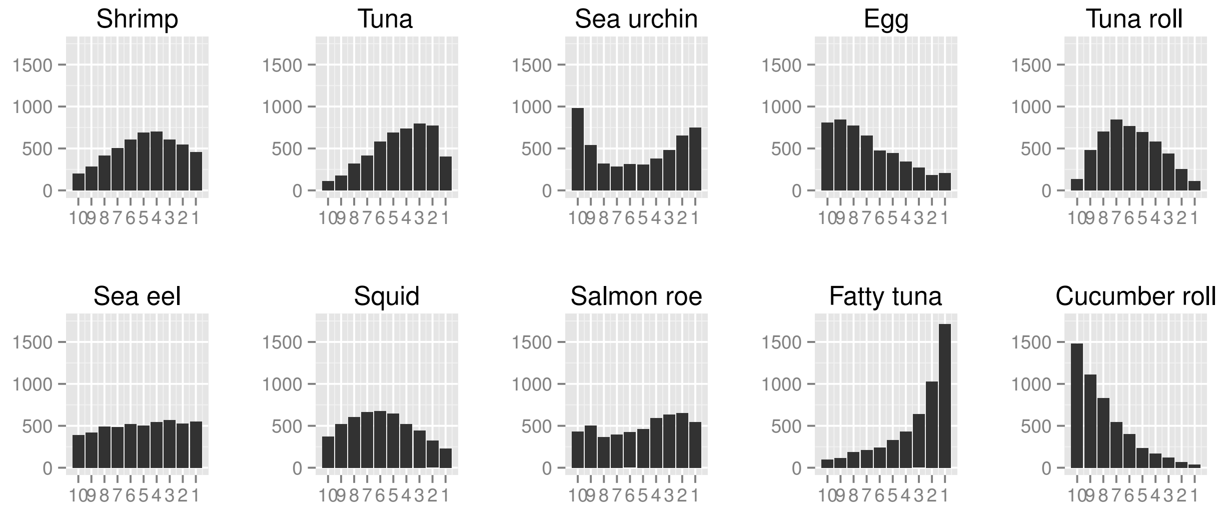


Figure 5.3: Marginal rank position distributions of all 10 types of sushi.

metric will be the TVD averaged over all alternatives, where Q is estimated from a training set and P is the empirical distribution of the test set.

5.3 Sushi Pairwise Preferences

Another natural question about rank data is the nature of the pairwise preferences, since (1) pairwise preferences are the smallest unit of rank data, (2) pairwise preferences are often considered when talking about desirable properties of ranking algorithms (e.g. Condorcet), and (3) looking at pairwise preferences instead of full rank preferences allows us to estimate a polynomial $O(m^2)$ number of probabilities instead of a factorial number.

I construct a pairwise preference matrix (Figure 5.4) to visualize the answer to the question *what proportion of agents prefer alternative i (row) over alternative j (column)?*. From the figure we see that every entry in fatty tuna's row is greater than 70%. This means that in any pairwise comparison between fatty tuna and any other sushi variant, at least 70% of the agents would prefer fatty tuna. Fatty tuna would hence be declared the Condorcet winner as it dominates every other sushi in pairwise preferences (1785). Extending that concept, the full Condorcet ranking is the

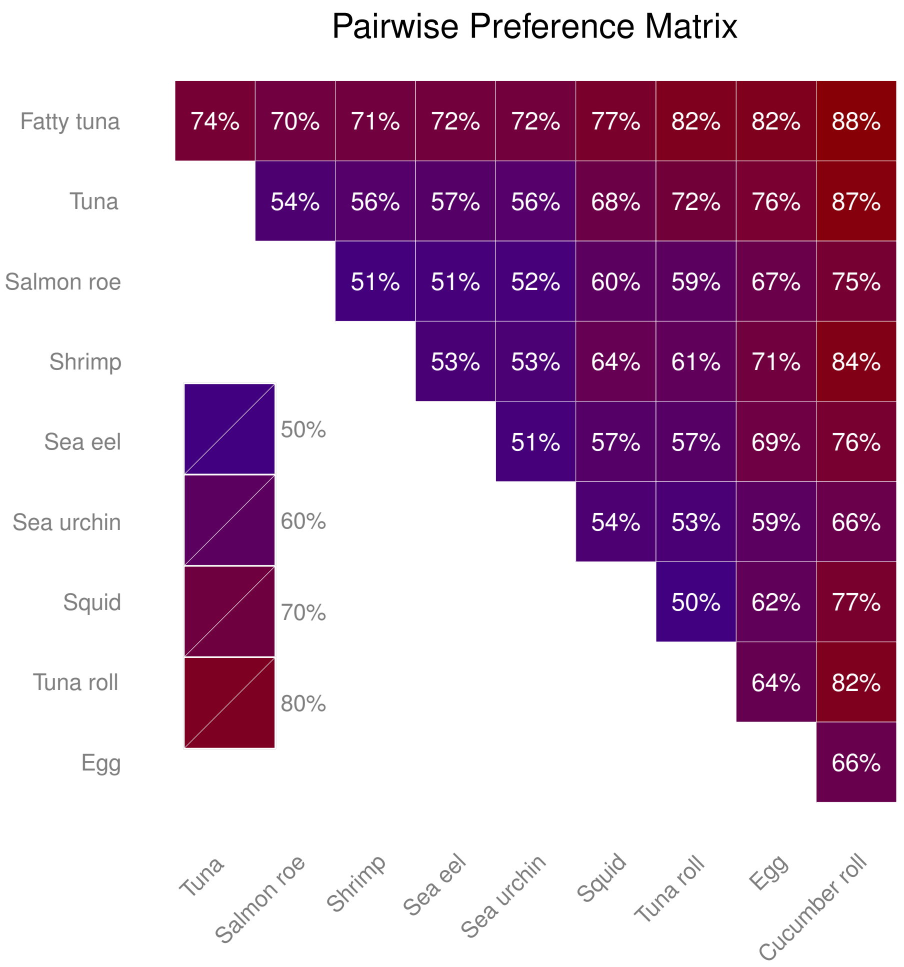


Figure 5.4: The pairwise preference matrix for the sushi dataset (only the upper right triangle is shown for visualization clarity). Each entry is the probability that the sushi variant in the row is preferred over the sushi variant in the column.

following:

$$\begin{aligned}&\text{fatty tuna} \succ \text{tuna} \succ \text{salmon roe} \succ \text{shrimp} \succ \text{sea eel} \succ \ldots \\ &\ldots \text{sea urchin} \succ \text{squid} \succ \text{tuna roll} \succ \text{egg} \succ \text{cucumber roll}\end{aligned} \tag{5.2}$$

This is a Condorcet ranking for the following reason: in any pairwise match-up between two sushi variants, a majority of the agents will prefer the one earlier in the preference ranking over the one later in the preference ranking. I note that the preference between squid and tuna roll is very close, with 2503 agents preferring squid over tuna roll and 2497 agents preferring tuna roll over squid. The sushi variants in Figure 5.4 are ordered according to their Condorcet ranking.

In Section 6.3, I will use TVD ($\delta(P, Q)$) to assess the quality of the prediction, where P is the true preference matrix for a hold-out test set of the rank data, and Q is the predicted preference matrix for the model trained on the training set. This measures the ability of a model to predict pairwise preferences in the dataset.

5.4 Sushi Recommendations

An interesting application of a ranking model is that given a sushi that an agent likes, recommendations can be made about other sushi variants the agent might like.

This requires that for alternatives $c_1, c_2, \ldots, c_{10}$, observing that the agent ranked c_1 first will give a distribution of second preferences (on alternatives $c_3, \ldots, c_{10}$) that is different from that when ranking c_2 first. A good model can capture these patterns and make more appropriate recommendations for an agent given the agent's top choice.

In Figure 5.5, I use four pairs of sushi variants to illustrate how a first preference can predict a second preference. The four pairs are sea urchin / egg, fatty tuna / squid, salmon roe / tuna roll, and shrimp / tuna.

Those who ranked squid first are much more likely to rank sea eel, shrimp, egg, and cucumber roll second, than those who ranked fatty tuna first. This is an intuitive result because squid, sea eel, shrimp, egg, and cucumber roll are the non-raw options out of the 10 alternatives. Those who ranked tuna first are much more likely to rank fatty tuna and tuna roll second, than those who ranked shrimp first. This is also an intuitive result.

Using a simulated Fisher Exact Test and drawing 100,000 random permutations, the differences in

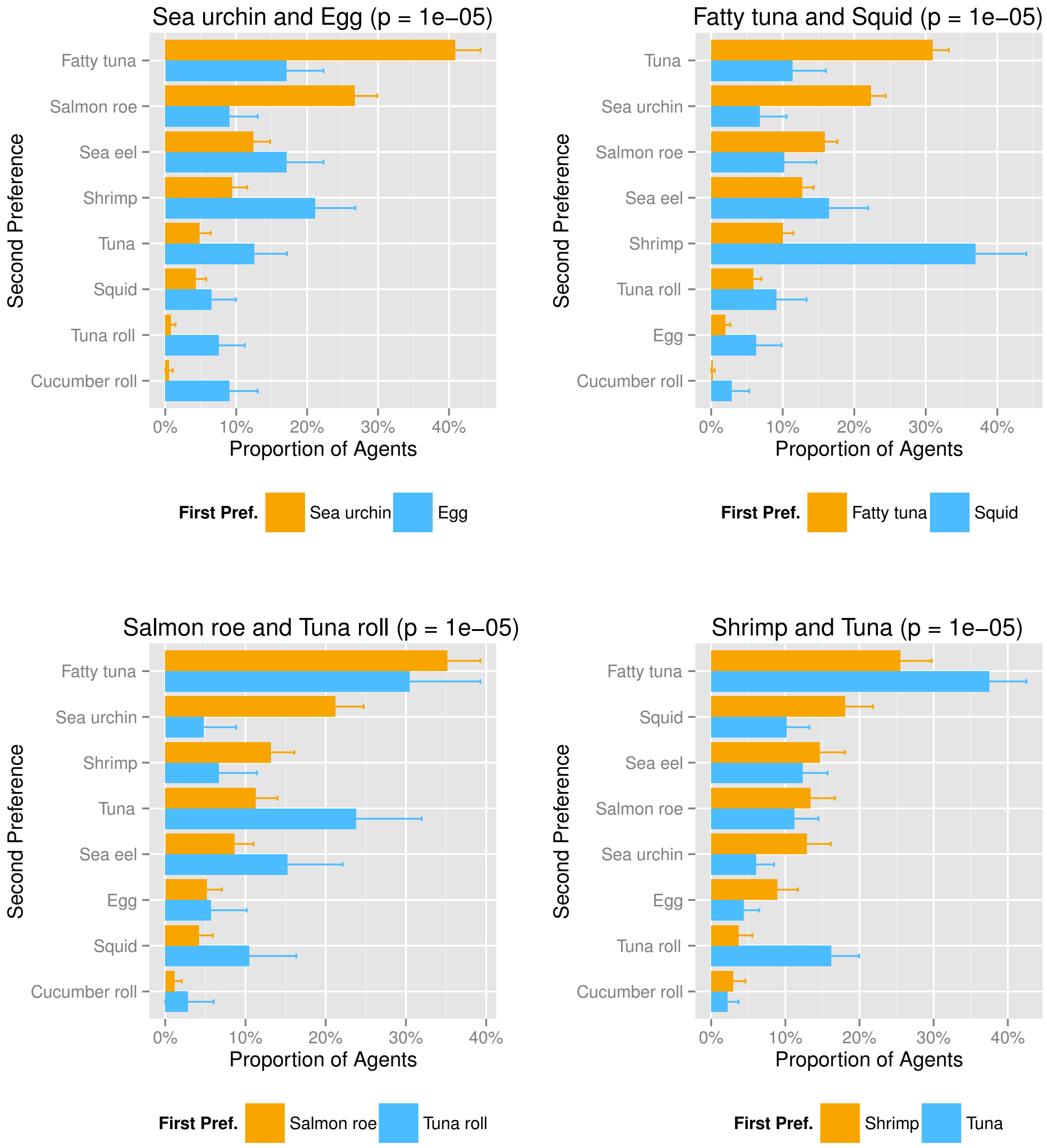
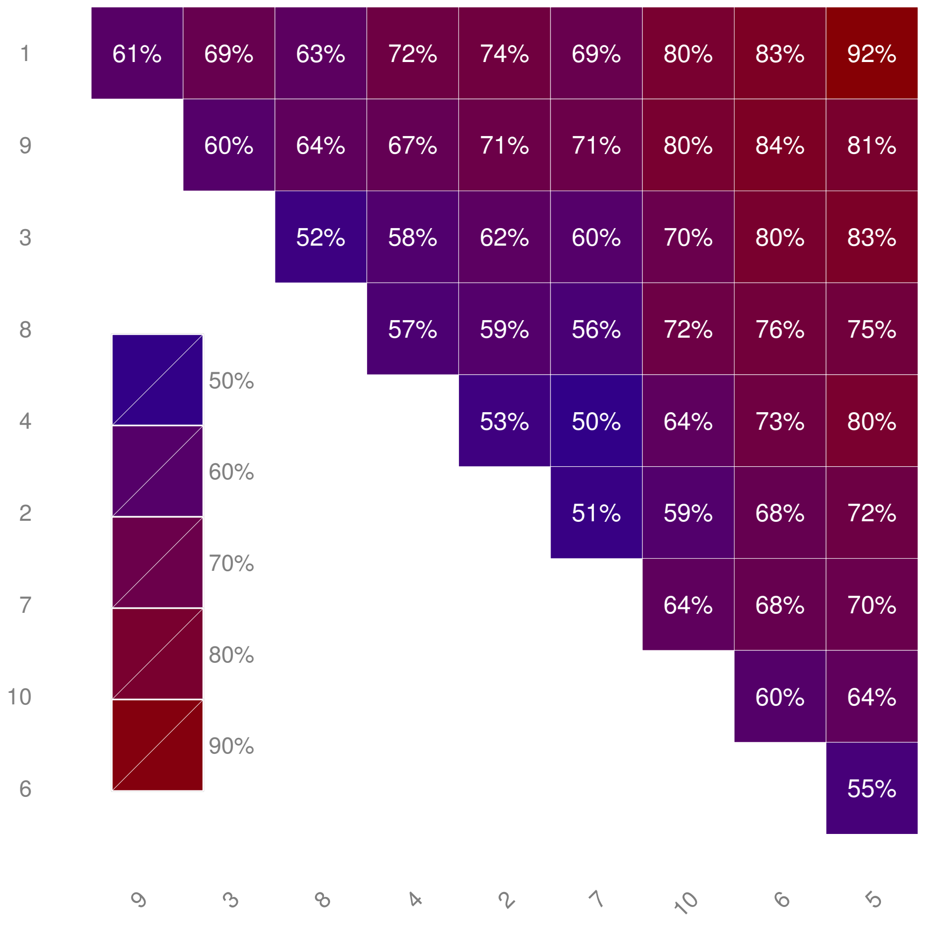


Figure 5.5: Each graph compares the distribution of second-preference sushi between two types of agents: those who listed the first sushi of a pair as first-preference, and those who listed the second sushi of a pair. The *p*-values are from a (simulated) Fisher Exact Test with 100,000 simulated draws. Error bands show 95% confidence intervals for proportions.

the distributions in these four examples is highly significant (with $\alpha = 0.05$). In fact, 40 of all 45 pairs of sushi returns a significant result – thus, knowing the agent's first-preference sushi informs us of the second preference.

I focus on the case of predicting the second preference of an agent given the agent's first preference, and measure accuracy using TVD. I calculate the TVD on all possible first preferences, and then weigh the TVDs by the count of each first occurrence. I present these results in Section 6.4.

5.5 Election Dataset



	9	3	8	4	2	7	10	6	5
1	61%	69%	63%	72%	74%	69%	80%	83%	92%
9		60%	64%	67%	71%	71%	80%	84%	81%
3			52%	58%	62%	60%	70%	80%	83%
8				57%	59%	56%	72%	76%	75%
4					53%	50%	64%	73%	80%
2						51%	59%	68%	72%
7							64%	68%	70%
10								60%	64%
6									55%

Figure 5.6: The pairwise preference matrix for the election dataset.

This dataset is one of 87 ranked-ballot election datasets collected by Nicolaus Tideman (2006) and contains *top rankings* of 10 candidates from 380 voters from a British organization. The objective was to select three candidates onto a committee. Voters were allowed to select any number of candidates to rank – it would be implied that any candidate that is ranked is preferred over any candidate that is not ranked. The candidates are anonymized.

Figure 5.6 illustrates the pairwise probabilities in this dataset. From the pairwise matrix visualization, we can see that candidates 1, 3, 9 would win this election under the Condorcet rule, since a majority of the voters would prefer those three candidates over any of the other seven candidates in pairwise elections. Interestingly, there is no Condorcet ranking in this dataset because of the presence of Condorcet's paradox (1785)– there are three candidates where the collective preference is cyclical: 52.8% of voters prefer candidate 4 over candidate 2, 51.2% of voters prefer candidate 2 over candidate 7, and 50.2% of voters prefer candidate 7 over candidate 4.

Table 5.1 illustrates the common ballots in this election. We see that the top three most common ballots agree with selecting candidates 1, 3, and 9 as winners. However, one must be careful as these top three ballots represent less than 10% of the total ballots selected. Moreover, looking at the most common ballots is most likely an unreasonable method to learn about the choices of voters because of sparsity – there are total of 6235301 possible ballots in a top ranking election of 10 candidates, yet only 252 of these ballots are represented here.

To calculate the 6235301 number, we consider all possible top ranking ballots. For an election of m candidates, the total number of ballots is $\sum_{k=0}^{m-1} m!/k!$, where k is the total number of candidates ranked. I note that in top ranking, ranking $k = m - 1$ candidates is equivalent to ranking $k = m$ candidates. Also, I include the case where no candidates are ranked. I further note that $\sum_{k=1}^{m} m!/k! = \lfloor m!(e-1) \rfloor$, so the total number of top ranking ballots is $O(m!)$ (OEIS, 2011).

As mentioned in Section 3.2, looking at the distribution of permutations or rank position distributions directly poses an issue because of sparsity. However, the distribution over ranks can still be estimated well using random utility models because of the introduction of a latent utility space. With a latent utility space, the model can "smooth" over possible rankings and let data "borrow" information from each other. For example, the top three rankings ($\{3 \succ 1 \succ$ all others$\}$, $\{1 \succ 3 \succ$ all others$\}$, and $\{1 \succ 3 \succ 9 \succ$ all others$\}$) are very similar and a good model should incorporate this information and deal with sparsity.

Table 5.1: The thirteen most common ballots (all ballots that at least 4 voters submitted).

Count	**Ranking**
13	$3 \succ 1 \succ$ all others
9	$1 \succ 3 \succ$ all others
9	$1 \succ 3 \succ 9 \succ$ all others
6	$2 \succ 8 \succ$ all others
5	$1 \succ 5 \succ 8 \succ$ all others
4	$1 \succ 3 \succ 9 \succ 7 \succ$ all others
4	$1 \succ 8 \succ 4 \succ$ all others
4	$1 \succ 9 \succ 3 \succ$ all others
4	$3 \succ 6 \succ 8 \succ$ all others
4	$4 \succ$ all others
4	$7 \succ 9 \succ 3 \succ$ all others
4	$9 \succ$ all others
4	$9 \succ 8 \succ 7 \succ$ all others

5.6 Nascar Dataset

This dataset contains race data from the 2002 NASCAR Winston Cup Series (Hunter, 2004). The season contained 36 races involving 83 drivers. Each race involved either 42 or 43 drivers.

The Winston Cup champion in 2002 was Tony Stewart of Indiana, who achieved the title for gaining the most points under the Championship points system (Wikipedia, 2014a). The Championship points system is each driver's score under a modified positional voting rule. In a positional voting system, each driver is given points according to their rank position in each race. The final ranking is made by ordering the alternatives by total number of accumulated points, including additional bonus points for leading laps or winning a race (Wikipedia, 2014b).

While a positional voting rule is adequate for giving a final rank order, it does not reveal potentially useful information, such as the probability of a driver beating another in a pairwise matchup (illustrated in Figure 5.7), or how consistently a driver performs well. This is when statistical rank aggregation methods, such as RUMs, become useful. I will compare the ability of NPRUMs to predict pairwise matchups and fit the data. The experiments will cover only a subset of the Nascar dataset (only those drivers who have participated in 20 to 30 races), since some baseline RUM metrics do not converge on the full dataset.

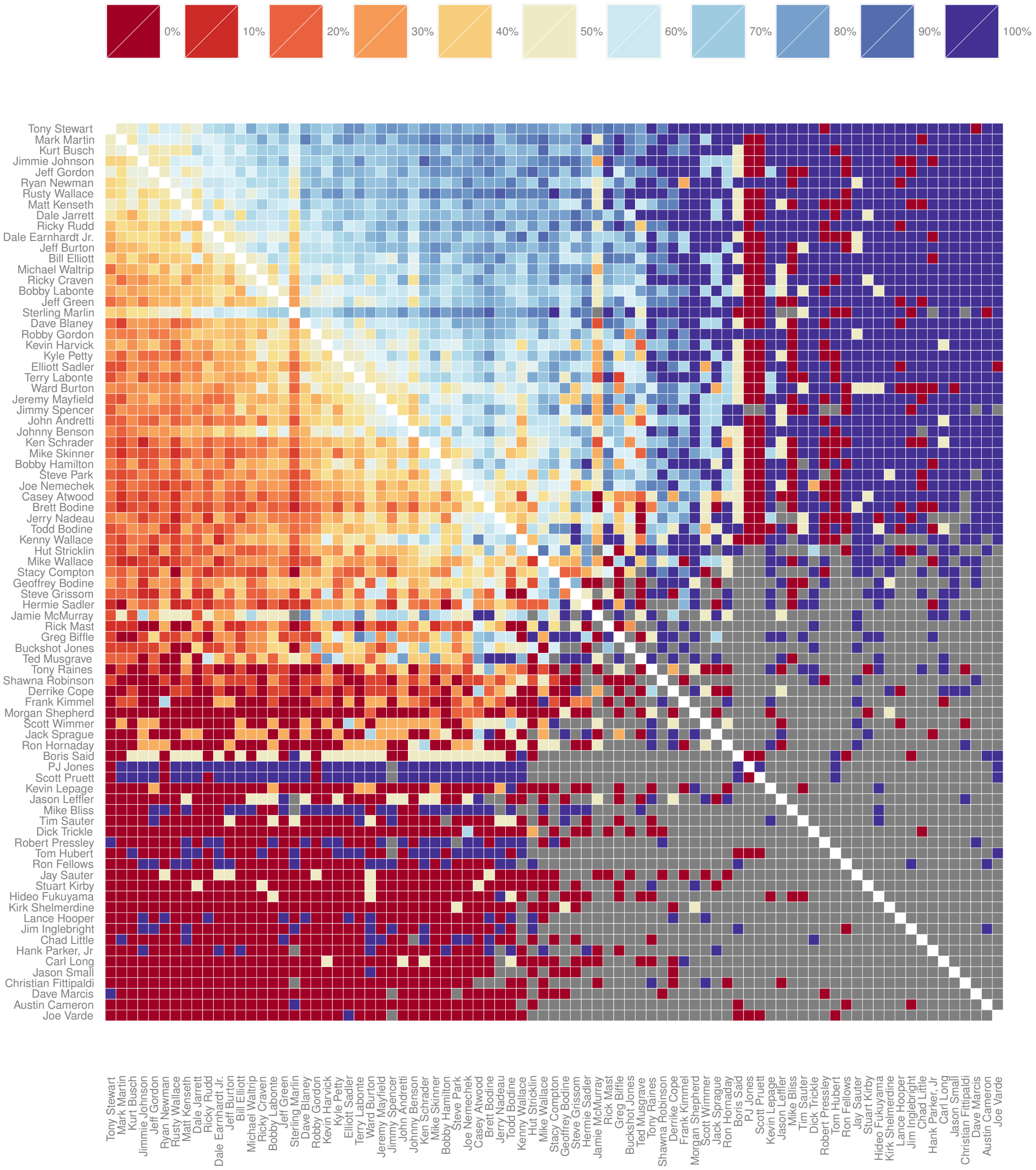


Figure 5.7: Pairwise win matrix of Nascar dataset. Drivers are ordered by final ranking in the 2002 Nascar Tournament, which was decided by a modified positional voting rule. Drivers ranked in the top half have all participated in 20 or more races, while almost all drivers in the bottom half have participated in 10 races or fewer. Gray squares indicate that the pair of racers have never faced each other in the same match.

Chapter 6

Experimental Results

I evaluate various RUM methods on the datasets in Table 6.1. Via experiments, I compare the ability of RUMs to:

(i) Capture heterogeneity and correlation of utilities

(ii) Predict RPDs via smoothing

(iii) Predict out-of-sample data and pairwise matrices

(iv) Complete the rank order given partial rank information

Table 6.1: Datasets used for evaluation. † denotes a subset of the full data

	Rank Data Type	m	n
Election (Tideman, 2006)	Top Partial	10	380
Nascar (Hunter, 2004)	Sub Partial	7†	36
Sushi (Kamishima, 2003)	Full	10	5000

All code and data, with the exception of the sushi dataset,[1] is available via the R package `StatRank` (Chen and Azari Soufiani, 2013).

[1] I was not granted permission to redistribute the sushi dataset due to the terms that were shown to the respondents. The sushi dataset is available at http://www.kamishima.net/sushi/.

6.1 Capturing Heterogeneity and Correlation of Utilities

6.1.1 Heterogeneity

The heterogeneity of the utility distribution for an alternative captures information that reveals properties about the rank dataset. In the case of user preferences or voting in elections, heterogeneity represents diversity of opinion. In other rank data applications, heterogeneity may represent different factors affecting performance. To understand this heterogeneity, I fit various RUMs to 5000 data points of the sushi dataset and plot the estimated marginal utility distributions for five alternatives in Figures 6.1-6.5.

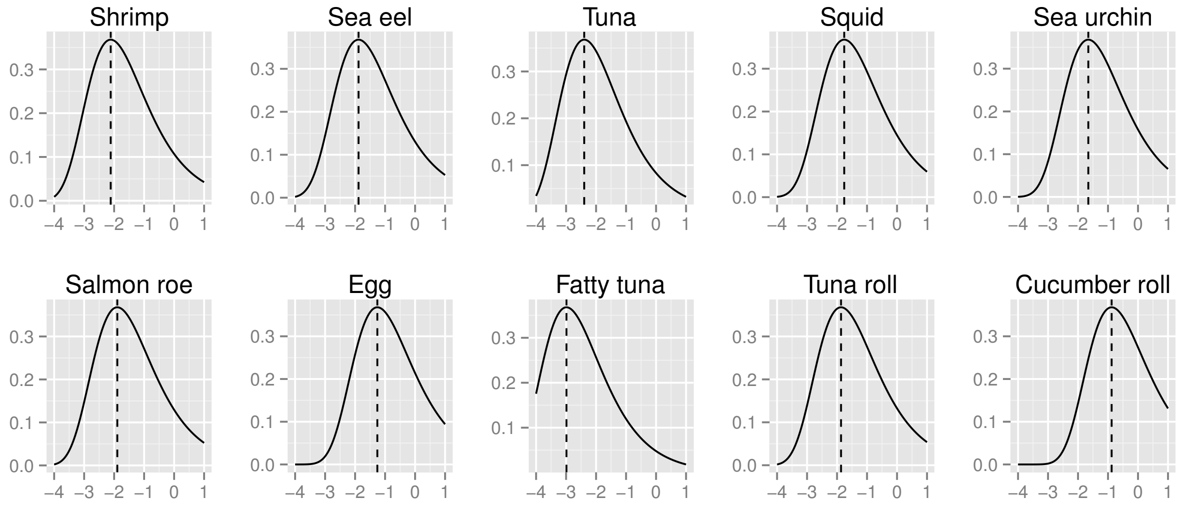


Figure 6.1: Estimated Gumbel densities on utilities in the Plackett-Luce RUM.

Generally, more expressive RUMs can encode a wider range of features. With more parameters, a model can go beyond other models such as the Plackett-Luce RUM by capturing more than just the location parameter of a utility distribution. Some models can also capture multimodality and differing variances across alternatives. The most notable example of features in a utility distribution is that of the sea urchin sushi in Figures 6.3 and 6.4. I explore these features more in Section 6.1.2.

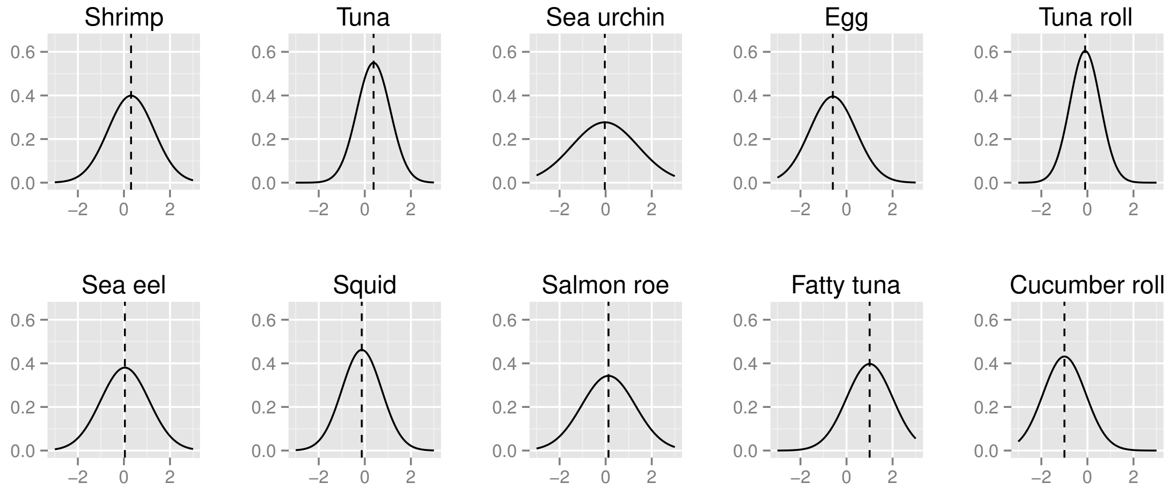


Figure 6.2: Estimated densities on utilities in the Normal RUM.

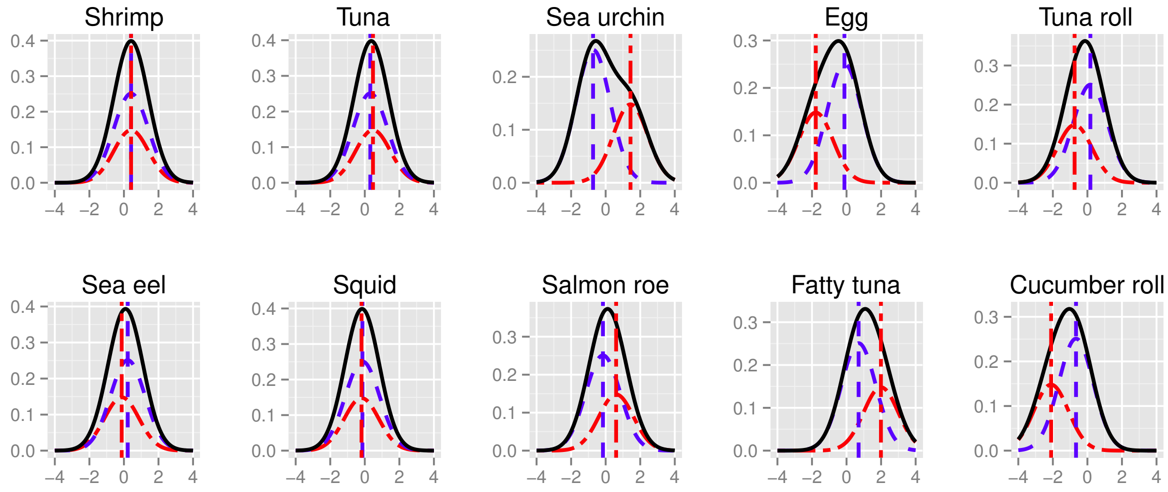


Figure 6.3: Estimated densities on utilities in the Multitype Normal RUM.

Comparing the empirical RPD and NPRUM within Figures 6.4 and 6.5, we see that the estimated utility distributions are expressive enough to share common features with the empirical RPDs. The utility distribution becomes more expressive with more parameters, and is most expressive when fit using NPRUM.

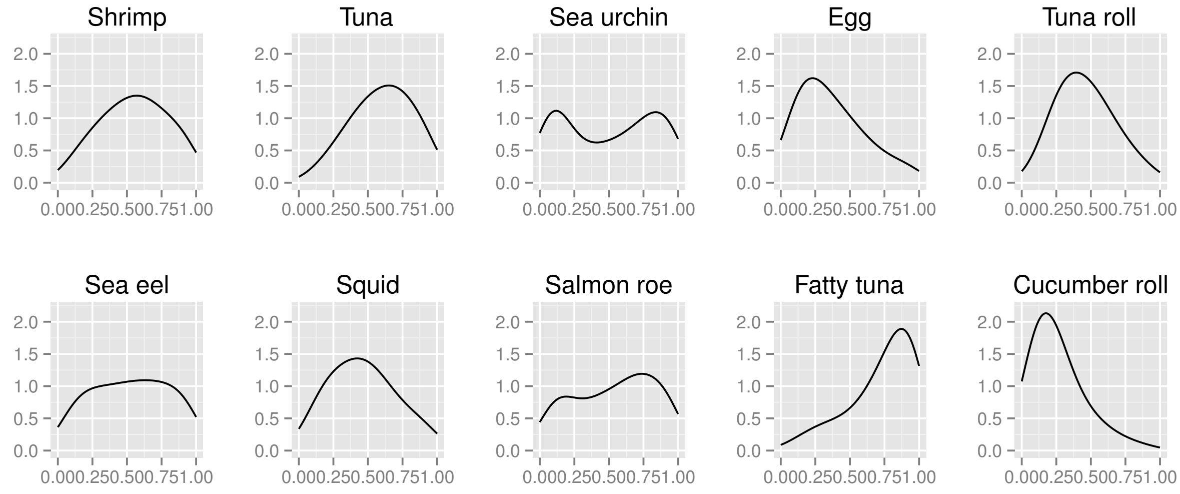


Figure 6.4: Estimated densities on utilities in the Nonparametric RUM ($h = 0.11$). Bandwidth chosen through cross-validation (see Section 6.3).

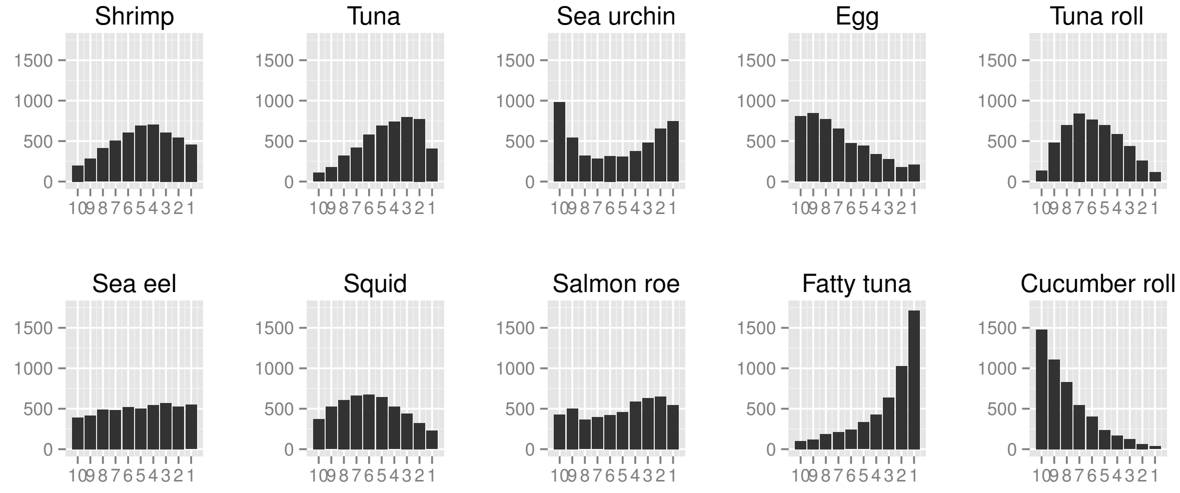


Figure 6.5: Marginal rank position distributions of the sushi dataset (copy of Figure 5.3)

6.1.2 Utility Correlation

A key benefit of NPRUM over existing RUM methods is NPRUM's ability to capture the correlation structure between utilities. Modeling correlation allows us to better understand rank data (e.g. agents' taste preferences), and will assist us in rank completion later on in Section 6.4. Figure 6.6 illustrates this correlation structure for two pairs of sushi.

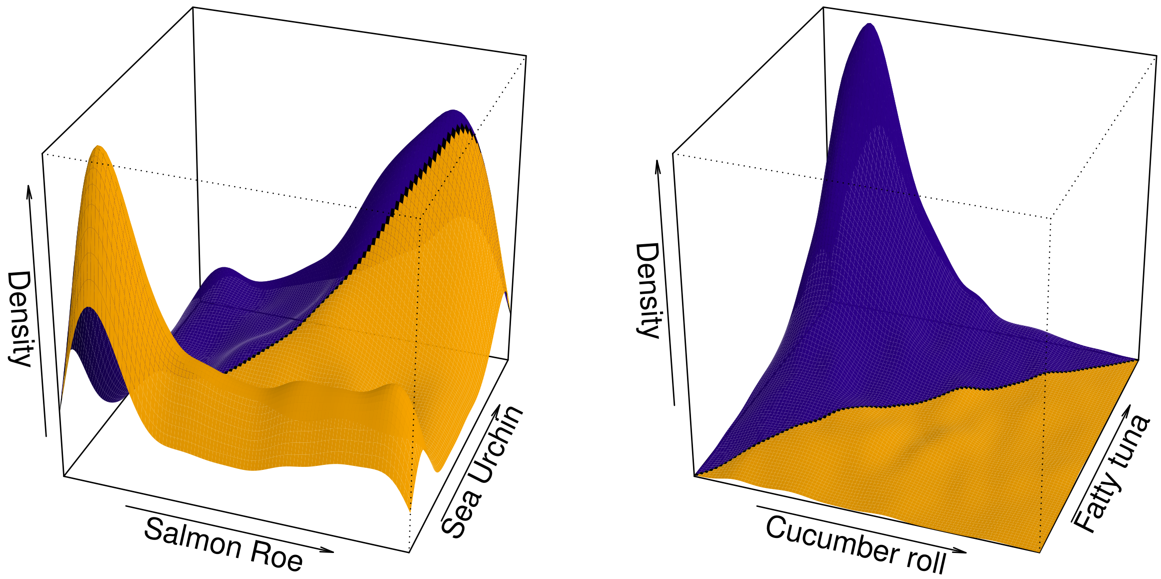


Figure 6.6: Joint distribution for two sets of positively correlated (salmon roe and sea urchin) and negatively correlated (cucumber roll and fatty tuna) sushi. The orange region represents the preference salmon roe over sea urchin or cucumber roll over fatty tuna.

The two modes in the joint distribution of salmon roe and sea urchin utility correspond to two different types of agents. One type ranks both high, while the other ranks both low. Similarly, we see agents that tend to like fatty tuna tend to dislike cucumber roll sushi. Figure 6.7 shows the pairwise correlation preferences between all pairs of sushi. A positive correlation means that the sushi tend to be liked or disliked in tandem, whereas a negative correlation means that an agent may like one but not the other.

6.2 RPD Prediction via Smoothing

As discussed in Section 3, estimating the rank position distribution (RPD) of rank data is hard with small n. However, the RPD is very useful in many contexts. For example, we might want to ask *What will be the demand for this sushi?*

We can use NPRUMs to smooth out noise. After fitting the RUM, I regenerate rank data by drawing a large number of samples from the model. By comparing the Figures 6.8 and 6.10 with

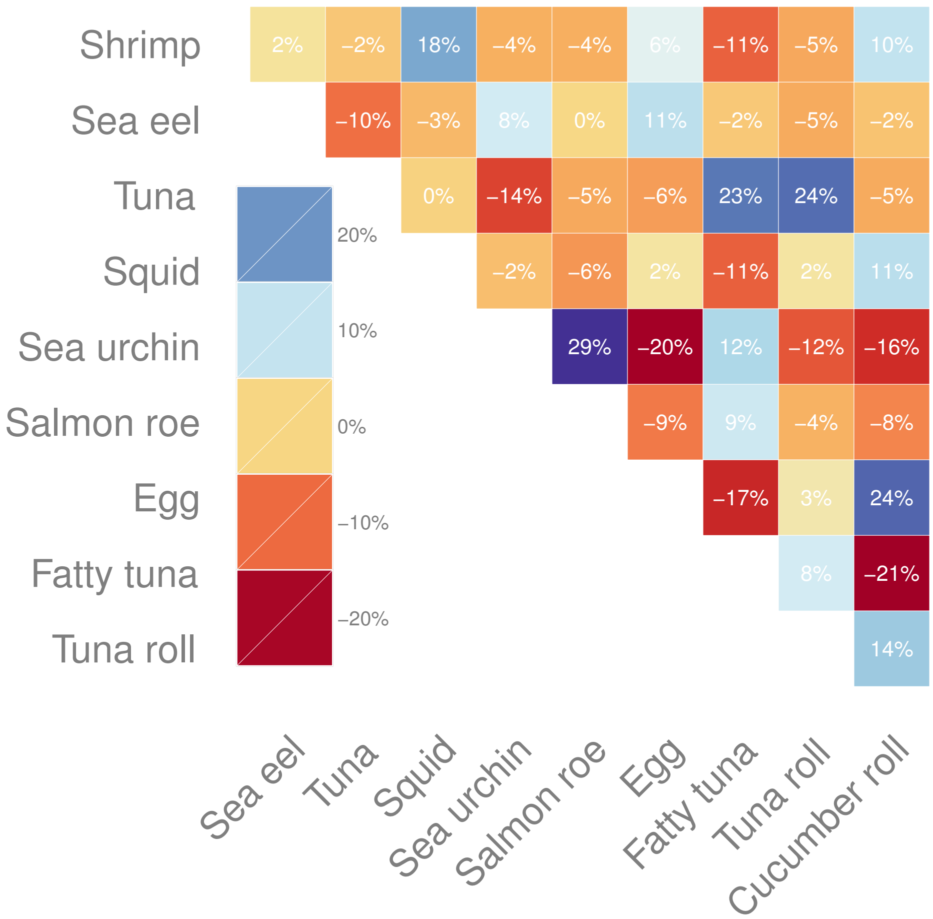


Figure 6.7: Pairwise correlation matrix for observed utilities of all ten sushi variants.

the actual RPD in Figure 6.5, we see that the regenerated RPD is a better estimate of the RPD than the empirical RPD.

In order to explore this concretely, I compare NPRUM with the following other RUMs in their ability to estimate RPDs:

- **Empirical:** Unsmoothed RPD as a baseline.
- **Plackett-Luce:** Gumbel RUM
- **2 x Normal Fixed Variance (FV):** Each agent is in one of two “types” with a certain

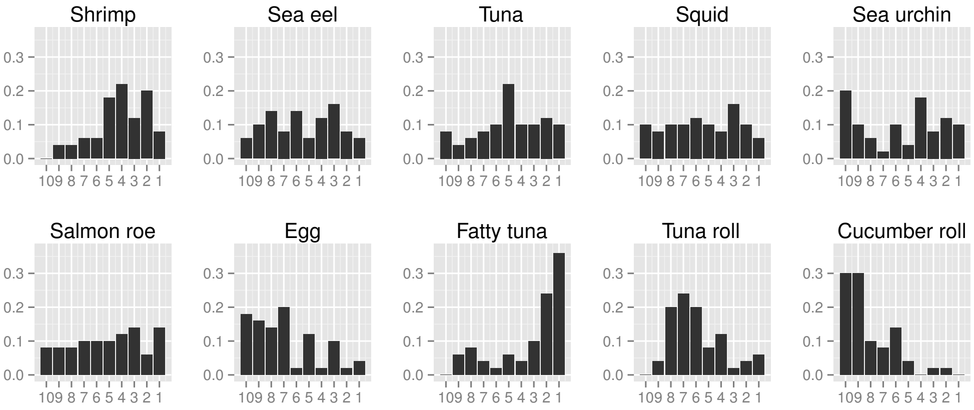


Figure 6.8: The empirical RPD of first 50 sushi agents

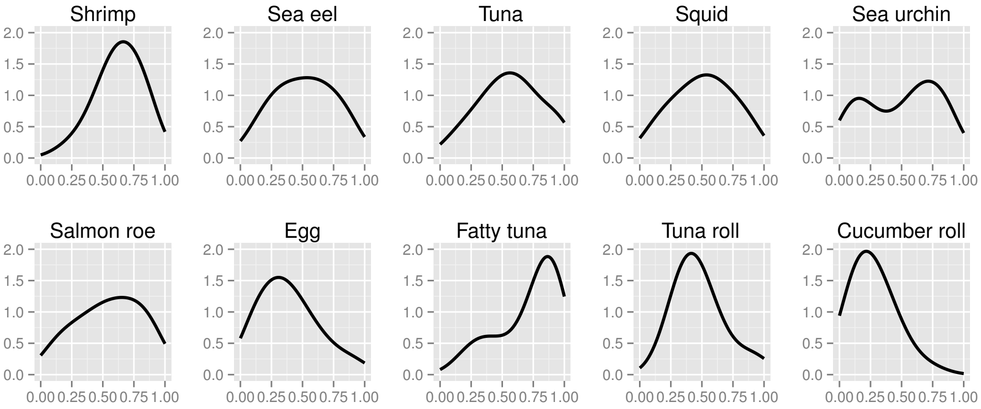


Figure 6.9: The NPRUM fit on first 50 sushi agents. In this graph $h = 0.12$, the best performing bandwidth from the experimental smoothing results from Figure 6.11.

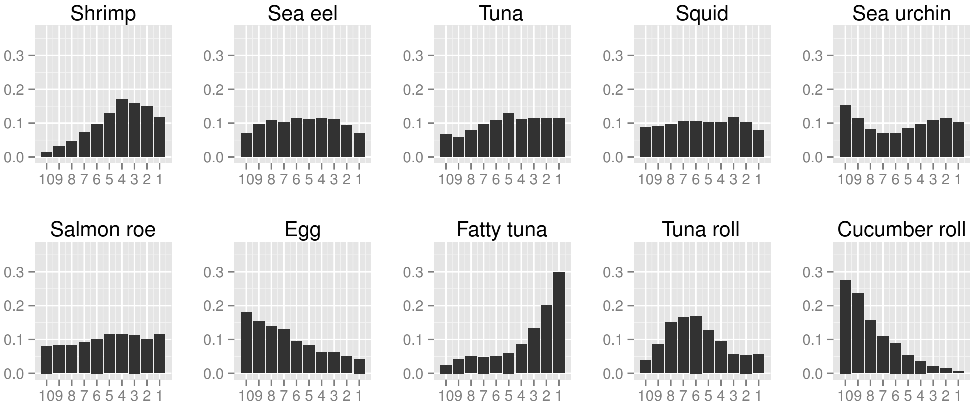


Figure 6.10: The regenerated RPD. This is the RPD on 5000 *simulated* agents drawn from NPRUM fit on first 50 sushi agents.

probability. The two types each have a different multivariate normal distribution (with the identity covariance matrix) for the joint utility density. (Azari Soufiani et al., 2013)

- **Normal Different Variance (DV):** The alternatives each have independent normally-distributed utilities with different variances, estimated from the data. (Azari Soufiani et al., 2012)

I fit each of these models on the noisy data, then regenerate 5000 samples from the models and calculate the regenerated RPDs on the new data.

I measure the success of smoothing by comparing the regenerated RPD from a random $n = 25, 50, 75$ or 100 agents from the sushi dataset with the RPD of the remaining $5000 - n$. I use total variation distance ($\delta(P, Q)$) between the RPDs as my metric, where Q is the regenerated RPD of the original n agents, and P is the RPD of the remaining $5000 - n$ agents. I present the results of this experiment in Figure 6.11. In the case where $n = 50$, a bandwidth of 0.12 outperforms all other bandwidths with a TVD of 0.0856 ± 0.0026 (95% interval), which is 7% and 47% less than the TVDs of Normal RUM and empirical, respectively. The nonparametric RUM for $n = 50$ outperforms all other RUMs with statistical significance ($\alpha = 0.05$) at bandwidths $h \in \{0.10, \ldots, 0.16\}$. NPRUM's advantage is more pronounced with a smaller n, when there is more noise in the empirical RPD.

6.3 RUM Comparison Results

To compare predictive and estimation capabilities of various RUM models, I adopt two metrics. The first metric, average log-likelihood, evaluates both in-sample and out-of-sample fit. The second metric measures error in estimating the pairwise matrix P. I also include the error metrics for the "Empirical" model, where the model matrix is exactly the preference matrix of the training dataset. I use cross-validation to determine the best bandwidth parameter to use for NPRUM. As shown in Figure 6.12, a larger h leads to more smoothing of the marginal utility distribution.

I run each model and dataset pair for 20 repetitions and 20 iterations each, with 80% of the data

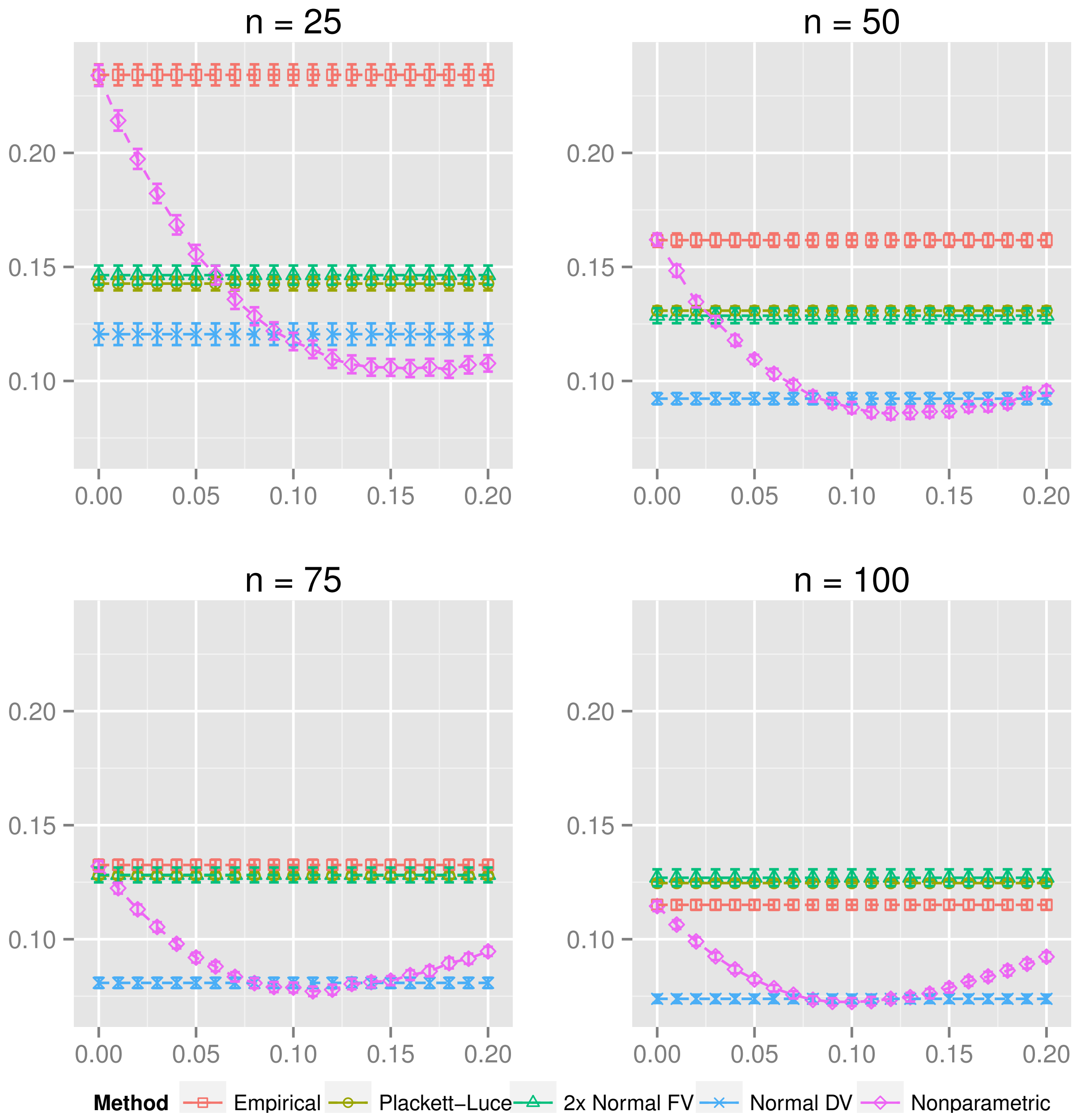


Figure 6.11: Out-of-sample RPD prediction performance. x-axis is bandwidth (h). y-axis is TVD. 75 repetitions are done for each data point. Error bars represent 95% confidence intervals. n represents the number of agents for which RPD was smoothed.

used as a training set and 20% used as the test set. While the methods converge in fewer than 10 iterations, I chose to run 20 iterations for all methods for a fair time comparison. Methods are run on a random 100 agents from the sushi and the election dataset. I report the mean and standard

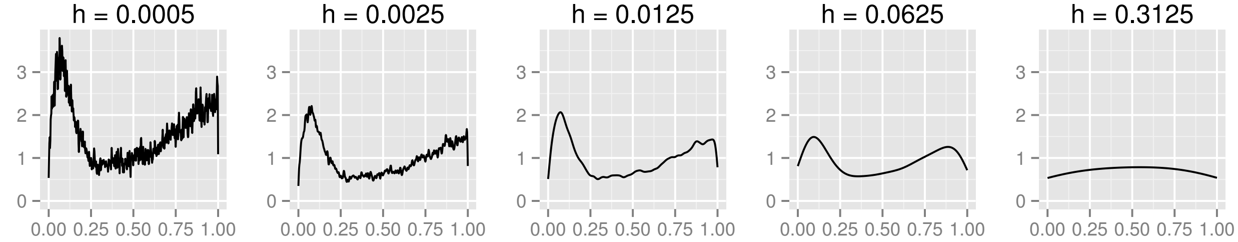


Figure 6.12: The choice of h affects the underlying nonparametric random utility distribution. If h is too low, there are spurious artifacts. If h is too high, it drowns out the features of the distribution.

error for each predictive metric. The results are shown in Table 6.2.

Table 6.2: (top) Average log likelihood. (bottom) Total variation distance between pairwise matrices. Numbers in bold are significantly better than other methods. * means that the method does not converge

	Election		Nascar		Sushi	
Method	**Train**	**Test**	**Train**	**Test**	**Train**	**Test**
Plackett-Luce	-5.95 (1e-02)	-5.98 (3e-02)	-3.97 (1e-02)	-4.43 (5e-02)	-14.25 (6e-03)	-14.37 (1e-02)
Normal FV	-7.73 (1e-02)	-7.44 (3e-02)	-5.37 (1e-01)	-6.89 (3e-01)	-14.06 (7e-03)	-14.06 (1e-02)
2 x Normal FV	-8.16 (2e-02)	-8.41 (3e-02)	**-3.79 (8e-03)**	-4.17 (3e-02)	-13.87 (1e-02)	-14.21 (2e-02)
Normal DV	-7.73 (1e-02)	-7.66 (2e-02)	*	*	**-13.66 (8e-03)**	-13.96 (1e-02)
NP ($h = .11$)	**-2.65 (1e-02)**	**-2.70 (3e-02)**	-3.83 (6e-03)	**-3.96 (2e-02)**	-13.75 (6e-03)	**-13.86 (1e-02)**
Plackett-Luce	15.29 (4e-02)	14.51 (6e-02)	7.31 (6e-02)	5.83 (3e-02)	2.28 (1e-02)	4.35 (3e-02)
Normal FV	5.49 (2e-02)	6.16 (4e-02)	2.42 (2e-02)	3.07 (2e-02)	4.86 (1e-02)	5.85 (4e-02)
2 x Normal FV	4.65 (3e-02)	5.64 (5e-02)	2.55 (8e-03)	2.80 (2e-02)	4.57 (2e-02)	4.94 (4e-02)
Normal DV	2.76 (6e-02)	5.27 (7e-02)	*	*	2.95 (5e-02)	5.29 (6e-02)
NP ($h = .11$)	2.12 (6e-03)	**4.45 (4e-02)**	1.86 (8e-03)	**2.71 (2e-02)**	0.93 (4e-03)	**3.63 (2e-02)**
Empirical	**0 (0)**	4.68 (4e-02)	**0 (0)**	3.19 (2e-02)	**0 (0)**	3.86 (3e-02)

Table 6.3: Runtime (seconds). Numbers in bold are significantly better than other methods. * means the method does not converge.

Method	**Election**	**Nascar**	**Sushi**
Plackett-Luce	28390 (2e+02)	930 (8e+00)	**150 (1e+00)**
Normal FV	28570 (1e+02)	920 (3e+00)	13680 (7e+01)
2x Normal FV	39120 (2e+02)	1910 (9e+00)	22280 (1e+02)
Normal DV	27570 (1e+02)	*	13610 (7e+01)
NP (h = .11)	**210 (1e+00)**	**60 (3e-01)**	180 (8e-01)

The nonparametric method outperforms the parametric RUMs on every out-of-sample metric and for all of the datasets. In the Sushi data, the Normal DV outperforms the NPRUM on in-sample log-likelihood, but NPRUM outperforms Normal DV on out-of-sample log-likelihood, which is evidence that the Normal DV may have overfit to the training set. The same explanation goes for 2x Normal FV on the in-sample log-likelihood on the Nascar dataset. In the same data, the same behavior is evident when comparing 2x Normal FV to Normal FV. The 2x Normal FV outperforms in training, but not in the test set.

As we see from Table 6.3, the nonparametric method also takes significantly less time than any other method in all datasets except for PL on sushi. Estimation of parameters for PL model for Nascar and Sushi data was done with Hunter's minorize-maximization algorithm (2004) which is faster than the general Monte Carlo EM algorithm proposed by Azari et al. (2012).

6.4 Rank Completion

Nonparametric RUMs are also applicable to rank completion, a recommendation problem where one may want to predict the full rankings for an agent given observed partial rankings.

Specifically, one may want to predict the agent's second-ranked sushi given the agent's top-ranked sushi. From the n-agent training set, I estimate the conditional distribution $\Pr(\pi(2)|\pi(1))$ for each first-ranked distribution. I calculate the TVD between this predicted conditional distribution and the actual conditional distribution on the $5000 - n$ agents used as test data. I take the average of the conditional TVDs as the performance metric, weighted by the frequency of each first-ranked alternative.

Figure 6.13 shows that the NPRUM model outperforms existing RUM methods at this rank completion problem. Interestingly, the parametric RUMs barely improve when the sample size is increased from $n = 50$ to $n = 100$. NPRUM's advantage widens with more data because NPRUM is the only existing RUM able to capture correlation, which is vital for rank completion.

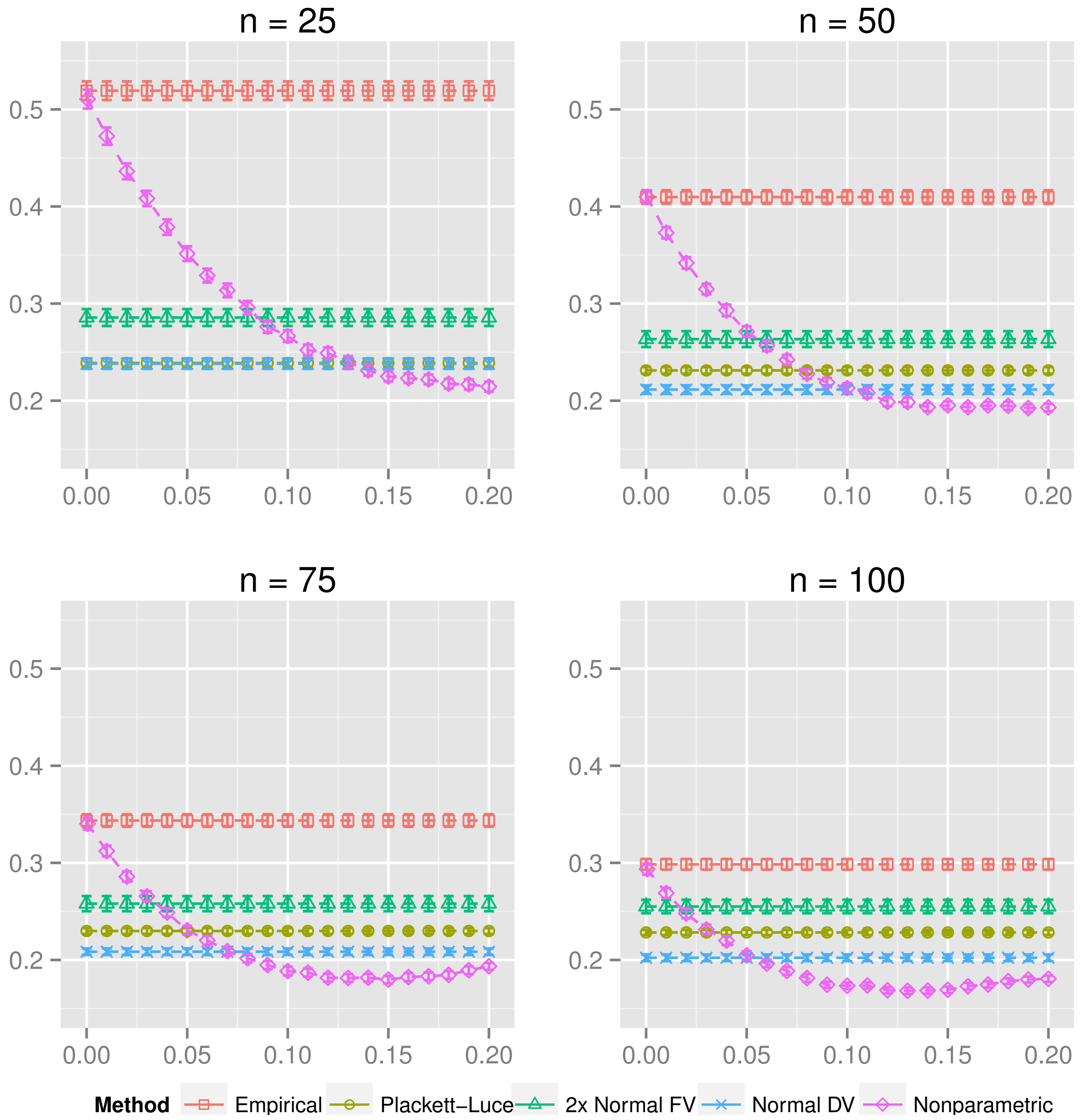


Figure 6.13: Out-of-sample rank completion performance. x-axis is bandwidth (h). y-axis is weighted mean TVD. 100 repetitions are done for each data point. Error bars represent 95% confidence intervals. n represents the number of agents used as training for rank completion.

6.5 Summary

Table 6.4 summarizes five implemented methods with respect to the out-of-sample evaluation metrics discussed above. NPRUM outperforms the best benchmark method by 1% in average log-likelihood, 6% in pairwise preference matrix TVD, 7% in rank smoothing TVD, and 14% in rank completion TVD. These differences are statistically significant ($\alpha = 0.05$), and demonstrate NPRUMs superior performance in fitting the data, predicting pairwise comparisons, smoothing noisy data, and predicting second ranks given the first rank.

Table 6.4: Performance summary of five implemented solutions on the Sushi dataset. Standard errors are included in parentheses. Smoothing is evaluated with $n = 50$ and completion is evaluated with $n = 100$. Bandwidth is set to the $h = 0.11$ used in Section 6.3.

Method	Average LL	Pairwise	Smoothing	Completion
Empirical	N/A	3.86 (3e-02)	0.162 (1e-03)	0.299 (3e-03)
Plackett-Luce	-14.37 (1e-02)	4.35 (3e-02)	0.131 (7e-04)	0.228 (1e-03)
Normal DV	-13.96 (1e-02)	5.29 (6e-02)	0.092 (9e-04)	0.202 (2e-03)
2 x Normal FV	-14.21 (2e-02)	4.94 (4e-02)	0.129 (2e-03)	0.255 (4e-03)
NP ($h = 0.11$)	**-13.86** (1e-02)	**3.63** (2e-02)	**0.086** (9e-04)	**0.174** (2e-03)

Chapter 7

Discussion

In this chapter, I discuss the advantages and disadvantages of NPRUMs. Naturally, increasing the model complexity comes with challenges in regards to estimation and inference. But on the other hand, it brings more expressive descriptions of the dataset with new capabilities and increased predictive power.

7.1 Distributional assumptions

NPRUM's weak modeling assumptions make them more generally applicable than parametric RUMs. However, assumptions are useful in certain settings, and RUMs with the correct assumptions may perform better than other models. For example, PL outperforms Normal for Election data, but Normal outperforms PL for Sushi data (see Table 6.2). Still, NPRUM outperforms both PL and Normal in the datasets, indicating that NPRUM's weak assumptions work better than the strong ones of PL and Normal RUM.

7.2 Estimation

The Monte Carlo EM algorithm for NPRUM is based on the Monte Carlo EM algorithm for the exponential family of RUMs introduced by Azari et al. (2012). I compare the time and space complexities of Monte Carlo EM in the parametric and nonparametric settings.

7.2.1 Time Complexity

In the E-step, sampling from the truncated parametric and nonparametric distributions can be accomplished via similar techniques. This leads to a similar complexity for each iteration. In practice, NPRUM is faster than existing methods (Table 6.2). I believe this is because the intermediary iterations are not limited by distributional assumptions, so NPRUM can take a quicker path to convergence within the E-step.

In the M-step, fitting utility densities for Exponential Family distributions (Morris, 1982) is simple because of the relationship between the sufficient statistics and the MLE parameters. Fitting the nonparametric model is more difficult as it requires kernel density estimation, a choice of kernel (fixed at Gaussian for this thesis), and a bandwidth. Identifying the distribution in the M-step of a parametric RUM is $O(mn)$, while identifying the KDE in the M-step of NPRUM is $O(dmn)$, where d is the number of evaluation points desired in a dimension, and can be a large constant.

7.2.2 Space Complexity

Representing a parametric RUM requires m location parameters for a Plackett-Luce model or $2m$ parameters for a Normal model. In contrast, the nonparametric model needs to be represented by the original vectors of utilities from the rank data, which is proportional in size to the data. Parametric RUMs are $O(m)$ in space complexity while NPRUMs are $O(mn)$. The other option for representing NPRUM is storing values of the density function on a lattice grid – but this quickly becomes infeasible with many alternatives since it is exponential in m, leading to the curse of dimensionality.

7.3 Inference

Tasks such as identifying the maximal posterior probability ranking and specifying the distribution over ranks are intractable because of the $m!$ size of the permutation space. However, RUM properties such Luce's Choice Axiom in the Plackett-Luce RUM lend conveniences to inference. Specifically, maximal posterior probability rankings and distributions over ranks can be identified easily. Pairwise preferences are also found easily in Normal and PL RUMs. NPRUM must instead rely on Monte-Carlo and resampling methods to perform these inferential tasks – sampling from multivariate kernel density estimates is easy, but integration and summarization is difficult.

Chapter 8

Conclusions

In this thesis, I introduced a nonparametric random utility model, and demonstrated that it outperforms existing methods on multiple real-world datasets. The evaluation has been done for multiple predictive metrics, including rank position distribution prediction, out-of-sample average log-likelihood, and rank completion. Results include statistically significant improvements in predictive average log-likelihood for all three datasets, and a 14% performance improvement in rank completion TVD on the sushi dataset.

In providing a comprehensive study of various RUMs, I observed that certain models consistently outperform other models on different types of datasets. Nonparametric models are flexible enough to capture the best features in every setting, leading to superior performance.

NPRUMs also outperform existing RUMs in regard to rank completion. This is due to a more expressive latent utility model that accounts for features such as correlation, which is imperative in any rank model that seeks to complete ranks given partial data.

This approach is the first random utility model method on full rank data to forgo distributional assumptions. I demonstrate that by approaching this carefully, new interpretations can be achieved and new applications will be enabled.

8.1 Extensions

I present possible research extensions for nonparametric random utility models, both in the context of this thesis and in the larger context of rank aggregation.

Extension of RUMs to incorporate ties. Common rank data, such as chess matches and football games, include tied ranks. However, random utility models cannot currently incorporate tied rank data, because the generative model involves ordering draws from a continuous space. Incorporating ties will extend the usefulness of RUMs in modeling domains that involve tied rankings.

Continued development of `StatRank` R package. In the interest of making the results reproducible and the methods useful for general application, I have released `StatRank`, an R package available on CRAN with productionalized versions of all of my functions. This includes (1) estimation methods for PL, normal, multitype, and nonparametric RUMs (2) Nascar and Election datasets (3) various evaluation metrics (4) useful evaluation functions (e.g. pairwise preference matrices and likelihood functions) (5) visualization functions used to make a number of figures in this thesis.

Application to pairwise comparison settings. The NPRUM algorithm also accepts pairwise data as input, which allows it to be compared against other pairwise methods such as Elo, Bradley-Terry, Zemel, Thurstone, Rank Centrality, Glicko, and TrueSkill. NPRUM's superior predictive ability on pairwise preferences compared to other RUMs suggests its usefulness in predicting winners in paired comparison data (e.g. a recent Kaggle competition to predict the outcomes of the 2014 NCAA March Madness tournament). Paired comparison data is common in sports and game analysis.

Optimization with respect to applied metrics. All of the metrics used to evaluate NPRUMs in this thesis (rank estimation, log likelihood, rank completion, pairwise preferences) are related to how well NPRUM predicts the structure of out-of-sample rank data. I have shown that NPRUM outperforms existing RUMs in capturing out-of-sample structure, but other metrics are more appropriate for some rank data applications, such as accuracy of sports match prediction (predicting spreads or brackets), meta-search results (using metrics such as normalized discounted cumulative gain), and aggregating crowd-sourced ranks (using metrics such as Kendall's tau).

References

Nir Ailon. Aggregation of partial rankings, p-ratings and top-m lists. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2007.

A. Ammar and D. Shah. Ranking: Compare, don't score. In *Allerton'11: Proceedings of the 49th annual Allerton conference on Communication, Control, and Computing*, pages 776–783, 2011.

Hossein Azari Soufiani, David C. Parkes, and Lirong Xia. Random utility theory for social choice. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, pages 126–134, Lake Tahoe, NV, USA, 2012.

Hossein Azari Soufiani, Hansheng Diao, Zhenyu Lai, and David C. Parkes. Generalized random utility models with multiple types. In *Proceedings of the Annual Conference on Neural Information Processing Systems (NIPS)*, pages 73–81, Lake Tahoe, NV, USA, 2013.

Steven Beggs, Scott Cardell, and Jerry Hausman. Assessing the potential demand for electric cars. *Journal of econometrics*, 17(1):1–19, 1981.

Henry David Block and Jacob Marschak. Random orderings and stochastic theories of responses. In *Contributions to Probability and Statistics*, pages 97–132, 1960.

Edwin Bonilla, Shengbo Guo, and Scott Sanner. Gaussian process preference elicitation. In *Advances in Neural Information Processing Systems 23*, pages 262–270. 2010.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th international conference on Machine learning (ICML-07)*, pages 129–136, Corvalis, Oregon, USA, 2007.

Josu Ceberio, Alexander Mendiburu, and Jose Antonio Lozano. The plackett-luce ranking model on permutation-based optimization problems. In *Evolutionary Computation (CEC), 2013 IEEE Congress on*, pages 494–501. IEEE, 2013.

William Chen and Hossein Azari Soufiani. *StatRank: Statistical Rank Aggregation: Inference, Evaluation, and Visualization*, 2013. URL http://CRAN.R-project.org/package=StatRank. R package version 0.0.2.

Yen-Liang Chen and Li-Chen Cheng. A novel collaborative filtering approach for recommending ranked items. *Expert systems with applications*, 34(4):2396–2405, 2008.

Marquis de Condorcet. *Essai sur l'application de l'analyse à la probabilité des décisions rendues à la pluralité des voix*. Paris: L'Imprimerie Royale, 1785.

Vincent Conitzer. *Computational aspects of preference aggregation*. PhD thesis, Carnegie Mellon University, 2006.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):pp. 1–38, 1977. ISSN 00359246. URL http://www.jstor.org/stable/2984875.

Cynthia Dwork, Ravi Kumar, Moni Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proceedings of the 10th World Wide Web Conference*, pages 613–622, 2001.

Arpad E. Elo. *The rating of chessplayers, past and present*. Arco Pub., New York, 1978. ISBN 0668047216 9780668047210.

Andrew Gelman and Donald B Rubin. Inference from iterative simulation using multiple sequences. *Statistical science*, pages 457–472, 1992.

Mark E Glickman. Parameter estimation in large dynamic paired comparison experiments. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 48(3):377–394, 1999.

Isobel Claire Gormley and Thomas Brendan Murphy. Analysis of Irish third-level college applications data. *Journal of the Royal Statistical Society Series A*, 169(2):361–379, 2006.

Isobel Claire Gormley and Thomas Brendan Murphy. A latent space model for rank data. In *Statistical Statistical Network Analysis: Models, Issues and New Directions. LNCS*, volume 4503, pages 90–107, 2007.

Isobel Claire Gormley and Thomas Brendan Murphy. Exploring voting blocs within the irish exploring voting blocs within the irish electorate: A mixture modeling approach. *Journal of the American Statistical Association*, 103(483):1014–1027, 2008.

John Guiver and Edward Snelson. Bayesian inference for Plackett-Luce ranking models. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML-09, pages 377–384, Montreal, Quebec, Canada, 2009.

Ralf Herbrich, Tom Minka, and Thore Graepel. Trueskill: A bayesian skill rating system. In *Advances in Neural Information Processing Systems*, pages 569–576, 2006.

David R. Hunter. MM algorithms for generalized Bradley-Terry models. In *The Annals of Statistics*, volume 32, pages 384–406, 2004.

Toshihiro Kamishima. Nantonac collaborative filtering: Recommendation based on order responses. In *Proceedings of the Ninth International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 583–588, Washington, DC, USA, 2003.

Amy N Langville and Carl Dean Meyer. *Who's# 1?: the science of rating and ranking.* Princeton University Press, 2012.

Seok Kee Lee, Yoon Ho Cho, and Soung Hie Kim. Collaborative filtering with ordinal scale-based implicit ratings for mobile music recommendations. *Information Sciences*, 180(11):2142–2155, 2010.

Tyler Lu and Craig Boutilier. Learning Mallows models with pairwise preferences. In *Proceedings of* *the Twenty-Eighth International Conference on Machine Learning (ICML 2011)*, pages 145–152, Bellevue, WA, USA, 2011.

R.Duncan Luce. The choice axiom after twenty years. *Journal of Mathematical Psychology*, 15(3): 215 – 233, 1977. ISSN 0022-2496. doi: http://dx.doi.org/10.1016/0022-2496(77)90032-3. URL http://www.sciencedirect.com/science/article/pii/0022249677900323.

Robert Duncan Luce. *Individual Choice Behavior: A Theoretical Analysis*. Wiley, 1959.

John I. Marden. *Analyzing and modeling rank data*. Chapman & Hall, 1995.

Daniel McFadden. Conditional logit analysis of qualitative choice behavior. In *Frontiers of Econometrics*, pages 105–142, New York, NY, 1974. Academic Press.

Carl N. Morris. Natural Exponential Families with Quadratic Variance Functions. *Annals of Statistics*, 10(1):65–80, 1982.

R. M. Neal. Software for slice sampling, March 2008. URL http://www.cs.toronto.edu/~radford/slice.software.html.

Radford M. Neal. Slice sampling. *Annals of Statistics*, 31:705–767, 2003.

SW Nombekela, MR Murphy, HW Gonyou, and JI Marden. Dietary preferences in early lactation cows as affected by primary tastes and some common feed flavors. *Journal of dairy science*, 77 (8):2393–2399, 1994.

OEIS. The on-line encyclopedia of integer sequences, 2011. URL http://oeis.org/A002627. [Online; accessed 29-March-2014].

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. The pagerank citation ranking: Bringing order to the web. 1999.

Emanuel Parzen. On estimation of a probability density function and mode. In *The Annals of Mathematical Statistics*, pages 847–1226, 1962.

Robin L. Plackett. The analysis of permutations. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 24(2):193–202, 1975.

Murray Rosenblatt. Remarks on some nonparametric estimates of a density function. In *The Annals of Mathematical Statistics*, pages 569–878, 1956.

Daniel Tarlow, Ryan Prescott Adams, and Richard S. Zemel. Randomized optimum models for structured prediction. In *Proceedings of the Fifthteenth International Conference on Artificial Intelligence and Statistics (AISTATS-11)*, pages 1221–1229, La Palma, Canary Islands, 2012.

Louis Leon Thurstone. A law of comparative judgement. *Psychological Review*, 34(4):273–286, 1927.

Nicolaus Tideman. *Collective Decisions and Voting: The Potential for Public Choice*. Ashgate Publishing, 2006.

Michel Truchon. Borda and the maximum likelihood approach to vote aggregation. *Mathematical Social Sciences*, 55(1):96–102, 2008.

Wikipedia. 2002 nascar winston cup series — wikipedia, the free encyclopedia, 2014a. URL http://en.wikipedia.org/wiki/2002_NASCAR_Winston_Cup_Series. [Online; accessed 29-March-2014].

Wikipedia. Nascar rules and regulations — wikipedia, the free encyclopedia, 2014b. URL http://en.wikipedia.org/wiki/NASCAR_rules_and_regulations. [Online; accessed 29-March-2014].

Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. Listwise Approach to Learning to Rank: Theorem and Algorithm. In *Proceedings of the Twenty-Fifth International Conference on Machine Learning (ICML-08)*, Helsinki, Finland, 2008.