# Efficient Learning from Comparisons

THÈSE N$^{O}$ 8637 (2018)

PRÉSENTÉE LE 1$^{ER}$ JUIN 2018

À LA FACULTÉ INFORMATIQUE ET COMMUNICATIONS
LABORATOIRE POUR LES COMMUNICATIONS INFORMATIQUES ET LEURS APPLICATIONS 4
PROGRAMME DOCTORAL EN INFORMATIQUE ET COMMUNICATIONS

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Lucas MAYSTRE

acceptée sur proposition du jury:

Prof. M. C. Gastpar, président du jury
Prof. M. Grossglauser, directeur de thèse
Prof. B. Prabhakar, rapporteur
Prof. D. Shah, rapporteur
Prof. M. Vojnovic, rapporteur
Prof. M. Jaggi, rapporteur

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE
Suisse
2018

# Acknowledgments

This thesis would not have been possible without the contributions, explicit or implicit, of many. First and foremost, I would like to thank my advisor, Matthias Grossglauser, for his mentorship. His vision and sharp insights have helped shape much of the research presented in this thesis. I am also particularly grateful for his never-ending generosity and enthusiasm.

Next, I would like to thank the members of my jury committee: Martin Jaggi, Michael Gastpar, Balaji Prabhakar, Devavrat Shah and Milan Vojnović. I thank them for the time they made in their busy schedules to read my dissertation carefully and provide useful comments. Special thanks go to Patrick Thiran, for taking interest in my work and for his valued feedback.

I have been fortunate to be surrounded by fantastic staff at the lab. Holly Cogliati-Bauereis provided invaluable help by proof-reading and improving my manuscripts. Patricia Hjelt, Angela Devenoge and Danielle Alvarez assisted me with the administrative tasks—always with utmost kindness. Carlos Perez helped me with cutting-edge IT infrastructure needs. I would like to thank them all.

It was a joy to work with a group of kind-hearted and clever people who gave supportive advice throughout my time at EPFL. Brunella, many thanks for your friendship and for inspiring me to always give the best of myself. Young-Jun, I have learned a lot from and with you, thank you for your benevolence. Victor, thank you for your enthusiasm and your constant positive attitude. Christina and William, thank you for being the best office mates I could wish for. Thank you Farid, Mohamed, Julien, Runwei, Vincent, Sébastien, Farnood, Daniyar, Aswin, Elisa, Emti, and Régis!

Finally, I would like to express my gratitude to my family, who supported me every step of the way. Thank you for your trust in me. A big heartfelt thank you goes to Ksenia, for making everything in life enjoyable.

*Lausanne, May 15, 2018* L. M.

# Abstract

Humans are comparison machines: comparing and choosing an item among a set of alternatives (such as objects or concepts) is arguably one of the most natural ways for us to express our preferences and opinions. In many applications, the analysis of data consisting of comparisons enables finding valuable information. But datasets often contain inconsistent comparison outcomes, because human preferences shift and observations are tainted by noise. A principled approach to dealing with intransitive data is to posit a probabilistic model of comparisons. In this thesis, we revisit Luce's choice model, the study of which began almost a century ago, in the context of large-scale online data collection. We set out to learn a ranking over a set of items from comparisons in a *computationally*, *statistically* and *data efficient* way.

First, we consider the algorithmic problem of estimating model parameters from choice data, and we seek to improve upon the computational and statistical efficiency of existing methods. Our contribution is to show that it is possible to express the maximizer of the model's likelihood function as the stationary distribution of a Markov chain. This enables the use of fast linear solvers or well-studied iterative methods for Markov chains for parameter inference in Luce's model.

Second, we develop a data-efficient method for learning a ranking, by adaptively choosing pairs of items to compare, based on previous comparison outcomes. We begin by showing that Quicksort, a widely-known sorting algorithm, works well even if comparison outcomes are noisy. Under distributional assumptions on model parameters, we provide asymptotic bounds on the quality of the ranking it recovers. Building on this result, we use sorting algorithms as a basis for a simple, practical active-learning method that performs well on real-world datasets, at a small fraction of the computational cost of competing methods.

Third, we focus on structured choices in a network. In particular, we study a model where users navigate in a network (e.g., following links on the Web) and set out to estimate transition probabilities along the edges of the network from limited observations. We show that if transitions follow Luce's axiom, their probability can be inferred using only data consisting of the (marginal) traffic at each node of the network. We propose

a robust inference algorithm that admits a computationally-efficient implementation. Our method scales to networks with billions of nodes and achieves good predictive performance on clickstream data.

Beyond human preferences, probabilistic models of pairwise comparisons can also be applied to sports. Consider football: two teams are compared against each other, and the better one wins. In the last part of this thesis, we look at a concrete application of pairwise comparison models and tackle the task of predicting outcomes of matches between national football teams. These teams play only a few matches every year, hence it is difficult to accurately assess their strength. Noting that national team players also compete against each other in clubs, we propose a way to overcome this challenge by taking into account outcomes of matches between clubs, of which there are plenty. We do so by embedding all matches in *player space*, and devise a computationally-efficient inference procedure. The resulting model predicts international tournament results more accurately than those using only national team results.

**Keywords** comparisons, choices, rankings, probabilistic models, statistical inference, algorithms, machine learning, active learning, networks

# Résumé

Nous, humains, sommes des machines à comparer. Faire une comparaison et choisir un objet ou un concept parmi un ensemble d'alternatives est sans doute l'une des façons les plus naturelles d'exprimer nos préférences et nos opinions. Dans le cadre de beaucoup d'applications pratiques, l'analyse de données sous forme de comparaisons permet de trouver des informations précieuses. Mais les jeux de données recueillis contiennent souvent des résultats de comparaisons en contradiction les uns avec les autres, parce que nos préférences changent et que les comparaisons observées sont contaminées par du bruit. Une approche raisonnée pour traiter de telles données intransitives consiste à postuler un modèle probabiliste de comparaisons. Dans cette thèse, nous revisitons le modèle de choix proposé par Luce (dont l'étude remonte à près d'un siècle) dans le contexte de la collecte de données en ligne et à grande échelle. Notre but est d'apprendre un classement sur un ensemble d'objets à partir de comparaisons d'une façon *efficace* : statistiquement, en matière de ressources de calcul et sur le plan de la quantité de données.

Tout d'abord, nous examinons le problème algorithmique de l'estimation des paramètres du modèle à partir de données sous forme de comparaisons et cherchons à améliorer l'efficacité statistique et calculatoire des méthodes existantes. Notre contribution consiste à montrer qu'il est possible d'exprimer les paramètres qui maximisent la vraisemblance du modèle par la distribution stationnaire d'une chaîne de Markov. Ceci ouvre la voie à l'utilisation de programmes de résolution d'équations linéaires rapides ou à l'utilisation de méthodes itératives pour chaînes de Markov pour estimer les paramètres du modèle de Luce.

Deuxièmement, nous développons une méthode économe en données pour apprendre un classement. Cette méthode consiste à choisir des paires d'objets à comparer de façon adaptive, en fonction des résultats de comparaisons observés précédemment. Nous commençons par montrer que Quicksort, un algorithme de tri connu, fonctionne bien même si les résultats des comparaisons sont bruités. Sous certaines hypothèses sur la distribution des paramètres du modèle, nous fournissons des bornes asymptotiques sur la qualité du classement retourné par Quicksort. En nous appuyant sur ce résultat, nous utilisons des algorithmes de tri comme point de départ d'une méthode simple et pratique

d'apprentissage actif. Celle-ci donne de bons résultats sur des jeux de données du monde réel tout en utilisant seulement une petite fraction des ressources de calcul nécessaires aux méthodes concurrentes.

Troisièmement, nous nous penchons sur un problème de choix structurés dans un réseau. Plus précisément, nous étudions un modèle où des utilisateurs naviguent sur un réseau (par exemple en suivant des liens sur le Web) et entreprenons d'apprendre les probabilités de transition sur les arêtes à partir d'observations limitées. Nous montrons que si les transitions suivent l'axiome de Luce, leur probabilité peut être déduite du trafic (marginal) à chaque nœud. Nous proposons un algorithme d'estimation des paramètres qui est robuste et qui admet une implémentation efficace en ressources de calcul. Notre méthode peut s'appliquer à des réseaux composés de milliards de nœuds et atteint de bons résultats pour la prédiction de flux de clics sur le Web.

Au-delà des préférences humaines, les modèles probabilistes de comparaisons par paire peuvent aussi s'appliquer au sport. Pensez au football : deux équipes se comparent l'une à l'autre, et la meilleure des deux gagne. Dans la dernière partie de cette thèse, nous considérons un cas pratique et nous nous attaquons au problème de prédire les résultats de matchs entre équipes nationales. Ces équipes ne jouent que quelques matchs chaque année et de ce fait il est difficile de juger de leur force de façon précise. En observant que les joueurs appelés en sélection nationale jouent aussi les uns contre les autres dans leur club respectif, nous proposons une façon de surmonter cette difficulté en prenant en compte les matchs entre clubs (desquels il est facile d'obtenir une grande quantité). Notre méthode se base sur une projection des matchs dans un *espace des joueurs* et s'appuie sur une procédure d'apprentissage économe en temps de calcul. Le modèle qui en résulte prédit les résultats de tournois internationaux d'une façon plus précise que d'autres modèles n'utilisant que les matchs entre équipes nationales.

**Mots-clés** comparaisons, choix, classements, modèles probabilistes, inférence statistique, algorithmes, apprentissage automatique, apprentissage actif, réseaux

# Contents

# Mathematical Notation

| Symbol | Description |
|---|---|
| $x$ | Plain lowercase letters denote scalar values. |
| $\boldsymbol{x} = [x_i]$ | Boldface lowercase letter denote column vectors. |
| $\boldsymbol{X} = [x_{ij}]$ | Boldface uppercase letters denote matrices. |
| $\mathcal{X}$ | Calligraphic uppercase letters denote sets. |
| $\mathbf{R}, \mathbf{R}_{>0}, \mathbf{N}$ | Number types: real, positive real and natural numbers, respectively. |
| $[N]$ | Set of consecutive natural numbers $\{1, \ldots, N\}$. |
| $i \succ j$ | Pairwise comparison outcome "$i$ wins over $j$". |
| $i \succeq \mathcal{A}$ | Multiway comparison outcome "$i$ is chosen among alternatives $\mathcal{A}$". |
| $\mathbf{P}[\mathcal{A}]$ | Probability of the event $\mathcal{A}$. |
| $1_{\{\mathcal{A}\}}$ | Indicator variable of the event $\mathcal{A}$. |
| $\mathbf{E}[x]$ | Expectation of the random variable $x$. |
| $\mathbf{Var}[x]$ | Variance of the random variable $x$. |
| $\mathbf{Cov}[x, y]$ | Covariance of the random variables $x$ and $y$. |
| $O(f(x))$ | $g(x) = O(f(x)) \iff \limsup_{x\to\infty} \lvert g(x)\rvert / f(x) < \infty$. |
| $o(f(x))$ | $g(x) = o(f(x)) \iff \lim_{x\to\infty} g(x)/f(x) = 0$. |
| $\Omega(f(x))$ | $g(x) = \Omega(f(x)) \iff f(x) = O(g(x))$. |
| $\omega(f(x))$ | $g(x) = \omega(f(x)) \iff f(x) = o(g(x))$. |

| Distribution | Domain | Density function $f(x)$ |
|---|---|---|
| $\mathrm{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ | $\mathbf{R}^D$ | $\frac{1}{\sqrt{2\pi\lvert\boldsymbol{\Sigma}\rvert}} \exp\left[-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})\right]$ |
| $\mathrm{U}(a, b)$ | $[a, b]$ | $\frac{1}{b-a}$ |
| $\mathrm{Beta}(\alpha, \beta)$ | $[0, 1]$ | $\frac{1}{B(\alpha,\beta)} x^{\alpha-1}(1-x)^{\beta-1}$ |
| $\mathrm{Exp}(\lambda)$ | $\mathbf{R}_{>0}$ | $\lambda \exp(-\lambda x)$ |
| $\mathrm{Gamma}(\alpha, \beta)$ | $\mathbf{R}_{>0}$ | $\frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha-1} \exp(-\beta x)$ |
| $\mathrm{Gumbel}(\mu, \beta)$ | $\mathbf{R}$ | $\frac{1}{\beta} \exp\{-[z + \exp(-z)]\}$, where $z = \frac{x-\mu}{\beta}$ |

# 1 Introduction

## 1.1 Motivation

Making a choice is a fundamental way for us to express our opinions and preferences. We choose the music we listen to and the movies we watch, and we choose the place where we live and the political candidate we vote for. We constantly compare alternatives in order to discern the one that best suits us. Unsurprisingly, a great understanding of collective and personal opinions can be gained by observing the outcomes of comparisons that we make.

The idea of analyzing human choices has been a longstanding topic of interest to researchers and practitioners across a wide range of disciplines, including psychology, sociology and economics. To give just one example among many, discrete choice analysis (DCA) has become an essential item in the econometrician's toolbox. DCA has important applications: for example, it accurately predicted the impact of a new metro line in the San Francisco Bay area on the usage of various modes of transport [McFadden et al., 1977]. The theory and methods developed in this context resulted in a Nobel prize for its main inventor [McFadden, 2001].

This thesis is part of the quest to improve the analysis of human choices. We are interested in the problem of extracting concise information (e.g., about our preferences) from raw *choice data*, i.e., observations that discriminate one out of several alternatives. Concretely, a typical task of interest would be to obtain a ranking of all alternatives from most to least preferred, often by means of numerical scores that describe the utility of each alternative and that are predictive of future choices. Even though research on choice models has produced a number of well-established methods, modern online applications (of which we give examples shortly) call for new approaches that can cope with large-scale data. Indeed, both the large number of *observations* and the large number of *alternatives* that are typical in modern applications raise new challenges: it becomes important to develop methods that are efficient—not only in order to quickly process all observations,

but also in order to end up with sufficient information about every alternative. This notion of *efficiency* is the guiding thread of this thesis and will be expanded upon in Section 1.3.

**Why Study Choice Data?** If we are ultimately interested in, say, a numerical utility score for each alternative, a sensible question to ask is: Why not *directly* ask for such a score? Two important reasons come to mind.

1. It is particularly natural and easy for humans to make comparisons. A popular theory in social psychology even states that comparing ourselves to others is one the primary ways in which we learn about and define ourselves, our beliefs and opinions [Festinger, 1954]. Arguably, it is more difficult for us to give meaningful and consistent numerical scores. What does a "3.5 star" rating on a restaurant really mean? In a world where everything is relative, an absolute rating might just be the wrong abstraction.

2. In some cases, it is possible to observe choices *implicitly*, simply by recording the actions that we take and the context in which we take them. This makes the process of collecting choices much less obtrusive than *explicitly* asking for feedback. In practice, it means that it is often possible to access much larger datasets, potentially leading to more accurate models.

**Dealing with Inconsistent Data** At first sight, the task of understanding opinions from comparison data might appear to be easy. And indeed it would be, if observed choices were a perfect reflection of a single set of opinions. However, when we start looking at data collected "in the wild", it becomes quickly apparent that comparison outcomes are not always consistent with each other: faced with the same alternatives, we sometimes appear to be making different choices. This is due to a multitude of factors: For example, ($a$) parts of the context in which the choice is made might not be observed, yet they might significantly influence the outcome; ($b$) if we try to summarize collective preferences based on individual choices, we can obviously expect some level of disagreement among individuals, even if some trends are shared; and ($c$) errors sometimes creep into the data, due to erroneous measurements or imperfect interpretations. A premise of this thesis is that these inconsistencies are unavoidable. But they can be dealt with in a principled way, using a probabilistic model. In a nutshell, this approach states that, given a set of alternatives, *any* comparison outcome is possible, but some outcomes are more likely than others, depending on the underlying preferences. The task is then reduced to finding preferences that explain the observations well. This approach has been dominant in the field and is the one that we adopt in this thesis. It will be explained further in Section 1.2.

**Modern Applications** Choice models have a long and rich history, but there has recently been a resurgence of interest in the context of large-scale online data collection. Indeed, the Web makes it easy for organizations to reach users throughout the world and to record their interactions with the organization's services. Let us consider three examples.

- Commercial online service-providers have increasingly relied on recommender systems (i.e., systems that learn user preferences) in order to increase user engagement and drive up sales. Spotify and Netflix, two popular services that stream music and videos, respectively, learn preferences based on implicit observations about the users' choices (which songs or movies they listen to). Amazon, a large e-commerce site, suggests personalized recommendations based on users' previous purchases.

- Scientists have built online platforms that enable them to collect large amounts of comparison data in order to answer challenging psychological and sociological research questions. For example, the GIFGIF project[1] aims at understanding the emotional content of animated GIF images, by showing users a pair of images and asking them the question: "Which image better expresses [happiness, shame, ...]?" The Place Pulse project[2] seeks to understand how different city neighborhoods are perceived, by using similar pairwise comparison questions. In both cases, comparisons are a natural way to elicit feedback from users. These two projects have each collected millions of data points over thousands of objects, and they resulted in fascinating findings that were previously out of reach using traditional methods.

- Pairwise comparisons are at the heart of *wiki surveys*, a novel surveying method developed by Salganik and Levy [2015]. Wiki surveys attempt to bridge the gap between questionnaires, which scale well but do not enable new information to emerge, and interviews, which are expensive to conduct but can lead to serendipitous discoveries. For example, the administration of New York City has used this service to gather feedback on a sustainability plan. Users could either propose new ideas or answer comparison questions of the type "Which [of the following two ideas] do you think is better for creating a greener, greater New York City?" The service makes it possible to simultaneously elicit new ideas and prioritize existing ones. At the time of writing, 11 739 surveys were created on `http://www.allourideas.org/`, totaling 17.8 million votes over 631 682 ideas.

**Beyond Preferences: Applications to Sport** Finally, we note that the very same methods used to model human choices can also be used to address problems that might first appear to be conceptually very different. In this thesis, we consider the problem

[1] See: `http://www.gif.gf/`.
[2] See: `http://pulse.media.mit.edu/`.

of predicting the outcome of football matches given historical data. In football, two teams are compared against each other during a match, at the end of which one of them wins. Using our previous terminology, we can frame the teams as alternatives being compared, and the winner as the outcome of the comparison. It is interesting to note that, historically, the main models and ideas used in this thesis have been developing simultaneously in the context of analyzing human choices, as well as sports outcomes, as we will see in the next section.

## 1.2 Probabilistic Models of Choice

In this section, we introduce the statistical models and associated methods that will be used or referred to throughout this thesis. We take a historical perspective: the context in which these models and methods were developed is fascinating. Although this section only gives a brief overview of the developments, it contains pointers for the reader interested in more comprehensive information.

### 1.2.1 Thurstone's Model

In 1927, Thurstone published an article that is widely regarded as foundational in the field of probabilistic models of comparison outcomes [Thurstone, 1927a]. He was interested in the problem of measurement in psychology and developed a method that explains the responses of human subjects to comparisons between two stimuli among $N$. In order to capture the fact that the response to a stimulus can vary, Thurstone suggested modeling the perceived value of a stimulus $i$ during some experiment by means of a *random* variable $x_i \in \mathbf{R}$. The outcome of the comparison between stimuli $i$ and $j$ is given by comparing a realization of the corresponding two random variables, i.e., by the event $x_i > x_j$. He further postulated that these random variables follow a jointly Gaussian distribution $\boldsymbol{x} \sim \mathrm{N}(\boldsymbol{\theta}, \boldsymbol{\Sigma})$. Denoting by $i \succ j$ the event "$i$ wins over $j$",

$$\mathbf{P}\left[i \succ j\right] = \mathbf{P}\left[x_i > x_j\right] = \Phi\left(\frac{\theta_i - \theta_j}{\sqrt{\Sigma_{ii} + \Sigma_{jj} - 2\Sigma_{ij}}}\right),$$

where $\Phi(\cdot)$ is the cumulative density function of the standard normal distribution and the last equality is obtained by noting that $x_i - x_j \sim \mathrm{N}(\theta_i - \theta_j, \Sigma_{ii} + \Sigma_{jj} - 2\Sigma ij)$. Thurstone considered several variants of the model that place successively more restrictive assumptions on the covariance matrix $\boldsymbol{\Sigma}$. The variant that is perhaps most widely-used nowadays is obtained by setting $\boldsymbol{\Sigma} = \frac{1}{2}\boldsymbol{I}$. In this case,

$$\mathbf{P}\left[i \succ j\right] = \Phi(\theta_i - \theta_j). \tag{1.1}$$

The vector of $N$ parameters $\boldsymbol{\theta} = [\theta_1 \ \cdots \ \theta_N]^\top \in \mathbf{R}^N$ governs the probabilities of all $\binom{N}{2}$ possible pairwise comparisons. Intuitively, $\theta_i$ can be interpreted as the *score* of stimulus

$i$, and the probability of observing a comparison outcome for $i$ and $j$ that is consistent with the true order increases with the distance $\theta_i - \theta_j$. Note that as (1.1) only involves pairwise distances, the parameters $\boldsymbol{\theta}$ are identifiable only up to a constant. In order to resolve this ambiguity, the parameters are often chosen such that $\sum_i \theta_i = 0$.

Perhaps the first application that Thurstone had in mind relates to psychophysics. Imagine being given two balls and asked the question: "Which of these two balls is heavier"? Given a collection of observations of this sort (some of which are possibly inconsistent), model (1.1) could be used to embed the stimuli on a real-valued scale (compactly summarizing all the data) by means of estimating the parameters $\boldsymbol{\theta}$.

**Application to Social Values** Thurstone quickly realized that there was potential beyond psychophysics. The same year, he published a study in which the method is applied to social values [Thurstone, 1927b]. This study seeks to understand the seriousness of 19 different criminal offenses in the United States, including crimes such as *bootlegging*, *arson*, *seduction* and *homicide*. Subjects (266 undergraduate students) were instructed to answer a questionnaire containing pairwise comparison questions of the type: "Which crime is more serious, $i$ or $j$?" The study perfectly illustrates the advantage of eliciting feedback in the form of comparisons rather than absolute ratings. Arguably, it would have been very difficult for the subjects to give a numerical score to each crime in a consistent way: in absolute terms, many crimes are extremely serious, and only relative judgments can make nuances appear.

Based on the outcomes of comparisons and using (1.1), Thurstone used a least-squares procedure to estimate the parameters $\boldsymbol{\theta}$. This enabled the representation of crimes on a global scale from least to most serious, in a way that reflected the subjects' opinions. Using the data tabulated in his 1927b paper, we could replicate[3] the analysis he performed. Figure 1.1 displays the resulting scale.

**A Note on Inference** The first approaches to learning the parameter vector $\boldsymbol{\theta}$ from data relied on least-squares fitting [Thurstone, 1927b, Mosteller, 1951]. Nowadays, maximum-likelihood estimation is more common. Indeed, the likelihood (1.1) is log-concave in $\boldsymbol{\theta}$, and standard off-the-shelf convex solvers can be used to find the maximizer. It is also interesting to note that Thurstone's model lends itself particularly well to approximate Bayesian inference methods [Chu and Ghahramani, 2005a,c]. An important quantity in Bayesian inference is the *marginal likelihood*, typically of the form

$$\int_{\mathbf{R}^N} \mathbf{P}\left[i \succ j \mid \boldsymbol{\theta}\right] \mathrm{N}(\boldsymbol{\theta} \mid \boldsymbol{\mu}, \boldsymbol{S}) d\boldsymbol{\theta}.$$

[3]The method used to fit the parameters of the model differs slightly from that of Thurstone [1927b]. However, the differences with the original plot are almost not perceptible.

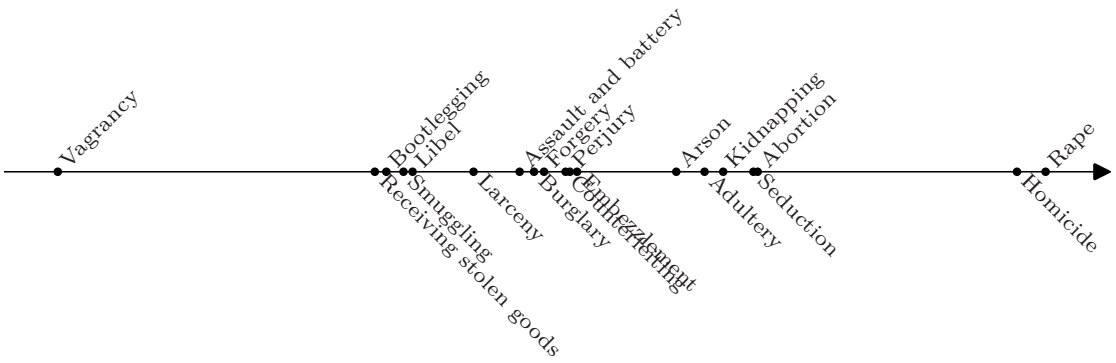



Figure 1.1 – Scale of seriousness of offenses (from left to right: least to most serious), using the data in Thurstone [1927b]. Nineteen crimes are embedded on a scale using Thurstone's model, based on 266 students' answers to 171 pairwise comparisons each.

In the case of Thurstone's model, this integral admits a simple closed-form solution. See, e.g., Rasmussen and Williams [2006, Section 3.9].

### 1.2.2 Bradley–Terry Model

Almost concurrently to Thurstone, Zermelo proposed (in German) a method for ranking chess players based on match outcomes [Zermelo, 1928]. He set out to address two problems: (*a*) coping with *unbalanced* tournaments, where players play an unequal number of games and against different sets of opponents, and (*b*) estimating the *relative strength* of players in a way that is predictive of future match outcomes. To this end, he introduced a probabilistic model of game outcomes. In his model, every player $i \in [N]$ is characterized by a latent strength parameter $\gamma_i \in \mathbf{R}_{>0}$. The probability of player $i$ winning against player $j$ is a function of their relative strength:

$$\mathbf{P}\left[i \succ j\right] = \frac{\gamma_i}{\gamma_i + \gamma_j}. \tag{1.2}$$

Note that the parameters $\boldsymbol{\gamma}$ are identifiable only up to a multiplicative factor; for this reason, it is often assumed that $\sum_i \gamma_i = 1$. Zermelo suggested finding the parameters $\boldsymbol{\gamma}$ by maximizing their likelihood given the observed data, an idea that was very advanced at the time. He formulated a necessary and sufficient condition[4] for the existence of a unique maximum-likelihood estimate, developed an iterative algorithm[5] to find it and proved the algorithm's convergence. Overall, his treatment of the model is very thorough and complete; unfortunately it appears to have been largely ignored for about 50 years

[4]The maximum-likelihood estimate exists if and only if there is no way to partition all players into two disjoint non-empty subsets $A, B \subset [N]$, such that there is no player in $A$ that has won a match against a player in $B$. See also Theorem 2.1.

[5]Interestingly, the same algorithm was later rediscovered multiple times in different contexts [Bradley and Terry, 1952, Ford, 1957, Dykstra, 1960, Hastie and Tibshirani, 1998, Hunter, 2004, Caron and Doucet, 2012].

[David, 1988]. See Glickman [2013] for a compelling introduction to Zermelo's paper. Finally, note that the chess rating system presently in use by the World Chess Federation is directly based on Zermelo's model [Elo, 1978].

**Relation to Thurstone's Model** Over two decades later, Bradley and Terry [1952], apparently unaware of Zermelo's work, rediscovered the model in the context of the rank analysis of experiments based on pairwise comparisons, linking the model back to the analysis of human opinions. The connection to Thurstone's model began becoming clear in Bradley [1953], where Bradley shows that by setting $\theta_i = \log \gamma_i$ for all $i$, the probability (1.2) can be rewritten as

$$\mathbf{P}\left[i \succ j\right] = \frac{1}{1 + \exp[-(\theta_i - \theta_j)]}. \tag{1.3}$$

Hence, the Bradley–Terry model (as it is commonly referred to) is another instance of a generalized linear model [Agresti, 2015] for pairwise comparisons: the outcome probability depends on the distance $\theta_i - \theta_j$ between the two parameters corresponding to the score of the alternatives. Yellot [1977] further expanded the connection, by showing that $\mathbf{P}\left[i \succ j\right]$ in (1.3) can be rewritten as $\mathbf{P}\left[x_i > x_j\right]$ for independent random variables $\{x_k : k \in [N]\}$ such that $x_k \sim \text{Gumbel}(\theta_k, 1)$, that is, $\mathbf{P}\left[x_k \leq y\right] = \exp\{-\exp[-(y - \theta_k)]\}$. Outcomes can therefore also be thought of as the comparison of the realizations of two random variables centered around the alternatives' scores, similarly to Thurstone's model, which gave rise to a *random utility* interpretation. Finally, Stern [1992] showed that both Thurstone's and the Bradley–Terry model can be unified under a more general model using the gamma distribution. In practice, both models give quantitatively similar results in most cases [Tsukida and Gupta, 2011]. Figure 1.2 illustrates the probabilities (1.1) and (1.3) as a function of $\theta_i - \theta_j$.

### 1.2.3 Luce's Choice Axiom

The two models discussed above are limited to comparisons between *pairs* of items. How to generalize these models to *multiway* comparisons? Given a set of alternatives $\mathcal{A} \subseteq [N]$ and an item $i \in \mathcal{A}$, denote by $i \succeq \mathcal{A}$ the event "$i$ is chosen among alternatives $\mathcal{A}$". A natural way to extend the Bradley–Terry model (1.2) to choices among arbitrarily many alternatives is as follows:

$$\mathbf{P}\left[i \succeq \mathcal{A}\right] = \frac{\gamma_i}{\sum_{j \in \mathcal{A}} \gamma_j}. \tag{1.4}$$

Simply put, the probability of a choice is always proportional to the strength $\gamma_i$ of the item $i$, no matter what the set of alternatives is. This choice model is due to Luce [1959], who showed that it is closely related to the following property.

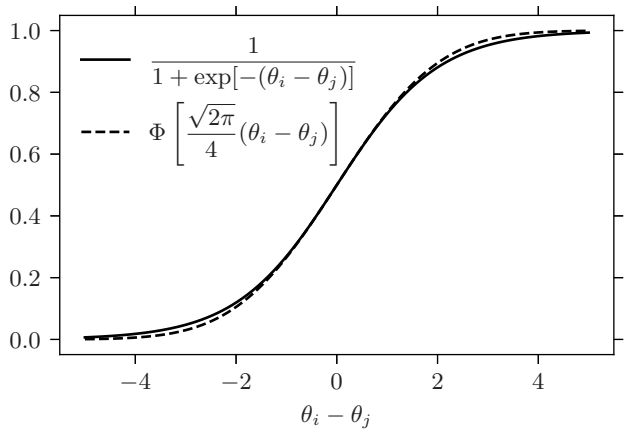



Figure 1.2 – Comparison of the Bradley–Terry model (solid line) and a rescaled version of Thurstone's model (dotted line). The scaling constant is chosen such that the slope at the origin matches.

**Definition** (independence of irrelevant alternatives)**.** A probabilistic choice model is said to satisfy the *independence from irrelevant alternatives* (IIA) property if for any $\mathcal{A} \subseteq [N]$ and any $i, j \in \mathcal{A}$,

$$\frac{\mathbf{P}\left[j \succeq \mathcal{A}\right]}{\mathbf{P}\left[i \succeq \mathcal{A}\right]} = \frac{\mathbf{P}\left[j \succ i\right]}{\mathbf{P}\left[i \succ j\right]}.$$

The IIA property is essentially equivalent[6] to Luce's *choice axiom* [1959, p. 6], and in this thesis we will refer to these two concepts interchangeably. Luce's fundamental contribution was to show that the IIA property enables an axiomatic characterization of the choice probabilities.

**Proposition 1.1** (Luce, 1959)**.** *A choice model satisfies the IIA property if and only if there exists a vector* $\boldsymbol{\gamma} \in \mathbf{R}_{>0}$ *such that the choice probabilities are given by* (1.4)*.*

*Proof.* It is trivial to verify that choice probabilities given by (1.4) satisfy IIA for any $\boldsymbol{\gamma}$. We will now show that the IIA property actually implies the existence of such a parametric representation of choice probabilities. Let $z \in [N]$ be an arbitrary alternative, and let

$$\gamma_i = \begin{cases} \mathbf{P}\left[i \succ z\right] / \mathbf{P}\left[z \succ i\right] & \text{if } i \neq z, \\ 1 & \text{otherwise.} \end{cases}$$

[6] Luce [1959] introduces the IIA property as a consequence of the choice axiom, which is slightly more general: its formulation permits $\mathbf{P}\left[i \succeq \mathcal{A}\right] = 0$, a technicality that we will not consider in this thesis.

Let $\mathcal{A} \subseteq [N]$ be any (non-empty) set of alternatives, and let $\mathcal{B} = \mathcal{A} \cup \{z\}$. By IIA,

$$\begin{aligned}
&\mathbf{P}\left[i \succeq \mathcal{B}\right] = \gamma_i \mathbf{P}\left[z \succeq \mathcal{B}\right] \qquad \forall i \in \mathcal{B} \\
&\quad \implies \frac{\mathbf{P}\left[j \succeq \mathcal{B}\right]}{\mathbf{P}\left[i \succeq \mathcal{B}\right]} = \frac{\gamma_j}{\gamma_i} = \frac{\mathbf{P}\left[j \succ i\right]}{\mathbf{P}\left[i \succ j\right]} = \frac{\mathbf{P}\left[j \succeq \mathcal{A}\right]}{\mathbf{P}\left[i \succeq \mathcal{A}\right]} \qquad \forall i, j \in \mathcal{A} \\
&\quad \implies \mathbf{P}\left[j \succeq \mathcal{A}\right] = \frac{\gamma_j}{\gamma_i} \mathbf{P}\left[i \succeq \mathcal{A}\right] \qquad \forall i, j \in \mathcal{A}.
\end{aligned}$$

Furthermore, as $\sum_{j \in \mathcal{A}} \mathbf{P}\left[j \succeq \mathcal{A}\right] = 1$, we have, for all $i \in \mathcal{A}$,

$$1 = \sum_{j \in \mathcal{A}} \frac{\gamma_j}{\gamma_i} \mathbf{P}\left[i \succeq \mathcal{A}\right] \implies \mathbf{P}\left[i \succeq \mathcal{A}\right] = \frac{\gamma_i}{\sum_{j \in \mathcal{A}} \gamma_j},$$

which concludes the proof. □

Independence of irrelevant alternatives is a powerful property, as it leads to a choice model that represents *combinatorially* many choice probabilities by using only $N$ parameters. This makes tractable the problem of learning the choice probabilities from a possibly small number of observations, but it inevitably restricts the expressivity of the model. In cases where some alternatives are very similar, IIA can turn out to be unrealistic, as shown by Debreu [1960] in a simple example. In the context of modern applications with a large number of items, which is the focus of this thesis, we believe that this trade-off is acceptable (and perhaps even necessary).

**Extension to Rankings** Numerous extensions of Luce's choice model have been proposed in the literature, enabling the analysis of observations beyond those of the type "one out of $K$". Perhaps one of the most widely-used extensions relates to ranking. Letting $i(r)$ be the item of rank $r$, Plackett [1975] suggested modeling the probability of a ranking on $K \leq N$ items as

$$\mathbf{P}\left[i(1) \succ i(2) \succ \cdots \succ i(K)\right] = \prod_{r=1}^{K-1} \frac{\gamma_{i(r)}}{\sum_{s=r}^{K} \gamma_{i(s)}},$$

for some $\boldsymbol{\gamma} \in \mathbf{R}_{>0}$. This can be seen as $K - 1$ independent choices made using Luce's model, iteratively, over the remaining alternatives. Therefore, this model is referred to as the *Plackett–Luce* model.

**Random Utility Models** Finally, we note that all the models discussed so far can be analyzed and generalized in the framework of *random utility models* [Train, 2009]. This enables, for example, an extension of Thurstone's model to multiway comparisons, in a similar way to Luce's extension of the Bradley–Terry model. In this framework, the

choice probabilities are defined by the probability of certain orderings of a collection of random variables (representing the utility of the alternatives).

## 1.3 Outline and Contributions

In this thesis, we address the problem of *efficiently* finding a ranking over a set of items (usually, by means of estimating choice model parameters). Efficiency is the guiding thread.

- As the size of datasets grows large, it becomes important to develop inference methods that are *computationally* efficient, without sacrificing their *statistical* efficiency, i.e., their accuracy.
- As the number of distinct items grows large, it becomes important to sample observations judiciously, such that the observations bring as much information as possible; we will refer to this as *data* efficiency.

In Chapter 2, we focus on algorithms for parameter inference and develop two procedures for models based on Luce's choice axiom. We do so by casting the inference problem as that of finding the stationary distribution of a Markov chain, an approach already suggested by Negahban et al. [2012] in the context of pairwise comparisons. Finding the stationary distribution of a Markov chain is a well-studied problem, and fast solvers are commonly available. We first show how the Markov chain can be derived from the likelihood function, a key insight that enables the generalization of Negahban et al.'s ideas to other models based on Luce's choice axiom. The first algorithm, LSR, finds a *spectral* estimate of model parameters by solving a homogeneous Markov chain: it is computationally very efficient and the estimate turns out to be more accurate than those obtained using competing methods with a similar running time. The second algorithm, I-LSR, finds the maximum-likelihood estimate (MLE) by solving a non-homogeneous Markov chain. The MLE is statistically more efficient than the spectral estimate but is also computationally more expensive. Even then, I-LSR turns out to be significantly faster than other commonly used algorithms for finding the MLE.

In Chapter 3, we shift our attention to the task of "intelligently" collecting pairwise comparison outcomes, based on the observed outcomes of previous comparisons. Supposing that we can adaptively choose which pair of items to query at every point in time, we seek to maximize the information obtained about the model (in particular, about the ranking of the $N$ items) in addition to minimizing the number of queries. In the machine-learning literature, this is known as the *active-learning* problem [Settles, 2012]. We start by analyzing Quicksort [Hoare, 1962], a popular sorting algorithm that computes a ranking when comparisons are always consistent with the true order. Under natural assumptions on the distribution of Bradley–Terry model parameters (that characterize the difficulty

of rankings), we show that Quicksort is remarkably resilient to inconsistent comparison outcomes. This leads to a practical and data-efficient sampling strategy that repeatedly runs a sorting algorithm until a given comparison budget is exhausted. With respect to competing active-learning strategies, our method achieves similar data-efficiency but is significantly less computationally expensive.

In Chapter 4 we consider a setting in which choices happen in a network, inspired by the work of Kumar et al. [2015]. We want to understand how users navigate in a network (e.g., which links they click on the Web), assuming that we have access to the aggregate traffic at each node in the network but not to the individual choices (i.e., the actual transitions). If transitions satisfy Luce's choice axiom, we show that the aggregate traffic is a sufficient statistic for the transition probabilities. Next, we develop an inference algorithm that ($a$) is robust to various ill-posed scenarios and ($b$) can be implemented efficiently. For example, the algorithm successfully scales to a snapshot of the WWW hyperlink graph containing billions of nodes. Finally, using real-world clickstream data, we demonstrate that our method is able to estimate transition probabilities well, despite the strong assumptions implied by Luce's axiom.

Lastly, in Chapter 5, we leave the realm of human opinions and switch over to an application in sports. In particular, we examine the problem of predicting the outcome of football matches between national teams. This problem is challenging, because national teams play only a few games every year, hence their strength is difficult to estimate based solely on the outcomes of the matches they play. Observing that most players in national teams play against each other in club competitions, we seek to take advantage of the (comparatively) large number of matches between clubs in order to improve the predictions. To this end, we embed all matches in a *player space* and use a kernel method to ensure that the model inference is computationally tractable. We evaluate the resulting prediction by using data from the last three European championships, and we find that those based on the joint model are more accurate than those based solely on the results between national teams.

# 2 Parameter Inference

In this chapter[1], we study the problem of inferring parameters of models derived from Luce's choice axiom. We begin by showing that the maximum-likelihood estimate (MLE) can be expressed as the stationary distribution of a Markov chain. This conveys insight into several recently proposed spectral inference algorithms. We then take advantage of this perspective and formulate a new spectral algorithm that generalizes and improves upon prior work. With a simple adaptation, this algorithm can be used iteratively, producing a sequence of estimates that converges to the MLE. The MLE version runs faster than competing approaches on a benchmark of five datasets.

## 2.1 Introduction

Markov chains have been used in recent work to aggregate inconsistent outcomes of pairwise comparisons and (partial) rankings [Dwork et al., 2001, Negahban et al., 2012, Azari Soufiani et al., 2013]. The idea is to build a Markov chain that is biased towards items that have won comparisons often, and to reduce the problem of ranking items to that of finding the *stationary distribution* of the chain (the ranking is then induced by the items' stationary probabilities). In this chapter, we highlight a connection between the MLE of models based on Luce's choice axiom and the stationary distribution of a Markov chain parametrized by the observed choices. By formalizing this link, we unify previous algorithms and explicate them from an ML inference perspective. Furthermore, the link suggests two new algorithms for parameter inference in Luce's general choice model. First, we develop a simple, consistent, and computationally efficient spectral algorithm that is applicable to a wide range of models derived from Luce's choice axiom. The exact formulation of the Markov chain used in the algorithm is distinct from related work [Negahban et al., 2012, Azari Soufiani et al., 2013] and achieves a significantly better statistical efficiency at no additional computational cost. Second, we observe that, with a small adjustment, the algorithm can be used iteratively, and it then converges

[1]This chapter is based on Maystre and Grossglauser [2015].

to the MLE. An evaluation on five real-world datasets reveals that it runs consistently faster than competing approaches and has a more predictable performance that does not depend on the structure of the data. The key step, finding a stationary distribution, can be offloaded to commonly available linear-algebra primitives, which makes our algorithms scale well. The method we propose is intuitively pleasing, simple to understand and implement, and it outperforms the state of the art. Therefore, we believe that it is highly useful to practitioners.

**Outline of the Chapter** We begin by introducing some notations and presenting a few useful facts about the MLE and about Markov chains. In Section 2.2, we discuss related work. In Section 2.3, we present our algorithms and, in Section 2.4, we evaluate them on synthetic and real-world data.

### 2.1.1 Maximum-Likelihood Estimate

Suppose that we collect $M$ independent choice observations in the multiset $\mathcal{D} = \{(c_m, \mathcal{A}_m) : m = 1, \dots, M\}$. Each observation consists of a choice $c_m$ among a set of alternatives $\mathcal{A}_m$; we say that $i$ *wins over* $j$ and denote by $i \succ j$ whenever $i, j \in \mathcal{A}_m$ and $c_m = i$. We postulate that the choices are generated from Luce's choice model and, for simplicity, we denote the model parameter associated with item $c_m$ by $\gamma_m$. From (1.4), it follows that the log-likelihood of parameters $\boldsymbol{\gamma}$ given observations $\mathcal{D}$ is given by

$$\ell(\boldsymbol{\gamma}) = \sum_{m=1}^{M} \Big[ \log \gamma_m - \log \sum_{j \in \mathcal{A}_m} \gamma_j \Big]. \tag{2.1}$$

In order to ensure that the parameters are likelihood-identifiable, we assume without loss of generality that $\sum_i \gamma_i = 1$. Next, we introduce a new object.

**Definition** (comparison graph)**.** The *comparison graph* $\mathcal{G}_\mathcal{D} = (\mathcal{V}, \mathcal{E})$ is a directed graph with $\mathcal{V} = [N]$ and $(j, i) \in \mathcal{E}$ if and only if $i$ wins at least once over $j$ in $\mathcal{D}$.

The existence and uniqueness of the MLE is completely determined by the connectivity of $\mathcal{G}_\mathcal{D}$, as the following well-known theorem shows.

**Theorem 2.1** (Zermelo, 1928, Ford, 1957, Hunter, 2004)**.** *The likelihood function* (2.1) *admits a unique maximizer* $\boldsymbol{\gamma}^\star \in \mathbf{R}^N_{>0}$ *such that* $\sum_i \gamma_i^\star = 1$ *if and only if* $\mathcal{G}_\mathcal{D}$ *is strongly connected.*

Throughout this chapter, we assume that $\mathcal{G}_\mathcal{D}$ is strongly connected. In practice, if this assumption does not hold, we can consider each strongly-connected component separately. Finally, note that even though $\ell(\boldsymbol{\gamma})$ admits a unique maximizer, it is not concave. However,

reparametrizing the model using $\theta_i \doteq \log \gamma_i$, the log-likelihood becomes

$$\ell(\boldsymbol{\theta}) = \sum_{m=1}^{M} \left[ \theta_m - \log \sum_{j \in \mathcal{A}_m} \exp \theta_j \right],$$

which is strictly concave in $\boldsymbol{\theta}$ (when $\mathcal{G}_{\mathcal{D}}$ is strongly connected). Furthermore, for all $i \in [N]$,

$$\frac{\partial \ell}{\partial \gamma_i} = \frac{\partial \ell}{\partial \theta_i} \cdot \frac{\partial \theta_i}{\partial \gamma_i} = \frac{\partial \ell}{\partial \theta_i} \cdot \frac{1}{\gamma_i} \quad \Longrightarrow \quad \frac{\partial \ell}{\partial \theta_i} = 0 \iff \frac{\partial \ell}{\partial \gamma_i} = 0.$$

As the strictly concave function $\ell(\boldsymbol{\theta})$ has a single stationary point (i.e., a single point where the gradient is zero), it follows that $\ell(\boldsymbol{\gamma})$ has a single stationary point at $\boldsymbol{\gamma}^\star$.

### 2.1.2 Markov Chains

We represent a finite, continuous-time Markov chain on $N$ states by a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = [N]$ and $\mathcal{E}$ is the set of transitions with positive rate[2]. If $\mathcal{G}$ is strongly connected, the Markov chain is said to be ergodic and admits a unique *stationary distribution* $\boldsymbol{\pi} \in \mathbf{R}^N_{>0}$, $\sum_i \pi_i = 1$. The *global balance equations* relate the transition rates $\{\lambda_{ij}\}$ to the stationary distribution as follows:

$$\sum_{j \neq i} \pi_i \lambda_{ij} = \sum_{j \neq i} \pi_j \lambda_{ji} \quad \forall i. \tag{2.2}$$

The stationary distribution is therefore invariant to changes in the time scale, i.e., to a rescaling of the transition rates. Given transition rates $\boldsymbol{\Lambda} = [\lambda_{ij}]$, finding the stationary distribution $\boldsymbol{\pi}$ can be implemented in several different ways. We distinguish implementations based on whether they consider a continuous-time or a discrete-time perspective on Markov chains.

**Continuous-Time Perspective** Let $\boldsymbol{Q}$ be the infinitesimal generator matrix of the Markov chain, i.e., $q_{ij} \doteq \lambda_{ij}$ and $q_{ii} \doteq -\sum_j \lambda_{ij}$. The stationary distribution satisfies $\boldsymbol{\pi}^\top \boldsymbol{Q} = \mathbf{0}$; this is simply a matrix formulation of the global balance equations (2.2). Therefore, one approach to finding the steady-state distribution is to compute the rank-1 left null space of $\boldsymbol{Q}$. This can be done, e.g., by LU decomposition, a basic linear-algebra primitive. In the case where $\boldsymbol{Q}$ is dense, the running time of a typical implementation is $O(N^3)$, but highly optimized parallel implementations such as that provided by LAPACK [Anderson et al., 1999] are commonly available. In the sparse case, LU decomposition can be done significantly faster using adapted algorithms, such as that of Demmel et al. [1999].

[2]Our exposition of Markov chains is succinct, and the interested reader is encouraged to consult Levin et al. [2008] for a more thorough exposition.

**Discrete-Time Perspective** Let $\epsilon < 1/\max_i |q_{ii}|$, then $\boldsymbol{P} = \boldsymbol{I} + \epsilon \boldsymbol{Q}$ is the transition matrix of a discrete-time Markov chain that satisfies $\boldsymbol{\pi}^\top \boldsymbol{P} = \boldsymbol{\pi}^\top$. In this case, finding the steady-state distribution is equivalent to finding the left eigenvector associated with the leading eigenvalue of the transition matrix $\boldsymbol{P}$. This is also a well-studied linear algebra problem for which plenty of efficient, off-the-shelf algorithms exist. For example, power iteration methods can find the eigenvector in a few (sparse) matrix multiplications. Beyond these well-known algorithms, recently proposed randomized approaches such as that of Halko et al. [2011] make it possible to scale to very-large problem sizes ($N \sim 10^6$ or more).

Both the continuous-time and the discrete-time perspectives yield exactly the same resulting stationary distribution, and the algorithms presented in this chapter are oblivious to this choice.

## 2.2 Related Work

Spectral methods applied to ranking and scoring items from noisy choices have a long-standing history. To the best of our knowledge, Saaty [1980] was the first to suggest using the leading eigenvector of a matrix of inconsistent pairwise judgments to score alternatives. Two decades later, Page et al. [1998] developed PageRank, an algorithm that ranks Web pages according to the stationary distribution of a random walk on the hyperlink graph. In the same vein, Dwork et al. [2001] proposed several variants of Markov chains for aggregating heterogeneous rankings. Their idea was to construct a random walk that is biased towards high-ranked items, and use the ranking induced by the stationary distribution. More recently, Negahban et al. [2012] presented Rank Centrality, an algorithm for aggregating pairwise comparisons close in spirit to that of Dwork et al. [2001]. When the data are generated under the Bradley–Terry model, this algorithm asymptotically recovers model parameters with only $\omega(N \log N)$ pairwise comparisons (when comparison pairs are chosen uniformly at random). For the more general case of rankings under the Plackett–Luce model, Azari Soufiani et al. [2013] propose to break rankings into pairwise comparisons and to apply an algorithm similar to Rank Centrality. The authors show that the resulting estimator is statistically consistent. Lastly, Fogel et al. [2014] take a seriation approach to ranking from pairwise comparisons and develop a different type of spectral algorithm. Interestingly, many of these spectral algorithms can be related to the method of moments, a broadly applicable alternative to maximum-likelihood estimation [Casella and Berger, 2002, Section 7.2.1].

The history of algorithms for maximum-likelihood inference under Luce's model goes back even further. In the special case of pairwise comparisons, the same iterative algorithm was independently discovered by Zermelo [1928], Ford [1957] and Dykstra [1960]. Much later, this algorithm was explained by Hunter [2004] as an instance of minorization-maximization (MM) algorithm and extended to the more general choice model. Today, Hunter's MM

algorithm is the *de facto* standard for ML inference in Luce's model. As the likelihood can be written as a concave function, off-the-shelf optimization procedures such as the Newton-Raphson method can also be used, although they have been reported to be slower and less practical [Hunter, 2004]. Recently, Kumar et al. [2015] looked at the problem of finding the transition matrix of a Markov chain, given its stationary distribution. The problem of inferring Luce's model parameters from data can be reformulated in their framework, and the MLE is the solution to the inversion of the stationary distribution. Their work stands out as the first to link ML inference to Markov chains, albeit very differently from the way presented in this chapter.

Beyond algorithms, properties of the maximum-likelihood estimator in this model were studied extensively. Hajek et al. [2014] consider the Plackett–Luce model for $K$-way rankings. They give an upper bound to the estimation error and show that the MLE is minimax-optimal. In summary, they show that only $\omega(N/K \log N)$ samples are enough to drive, as $N$ increases, the mean-square error down to zero. Rajkumar and Agarwal [2014] consider the Bradley–Terry model for pairwise comparisons. They show that the ML estimator is able to recover the correct ranking, even when the data are generated as per another model, e.g., Thurstone's [Thurstone, 1927b], as long as a so-called *low-noise* condition is satisfied. Some authors also propose Bayesian inference methods as an alternative to likelihood maximization. Caron and Doucet [2012] present a Gibbs sampler, and Guiver and Snelson [2009] present an approximate inference algorithm based on expectation propagation.

We provide a unifying perspective on recent advances in spectral algorithms [Negahban et al., 2012, Azari Soufiani et al., 2013] from a maximum-likelihood estimation perspective. It turns out that this perspective enables us to make contributions on both sides: We develop an improved and more general spectral ranking algorithm, and we propose a faster procedure for ML inference by using this algorithm iteratively.

## 2.3 Algorithms

We begin by expressing the MLE under the choice model as the stationary distribution of a Markov chain. We then take advantage of this formulation to propose novel algorithms for model inference. Although our derivation is made in the general choice model, we also discuss implications for the special cases of pairwise data in Section 2.3.3, $K$-way ranking data in Section 2.3.4, and pairwise comparisons with ties in Section 2.3.5.

### 2.3.1 MLE as a Stationary Distribution

For each item $i \in [N]$, we define two sets of indices. Let $\mathcal{W}_i \doteq \{m : i \in \mathcal{A}_m, c_m = i\}$ and $\mathcal{L}_i \doteq \{m : i \in \mathcal{A}_m, c_m \neq i\}$ be the indices of the observations where item $i$ wins over and

loses against the alternatives, respectively. We start from the log-likelihood $\ell(\boldsymbol{\gamma})$ in (2.1); the optimality condition $\nabla\ell(\boldsymbol{\gamma}^\star) = \mathbf{0}$ implies

$$\frac{\partial\ell(\boldsymbol{\gamma})}{\partial\gamma_i}\Big|_{\boldsymbol{\gamma}=\boldsymbol{\gamma}^\star} = \sum_{m\in W_i}\left[\frac{1}{\gamma_i^\star} - \frac{1}{\sum_{j\in\mathcal{A}_m}\gamma_j^\star}\right] - \sum_{m\in\mathcal{L}_i}\frac{1}{\sum_{j\in\mathcal{A}_m}\gamma_j^\star} = 0 \quad \forall i \tag{2.3}$$

$$\iff \sum_{j\neq i}\left[\sum_{m\in\mathcal{W}_i\cap\mathcal{L}_j}\frac{\gamma_j^\star}{\sum_{k\in\mathcal{A}_m}\gamma_k^\star} - \sum_{m\in\mathcal{W}_j\cap\mathcal{L}_i}\frac{\gamma_i^\star}{\sum_{k\in\mathcal{A}_m}\gamma_k^\star}\right] = 0 \quad \forall i. \tag{2.4}$$

In order to go from (2.3) to (2.4), we multiply by $\gamma_i^\star$ and rearrange the terms. To simplify the notation, let us further introduce the function

$$f(\mathcal{S},\boldsymbol{\gamma}) \doteq \sum_{\mathcal{A}\in\mathcal{S}}\frac{1}{\sum_{i\in\mathcal{A}}\gamma_i},$$

which takes observations $\mathcal{S}\subseteq\mathcal{D}$ and an instance of model parameters $\boldsymbol{\gamma}$, and returns a non-negative real number. Let $\mathcal{D}_{i\succ j} \doteq \{(c_m,\mathcal{A}_m)\in\mathcal{D} : m\in\mathcal{W}_i\cap\mathcal{L}_j\}$, i.e., the set of observations where $i$ wins over $j$. Then (2.4) can be rewritten as

$$\sum_{j\neq i}\gamma_i^\star\cdot f(\mathcal{D}_{j\succ i},\boldsymbol{\gamma}^\star) = \sum_{j\neq i}\gamma_j^\star\cdot f(\mathcal{D}_{i\succ j},\boldsymbol{\gamma}^\star) \quad \forall i. \tag{2.5}$$

This formulation conveys a new viewpoint on the MLE. It is easy to recognize the global balance equations (2.2) of a Markov chain on $N$ states (representing the items), with transition rates $\lambda_{ji} = f(\mathcal{D}_{i\succ j},\boldsymbol{\gamma}^\star)$ and stationary distribution $\boldsymbol{\gamma}^\star$. These transition rates have an interesting interpretation: $f(\mathcal{D}_{i\succ j},\boldsymbol{\gamma})$ is the count of how many times $i$ wins over $j$, weighted by the strength of the alternatives. At this point, it is useful to observe that for any parameters $\boldsymbol{\gamma}$, $f(\mathcal{D}_{i\succ j},\boldsymbol{\gamma}) > 0$ if and only if $(j,i)\in\mathcal{E}$. Combined with the assumption that $\mathcal{G}$ is strongly connected, it follows that any $\boldsymbol{\gamma}$ parametrizes the transition rates of an ergodic (homogeneous) Markov chain. The ergodicity of the inhomogeneous Markov chain, where the transition rates are constantly updated to reflect the current distribution over states, is shown by the following theorem.

**Theorem 2.2.** *The Markov chain with inhomogeneous transition rates $\lambda_{ji} = f(\mathcal{D}_{i\succ j},\boldsymbol{\gamma})$ converges to the maximum-likelihood estimate $\boldsymbol{\gamma}^\star$, for any initial distribution in the open probability simplex.*

*Proof.* Let $\boldsymbol{Q}(\boldsymbol{\gamma})$ be the infinitesimal generator matrix of the Markov chain $\boldsymbol{\gamma}(t)$. The dynamics of the Markov chain are described by the differential equation

$$\frac{d\boldsymbol{\gamma}^\top}{dt} = \boldsymbol{\gamma}^\top\boldsymbol{Q}(\boldsymbol{\gamma}). \tag{2.6}$$

By construction, the invariant distributions of the Markov chain coincide with the maximizers of the log-likelihood (2.1). Hence, we know that $\boldsymbol{\gamma}^\star$ is the unique equilibrium point for (2.6), i.e., satisfying $\boldsymbol{\gamma}^\top\boldsymbol{Q}(\boldsymbol{\gamma}) = \mathbf{0}$. We will now show that this point is globally

**Algorithm 2.1** Luce Spectral Ranking.

**Require:** observations $\mathcal{D}$
$\boldsymbol{\Lambda} \leftarrow \mathbf{0}_{N\times N}$
**for** $(i, \mathcal{A}) \in \mathcal{D}$ **do**
  **for** $j \in \mathcal{A} \setminus \{i\}$ **do**
    $\lambda_{ji} \leftarrow \lambda_{ji} + N/|\mathcal{A}|$
  **end for**
**end for**
**return** stationary distribution of Markov chain with transition rates $\boldsymbol{\Lambda}$

and asymptotically stable, i.e., $\boldsymbol{\gamma}(t) \to \boldsymbol{\gamma}^\star$ as $t \to \infty$ for any $\boldsymbol{\gamma}(0)$ in the open probability simplex. To this end, it suffices to show that $V(\boldsymbol{\gamma}) = -\ell(\boldsymbol{\gamma}) + \ell(\boldsymbol{\gamma}^\star)$ is a Lyapunov function for the dynamical system (2.6). First, we have that $V(\boldsymbol{\gamma}^\star) = 0$ and $V(\boldsymbol{\gamma}) > 0$ for all $\boldsymbol{\gamma} \neq \boldsymbol{\gamma}^\star$ (by definition of the MLE). Second, we note that $\boldsymbol{\gamma}^\top \boldsymbol{Q}(\boldsymbol{\gamma}) = \operatorname{diag}(\boldsymbol{\gamma})\, \nabla\ell(\boldsymbol{\gamma})$. Hence,

$$\frac{dV}{dt} = (\nabla V)^\top \frac{d\boldsymbol{\gamma}}{dt} = -[\nabla\ell(\boldsymbol{\gamma})]^\top \operatorname{diag}(\boldsymbol{\gamma})\, \nabla\ell(\boldsymbol{\gamma}) < 0,$$

for all $\boldsymbol{\gamma} \neq \boldsymbol{\gamma}^\star$. Third, $\ell(\boldsymbol{\gamma})$ grows unboundedly as $\boldsymbol{\gamma}$ approaches the boundary of the probability simplex [Hunter, 2004, Lemma 1] and therefore $V(\boldsymbol{\gamma})$ does so as well. The result then follows by applying the Barbashin-Krasovskii theorem, a standard result found, e.g., in Khalil [1996, Chapter 3]. □

### 2.3.2 Approximate and Exact ML Inference

We approximate the Markov chain described in (2.5) by considering a priori that all alternatives have equal strength. That is, we set the transition rates $\lambda_{ji} \doteq f(\mathcal{D}_{i\succ j}, \boldsymbol{\gamma})$ by fixing $\boldsymbol{\gamma}$ to $[1/N \ \cdots \ 1/N]^\top$. For $i \neq j$, the contribution of $i$ winning over $j$ to the rate of transition $\lambda_{ji}$ is $N/|\mathcal{A}|$. In other words, for each observation, the winning item is rewarded by a fixed amount of incoming rate that is evenly split across the alternatives (the chunk allocated to itself is discarded). We interpret the stationary distribution $\bar{\boldsymbol{\gamma}}$ as an estimate of model parameters. Algorithm 2.1 summarizes this procedure, called *Luce Spectral Ranking* (LSR). If we consider a growing number of observations, LSR converges to the true model parameters $\boldsymbol{\gamma}'$, even in the restrictive case where the sets of alternatives are fixed.

**Theorem 2.3.** *Let $\mathcal{U} = \{\mathcal{A}_n\}$ be a collection of sets of alternatives such that for any partition of $\mathcal{U}$ into two non-empty sets $\mathcal{S}$ and $\mathcal{T}$, $(\cup_{\mathcal{A}\in\mathcal{S}}\mathcal{A}) \cap (\cup_{\mathcal{A}\in\mathcal{T}}\mathcal{A}) \neq \varnothing$. Let $M_n$ be the number of choices observed over alternatives $\mathcal{A}_n$. Then $\bar{\boldsymbol{\gamma}} \to \boldsymbol{\gamma}'$ as $M_n \to \infty$ $\forall n$.*

*Proof.* Let $M \to \infty$ be a shorthand for $M_n \to \infty$ $\forall n$. The condition on $\mathcal{U}$ is equivalent to stating that the hypergraph $H = (\mathcal{V}, \mathcal{U})$, with $\mathcal{V} = [N]$, is connected. It implies that,

asymptotically, the comparison graph $\mathcal{G}_\mathcal{D}$ is strongly connected. Indeed, for a given set of alternatives $\mathcal{A}_n$, let $i, j \in \mathcal{A}_n$. The probability that $(j, i) \in \mathcal{E}$ is

$$1 - \left(1 - \frac{\gamma'_i}{\sum_{k\in\mathcal{A}_n} \gamma'_k}\right)^{M_n} > 1 - (1 - \gamma'_i)^{M_n} \xrightarrow{M_n\to\infty} 1,$$

where we use the fact that $\gamma'_i > 0$ for all $i$. Therefore, asymptotically, every alternative set $\mathcal{A}_n$ forms a clique in $\mathcal{G}_\mathcal{D}$. By assumption of connectivity on the hypergraph $\mathcal{H}$, the comparison graph is strongly connected.

Now that we know that the Markov chain is ergodic, we will show that the stationary distribution matches the true model parameters. Let $c_m^n$ be a random variable denoting the item chosen in the $m$-th observation over alternatives $\mathcal{A}_n$, and let $1_{\{\mathcal{X}\}}$ be the indicator variable for the event $\mathcal{X}$. By the law of large numbers, for any item $i \in \mathcal{A}_n$,

$$\lim_{M_n\to\infty} \frac{1}{M_n} \sum_{m=1}^{M_n} 1_{\{c_m^n = i\}} = \frac{\gamma'_i}{\sum_{k\in A_n} \gamma'_k}. \tag{2.7}$$

Now consider two items $i$ and $j$. If they have never been compared, $\lambda_{ij} = \lambda_{ji} = 0$. Otherwise, suppose that they have been compared in alternative sets whose indices are in $\mathcal{B} = \{n : i, j \in \mathcal{A}_n\}$. By construction of the transition rates in LSR, we have that

$$\frac{\lambda_{ij}}{\lambda_{ji}} = \frac{\sum_{n\in\mathcal{B}} \sum_{m=1}^{M_n} 1_{\{c_m^n = j\}}\ N/|\mathcal{A}_n|}{\sum_{n\in\mathcal{B}} \sum_{m=1}^{M_n} 1_{\{c_m^n = i\}}\ N/|\mathcal{A}_n|}.$$

From (2.7) it follows that

$$\lim_{M\to\infty} \frac{\lambda_{ij}}{\lambda_{ji}} = \frac{\sum_{n\in\mathcal{B}} (\gamma'_j / \sum_{k\in A_n} \gamma'_k)\ N/|\mathcal{A}_n|}{\sum_{n\in\mathcal{B}} (\gamma'_i / \sum_{k\in A_n} \gamma'_k)\ N/|\mathcal{A}_n|} = \frac{\gamma'_j}{\gamma'_i}.$$

Therefore, when $M \to \infty$,

$$\sum_{j\neq i} \gamma'_i \lambda_{ij} = \sum_{j\neq i} \gamma'_i \left(\frac{\gamma'_j}{\gamma'_i} \lambda_{ji}\right) = \sum_{j\neq i} \gamma'_j \lambda_{ji} \quad \forall i.$$

We recognize the global balance equations (2.2), and it follows that $\boldsymbol{\gamma}'$ is the stationary distribution of the Markov chain. □

Starting from the LSR estimate, we can iteratively refine the transition rates of the Markov chain and obtain a sequence of estimates. By (2.5), the only fixed point of this iteration is the MLE $\boldsymbol{\gamma}^\star$. We call this procedure I-LSR and describe it in Algorithm 2.2.

LSR (or one iteration of I-LSR) entails (*a*) filling a matrix of (weighted) pairwise counts and (*b*) finding a stationary distribution. Let $D \doteq \sum_m |\mathcal{A}_m|$, and let $S$ be the running

**Algorithm 2.2** Iterative Luce Spectral Ranking.

**Require:** observations $\mathcal{D}$
$\boldsymbol{\gamma} \leftarrow [1/N \ \cdots \ 1/N]^\top$
**repeat**
  $\boldsymbol{\Lambda} \leftarrow \mathbf{0}_{N \times N}$
  **for** $(i, \mathcal{A}) \in \mathcal{D}$ **do**
    **for** $j \in \mathcal{A} \setminus \{i\}$ **do**
      $\lambda_{ji} \leftarrow \lambda_{ji} + 1/\sum_{k \in \mathcal{A}} \gamma_k$
    **end for**
  **end for**
  $\boldsymbol{\gamma} \leftarrow$ stationary distribution of Markov chain with transition rates $\boldsymbol{\Lambda}$
**until** convergence

time of finding the stationary distribution. Then LSR has running time $O(D + S)$. As a comparison, one iteration of the MM algorithm [Hunter, 2004] is $O(D)$. Finding the stationary distribution can be implemented in different ways. For example, in a sparse regime where $D \ll N^2$, the stationary distribution can be found with the power method in a few $O(D)$ sparse matrix multiplications. In practice, it is not clear whether $D$ or $S$ turns out to be dominant in the running time.

### 2.3.3 Bradley–Terry Model

A widely-used special case of Luce's choice model occurs when all sets of alternatives contain exactly two items, i.e., when the data consist of pairwise comparisons. This model was proposed by Zermelo [1928] and later by Bradley and Terry [1952]. As the stationary distribution is invariant to changes in the time scale, we can rescale the transition rates and set $\lambda_{ji} \doteq |\mathcal{D}_{i \succ j}|$ when using LSR on pairwise data. Let $\mathcal{S}$ be the set containing the pairs of items that are compared at least once. In the case where each pair $(i, j) \in \mathcal{S}$ are compared exactly $C$ times, LSR is strictly equivalent to a continuous-time Markov-chain formulation of Rank Centrality [Negahban et al., 2012]. In fact, our derivation justifies Rank Centrality as an approximate ML inference algorithm for the Bradley–Terry model. Furthermore, we provide a principled extension of Rank Centrality to the case where the number of observed comparisons is unbalanced. Rank Centrality considers transition rates proportional to the *ratio* of wins, whereas (2.5) justifies making transition rates proportional to the *count* of wins.

Negahban et al. [2012] also provide an upper bound on the error rate of Rank Centrality, which essentially shows that the error rate is minimax-optimal. Because the two estimators are equivalent in the setting of balanced pairwise comparisons, the bound also applies to LSR.

### 2.3.4 Plackett–Luce Model

Another case of interest is when observations do not consist of only a single choice, but of a ranking over the alternatives. We now suppose that we have a dataset of $M$ observations consisting of $K$-way rankings, $2 \leq K \leq N$. For conciseness and without loss of generality, we suppose that $K$ is the same for all observations. Let one such observation be $i(1) \succ \cdots \succ i(K)$, where $i(r)$ is the item with $r$-th rank. The Plackett–Luce model (c.f. Section 1.2.3) posits

$$\mathbf{P}\left[i(1) \succ \cdots \succ i(K)\right] = \prod_{r=1}^{K} \frac{\gamma_{i(r)}}{\sum_{s=r}^{K} \gamma_{i(s)}}.$$

A ranking can thus be interpreted as a sequence of $K-1$ independent choices: choose the first item, then choose the second among the remaining alternatives, etc. With this point of view in mind, LSR and I-LSR can easily accommodate data consisting of $K$-way rankings, by decomposing the $M$ observations into $M' = M(K-1)$ choices.

Azari Soufiani et al. [2013] provide a class of consistent estimators for the Plackett–Luce model, using the idea of breaking rankings into pairwise comparisons. Although they explain their algorithms from a generalized-method-of-moments perspective, it is straightforward to reinterpret their estimators as stationary distributions of particular Markov chains. In fact, for $K = 2$, their algorithm GMM-F is identical to LSR. When $K > 2$ however, breaking a ranking into $\binom{K}{2}$ pairwise comparisons implicitly makes the (incorrect) assumption that these comparisons are statistically independent. The Markov chain that LSR builds breaks rankings into pairwise rate contributions, but weights the contributions differently depending on the rank of the winning item. In Section 2.4, we show that this weighting turns out to be crucial. Our approach yields a significant improvement in statistical efficiency yet keeps the same attractive computational cost and ease of use.

### 2.3.5 Rao–Kupper Model

The link between the MLE and the stationary distribution of a Markov chain seemingly applies to other variants and extensions of Luce's choice model. For an illustration, we consider the model proposed by Rao and Kupper [1967], which extends the Bradley–Terry model to the case where a comparison between two items can result in a tie. This model is useful, e.g., for chess, where a significant fraction of comparison outcomes do not result in either a win or a loss. Letting $\alpha \in [1, \infty)$, the probabilities of $i$ winning over and tying

with $j$, respectively, are given by

$$p(i \succ j) = \frac{\gamma_i}{\gamma_i + \alpha\gamma_j},$$
$$p(i \equiv j) = \frac{\gamma_i\gamma_j(\alpha^2 - 1)}{(\gamma_i + \alpha\gamma_j)(\alpha\gamma_i + \gamma_j)}.$$

Informally, the parameter $\alpha$ controls the expected probability of observing a tie in the comparison of two items of equal strength. We assume that $\alpha$ is fixed, and derive an expression of the MLE $\boldsymbol{\gamma}^\star$. Let $a_{ji}$ be the number of times $i$ wins over $j$, and $t_{ij} = t_{ji}$ be the number of ties between $i$ and $j$. The log-likelihood can be written as

$$\begin{aligned}\ell(\boldsymbol{\gamma}) = &\sum_i \sum_{j\neq i} a_{ji} \left[\log\gamma_i - \log(\gamma_i + \alpha\gamma_j)\right] \\ &+ \sum_i \sum_{j>i} t_{ij} \Big[\log\gamma_i + \log\gamma_j + \log(\alpha^2 - 1) - \log(\gamma_i + \alpha\gamma_j) - \log(\alpha\gamma_i + \gamma_j)\Big].\end{aligned}$$

This function admits a unique MLE $\boldsymbol{\gamma}^\star$, and the optimality condition $\nabla\ell(\boldsymbol{\gamma}^\star) = \mathbf{0}$ implies

$$\begin{aligned}\frac{\partial\ell(\boldsymbol{\gamma})}{\partial\gamma_i}\Big|_{\gamma=\gamma^\star} &= \sum_{j\neq i} \left[a_{ji}\left(\frac{1}{\gamma_i^\star} - \frac{1}{\gamma_i^\star + \alpha\gamma_j^\star}\right) - a_{ij}\frac{\alpha}{\alpha\gamma_i^\star + \gamma_j^\star}\right. \\ &\qquad \left. + t_{ij}\left(\frac{1}{\gamma_i^\star} - \frac{1}{\gamma_i^\star + \alpha\gamma_j^\star} - \frac{\alpha}{\alpha\gamma_i^\star + \gamma_j^\star}\right)\right] = 0 \\ &\iff \sum_{j\neq i} \left[a_{ji}\frac{\alpha\gamma_j^\star}{\gamma_i^\star + \alpha\gamma_j^\star} - a_{ij}\frac{\alpha\gamma_i^\star}{\alpha\gamma_i^\star + \gamma_j^\star} + t_{ij}\frac{\alpha(\gamma_j^\star)^2 - \alpha(\gamma_i^\star)^2}{(\gamma_i^\star + \alpha\gamma_j^\star)(\alpha\gamma_i^\star + \gamma_j^\star)}\right] = 0 \\ &\iff \sum_{j\neq i} \left[\frac{a_{ji} + t_{ji}\frac{\gamma_j^\star}{\alpha\gamma_i^\star + \gamma_j^\star}}{\gamma_i^\star + \alpha\gamma_j^\star}\gamma_j^\star \;-\; \frac{a_{ij} + t_{ij}\frac{\gamma_i^\star}{\gamma_i^\star + \alpha\gamma_j^\star}}{\alpha\gamma_i^\star + \gamma_j^\star}\gamma_i^\star\right] = 0.\end{aligned}$$

Therefore, the MLE can be interpreted as the stationary distribution of a Markov chain with transition rates

$$\lambda_{ij} = \frac{a_{ij} + t_{ij}\frac{\gamma_i^\star}{\gamma_i^\star + \alpha\gamma_j^\star}}{\alpha\gamma_i^\star + \gamma_j^\star}.$$

Given these transition rates, the extension of Algorithms 2.1 and 2.2 is straightforward. For example, for LSR, the transition rates simplify to $\lambda_{ij} \propto a_{ij} + t_{ij}(1 + \alpha)^{-1}$.

Beyond the Rao–Kupper model, we believe that our algorithms can be generalized to further models that are based on the choice axiom. However, this axiom is key, and other choice models (such as Thurstone's [1927a]) do not seem to admit the stationary-distribution interpretation we derive here.

## 2.4 Experimental Evaluation

In this section, we compare LSR and I-LSR to other inference algorithms, in terms of statistical efficiency and empirical performance. First, in order to measure the statistical efficiency of the estimators, we generate synthetic data from a known ground truth. Then, we look at five real-world datasets and investigate the practical performance of the algorithms in terms of accuracy, running time and convergence rate.

**Error Metric** As the probability of $i$ winning over $j$ depends on the ratio of strengths $\gamma_i/\gamma_j$, the strengths are typically logarithmically spaced. In order to evaluate the accuracy of an estimate $\boldsymbol{\gamma}$ to ground truth parameters $\boldsymbol{\gamma}'$, we therefore use a log transformation, reminiscent of the random-utility-theoretic formulation of the choice model. Define $\theta_i \doteq \log(\gamma_i) - \kappa$, with $\kappa$ chosen such that $\sum_i \theta_i = 0$, and let $\boldsymbol{\theta} = [\theta_i]$. We will consider the root-mean-squared error

$$E_{\mathrm{RMS}} = \|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_2/\sqrt{N}.$$

### 2.4.1 Statistical Efficiency

To assess the statistical efficiency of LSR and other algorithms, we follow the experimental procedure of Hajek et al. [2014]. We consider $N = 1024$ items, and draw $\boldsymbol{\theta}'$ uniformly at random in $[-2, 2]^N$. We generate $M = 64$ full rankings over the $N$ items from a Plackett–Luce model parametrized with $\boldsymbol{\gamma} \propto [e^{\theta_i}]$. For a given $K \in \{2^1, \ldots, 2^{10}\}$, we break down each of the full rankings as follows. First, we partition the items into $N/K$ subsets of size $K$ uniformly at random. Then, we store the $K$-way rankings induced by the full ranking on each of those subsets. As a result, we obtain $MN/K$ statistically independent $K$-way partial rankings. For a given estimator, these data produce an estimate $\boldsymbol{\theta}$, for which we record the root-mean-square error to $\boldsymbol{\theta}'$. We consider four estimators. The first two (LSR and ML) work on the ranking data directly. The remaining two follow Azari Soufiani et al. [2013] who suggest breaking down $K$-way rankings into $\binom{K}{2}$ pairwise comparisons. These comparisons are then used by LSR, resulting in Azari Soufiani et al.'s GMM-F estimator, and by an ML estimator (ML-F). In short, the four estimators vary according to $(a)$ whether they use as-is rankings or derived comparisons, and $(b)$ whether the model is fitted using an approximate spectral algorithm or using the exact MLE. Figure 2.1 plots $E_{\mathrm{RMS}}$ for increasing sizes of partial rankings, as well as a lower bound to the error of any estimator for the Plackett–Luce model (see Hajek et al. [2014] for details). We observe that breaking the rankings into pairwise comparisons (*-F estimators) incurs a significant efficiency loss over using the $K$-way rankings directly (LSR and ML). We conclude that by correctly weighting pairwise rates in the Markov chain, LSR distinctly outperforms the rank-breaking approach as $K$ increases. We also observe that the MLE is always more efficient. Spectral estimators such as LSR provide a computationally inexpensive,

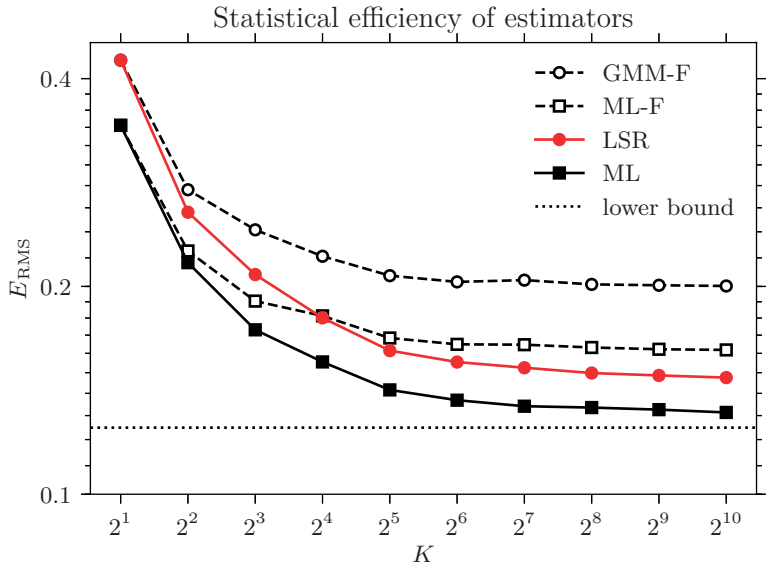



Figure 2.1 – Statistical efficiency of different estimators for increasing sizes of partial rankings. As $K$ grows, breaking rankings into pairwise comparisons becomes increasingly inefficient. LSR remains efficient at no additional computational cost.

asymptotically consistent estimate of parameters, but this observation justifies calling them *approximate* inference algorithms.

### 2.4.2 Empirical Performance

We investigate the performance of various inference algorithms on five real-world datasets. The NASCAR [Hunter, 2004] and sushi [Kamishima and Akaho, 2009] datasets contain multiway partial rankings. The YouTube[3], GIFGIF[4] and chess[5] datasets contain pairwise comparisons. Among these, the chess dataset is particular in that it features 45% of ties; in this case, we use the model proposed by Rao and Kupper [1967]. We preprocess each dataset by discarding items that are not part of the largest strongly connected component in the comparison graph, in order to ensure that the MLE is well-defined. For each dataset, the number of items $N$, the number of rankings $M$, as well as the size $K$ of a partial ranking after preprocessing are given in Table 2.1.

**Experimental Procedure** We run all experiments on a machine with a quad-core 2.0 GHz Haswell processor and 16 GB of RAM, running Mac OS X 10.9. For LSR and I-LSR, we use a slightly adapted version the Python code presented in Listing 2.1, which calls a

[3]See: `https://archive.ics.uci.edu/ml/machine-learning-databases/00223/`.
[4]See: `http://lucas.maystre.ch/gifgif-data`.
[5]See: `https://www.kaggle.com/c/chess`.

dense LU factorization routine. We implement the Rank Centrality (RC), GMM-F and MM [Hunter, 2004] algorithms in Python as well. For the Newton-Raphson algorithm, we implement the choice model on top of the popular `statsmodels` Python library[6] that provides a Newton-Raphson solver. We have compared our implementation of the MM algorithm to that of Hunter written in Matlab[7] and observed that ours has comparable running time. For the chess dataset, we use the Rao–Kupper model with $\alpha = \sqrt{2}$ for simplicity. Note that this parameter could also be estimated from the data, however in our experiments we focus on the performance of algorithms for estimating $\boldsymbol{\gamma}$.

Listing 2.1 – Python implementation of one iteration of I-LSR.

```
import numpy as np
import scipy.linalg as spl

def weighted_lsr(n, rankings, weights):
    chain = np.zeros((n, n), dtype=float)
    for ranking in rankings:
        sum_weights = sum(weights[x] for x in ranking)
        for i, winner in enumerate(ranking):
            val = 1.0 / sum_weights
            for loser in ranking[i+1:]:
                chain[loser, winner] += val
            sum_weights -= weights[winner]
    chain -= np.diag(chain.sum(axis=1))
    return statdist(chain)

def statdist(chain):
    lu, piv = spl.lu_factor(generator.T)
    res = spl.solve_triangular(lu[:-1,:-1], -lu[:-1,-1])
    res = np.append(res, 1.0)
    return res / res.sum()
```

We first compare the estimates produced by three approximate ML inference algorithms, LSR, GMM-F and RC. Note that RC applies only to pairwise comparisons, and that LSR is the only algorithm able to infer the parameters in the Rao-Kupper model. Also note that, in the case of pairwise comparisons, GMM-F and LSR are strictly equivalent. In Table 2.1, we report the root-mean-square deviation to the MLE $\boldsymbol{\theta}^\star$ and the running time $T$ of the algorithm.

The smallest value of $E_{\mathrm{RMS}}$ is highlighted in bold for each dataset. We observe that in the case of multiway partial rankings, LSR is almost four times more accurate than GMM-F on the datasets considered. In the case of pairwise comparisons, RC is slightly worse than LSR and GMM-F, because the number of comparisons per pair is not homogeneous (see Section 2.3.3). The running time of the three algorithms is comparable.

Next, we turn our attention to ML inference and consider three iterative algorithms: I-LSR, MM and Newton-Raphson. For Newton-Raphson, we use an off-the-shelf solver. Each algorithm is initialized with $\boldsymbol{\gamma}^{(0)} = [1/N \ \cdots \ 1/N]^\top$, and convergence is declared

[6] See: `http://statsmodels.sourceforge.net/`
[7] See: `http://sites.stat.psu.edu/~dhunter/code/btmatlab/`

Table 2.1 – Performance of approximate ML inference algorithms.

| | | | | LSR | | GMM-F | | RC | |
|---|---|---|---|---|---|---|---|---|---|
| Dataset | $N$ | $M$ | $K$ | $E_{\mathrm{RMS}}$ | $T$ [s] | $E_{\mathrm{RMS}}$ | $T$ [s] | $E_{\mathrm{RMS}}$ | $T$ [s] |
| NASCAR | 83 | 36 | 43 | **0.194** | 0.03 | 0.751 | 0.06 | — | — |
| Sushi | 100 | 5 000 | 10 | **0.034** | 0.22 | 0.130 | 0.19 | — | — |
| YouTube | 16 187 | 1 128 704 | 2 | **0.417** | 34.18 | **0.417** | 34.18 | 0.432 | 41.91 |
| GIFGIF | 5 503 | 95 281 | 2 | **1.286** | 1.90 | **1.286** | 1.90 | 1.295 | 2.84 |
| Chess | 6 174 | 63 421 | 2 | **0.420** | 2.90 | — | — | — | — |

when $E_{\mathrm{RMS}} < 0.01$. In Table 2.2, we report the number of iterations $I$ needed to reach convergence, as well as the total running time $T$ of the algorithm.

Table 2.2 – Performance of iterative ML inference algorithms.

| | | I-LSR | | MM | | Newton | |
|---|---|---|---|---|---|---|---|
| Dataset | $\xi$ | $I$ | $T$ [s] | $I$ | $T$ [s] | $I$ | $T$ [s] |
| NASCAR | 0.832 | 3 | **0.08** | 4 | 0.10 | — | — |
| Sushi | 0.899 | 2 | **0.42** | 4 | 1.09 | 3 | 10.45 |
| YouTube | 0.002 | 12 | **414.44** | 8 680 | 22 443.88 | — | — |
| GIFGIF | 0.408 | 10 | **22.31** | 119 | 109.62 | 5 | 72.38 |
| Chess | 0.007 | 15 | **43.69** | 181 | 55.61 | 3 | 49.37 |

The smallest total running time $T$ is highlighted in bold for each dataset. We observe that Newton-Raphson does not always converge, despite the log-likelihood being strictly concave[8]. I-LSR consistently outperforms MM and Newton-Raphson in running time. Even if the average running time per iteration is in general larger than that of MM, it needs considerably fewer iterations: For the YouTube dataset, I-LSR yields an increase in speed of over 50 times.

The slow convergence of minorization-maximization algorithms is known [Hunter, 2004], yet the scale of the issue and its apparent unpredictability is surprising. In Hunter's MM algorithm, updating a given $\gamma_i$ involves only parameters of items to which $i$ has been compared. Therefore, we speculate that the convergence rate of MM is dependent on the expansion properties of the comparison graph $\mathcal{G}_{\mathcal{D}}$. For an illustration, we consider the sushi dataset. To quantify the expansion properties, we look at the spectral gap $\xi$ of a simple random walk on $\mathcal{G}_{\mathcal{D}}$; intuitively, the larger the spectral gap is, the better the expansion properties are [Levin et al., 2008]. The original comparison graph is almost complete, and $\xi = 0.899$. By breaking each 10-way ranking into 5 independent pairwise

[8]On the NASCAR dataset, this has also been noted by Hunter [2004]. Computing the Newton step appears to be severely unstable for many real-world datasets. We believe that this instability can be addressed by a careful choice of starting point, step size, or by monitoring the numerical stability; however, these modifications are non-trivial and put an additional burden on the practitioner.

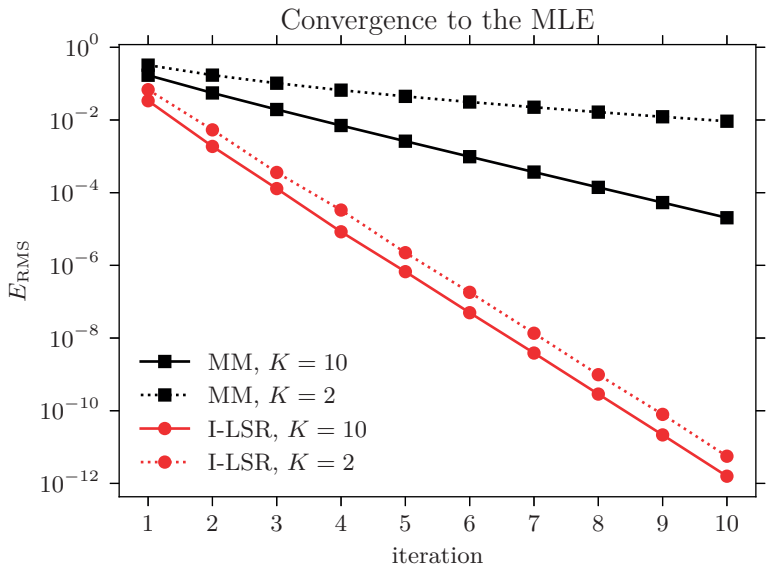



Figure 2.2 – Convergence rate of I-LSR and MM on the sushi dataset. When partial rankings ($K = 10$) are broken down into independent comparisons ($K = 2$), the comparison graph becomes sparser. I-LSR is robust to this change, whereas the convergence rate of MM significantly decreases.

comparisons, we effectively sparsify the comparison graph. As a result, the spectral gap decreases to 0.815. In Figure 2.2, we show the convergence rate of MM and I-LSR for the original ($K = 10$) and modified ($K = 2$) datasets. We observe that both algorithms display geometric convergence, however the rate at which MM converges appears to be sensitive to the structure of the comparison graph. In contrast, I-LSR is robust to changes in the structure. The spectral gap of each dataset is listed in Table 2.2.

## 2.5 Summary

In this chapter, we have developed a stationary-distribution perspective on the maximum-likelihood estimate of Luce's choice model. This perspective explains and unifies several recent spectral algorithms from an ML inference point of view. We have presented our own spectral algorithm, that works on a wider range of data, and shown that the resulting estimate significantly outperforms previous approaches in terms of accuracy. We have also shown that this simple algorithm, with a straightforward adaptation, can produce a sequence of estimates that converge to the ML estimate. On real-world datasets, our ML algorithm is always faster than the state of the art, at times by up to two orders of magnitude.

Beyond statistical and computational performance, we believe that a key strength of our algorithms is that they are simple to implement. As an example, our implementation of LSR fits in ten lines of Python code. The most complex operation—finding a stationary distribution—can be readily offloaded to commonly available and highly optimized linear-algebra primitives. As such, we believe that our contribution is useful for practitioners.

# 3 Active Learning

In this chapter[1], we develop a data-efficient method for learning a ranking from adaptively chosen pairwise comparisons (a setting known as *active learning*). Our goal is to recover the ranking accurately, but to sample the comparisons sparingly. If all comparison outcomes are consistent with the ranking, the optimal solution is to use an efficient sorting algorithm, such as Quicksort. But how do sorting algorithms behave if some comparison outcomes are inconsistent with the ranking? We give favorable guarantees for Quicksort for the Bradley–Terry model, under natural assumptions on the parameters. Furthermore, we empirically demonstrate that sorting algorithms lead to a very simple and effective active-learning strategy: repeatedly sort the items. This strategy performs as well as state-of-the-art methods (and much better than random sampling), at a minuscule fraction of the computational cost.

## 3.1 Introduction

Whereas pairwise comparison models and related inference algorithms have been extensively studied, the issue of *which pairwise comparisons to query* has received significantly less attention from the research community. To understand the potential benefits of adaptively selecting samples, consider the case where comparison outcomes are noiseless, i.e., consistent with a linear order on a set of $N$ items. If pairs of items are selected at random, it is necessary to collect $\Omega(N^2)$ comparisons to recover the ranking [Alon et al., 1994]. In contrast, by using an efficient sorting algorithm, $O(N \log N)$ adaptively chosen comparisons are sufficient. In this chapter, we demonstrate that sorting algorithms can also be helpful in the *noisy* setting, where some comparison outcomes are inconsistent with the ranking: despite errors, sorting algorithms tend to select informative samples. We focus on the Bradley–Terry (BT) model, that captures the intuitive notion that some pairs of items are easy to compare, but some are more difficult (c.f. Section 1.2.2).

[1]This chapter is based on Maystre and Grossglauser [2017b].

First, we study the output of a single execution of Quicksort when comparison outcomes are generated from a BT model, under the assumption that the distance between adjacent parameters is (stochastically) uniform across the ranking. We measure the quality of a ranking estimate by its displacement with respect to the ground truth, i.e., the sum of rank differences. We show that Quicksort's output is a good approximation to the ground-truth ranking: no method comparing every pair of items at most once can do better (up to constant factors). Furthermore, we show that by aggregating $O(\log^5 N)$ independent runs of Quicksort, it is possible to recover the exact rank for all but a vanishing fraction of the items. These theoretical results suggest that adaptive sampling is able to bring a substantial acceleration to the learning process.

Second, we propose a practical active-learning (AL) strategy that consists of repeatedly sorting the items. We evaluate our sorting-based method on three datasets and compare it to existing AL methods. We observe that *all* the strategies that we consider lead to better ranking estimates noticeably faster than random sampling. However, most strategies are challenging to operate and computationally expensive, thus hindering wider adoption [Schein and Ungar, 2007]. In this regard, sorting-based AL stands out, as $(a)$ it is computationally-speaking as inexpensive as random sampling, $(b)$ it is trivial to implement, and $(c)$ it requires no tuning of hyperparameters.

**Outline of the Chapter** After concluding this section with some preliminaries, we review related literature in Section 3.2. Next, in Section 3.3, we study the displacement of Quicksort's output under noisy comparisons. In Section 3.4, we empirically evaluate several AL strategies on three datasets. For clarity of presentation, we defer some proofs to Section 3.5.

### 3.1.1 Preliminaries and Notation

Without loss of generality, we assume that the $N$ items are enumerated by increasing preference[2], i.e., $i < j$ means that $j$ is (in expectation) preferred to $i$ for all $i, j \in [N]$. When $j$ is preferred to $i$ as a result of a pairwise comparison, we denote the observation by $i \prec j$. If $i < j$, we say that $i \prec j$ is a *consistent* outcome and $j \prec i$ an *inconsistent* (incorrect) outcome. We denote by $\mathrm{BT}(\boldsymbol{\theta})$ a Bradley–Terry model with parameters $\boldsymbol{\theta} = [\theta_1 \ \cdots \ \theta_N]^\top \in \mathbf{R}^N$. A ranking $\sigma$ is a function that maps an item to its rank, i.e., $\sigma(i) =$ rank of item $i$. The (ground-truth) identity ranking is denoted by id, i.e. $\mathrm{id}(i) = i$. To measure the quality of a ranking $\sigma$ with respect to the ground-truth, we consider the

[2]This convention greatly simplifies the notation throughout the chapter, but differs from that used in most of the preference-learning literature. In this chapter, the item with rank 1 is the *worst*.

*displacement*

$$\Delta(\sigma) = \sum_{i=1}^{N} |\sigma(i) - i|,$$

also known as Spearman's footrule distance. Another metric widely used in practice is the Kendall–Tau distance, defined as $K(\sigma) = \sum_{i<j} 1_{\{\sigma(i)>\sigma(j)\}}$. Diaconis and Graham [1977] show that both metrics are equivalent up to a factor of two, i.e.,

$$\Delta(\sigma)/2 \leq K(\sigma) \leq \Delta(\sigma).$$

Hence, bounds on $\Delta(\sigma)$ also hold for $K(\sigma)$ up to constant factors. Finally, we say that an event $A$ holds *with high probability* if $\mathbf{P}[A] \to 1$ as $N \to \infty$. For a random variable $X$ and a sequence of numbers $a_N$, we say that $X = O(a_N)$ with high probability if $\mathbf{P}[|X| \leq ca_N] \to 1$ as $N \to \infty$ for some constant $c$ that does not depend on $N$.

## 3.2 Related Work

**Passive Setting** Recently, there have been a number of results on the sample complexity of the BT model, based on the assumption that all pairs of items are chosen *before* any comparison outcome is revealed [Negahban et al., 2012, Hajek et al., 2014, Rajkumar and Agarwal, 2014, Vojnovic and Yun, 2016]. In general, these results reveal that choosing pairs of items uniformly at random is essentially optimal. Furthermore, they suggest that the ranking induced by the BT model cannot be recovered with less than $\Omega(N^2)$ comparisons. Our work shows that by *adaptively* selecting pairs based on observed outcomes, we observe substantial gains.

**Active Preference Learning** AL approaches for learning a ranking from noisy comparison outcomes have been studied under various assumptions. Braverman and Mossel [2008] examine a model where outcomes of pairwise comparisons are flipped with a small, constant probability. Ailon [2012] considers an adversarial setting (comparison outcomes can be arbitrary) and investigates AL in the context of finding a ranking that minimizes the number of inconsistent outcomes, also known as the minimum feedback-arc set problem on tournaments (MFAST). These theoretical studies imply, in their respective settings, that $O(N \log^K N)$ comparison outcomes are enough to recover a near-optimal ranking, for some constant $K$. Jamieson and Nowak [2011] propose an efficient active-ranking algorithm that is applicable if items can be embedded in $\mathbf{R}^D$ (e.g., using $D$ features) and assuming that admissible rankings satisfy some geometric constraints. Wang et al. [2014] study a collaborative preference-learning problem (each user is modeled by a different BT model) and show that a variant of uncertainty sampling—a well-known AL

strategy—works well for their problem. Here, we assume that we do not have access to item features and that comparison outcomes follow a single BT model.

**Bayesian Methods** From a practical standpoint, Bayesian methods provide an effective way to select informative samples [MacKay, 1992]. However, they can be difficult to scale if the number of items is large. Work on Bayesian active preference learning includes Chu and Ghahramani [2005a], Houlsby et al. [2012], Salimans et al. [2012] and Chen et al. [2013]. We compare our AL strategy to these methods in Section 3.4.

**Multi-Armed Bandit** The *dueling bandit* problem [Yue et al., 2009] is somewhat related to our work. In this problem, the goal is to identify the best item, based on noisy comparison outcomes, using as few adaptively chosen samples as possible. Two recent papers also extend the problem to that of recovering the entire ranking (instead of only the top element). The work of Szörényi et al. [2015] is the closest to ours, as it also uses the BT model. They show that a quasilinear number of comparisons is sufficient for recovering the true ranking (under some conditions on $\boldsymbol{\theta}$), a result that is similar to our Theorem 3.6. Heckel et al. [2016] investigate a non-parametric model and develop some theoretical guarantees. In contrast to these works, we study practical comparison budgets: we give theoretical guarantees for the output obtained from a single call to Quicksort, and in our experiments we never exceed $\approx 10$ calls.

**Quicksort** The Quicksort algorithm [Hoare, 1962] is one of the most widely studied sorting procedures. Quicksort has been shown to produce useful rankings beyond classic sorting problems. For example, Ailon et al. [2008] show that Quicksort produces (in expectation) a 3-approximation to the MFAST problem. Quicksort combined with BT comparison outcomes has also been proposed as a probabilistic ranking model [Ailon, 2008]. We take advantage of some of the properties of this ranking model in order to derive the theoretical results of Section 3.3.

## 3.3 Theoretical Results

In this section, we begin by studying the behavior and output of Quicksort under inconsistent comparison outcomes, without any assumptions on the noise generating process. Then, starting in Section 3.3.1, we focus on comparison outcomes generated by the BT model. For clarity, longer proofs are deferred to Section 3.5.

Quicksort (Algorithm 3.1) is best described as a recursive procedure. At each step of the recursion, a *pivot* item $p$ is chosen uniformly at random (line 3). Then, during the *partition* operation (lines 4–10), every other item is compared to $p$ and added to the set $\mathcal{L}$ or $\mathcal{R}$,

**Algorithm 3.1** Quicksort.

**Require:** set of items $\mathcal{V} \subseteq [N]$
**if** $|\mathcal{V}| < 2$ **then return** list($\mathcal{V}$) ▷ Terminating case.
$\mathcal{L} \leftarrow \varnothing, \mathcal{R} \leftarrow \varnothing$
$p \leftarrow$ element of $\mathcal{V}$ selected uniformly at random
**for** $i \in \mathcal{V} \setminus \{p\}$ **do**
    **if** $i \prec p$ **then** ▷ Pairwise comparison.
        $\mathcal{L} \leftarrow \mathcal{L} \cup \{i\}$
    **else**
        $\mathcal{R} \leftarrow \mathcal{R} \cup \{i\}$
    **end if**
**end for**
**return** Quicksort($\mathcal{L}$) $\cdot\, p \cdot$ Quicksort($\mathcal{R}$)

depending on the outcome of the comparison with the pivot. If all comparison outcomes are consistent, it is well-known that Quicksort terminates after sampling $O(N \log N)$ comparisons with high probability. What happens if we drop the consistency assumption? The following two lemmas state that these key properties remain valid, no matter which (and how many) comparison outcomes are inconsistent.

**Lemma 3.1.** *Quicksort always terminates and samples each of the $N(N-1)/2$ possible comparisons at most once.*

*Proof.* The proof is identical to the consistent setting. Consider the state of $\mathcal{L}$ and $\mathcal{R}$ at the end of a partition operation. Because $|\mathcal{L}| + |\mathcal{R}| = |\mathcal{V}| - 1$, the recursive calls are made on sets of items of strictly decreasing cardinality, and the algorithm terminates after a finite number of steps. Furthermore, suppose that Quicksort samples an outcome for the pair $(i, j)$. Then either $i$ or $j$ is the pivot in a partition operation. In either case, the pivot is not included in the recursive calls, which ensures that $(i, j)$ cannot be compared again. □

**Lemma 3.2.** *Quicksort samples $O(N \log N)$ comparisons with high probability.*

*Proof (sketch).* We follow a standard analysis of Quicksort [see, e.g., Dubhashi and Panconesi, 2009, Section 3.3.3]. With high probability, we choose a "good" pivot (i.e., one that results in a balanced partition) a constant fraction of the time. In this case, the depth of the call tree is $O(\log N)$. As there are at most $N$ comparisons at each level of the call tree, we conclude that Quicksort uses $O(N \log N)$ comparisons in total. With respect to the standard proof, ours requires additional work in order to formalize the notion of "good" pivot to the setting where comparison outcomes are not consistent with a linear order. □

Lemma 3.2 complements Theorem 3 in Ailon and Mohri [2010], which states that Quicksort samples $O(N \log N)$ in expectation. These results might suggest that *all* properties of Quicksort carry over to the noisy setting. This is not the case. For example, although Quicksort uses approximately $2N \ln N$ comparisons on average in the noiseless setting [Sedgewick and Wayne, 2011], this number can be distinctly different with inconsistent comparison outcomes[3].

Quicksort (and efficient sorting algorithms in general) infer most pairs of items' relative position by transitivity and rely heavily on the consistency of comparison outcomes. In the noisy case, it is therefore important to precisely understand the effect of an inconsistent outcome on the output of the algorithm; this effect extends beyond the pair of items whose comparison outcome was inconsistent. For this purpose, the next Lemma bounds the displacement of Quicksort's output as a function of the inconsistent outcomes.

**Lemma 3.3.** *Let $\mathcal{E}$ be the set of pairs sampled by Quicksort whose outcome is inconsistent with* id*. Let $\sigma$ be the output of Quicksort. Then,*

$$\Delta(\sigma) \leq 2 \sum_{(i,j) \in \mathcal{E}} |i - j|$$

*Proof (sketch).* Consider the first partition operation, with pivot $p$, resulting in partitions $\mathcal{L}$ and $\mathcal{R}$. Denote the set of pairs of items involved in errors made during this partition operation by $\mathcal{E}_1$. We can show that the displacement is bounded by

$$\Delta(\sigma) \leq \Delta_{\mathcal{L}}(\sigma) + \Delta_{\mathcal{R}}(\sigma) + 2 \sum_{(i,j) \in \mathcal{E}_1} |i - j|,$$

where $\Delta_{\mathcal{L}}(\sigma)$ and $\Delta_{\mathcal{R}}(\sigma)$ represent the displacement of the ordering induced by $\sigma$ on $\mathcal{L}$ and $\mathcal{R}$, respectively. In other words, the total displacement can be decomposed into a term that represents the "local" displacement due to the partition operation and into two terms that account for errors in the recursive calls. We obtain the desired result by recursively bounding $\Delta_{\mathcal{L}}(\sigma)$ and $\Delta_{\mathcal{R}}(\sigma)$. □

Informally, Lemma 3.3 states that the displacement can be bounded by a sum of "local shifts" due to the inconsistent outcomes and that the price to pay for any information inferred by transitivity is bounded by a factor two. Lemma 3.3 is a crucial component of our subsequent analysis of BT noise, and we believe that it can be useful in order to investigate Quicksort under a wide variety of other noise generating processes.

[3] E.g., if comparison outcomes are uniformly random, all items are "good" pivots with high probability, and the average number of comparisons will be closer to $N \log_2 N$ on average, for large $N$.

### 3.3.1 Poisson-Distributed Parameters

From here on, we assume that comparison outcomes are generated from $\mathrm{BT}(\boldsymbol{\theta})$. Clearly, any results on the displacement of a ranking estimated from samples of a BT model will depend on $\boldsymbol{\theta}$; it is easy to construct a model instance for which it is arbitrarily hard to recover the ranking, by choosing parameters sufficiently close to each other. Our approach is as follows. We postulate a family of distributions over $\boldsymbol{\theta}$, and we give bounds on the displacement that hold with high probability.

We suppose that comparison outcomes are (in expectation) *uniformly noisy across the ranking*: i.e., comparing two elements at the bottom is (a priori) as difficult as comparing two elements at the top or in the middle. This means that the probability distribution over parameters $\theta_1, \ldots, \theta_N$ results in (random) distances $|\theta_{i+k} - \theta_i|$ that depend only on $k$. One such distribution arises if the parameters are drawn from a Poisson point process of rate $\lambda$. That is,

$$\text{i.i.d. } x_1, \ldots, x_{N-1} \sim \mathrm{Exp}(\lambda), \qquad \theta_i = \sum_{n=1}^{i-1} x_n. \tag{3.1}$$

The average distance between two items separated by $k$ positions in the ordering is $\mathbf{E}\left[\theta_{i+k} - \theta_i\right] = k/\lambda$. Although the distance between adjacent items is constant in expectation, we let some parameters be arbitrarily close[4]. The parameter $\lambda$ indirectly controls the expected level of noise; a large $\lambda$ is likely to result in a larger number of inconsistent outcomes. Although the precise choice of this Poisson model is driven by tractability concerns, in Section 3.3.2 we argue that it is essentially equivalent to choosing the parameters independently and uniformly at random in the interval $[0, (N+1)/\lambda]$, when $\lambda$ is fixed and $N$ is large. We are now ready to state our main result.

**Theorem 3.4.** *Let $\boldsymbol{\theta}$ be sampled from a Poisson point process of rate $\lambda$. Let $\sigma$ be the output of Quicksort using comparison outcomes sampled from* $\mathrm{BT}(\boldsymbol{\theta})$*. Then,*

$$\Delta(\sigma) = O(\lambda^2 N), \tag{3.2}$$

$$\max_i |\sigma(i) - i| = O(\lambda \log N), \tag{3.3}$$

*with high probability.*

*Proof (sketch).* Let $z_{ij}$ be the indicator random variable of the event "the comparison between $i$ and $j$ results in an error", and let $d_{ij} = |\theta_i - \theta_j|$. The distance $d_{ij}$ is a sum of $|i-j|$ i.i.d. exponential random variables, i.e., $d_{ij} \sim \mathrm{Gamma}(|i-j|, \lambda)$, and we can show

[4] In particular, the expected minimum distance between two items (i.e., the min of $N$ exponential r.v.s) decreases as $(N\lambda)^{-1}$ as $N$ increases.

that

$$\mathbf{E}\left[z_{ij}\right] = \mathbf{E}\left[\frac{1}{1+\exp(d_{ij})}\right] \leq \mathbf{E}\left[\exp(-d_{ij})\right] = (1+1/\lambda)^{-|i-j|}.$$

Using Lemma 3.3 and the fact that every pair of items is compared at most once, we find

$$\mathbf{E}\left[\Delta\right] \leq 2\sum_{i<j} |i-j|\mathbf{E}\left[z_{ij}\right] \leq 2N\sum_{k=0}^{\infty} k(1+1/\lambda)^{-k} = 2N\lambda(\lambda+1).$$

The random variables $\{z_{ij}\}$ are not independent (they are independent when conditioned on $\boldsymbol{\theta}$) but, with some more work, we can show that $\mathbf{Var}\left[\Delta\right] = O(N)$. By using a Chebyshev bound, (3.2) follows.

In order to prove (3.3), we take advantage of a theorem due to Ailon [2008] which states that

$$\mathbf{P}\left[\sigma(i) < \sigma(j) \mid \boldsymbol{\theta}\right] = \mathbf{P}\left[i \prec j \mid \boldsymbol{\theta}\right],$$

even if $i$ and $j$ were not directly compared with each other. We use a Chernoff bound on $d_{ij}$ to show that the relative order between any two items separated by at least $O(\lambda \log N)$ positions is correct with high probability. The second part of the claim follows easily. □

Note that any method that compares each pair of items at most once results in a ranking estimate $\tau$ with displacement $\Delta(\tau) = \Omega(N)$ with high probability: As there is only a single (possibly inconsistent) comparison outcome between each pair of adjacent items, it is likely that a constant fraction of the items will be ranked incorrectly, resulting in a displacement that grows linearly in $N$. Hence, our bound on $\Delta(\sigma)$ shows that Quicksort is order-optimal (in $N$).

In light of Theorem 3.4, a natural question to ask is as follows. How many comparisons are needed in order to find the correct ranking? Finding the exact ranking is difficult: in fact, $\Omega(N)$ comparison outcomes are necessary in order to discriminate the closest pair of items reliably, as we show next. Suppose that we are given $K$ comparisons to find the relative order between $i$ and $j$, and define $e_{ij}$ as the event "more than half of the $K$ comparison outcomes between $i$ and $j$ are inconsistent".

**Proposition 3.5.** *Let $\boldsymbol{\theta}$ be sampled from a Poisson point process of rate $\lambda$. Then, there is a pair $i, j \in [N]$ and a constant $c > 0$ independent of $N$ such that if $K = o(\lambda N)$,*

$$\mathbf{P}\left[e_{ij}\right] \geq c$$

*with high probability.*

**Algorithm 3.2** Multisort.

**Require:** set of items $\mathcal{V} \subseteq [N]$, number of iterations $K$
$\mathcal{S} \leftarrow \varnothing$
**for** $k = 1, \ldots, K$ **do**
    $\sigma \leftarrow \text{Quicksort}(\mathcal{V})$
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{\sigma\}$
**end for**
**return** Copeland aggregation of $\mathcal{S}$

*Proof.* The distance between the two closest items is $d_{\min} = \min_i |\theta_{i+1} - \theta_i| = \min_n x_n$, i.e., the minimum of $N-1$ independent exponential random variables of rate $\lambda$. Therefore, $d_{\min} \sim \text{Exp}((N-1)\lambda)$, and for $N \geq 2$ with probability at least $1 - e^{-1/2} \approx 0.39$ we have $d_{\min} \leq (\lambda N)^{-1}$. Let $z_k$ be the indicator random variable for the event "the outcome of the $k$-th comparison is incorrect". Assuming that $d_{\min} \leq (\lambda N)^{-1}$ and that $\lambda N \geq 1/2$,

$$\begin{aligned}\mathbf{P}\left[z_k = 0\right] &\leq \frac{1}{1 + \exp[-1/(\lambda N)]} \leq \frac{1}{2 - 1/(\lambda N)} = \frac{1}{2} \cdot \left(1 + \frac{1}{2\lambda N - 1}\right) \\ &\leq \frac{1}{2} \exp\left[\frac{1}{2\lambda N - 1}\right],\end{aligned}$$

where we used the inequality $e^x \geq 1 + x$ twice. The probability of *correctly* identifying the relative order between the two closest items based on $K$ comparisons is

$$\begin{aligned}\mathbf{P}\left[\sum_{k=1}^{K} z_k \leq K/2\right] &\leq \sum_{\ell=1}^{K/2} \binom{K}{\ell} \mathbf{P}\left[z_k = 0\right]^K \leq \exp\left[\frac{K}{2\lambda N - 1}\right] \cdot 2^{-K} \sum_{\ell=1}^{K/2} \binom{K}{\ell} \\ &= \frac{1}{2} \exp\left[\frac{K}{2\lambda N - 1}\right].\end{aligned}$$

As $\mathbf{P}\left[e_{ij}\right] = 1 - \mathbf{P}\left[\sum_{k=1}^{K} z_k \leq K/2\right]$, it follows that, if $K = o(\lambda N)$, the probability of *incorrectly* identifying the relative order between the two closest items is bounded from below by a positive constant. □

As finding the *exact* ranking appears to be difficult, we instead focus on finding a ranking that matches the ground truth everywhere, except at a vanishing fraction of the items.

Multiple runs of Quicksort likely produce different outputs, because of the noisy comparison outcomes and because the algorithm itself is randomized (the pivot selection is random). By aggregating $K$ independent outputs of Quicksort, is it possible to produce a better ranking estimate? Similarly to Szörényi et al. [2015], we combine the $K$ outputs $\sigma_1, \ldots, \sigma_K$ into an aggregate ranking $\hat{\sigma}$ using Copeland's method. The method assigns, to each item, a score that corresponds to the number of items that it beats in a majority of the rankings, and it then ranks the items by increasing score [Copeland, 1951]. We call the procedure Multisort and describe it in Algorithm 3.2.

**Theorem 3.6.** *Let $\boldsymbol{\theta}$ be sampled from a Poisson point process of rate $\lambda$. Let $\hat{\sigma}$ be the output of Multisort using $K = O(\lambda^2 \log^5 N)$ and comparison outcomes sampled from* $\mathrm{BT}(\boldsymbol{\theta})$*. Then,*

$$\Delta(\hat{\sigma}) = o(\lambda N)$$

*with high probability.*

*Proof (sketch).* We use results on the order statistics of the distances $x_1, \ldots, x_{N-1}$ between successive items, as defined in (3.1), to partition the items into two disjoint subsets $\mathcal{B}$ and $\mathcal{G}$. The set $\mathcal{B}$ contains a vanishing $(1/\log^2 N)$-fraction of "bad" items that are difficult to order. The set $\mathcal{G}$ is such that the smallest distance $d_{ij}$ from any item $i \in \mathcal{G}$ to any other item $j \in [N]$ is bounded from below by $c/(\lambda \log^2 N)$. We can show that with $K = O(\lambda^2 \log^5 N)$, for any $i \in \mathcal{G}$ and $j \in [N]$ we have $i < j \iff \sigma(i) < \sigma(j)$ in a majority of the Quicksort outputs (with high probability). This implies that $\hat{\sigma}(i) = i$ for all $i \in \mathcal{G}$ with high probability. Using (3.3) for items in $\mathcal{B}$, we have

$$\Delta(\hat{\sigma}) = |\mathcal{B}| \cdot O(\lambda \log N) = O(\lambda N / \log N)$$

with high probability. □

Theorem 3.6 states that all but a vanishing fraction of items are correctly ranked using $O(\lambda^2 N \log^6 N)$ comparisons. This result should be compared to that of Rajkumar and Agarwal [2014] obtained in the passive setting, which suggests that $\Omega(N^2)$ comparisons are needed if samples are selected uniformly at random.

**Empirical Validation** In Figure 3.1, we illustrate Theorems 3.4 and 3.6 by running simulations for increasing $N$ and different values of $\lambda$. The bound on $\Delta(\sigma)$ is tight in $N$, but the dependence on $\lambda$ appears to be linear rather than quadratic. The bound on $\max_i |\sigma(i) - i|$ appears to be tight in $N$ and $\lambda$. Finally, we compare the Copeland aggregation of $K$ outputs of Quicksort with the ranking induced by the maximum-likelihood estimate (MLE), inferred from the outcomes of all the pairwise comparisons sampled by the $K$ runs. Although the ranking induced by the MLE does not benefit from the guarantees of Theorem 3.6, it performs better in practice. We will make use of this observation in Section 3.4.

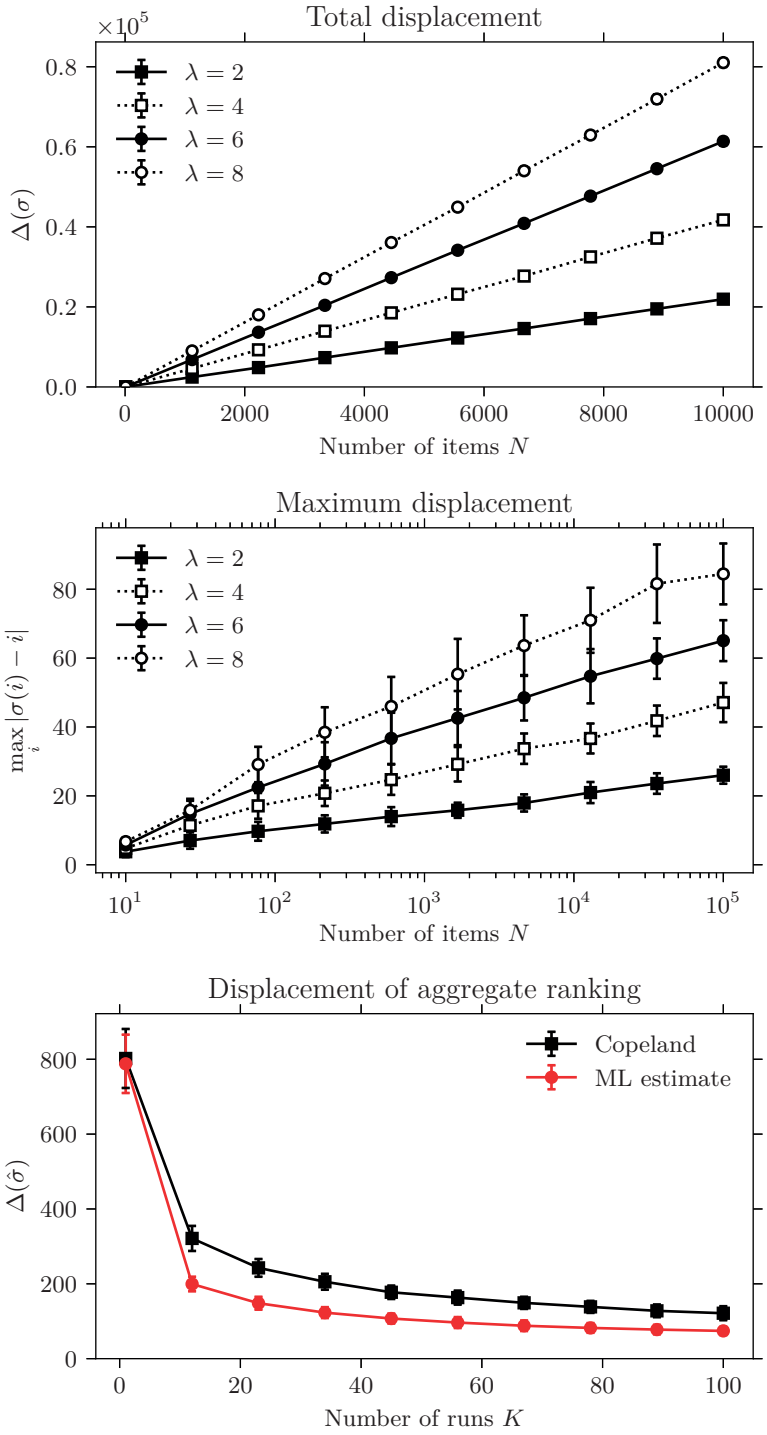



Figure 3.1 – Empirical validation of Theorem 3.4 and illustration of Theorem 3.6. Every simulation is repeated 50 times, and we report the mean and the standard deviation. Top and middle: total and maximum displacement (respectively) for increasing $N$ and different values of $\lambda$. Bottom: displacement of the aggregate ranking $\hat{\sigma}$ for increasing $K$, fixing $N = 200$ and $\lambda = 4$ and using two different aggregation rules.

### 3.3.2 Independent Uniformly Distributed Parameters

A different (perhaps more natural) assumption about the parameters $\boldsymbol{\theta}$ is to consider that they are drawn independently and uniformly at random over some interval. That is,

$$\text{i.i.d. } \bar{\theta}_1, \ldots, \bar{\theta}_N \sim \mathrm{U}(0, (N+1)/\lambda),$$

with $\theta_1, \ldots, \theta_N$ the order statistics of $\bar{\boldsymbol{\theta}}$, i.e., the random variables arranged in increasing order. From some elementary results on the joint distribution of order statistics [see, e.g., Arnold et al., 2008], we have that

$$|\theta_{i+k} - \theta_i| \sim (N+1)/\lambda \cdot \mathrm{Beta}(k, N-k+1),$$

i.e., $|\theta_{i+k} - \theta_i|$ is distributed as a Beta random variable rescaled between 0 and $(N+1)/\lambda$. Letting $f_{k,N}(x)$ be the probability density of $|\theta_{i+k} - \theta_i|$, we have, for any fixed $k$ and $\lambda$,

$$f_{k,N}(x) \propto x^{k-1}\left[1 - \frac{\lambda x}{N+1}\right]^{N-k} \xrightarrow{N\to\infty} x^{k-1}e^{-\lambda x}.$$

We recognize the functional form of the density of a Gamma$(k, \lambda)$ distribution. Hence, the Poisson model and the i.i.d. uniform model are essentially equivalent for fixed $\lambda$ and large $N$, and we can expect the results developed in Section 3.3.1 to hold in the i.i.d. uniform case as well.

## 3.4 Experimental Evaluation

In practice, the comparison budget for estimating a ranking from noisy data might typically be larger than that for a single call to Quicksort, and it might not exactly match the number of comparisons required to run a given number of calls to Quicksort to completion. Building upon the observations made at the end of Section 3.3.1, we suggest the following practical active-learning strategy: For a budget of $M$ pairwise comparisons, run the sorting procedure repeatedly until the budget is depleted (the last call might have to be truncated); then, retain only the set of $M$ comparison pairs and their outcomes and discard the rankings produced by the sorting procedure; the final ranking estimate is then induced from the MLE over the set of $M$ comparison outcomes.

In this section, we demonstrate the effectiveness of this sampling strategy on synthetic and real-world data. In particular, we show that it is comparable to existing AL strategies at a minuscule fraction of the computational cost.

### 3.4.1 Competing Sampling Strategies

To assess the relative merits of our sorting-based strategy, we consider three strategies that are representative of the state of the art in active preference learning.

**Uncertainty Sampling** Developed in the context of classification tasks, this popular active-learning heuristic suggests to greedily sample the point that lies closest to the decision boundary [Settles, 2012]. In the context of a ranking task, this corresponds to sampling the pair of items whose relative order is most uncertain. After $t$ observations, given an estimate of model parameters $\boldsymbol{\theta}^t$, the strategy selects the $(t{+}1)$-st pair uniformly at random in

$$\arg\min_{i \neq j} |\theta_i^t - \theta_j^t|.$$

This set can be computed in time $O(N \log N)$ by sorting the parameters. The parameters themselves need to be estimated, e.g., using (penalized) ML inference that in practice can be the dominating cost.

**Bayesian Methods** If we have access to a full posterior distribution $q^t(\boldsymbol{\theta})$ instead of a point estimate $\boldsymbol{\theta}^t$, we can take advantage of the extra information on the uncertainty of the parameters to improve the selection strategy. A principled approach to AL consists of sampling the point that maximizes the expected information gain [MacKay, 1992, Chu and Ghahramani, 2005a]. That is, the pair of items at iteration $t+1$ is selected in

$$\arg\max_{i \neq j} H(q^t) - \mathbf{E}\left[H(q^{t+1})\right], \tag{3.4}$$

where $H(\cdot)$ denotes the entropy function. A conceptually similar but slightly different selection strategy is given by Chen et al. [2013]. Letting $q_{ij}$ be the marginal distribution of $(\theta_i, \theta_j)$, the pair is selected in

$$\arg\max_{i \neq j} \mathbf{E}\left[\mathrm{KL}(q_{ij}^{t+1} \| q_{ij}^t)\right], \tag{3.5}$$

where $\mathrm{KL}(\cdot)$ denotes the Kullback–Leibler divergence. Computing the exact posterior is not analytically tractable for the BT model, but a Gaussian approximation can be found in time $O(N^3)$. Criteria (3.4) and (3.5) can be computed in constant time for each pair of items. The dominating cost is again that of estimating $\boldsymbol{\theta}$ (or, in this case, $q(\boldsymbol{\theta})$).

In addition to these existing AL strategies, we also include in our experiments a variation of our sorting-based strategy that uses Mergesort instead of Quicksort. In the noiseless setting, Mergesort is known to use on average $\approx 39$ % fewer comparisons than Quicksort

Table 3.1 – Time (in seconds) to select the $(N+1)$-st pair. See text for details.

| | $T$ [s] | | |
|---|---|---|---|
| Strategy | $N = 10^2$ | $N = 10^3$ | $N = 10^4$ |
| uncertainty | 0.05 | 0.5 | 11 |
| entropy | 0.3 | 40 | — |
| KL-divergence | 0.9 | 71 | — |
| Mergesort | $< 0.001$ | $< 0.001$ | $< 0.001$ |
| Quicksort | $< 0.001$ | $< 0.001$ | $< 0.001$ |
| random | $< 0.001$ | $< 0.001$ | $< 0.001$ |

per run [Knuth, 1998], but it does not benefit from the theoretical guarantees developed in Section 3.3.

### 3.4.2 Running Time

In this section, we briefly discuss the running time of the methods. We implement ML and Bayesian approximate inference algorithms for the BT model as a Python library (see Appendix A). For approximate Bayesian inference, we use a variant of the expectation-propagation algorithm outlined by Chu and Ghahramani [2005a]. All experiments are performed on a server with a 12-core Xeon X5670 processor running at 2.93 GHz. Numerical computations take advantage of the Intel Math Kernel Library.

We illustrate the running time of AL strategies as follows. For $N \in \{10^2, 10^3, 10^4\}$, we generate outcomes for $N$ comparisons pairs chosen uniformly at random among $N$ items. For each strategy, we then measure the time it takes to select the $(N+1)$-st pair of items adaptively. The results are presented in Table 3.1. Note that these numbers are intended to be considered as orders of magnitude, rather than exact values, as they depend on the particular combination of software and hardware that we use. The running time of the Bayesian AL strategies exceeds 10 hours for $N = 10^4$, and the calls were stopped ahead of completion. Our sorting-based methods, like random sampling, are the only AL strategies whose running time is constant for increasing $N$ (and for increasing $M$). In fact, their running time is negligible in comparison to the other strategies, including uncertainty sampling.

### 3.4.3 Data Efficiency

We now investigate three datasets and measure the displacement of rankings estimated from adaptively-chosen samples, as a function of the budget $M$. Note that in order to use uncertainty sampling and Bayesian methods, it is necessary to choose a regularization strength or prior variance in the inference step. Different values can result in drastically

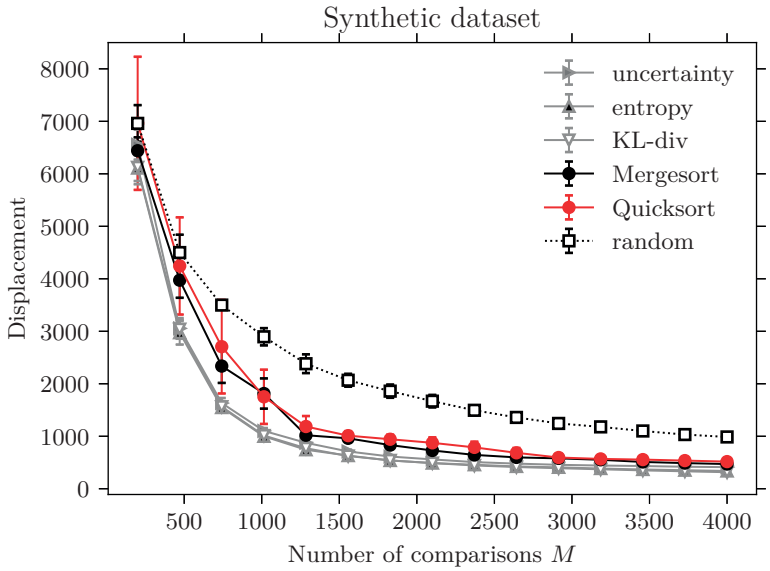



Figure 3.2 – Synthetic dataset with $\lambda = 5$ and $N = 200$. The experiment is repeated 10 times, and we report the mean and the standard deviation. Compared to random sampling, AL results in significantly better rankings for a given budget $M$.

different outcomes (in particular for uncertainty sampling) and, in practice, choosing a good value can be a significant challenge. In the following, we report results for the values that worked best *a posteriori*. Observe that, in contrast, our sorting-based approach is entirely parameter-free.

**Synthetic Dataset** We generate $N$ i.i.d. parameters $\theta_1, \dots, \theta_N$ uniformly in $[0, (N+1)/\lambda]$ and draw samples from BT($\boldsymbol{\theta}$). The ground-truth ranking is the one induced by the parameters. Figure 3.2 presents results for $N = 200$ and $\lambda = 5$. In comparison to random sampling, AL is very effective and results in significantly better ranking estimates for any given number of comparisons. The two Bayesian methods, though being the most computationally expensive, perform the best for all values of $M$ but are nearly indistinguishable from uncertainty sampling. The two sorting-based strategies perform similarly (with a small edge for Mergesort). They are slightly worse than the Bayesian methods but are still able to reap most of the benefits of active learning.

**Sushi Dataset** Next, we consider a dataset of sushi preferences [Kamishima and Akaho, 2009]. In this dataset, 5 000 respondents give a strict ordering over 10 different types of sushi. These 10 sushi are chosen among a larger set of $N = 100$ items. To suit our purposes, we decompose each 10-way partial ranking into pairwise comparisons, resulting

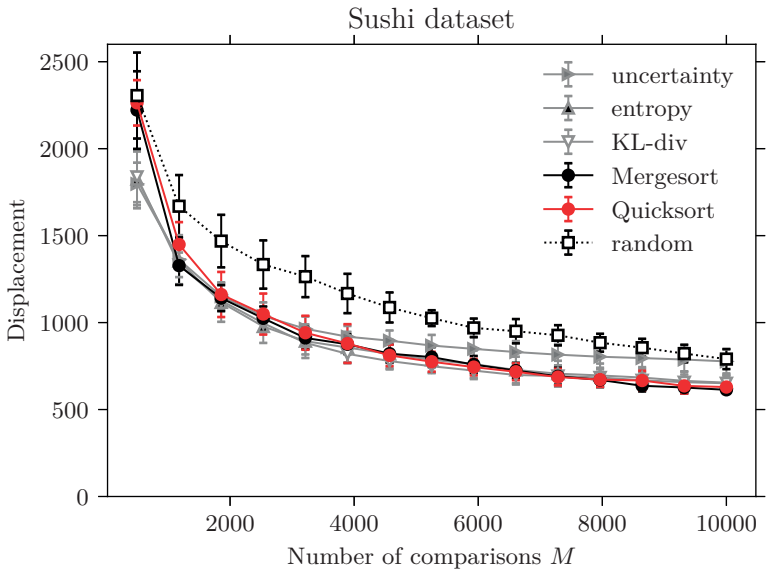



Figure 3.3 – Experimental results on the sushi dataset. Every experiment is repeated 10 times, and we report the mean and the standard deviation. Sorting-based and Bayesian AL strategies have near-identical performance starting from $M \approx 1\,000$.

in 225 000 comparison outcomes. We use all comparisons to fit a BT model that induces a ground-truth ranking[5].

The comparisons are dense, and there is at least one comparison outcome for almost all pairs. When an outcome for pair $(i, j)$ is requested, we sample uniformly at random over all outcomes observed for this pair. In the rare case where no outcome is available, we return $i \prec j$ with probability 1/2. This enables us to compare sampling strategies in a realistic setting, where the assumptions of the BT model do not necessarily hold anymore.

Results are shown in Figure 3.3. Once again, active learning performs noticeably better than random sampling. On this real-world dataset, the performance of our sorting-based strategies is indistinguishable from that of the Bayesian methods, after completing one entire call to the sorting procedure (slightly less than 1 000 comparisons). This result should be interpreted in light of the time needed to select all $10^4$ pairs: a fraction of a second for sorting-based strategies, and several hours for the Bayesian methods. Finally, we observe that the performance of uncertainty sampling progressively degrades as $M$ increases. A detailed analysis reveals that uncertainty sampling increasingly focuses on a small set of hard-to-discriminate pairs, symptomatic of a well-known issue [Settles, 2012].

[5]The BT-induced ranking is almost the same as that obtained using the Copeland score. The results are very similar if the Copeland aggregation is used as ground truth.

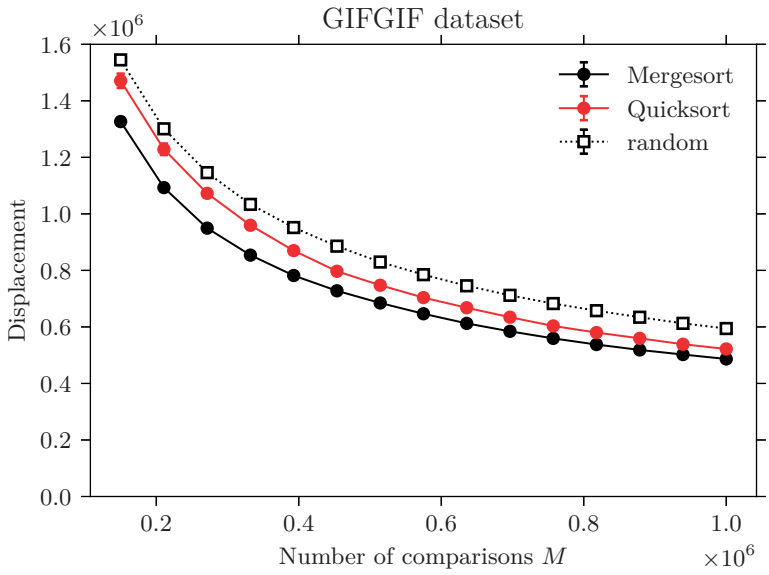



Figure 3.4 – Experimental results on the GIFGIF dataset. Every experiment is repeated 10 times, and we report the mean and the standard deviation. Most AL strategies are computationally too expensive—except for sorting-based methods.

**GIFGIF Dataset** GIFGIF[6] is a project of the MIT Media Lab that aims at explaining the emotions communicated by a collection of animated GIF images. Users of the website are shown a prompt with two images and a question, "Which better expresses $x$?" where $x$ is one of 17 emotions. The users can click on either image, or use a third option, *neither*. To date, over three million comparison outcomes have been collected. For the purpose of our experiment, we restrict ourselves to a single emotion, *happiness*; and we ignore outcomes that resulted in *neither*. We consider 106 887 comparison outcomes over $N = 6\,120$ items—a significant increase in scale, compared to the sushi dataset.

As the data, despite a relatively large number of comparisons, remain sparse (less than 20 comparisons per item on average), we proceed as follows. We fit a BT model by using all the available comparisons and use the induced ranking as ground truth. We then generate new, synthetic comparison outcomes from the BT model. In this sense, the experiment enables us to compare sampling strategies by using a large BT model with realistic parameters. The large number of items makes uncertainty sampling and the two Bayesian methods prohibitively expensive. We try a simplified, computationally less expensive version of uncertainty sampling where, at every iteration, each item is compared to its two closest neighbors, but this heuristic fails spectacularly: The resulting displacement is over 5× larger than random sampling for $M = 10^6$ and is therefore not reported here.

[6]See `http://www.gif.gf/`. Data available at `http://lucas.maystre.ch/gifgif-data`.

Figure 3.4 compares the displacement of random sampling to that of the two sorting-based sampling strategies for increasing $M$. The adaptive sampling approaches perform systematically better. After $10^6$ comparisons, the displacement of random sampling is 14 % and 23 % larger than that of Quicksort and Mergesort, respectively. Conversely, in order to reach any target displacement, Mergesort requires approximately $2\times$ fewer comparisons than random sampling.

## 3.5 Proofs

Section 3.5.1 contains the proofs of Lemmas 3.2 and 3.3. Section 3.5.2 presents the proof for our results on the displacement of the output of a single call to Quicksort (Theorem 3.4), and Section 3.5.3 shows our result on the displacement of the Copeland aggregation of multiple outputs (Theorem 3.6).

### 3.5.1 Lemmas 3.2 and 3.3

We start by briefly presenting a result from graph theory that will be useful in the proof of Lemma 3.2. A *tournament* is a directed graph obtained by assigning a direction to every edge of a complete graph. The *score sequence* of a tournament is defined as the nondecreasing sequence of the vertices' outdegrees. The following proposition is by Landau [1953].

**Proposition 3.7.** *Let* $(s_1, \dots, s_N)$ *with* $0 \leq s_1 \leq \cdots \leq s_N$ *be the score sequence of a tournament on* $N$ *vertices. Then,*

$$\frac{n-1}{2} \leq s_n \leq \frac{N+n-2}{2} \quad \forall\, n \in [N].$$

We use a tournament on $N$ vertices to represent the outcome of a comparison between each pair of items. In particular, we represent the outcome $i \prec j$ by an edge $(i, j)$. In this case, the outdegree of a vertex $i$ corresponds to the number of items which "won" in a comparison against $i$. Note that the comparison outcomes do not need to be transitive, i.e., the tournament can contain cycles.

The proof of Lemma 3.2 is adapted from standard results on Quicksort, see, e.g., Dubhashi and Panconesi [2009, Section 3.3.3]. These results are based on the fact that it is likely that the random choice of pivot leads to a well-balanced partition into subsets $\mathcal{L}$ and $\mathcal{R}$. In our setting, the comparison outcomes do not need to be consistent with an ordering of the items, therefore we cannot use the standard argument based on the pivot's *rank*. Instead, we use the tournament representation of the comparison outcomes and analyze the pivot's *out-degree* (using Proposition 3.7) to ensure that the partition is balanced often enough.

*Proof of Lemma 3.2.* We show that the maximum call depth of Quicksort is at most $\lceil 48 \log N \rceil$ with high probability. The statement follows by noting that at most $N$ comparisons are used at each level of the call tree.

By Lemma 3.1, Quicksort samples a comparison outcome for each pair of items at most once. Therefore, we can represent these (a priori unobserved) pairwise outcomes as a tournament $\mathcal{T} = ([N], \mathcal{A})$. At each step of the recursion, we select a pivot $p$ uniformly at random in the set $\mathcal{V}$ (line 3), and compare it to the rest of the items in the set (line 5). Let $\mathcal{T}_\mathcal{V}$ denote the subgraph of $\mathcal{T}$ induced by $\mathcal{V}$. Given that the comparison outcomes follow from the edges of the tournament, $\mathcal{L}$ is equal to the set of incoming neighbors of $p$ in $\mathcal{T}_\mathcal{V}$. (Correspondingly, $\mathcal{R}$ is equal to the set of the outgoing neighbors.) Hence, the outdegree of $p$ in $\mathcal{T}_\mathcal{V}$ determines how balanced the partition is. The probability that the outdegree of $p$ lies in the middle half of the score sequence is $1/2$, and if it does, Proposition 3.7 tells us that

$$\frac{|\mathcal{V}| - 7}{8} \leq \operatorname{outdeg}(p) \leq \frac{7|\mathcal{V}| - 5}{8}.$$

In this case, at the end of the partition $|\mathcal{L}|$ and $|\mathcal{R}|$ are of size at most $7|\mathcal{V}|/8$, and in at most $\log_{8/7}(N) \leq 8 \log N$ such partitions we get to a subset of size one and match the terminating case. Even though we do not select, every time, the pivot in the middle half, it is unlikely that more than $c \cdot 8 \log N$ recursions are needed (for some small constant $c$) to select the pivot in the middle range at least $8 \log N$ times. Let $z_d$ i.i.d $\sim \mathrm{Bern}(1/2)$ be the indicator variable for the event "the pivot is selected in the middle half at level of recursion $d$". Using a Chernoff bound, we have

$$\mathbf{P}\left[\sum_{d=1}^{\lceil 48 \log N \rceil} z_d \leq 8 \log N\right] \leq \frac{1}{N^2},$$

i.e., the depth of a leaf in the call tree is at most $\lceil 48 \log N \rceil$ with probability at least $1 - 1/N^2$. As there are at most $N$ leaves in the tree, the *maximum* depth is bounded by the same value with probability at least $1 - 1/N$. □

In order to prove Lemma 3.3, we introduce some additional notation. Let $\mathbf{S}_N$ be the set of all permutations on $[N]$. For any $\sigma \in \mathbf{S}_N$ and $\mathcal{V} \subseteq [N]$, let $\sigma_\mathcal{V} : \mathcal{V} \to \{1, \ldots, |\mathcal{V}|\}$ be the ordering induced by $\sigma$ on $\mathcal{V}$. We generalize the definition of displacement as

$$\Delta_\mathcal{V}(\sigma, \tau) = \sum_{i \in \mathcal{V}} |\sigma_\mathcal{V}(i) - \tau_\mathcal{V}(i)|.$$

For conciseness, we use the shorthand $\Delta_\mathcal{V}(\sigma) \doteq \Delta_\mathcal{V}(\sigma, \mathrm{id})$, where id is the identity permutation.

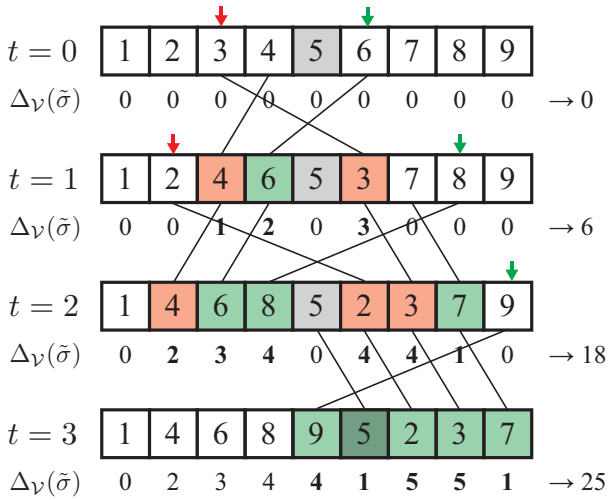



Figure 3.5 – Illustration of the decomposition of $\Delta_{\mathcal{V}}(\tilde{\sigma})$ into contributions of individual errors over a sequence of steps. In this example, $\mathcal{V} = \{1, \ldots, 9\}$, $p = 5$ and there are five errors. At step $t = 1$, we process the errors $(5, 3)$ and $(5, 6)$; at step $t = 2$, we process the errors $(5, 2)$ and $(5, 8)$, and finally, at step $t = 3$, we process the error $(5, 9)$. The shifts caused by an error are highlighted in red and green. In this example, $\Delta_{\mathcal{V}}(\tilde{\sigma}) = 25 < 2\sum_{(i,j)\in\mathcal{E}_{\mathcal{V}}} |i - j| = 26$.

*Proof of Lemma 3.3.* Denote by $\mathcal{A}$ the collection of working sets that were used as input to one of the recursive calls to Quicksort. For $\mathcal{V} \in \mathcal{A}$, let $\mathcal{E}_{\mathcal{V}}$ be the set of pairs sampled by Quicksort to partition $\mathcal{V}$ and which result in an error. Note that $\mathcal{E}_{\mathcal{V}} \cap \mathcal{E}_{\mathcal{V}'} = \varnothing$ for $\mathcal{V} \neq \mathcal{V}'$, and that $\bigcup_{\mathcal{V}} \mathcal{E}_{\mathcal{V}} = \mathcal{E}$. We will show that for all $\mathcal{V} \in \mathcal{A}$,

$$\Delta_{\mathcal{V}}(\sigma) \leq \Delta_{\mathcal{L}}(\sigma) + \Delta_{\mathcal{R}}(\sigma) + 2\sum_{(i,j)\in\mathcal{E}_{\mathcal{V}}} |i - j|, \tag{3.6}$$

where $\mathcal{L}, \mathcal{R} \in \mathcal{A}$ are the two sets obtained at the end of the partition operation. The lemma follows by taking $\mathcal{V} = [N]$ and recursively bounding $\Delta_{\mathcal{L}}(\sigma)$ and $\Delta_{\mathcal{R}}(\sigma)$.

Consider the partition operation on $\mathcal{V}$, with pivot $p$, resulting in partitions $\mathcal{L}$ and $\mathcal{R}$. Let $\tilde{\sigma}$ be the ordering on $\mathcal{V}$ that $(a)$ ranks $\mathcal{L}$ at the bottom, $p$ in the middle and $\mathcal{R}$ at the top, and $(b)$ matches the identity permutation on $\mathcal{L}$ and $\mathcal{R}$, i.e., $\Delta_{\mathcal{L}}(\tilde{\sigma}) = \Delta_{\mathcal{R}}(\tilde{\sigma}) = 0$. In a sense, $\tilde{\sigma}$ is the ordering that would be obtained if there were no further errors in the remaining recursive calls. Using the triangle inequality, we have that

$$\Delta_{\mathcal{V}}(\sigma) \leq \Delta_{\mathcal{V}}(\sigma, \tilde{\sigma}) + \Delta_{\mathcal{V}}(\tilde{\sigma}). \tag{3.7}$$

By definition of $\tilde{\sigma}$, we have that

$$\Delta_{\mathcal{V}}(\sigma, \tilde{\sigma}) = \Delta_{\mathcal{L}}(\sigma, \tilde{\sigma}) + \Delta_{\mathcal{R}}(\sigma, \tilde{\sigma}) = \Delta_{\mathcal{L}}(\sigma) + \Delta_{\mathcal{R}}(\sigma), \tag{3.8}$$

where the first equality follows from $(a)$, and the second follows from $(b)$.

Finally, we bound $\Delta_{\mathcal{V}}(\tilde{\sigma})$. Let $\mathcal{E}_{\mathcal{V}}^{-} = \{(p,i) \in \mathcal{E}_{\mathcal{V}} : i < p\}$, and similarly $\mathcal{E}_{\mathcal{V}}^{+} = \{(p,i) \in \mathcal{E}_{\mathcal{V}} : i > p\}$. Without loss of generality, we can assume that $\mathcal{V}$ consists of consecutive integers, and that $\kappa \doteq |\mathcal{E}_{\mathcal{V}}^{-}| \leq |\mathcal{E}_{\mathcal{V}}^{+}|$. We proceed as follows: starting from the ranking $\mathrm{id}_{\mathcal{V}}$, we progressively incorporate errors into the ranking, ending with $\tilde{\sigma}$ once all errors have been treated. To understand the effect of each error on $\Delta_{\mathcal{V}}(\tilde{\sigma})$, we look at errors in the following specific sequence.

1. At steps $t = 1, \ldots, \kappa$, we consider the $t$-th "smallest" errors in $\mathcal{E}_{\mathcal{V}}^{-}$ and $\mathcal{E}_{\mathcal{V}}^{+}$. That is, we process $(p,i) \in \mathcal{E}_{\mathcal{V}}^{-}$ and $(p,i') \in \mathcal{E}_{\mathcal{V}}^{+}$ such that $|p-i|$ and $|p-i'|$, respectively, are the smallest among errors not yet treated.

2. At steps $t = \kappa+1, \ldots, |\mathcal{E}_{\mathcal{V}}^{+}|$, we process the remaining errors in $\mathcal{E}_{\mathcal{V}}^{+}$, once again in increasing order of distance to $p$.

Figure 3.5 illustrates the state of the ranking at different steps on a concrete example. We start with the first case, i.e., $t \leq \kappa$. The effect of the errors $(p,i)$ and $(p,i')$ on $\Delta_{\mathcal{V}}(\tilde{\sigma})$ is as follows.

- All items $j < i$ and $j > i'$ are not affected by the two errors: their position remains the same.
- The position of the pivot $p$ remains the same, as the two errors balance out.
- Item $i$ is shifted by $|p-i|+1$ positions to the right, just right of $p$. Similarly, item $i'$ is shifted by $|p-i'|+1$ positions to the left, just left of $p$.
- The $|p-i|-1$ items that are between $p$ (excluded) and $i$ are shifted by 1 position to the left. Similarly, the $|p-i'|-1$ items that are between $p$ and $i'$ are shifted by 1 position to the right.

Hence, the two errors contribute $2(|p-i|+|p-i'|)$ towards $\Delta_{\mathcal{V}}(\tilde{\sigma})$. Now consider the second case, when $t > \kappa$. The effect of an error $(p,i)$ is as follows.

- All items $j > i$ and all the items on the left of $p$ are not affected by the error: their position remains the same.
- The (at most) $|p-i|$ items that are between $p$ (included) and $i$ are shifted by 1 position to the right.
- Item $i$ is shifted by at most $|p-i|$ positions to the left, just left of $p$.

As a result, the error contributes at most $2|p - i|$ to the displacement. Adding up the contributions of all the errors, it follows that

$$\Delta_{\mathcal{V}}(\tilde{\sigma}) \leq 2 \sum_{(i,j) \in \mathcal{E}_{\mathcal{V}}} |i - j|. \tag{3.9}$$

Combining (3.8) and (3.9) using (3.7) we obtain (3.6), which concludes the proof. □

### 3.5.2 Theorem 3.4

From now on, we focus on parameters drawn from a Poisson process of rate $\lambda$, as described in Section 3.3.1. We consider a worst-case scenario and assume that Quicksort samples a comparison outcome for every pair of items. Let $z_{ij}$ be the indicator random variable of the event "the comparison between $i$ and $j$ resulted in an error". By Lemma 3.3, we have

$$\Delta(\sigma) \leq 2 \sum_{i<j} |i - j| z_{ij} \tag{3.10}$$

In the following, we will bound some of the statistical properties of the random variables $\{z_{ij}\}$. We start with a lemma that bounds their mean.

**Lemma 3.8.** *For any* $1 \leq i < j \leq N$,

$$\mathbf{E}\left[z_{ij}\right] \leq \left(\frac{\lambda}{\lambda + 1}\right)^{j-i}.$$

*Proof.* Let $d_{ij} = \theta_i - \theta_j$ be the (random) distance between items $i$ and $j$. This distance is a sum of $k = j - i$ independent exponential random variables, and therefore $d_{ij} \sim \text{Gamma}(k, \lambda)$. The comparison outcome is generated as per the BT model; conditioned on the distance $d_{ij}$, the random variable $z_{ij}$ is a Bernoulli trial with probability $[1 + \exp(d_{ij})]^{-1}$. Therefore, we have that

$$\mathbf{E}\left[z_{ij}\right] \leq \mathbf{E}\left[\exp(-d_{ij})\right] = \left(\frac{\lambda}{\lambda + 1}\right)^{k}$$

□

Next, we bound their covariance. Note that the random variables $\{z_{ij}\}$ are in general *not* unconditionally independent. They become independent only when conditioned on $\boldsymbol{\theta}$.

**Lemma 3.9.** *For any $1 \leq i < j \leq N$ and any $1 \leq u < v \leq N$, let $\mathcal{A} = \{i \,..\, j-1\}$ and $\mathcal{B} = \{u \,..\, v-1\}$.*

$$\mathbf{Cov}\left[z_{ij}, z_{uv}\right] \leq \begin{cases} 0 & \text{if } \mathcal{A} \cap \mathcal{B} = \varnothing, \\ \left(\frac{\lambda}{\lambda+1}\right)^{j-i} & \text{if } \mathcal{A} = \mathcal{B}, \\ \left(\frac{\lambda+1}{\lambda+2}\right)^{j-i+v-u} & \text{otherwise.} \end{cases}$$

*Proof.* If $\mathcal{A}$ and $\mathcal{B}$ are disjoint, the distances $d_{ij}$ and $d_{uv}$ are independent random variables. Conditioned on the distances, the comparison outcomes are independent Bernoulli trials, and we conclude that $z_{ij}$ and $z_{uv}$ are independent. In the two remaining cases, we bound $\mathbf{E}\left[z_{ij}z_{uv}\right] \geq \mathbf{Cov}\left[z_{ij}, z_{uv}\right]$. If $\mathcal{A} = \mathcal{B}$, then $z_{ij} = z_{uv}$ and we have

$$\mathbf{E}\left[z_{ij}z_{uv}\right] = \mathbf{E}\left[z_{ij}^2\right] = \mathbf{E}\left[z_{ij}\right]$$

and we apply Lemma 3.8. Finally, if $\mathcal{A}$ and $\mathcal{B}$ are neither equivalent nor disjoint, the two comparison outcomes are independent Bernoulli trials conditioned on the distances $d_{ij}$ and $d_{uv}$, but the distances are not independent. Consider the case where $i < u < j < v$. Even though $d_{ij}$ and $d_{uv}$ are dependent, the distances $d_{iu}$, $d_{uj}$, $d_{jv}$ are independent Gamma random variables of rate $\lambda$ and shape $u - i$, $j - u$ and $v - j$, respectively, and

$$\begin{aligned} \mathbf{E}\left[z_{ij}z_{uv}\right] &\leq \mathbf{E}\left[\exp\{-(d_{iu} + d_{uj}) - (d_{uj} + d_{jv})\}\right] \\ &= \left(\frac{\lambda}{\lambda+1}\right)^{u-i}\left(\frac{\lambda}{\lambda+2}\right)^{j-u}\left(\frac{\lambda}{\lambda+1}\right)^{v-j} \leq \left(\frac{\lambda+1}{\lambda+2}\right)^{j-i+v-u} \end{aligned}$$

The other cases are treated analogously. □

Lemmas 3.8 and 3.9 will be useful in proving the first part of Theorem 3.4. For the second part, we need a result from Ailon [2008], which characterizes the pairwise marginals of the distribution over rankings induced by Quicksort with comparisons sampled from a BT model.

**Theorem 3.10** (Ailon, 2008, Theorem 4.1)**.** *Let $\sigma$ be the output of Quicksort using comparison outcomes sampled from* $\mathrm{BT}(\boldsymbol{\theta})$*. Then, for any $i, j \in [N]$,*

$$\mathbf{P}\left[\sigma(i) < \sigma(j) \mid \boldsymbol{\theta}\right] = \mathbf{P}\left[i \prec j \mid \boldsymbol{\theta}\right]$$

Note that the result is non-trivial as $i$ and $j$ might not have been directly compared to each other: their relative position might have been deduced by transitivity from other comparison outcomes. We are now ready to prove Theorem 3.4.

*Proof of Theorem 3.4.* We begin with the first part of the theorem, which bounds the displacement $\Delta(\sigma)$. For clarity of exposition, we use the notation $z_{i\to k}$ instead of $z_{ij}$ if $j = i + k$. Using (3.10) and Lemma 3.8, we can bound the expected displacement as

$$\mathbf{E}\left[\Delta\right] \leq \sum_{i=1}^{N-1}\sum_{k=1}^{N-i} 2k\mathbf{E}\left[z_{i\to k}\right] \leq N\sum_{k=1}^{\infty} 2k\left(\frac{\lambda}{\lambda+1}\right)^k = 2N\lambda(\lambda+1).$$

In a similar way, using Lemma 3.9, we can bound the variance of the displacement as

$$\begin{aligned}\mathbf{Var}\left[\Delta\right] &\leq \sum_{i=1}^{N-1}\sum_{k=1}^{N-i} 4k^2\mathbf{Var}\left[z_{i\to k}\right] + 2\sum_{i=1}^{N-1}\sum_{k=1}^{N-i} 2k\sum_{u=i+1}^{i+k}\sum_{\ell=1}^{N-u} 2\ell\mathbf{Cov}\left[z_{i\to k}, z_{u\to \ell}\right] \\ &\leq N\sum_{k=1}^{\infty} 4k^2\left(\frac{\lambda}{\lambda+1}\right)^k + 2N\sum_{k=1}^{\infty} 2k^2\left(\frac{\lambda+1}{\lambda+2}\right)^k \cdot \sum_{\ell=1}^{\infty} 2\ell\left(\frac{\lambda+1}{\lambda+2}\right)^\ell \\ &\leq 1500N(\lambda^5+1).\end{aligned}$$

Combining the bounds for the mean and the variance with Chebyshev's inequality, we have that

$$\mathbf{P}\left[\Delta(\sigma) \geq 50N(\lambda^2+1)\right] \leq \lambda/N,$$

which concludes the proof of the first part of the claim.

The second part of the theorem bounds the maximum displacement for any single item. We start by showing that with high probability, there is no pair of items separated by at least $O(\lambda \log N)$ positions that is "flipped" in the output of Quicksort. Let $i$ and $j$ be two items such that $i < j$ and let $k = |i - j|$. Then $d_{ij} \sim \text{Gamma}(k, \lambda)$, and using a Chernoff bound we obtain

$$\mathbf{P}\left[d_{ij} \leq k/(e\lambda)\right] \leq \exp(-k/e).$$

If $k \geq 3(\lambda+1)e\log N$, we find that

$$\mathbf{P}\left[d_{ij} \leq k/(e\lambda)\right] \leq \mathbf{P}\left[d_{ij} \leq 3\log N\right] \leq N^{-3}. \tag{3.11}$$

Using the fact that the pairwise marginals of Quicksort match the pairwise comparison outcome probabilities (Theorem 3.10), we find

$$\mathbf{P}\left[\sigma(j) < \sigma(i)\right] = \mathbf{P}\left[j \prec i\right] \leq \exp(-3\log N) = N^{-3}. \tag{3.12}$$

Combining (3.11) and (3.12), and using a union bound over the $\binom{N}{2}$ pairs, we see that with probability $1-1/N$ there is no pair of items $(i, j)$ separated by at least $3(\lambda+1)e\log N$ position with $i < j$ but $\sigma(j) < \sigma(i)$. Finally, suppose that there is an $i$ such $|\sigma(i) - i| = k$. Without loss of generality, we can assume that $i < \sigma(i)$. This means that there are $k$

items larger than $i$ that are on the left of $i$ in $\sigma$. In particular, there is an item $j > i$ such that $|i - j| \geq k$ and $\sigma(j) < \sigma(i)$. This concludes the proof. □

### 3.5.3 Theorem 3.6

In order to prove Theorem 3.6, we first need a basic result on the order statistics of exponential random variables. Let $x_1, \ldots, x_N$, be i.i.d. exponential random variables of rate $\lambda$. Let $x_{(1)}, \ldots, x_{(N)}$ be their order statistics, i.e., the random variables arranged in increasing order. Then,

$$x_{(n)} = \sum_{i=1}^{n} \frac{1}{N - i + 1} y_i, \tag{3.13}$$

where $y_1, \ldots, y_N$ are i.i.d. exponential random variables of rate $\lambda$ [see, e.g., Arnold et al., 2008, Section 4.6].

*Proof of Theorem 3.6.* We consider the order statistics of the $N - 1$ i.i.d. exponential random variables $x_1, \ldots, x_{N-1}$ which define the distances between neighboring items. Let $N' = \lceil N/\log^2 N \rceil$, and denote by $\mathcal{B} \subset [N]$ the set of items at both ends of $x_{(1)}, \ldots, x_{(N'-1)}$. These "bad" items are close to their nearest neighbor, and we simply invoke Theorem 3.4 to claim that each of these items is shifted by at most $O(\lambda \log N)$ positions with high probability. Consider now the "good" items, i.e., those in $\mathcal{G} = [N] \setminus \mathcal{B}$. Using (3.13) and for $N$ large enough,

$$\mathbf{P}\left[x_{(N')} \leq 1/(e\lambda \log^2 N)\right] \leq \mathbf{P}\left[\sum_{i=1}^{N'} y_i/N \leq 1/(e\lambda \log^2 N)\right] \leq \exp(-N'/e) \leq 1/N.$$

The second-to-last inequality follows from a Chernoff bound similar to that used in the proof of Theorem 3.4. Therefore, with high probability, all items in $\mathcal{G}$ are at distance larger than $c/(\lambda \log^2 N)$ from their nearest neighbor for some constant $c$.

We will now show that after $K = O(\lambda^2 \log^5 N)$ runs of Quicksort, $\hat{\sigma}(i) = i$ with high probability for all $i \in \mathcal{G}$. Let $i \in \mathcal{G}$, $j \in [N]$ be a pair of items, and without loss of generality assume that $i < j$. Let $t_k$ be the indicator random variable for the event "$\sigma(i) < \sigma(j)$ in the $k$-th run of Quicksort", and let $p = \mathbf{P}[t_k = 1]$. Then, using Theorem 3.10,

$$\begin{aligned} p - \frac{1}{2} = \mathbf{P}[i \prec j] - \frac{1}{2} &= \frac{1 - \exp(-d_{ij})}{2[1 + \exp(-d_{ij})]} \\ &\geq \frac{1 - \exp[-1/(e\lambda \log^2 N)]}{4} \geq \frac{1}{8e\lambda \log^2 N} \end{aligned}$$

with high probability. In the last inequality, we used the fact that $1 - e^{-x} \geq x/2$ for $x \in [0, 1]$. The random variables $t_1, \ldots, t_K$ are independent Bernoulli trials, and using a

Chernoff bound we obtain

$$\begin{aligned}\mathbf{P}\left[\hat{\sigma}(j) < \hat{\sigma}(i)\right] &= \mathbf{P}\left[\sum_{k=1}^{K} t_k \leq K/2\right] \\ &\leq \exp[-2K(p-1/2)^2] \leq \exp\left[-\frac{K}{32e^2\lambda^2 \log^4 N}\right].\end{aligned}$$

By choosing $K = 96e^2\lambda^2 \log^5 N$, we have $\mathbf{P}\left[\hat{\sigma}(j) < \hat{\sigma}(i)\right] \leq N^{-3}$, and using a union bound we see that with probability $1 - 1/N$ we have $\hat{\sigma}(i) = i$ for all $i \in \mathcal{G}$. Therefore, the total displacement is

$$\Delta(\hat{\sigma}) = \sum_{i \in \mathcal{B}} |\hat{\sigma}(i) - i| \leq |\mathcal{B}| \cdot 3(\lambda + 1)e \log N = O(\lambda N / \log N).$$

This concludes the proof. □

## 3.6 Summary

We have demonstrated that active learning can substantively accelerate the task of learning a ranking from noisy comparisons gains—both in theory and in practice. With the advent of large-scale crowdsourced ranking surveys, exemplified by GIFGIF and wiki surveys [Salganik and Levy, 2015], there is a clear need for practical AL strategies. However, existing methods are complex and computationally expensive to operate even for a reasonable number of items (a few thousands). We have shown that a deceptively simple idea—repeatedly sorting the items—is able to bring in all the benefits of active learning, is trivial to implement, and is computationally no more expensive that random sampling.

# 4 Choices in Networks

In this chapter[1], we address the problem of understanding how users navigate in a network of $N$ nodes. We consider a setting where only aggregate node-level traffic is observed and tackle the task of learning edge-transition probabilities. We cast it as a preference-learning problem and study a model where choices follow a variant of Luce's axiom. In this case, the $O(N)$ marginal counts of node visits are a sufficient statistic for the $O(N^2)$ transition probabilities. We show how to make the inference problem well-posed, regardless of the network's structure, and we develop an iterative algorithm that scales to networks that contain billions of nodes and edges. We apply the model to two clickstream datasets and show that it successfully recovers the transition probabilities by using only the network structure and marginal (node-level) traffic data. Finally, we also consider an application to mobility networks and apply the model to one year of rides on New York City's bicycle-sharing system.

## 4.1 Introduction

Consider the problem of estimating click probabilities for links between pages of a website, given a hyperlink graph and aggregate statistics on the number of times each page has been visited. Naively, we might expect that the probability of clicking on a particular link should be roughly proportional to the traffic of the link's target. However, this neglects important structural effects: a page's traffic is influenced by ($a$) the number of incoming links, ($b$) the traffic at the pages that link to it, and ($c$) the traffic absorbed by competing links. In order to successfully infer click probabilities, it is therefore necessary to disentangle the *preference* for a page (i.e., the intrinsic propensity of a user to click on a link pointing to it) from the page's *visibility* (the exposure it gets from pages linking to it). Building upon recent work by Kumar et al. [2015], we present a statistical framework that tackles a general formulation of the problem: Given a network (representing possible transitions between nodes) and the marginal traffic at each node, recover the transition

[1]This chapter is based on Maystre and Grossglauser [2017a].

probabilities. This problem is relevant to a number of scenarios (in social, information or transportation networks) where transition data is not available due to privacy concerns or monitoring costs, for example.

We begin by postulating the following model of traffic. Users navigate from node to node along the edges of the network, by making a choice between adjacent nodes at each step, which is reminiscent of the random-surfer model introduced by Brin and Page [1998]. Choices are assumed to be independent and generated according to a variant of Luce's model [Luce, 1959]: each node in the network is characterized by a latent *strength* parameter, and (stochastic) choice outcomes tend to favor nodes with greater strengths. In this model, estimating the transition probabilities amounts to estimating the strength parameters. Unlike in the setting in which choice models are traditionally studied [Train, 2009, Vojnovic and Yun, 2016], we do not observe distinct choices among well-identified sets of alternatives. Instead, we only have access to aggregate, marginal statistics about the traffic at each node in the network. In this setting, we make the following contributions.

1. We observe that marginal per-node traffic is a sufficient statistic for the strength parameters. That is, the parameters can be inferred from marginal traffic data without any loss of information.

2. We show that if the parameters are endowed with a prior distribution, the inference problem becomes well-posed, regardless of the network structure. This is a crucial step in making the framework applicable to real-world datasets.

3. We show that model inference can scale to very large datasets. We present an iterative EM-type inference algorithm that enables a remarkably efficient implementation—each iteration requires the computational equivalent of two iterations of PageRank.

We evaluate two aspects of our framework by using real-world networks. We begin by demonstrating that local preferences can indeed be inferred from global traffic: we investigate the accuracy of the transition probabilities recovered by our model on three datasets for which we have ground-truth transition data. First, we consider two hyperlink graphs that represent the English Wikipedia (over two million nodes) and a Hungarian news portal (approximately 40 000 nodes), respectively. We model clickstream data as a sequence of independent choices over the links available at each page. Given only the structure of the graph and the marginal traffic at every node, we estimate the number of transitions between nodes, and we find that our estimate matches ground-truth edge-level transitions accurately in both instances. Second, we consider the network of New York City's bicycle-sharing service. For a given ride, given a pick-up station, we model the drop-off station as a choice out of a set of locations. Our model yields promising results,

suggesting that our method can be useful beyond clickstream data. Next, we test the scalability of the inference algorithm. We show that the algorithm is able to process a snapshot of the WWW hyperlink graph that contains over a hundred billion edges using a single machine.

**Outline of the Chapter** In Section 4.2, we briefly review related literature. In Section 4.3, we formalize the network choice model and present some important statistical properties of the model. In Section 4.4, we propose a prior distribution that makes the inference problem well-posed and describe an inference algorithm that enables an efficient implementation. We evaluate the model and the inference algorithm in Section 4.5.

## 4.2 Related Work

A variant of the network choice model was recently introduced by Kumar et al. [2015], in an article that lays much of the groundwork for this chapter. Their generative model of traffic and the parametrization of transition probabilities based on Luce's axiom form the basis of our work. Kumar et al. define the *steady-state inversion* problem as follows: Given a directed graph $\mathcal{G}$ and a target stationary distribution, find transition probabilities that lead to the desired stationary distribution. This problem formulation assumes that $\mathcal{G}$ satisfies restrictive structural properties (strong-connectedness, aperiodicity) and is valid only asymptotically, when each user's sequence of choices is very long. Our formulation is, in contrast, more general. In particular, we eliminate any assumptions about the structure of $\mathcal{G}$ and cope with finite data in a principled way—in fact, our derivations are valid for choice sequences of any length. One of our contributions is to explain the steady-state inversion problem in terms of (asymptotic) maximum-likelihood inference in the network choice model. Furthermore, the statistical viewpoint that we develop also leads to (*a*) a robust regularization scheme that lets us handle graphs of arbitrary structure, and (*b*) a simple and efficient EM-type inference algorithm. These important extensions make the model easier to apply to real-world data.

**Luce's Choice Axiom** The general problem of estimating parameters of models based on Luce's axiom has received considerable attention, as discussed at length in Chapters 1 and 2. Of particular interest in the context of this work, Hunter [2004] develops an inference algorithm from the perspective of the minorization-maximization (MM) method. This method is easily generalized to other models that are based on Luce's axiom, and it yields simple, provably convergent algorithms for maximum-likelihood (ML) or maximum-a-posteriori point estimates. Caron and Doucet [2012] observe that, by introducing suitable latent variables, these MM algorithms can be further recast as expectation-maximization (EM) algorithms. They use this observation to derive Gibbs samplers for a wide family of models. We take advantage of this line of work, in Section 4.4,

when developing an inference algorithm for the network choice model. In recent years, several authors also analyzed the sample complexity of the ML estimate in Luce's choice model [Hajek et al., 2014, Vojnovic and Yun, 2016] and investigated alternative spectral inference methods [Negahban et al., 2012, Azari Soufiani et al., 2013]. Some of these results could be applied to our setting, but in general they require observing distinct choices among well-identified sets of alternatives[2].

**Network Analysis** Understanding the preferences of users in networks is of significant interest in many domains. For brevity, we focus mostly on literature related to hyperlink graphs. A method that has undoubtedly had a tremendous impact in this context is PageRank [Brin and Page, 1998]. PageRank computes a set of scores that are proportional to the amount of time a surfer, who clicks on links randomly and uniformly, spends at each node. These scores are based only on the structure of the graph. The network choice model presented in this chapter appears to be similar at first, but tackles a different problem. In addition to the structure of the graph, it uses the traffic at each page, and computes a set of scores that reflect the (non-uniform) probability of clicking on each link. Nevertheless, there are striking similarities in the implementation of the respective inference algorithms (see Section 4.5). The HOTness method proposed by Tomlin [2003] is somewhat related, but tries to tackle a harder problem. It attempts to estimate jointly the traffic and the probability of clicking on each link, by using a maximum-entropy approach. At the other end of the spectrum, BrowseRank [Liu et al., 2008] uses detailed data collected in users' browsers to improve on PageRank. Our method uses only marginal traffic data that can be obtained without tracking users. Finally, in the context of mobility analysis, we mention the works of Ashbrook and Starner [2003] and Kafsi et al. [2015], both of which use Markovian models to predict human mobility across in a network of locations.

## 4.3 Network Choice Model

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a directed graph on $N$ nodes (corresponding to items) and $M$ edges, with edge weights $w_{ij} > 0$ for all $(i, j) \in \mathcal{E}$. We denote the out-neighborhood of node $i$ by $\mathcal{N}_i^+$ and its in-neighborhood by $\mathcal{N}_i^-$. We consider the following choice process on $\mathcal{G}$. A user starts at a node $i$ and is faced with alternatives $\mathcal{N}_i^+$. The user chooses item $j$ and moves to the corresponding node. At node $j$, the user is faced with alternatives $\mathcal{N}_j^+$ and chooses $k$, and so on. At any time, the user can stop. Figure 4.1 gives an example of a graph and the alternatives available at a step of the process.

To define the transition probabilities, we follow Kumar et al. [2015] and posit a probabilistic model of choice, which extends that of Luce [1959]. For every node $i$ and every

[2]This is also the case for the spectral inference algorithm developed in Chapter 2 of this thesis, which cannot be applied in the setting studied in this chapter.

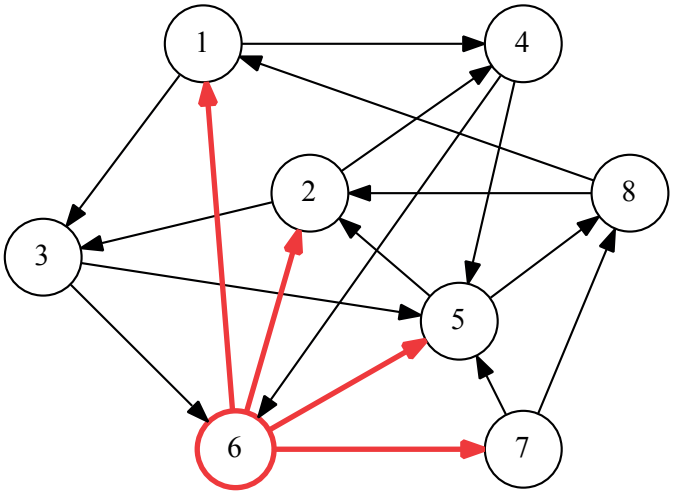



Figure 4.1 – An illustration of one step of the process. The user is at node 6 and can reach nodes $\mathcal{N}_6^+ = \{1, 2, 5, 7\}$.

$j \in \mathcal{N}_i^+$, the probability that $j$ is selected among alternatives $\mathcal{N}_i^+$ can be written as

$$p_{ij} = \frac{w_{ij}\gamma_j}{\sum_{k \in \mathcal{N}_i^+} w_{ik}\gamma_k}, \tag{4.1}$$

for some parameter vector $\boldsymbol{\gamma} = [\gamma_1 \ \cdots \ \gamma_N]^\top \in \mathbf{R}_{>0}^N$. Intuitively, the parameter $\gamma_i$ can be interpreted as the utility of item $i$. The edge weights are relevant in situations where the current context modulates the alternatives' utility; for example, they can be used to encode the position or prominence of a link on a page in a hyperlink graph, or the distance between two locations in a mobility network. Luce's original choice model is obtained by setting $w_{ij} \doteq 1$ for all $i, j$. Note that $p_{ij}$ depends only on the out-neighborhood of node $i$. As such, the choice process satisfies the Markov property, and we can think of the sequence of choices as a trajectory in a Markov chain.

In the context of this model, we can formulate the inference problem as follows. Given a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, edge weights $\{w_{ij}\}$ and data on the aggregate traffic at each node, find a parameter vector $\boldsymbol{\gamma}$ that fits the data.

**Notation** In some expressions, we use $\kappa$ to denote a constant that does not depend on the parameter vector $\boldsymbol{\gamma}$. Its value can change from line to line.

### 4.3.1 Sufficient Statistic

We begin by showing that $O(N)$ values summarizing the aggregate traffic at each node are a sufficient statistic of the transition counts. Let $c_{ij}$ denote the number of transitions that occurred along edge $(i, j) \in \mathcal{E}$. Starting from the transition probability defined

in (4.1), we can write the log-likelihood of $\boldsymbol{\gamma}$ given data $\mathcal{D} = \{c_{ij} : (i,j) \in \mathcal{E}\}$ as

$$\begin{aligned}\ell(\boldsymbol{\gamma}; \mathcal{D}) &= \sum_{(i,j)\in\mathcal{E}} c_{ij}\left[\log w_{ij}\gamma_j - \log \sum_{k\in\mathcal{N}_i^+} w_{ik}\gamma_k\right] \\ &= \sum_{j=1}^{N}\sum_{i\in\mathcal{N}_j^-} c_{ij}\log\gamma_j - \sum_{i=1}^{N}\sum_{j\in\mathcal{N}_i^+} c_{ij}\log\sum_{k\in\mathcal{N}_i^+} w_{ik}\gamma_k + \sum_{(i,j)\in\mathcal{E}} c_{ij}\log w_{ij}, \\ &= \sum_{i=1}^{N}\left[c_i^-\log\gamma_i - c_i^+\log\sum_{k\in\mathcal{N}_i^+} w_{ik}\gamma_k\right] + \kappa, \end{aligned} \tag{4.2}$$

where $c_i^- = \sum_{j\in\mathcal{N}_i^-} c_{ji}$ and $c_i^+ = \sum_{j\in\mathcal{N}_i^+} c_{ij}$ is the aggregate number of transitions arriving in and originating from $i$, respectively. This formulation of the log-likelihood exhibits a key feature of the model: the set of $2N$ counts $\{(c_i^-, c_i^+) : i \in \mathcal{V}\}$ is a sufficient statistic of the $M = O(N^2)$ counts $\{c_{ij} : (i,j) \in \mathcal{E}\}$ for the parameters $\boldsymbol{\gamma}$. The following theorem is an extension of a well-known result for Luce's choice model [Bühlmann and Huber, 1963].

**Theorem 4.1.** *The set of aggregate transitions $\{(c_i^-, c_i^+) : i \in \mathcal{V}\}$ is a minimally sufficient statistic for the parameters $\boldsymbol{\gamma}$.*

*Proof.* Let $f(\{c_{ij}\} \mid \boldsymbol{\gamma})$ be the discrete probability density function of the data under the model with parameters $\boldsymbol{\gamma}$. By Theorem 6.2.13 in Casella and Berger [2002], $\{(c_i^-, c_i^+)\}$ is a minimally sufficient statistic for $\boldsymbol{\gamma}$ if and only if, for any $\{c_{ij}\}$ and $\{d_{ij}\}$ in the support of $f$,

$$\frac{f(\{c_{ij}\} \mid \boldsymbol{\gamma})}{f(\{d_{ij}\} \mid \boldsymbol{\gamma})} \text{ is independent of } \boldsymbol{\gamma} \iff (c_i^-, c_i^+) = (d_i^-, d_i^+) \quad \forall i. \tag{4.3}$$

Taking the log of the ratio on the left-hand side and using (4.2), we find that

$$\log\frac{f(\{c_{ij}\} \mid \boldsymbol{\gamma})}{f(\{d_{ij}\} \mid \boldsymbol{\gamma})} = \sum_{i=1}^{N}\left[(c_i^- - d_i^-)\log\gamma_i - (c_i^+ - d_i^+)\log\sum_{k\in\mathcal{N}_i^+} w_{ik}\gamma_k\right] + \kappa.$$

From this, it is easy to see that the ratio of densities is independent of $\boldsymbol{\gamma}$ if and only if $c_i^- = d_i^-$ and $c_i^+ = d_i^+$, which verifies (4.3). □

In other words, it is enough to observe marginal information about the number of arrivals and departures at each node—we call this collective data the *traffic* at a node—and no additional information can be gained by observing the full choice process. This makes the model particularly attractive, because it means that it is unnecessary to track users across nodes. In several applications of practical interest, tracking users is undesirable, difficult, or outright impossible, due to (*a*) privacy reasons, (*b*) monitoring costs, or (*c*) lack of data in existing datasets.

Note that if we make the additional assumption that the flow in the network is conserved, then $c_i^- = c_i^+$. If users' typical trajectories are made of many hops, it is reasonable to approximate $c_i^-$ or $c_i^+$ by using this assumption, should one of the two quantities be missing.

### 4.3.2 Steady-State Inversion Problem

In recent work, Kumar et al. [2015] define the problem of *steady-state inversion* as follows: Given a strongly-connected directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with edge weights $\{w_{ij}\}$ and a target distribution over the nodes $\boldsymbol{\pi}$, find the transition matrix of a Markov chain on $\mathcal{G}$ with stationary distribution $\boldsymbol{\pi}$. As there are $M = O(N^2)$ degrees of freedom (the transition probabilities) for $N$ constraints (the stationary distribution), the problem is in most cases underdetermined. Following Luce's ideas, the transition probabilities are constrained to be proportional to a latent score of the destination node as per (4.1), thus reducing the number of parameters from $M$ to $N$. Denote by $\boldsymbol{P}(\boldsymbol{s})$ the Markov-chain transition matrix parametrized with scores $\boldsymbol{s}$. The score vector $\boldsymbol{s}$ is a solution for the steady-state inversion problem if and only if $\boldsymbol{\pi}^\top = \boldsymbol{\pi}^\top \boldsymbol{P}(\boldsymbol{s})$, or equivalently

$$\pi_i = \sum_{j \in \mathcal{N}_i^-} \frac{w_{ji} s_i}{\sum_{k \in \mathcal{N}_j^+} w_{jk} s_k} \pi_j \quad \forall i. \tag{4.4}$$

In order to formalize the connection between Kumar et al.'s work and ours, we express the steady-state inversion problem as that of asymptotic maximum-likelihood estimation in the network choice model. Suppose that we observe node-level traffic data $\mathcal{D} = \{(c_i^-, c_i^+) : i \in \mathcal{V}\}$ about a trajectory of length $T$ starting at an arbitrary node. We want to obtain an estimate of the parameters $\boldsymbol{\gamma}^\star$ by maximizing the average log-likelihood $\hat{\ell}(\boldsymbol{\gamma}) = \frac{1}{T}\ell(\boldsymbol{\gamma}; \mathcal{D})$. From standard convergence results for Markov chains [Kemeny and Snell, 1976], it follows that as $\mathcal{G}$ is strongly connected, $\lim_{T\to\infty} c_i^-/T = \lim_{T\to\infty} c_i^+/T = \pi_i$. Therefore,

$$\hat{\ell}(\boldsymbol{\gamma}) = \sum_{i=1}^N \left[ \frac{c_i^-}{T} \log \gamma_i - \frac{c_i^+}{T} \log \sum_{k \in \mathcal{N}_i^+} w_{ik}\gamma_k \right] \xrightarrow{T\to\infty} \sum_{i=1}^N \pi_i \left[ \log \gamma_i - \log \sum_{k \in \mathcal{N}_i^+} w_{ik}\gamma_k \right].$$

Let $\boldsymbol{\gamma}^\star$ be a maximizer of the average log-likelihood. When $T \to \infty$, the optimality condition $\nabla\hat{\ell}(\boldsymbol{\gamma}^\star) = \mathbf{0}$ implies, for all $i$,

$$\begin{aligned} \left.\frac{\partial \hat{\ell}(\boldsymbol{\gamma})}{\partial \gamma_i}\right|_{\boldsymbol{\gamma}=\boldsymbol{\gamma}^\star} &= \frac{\pi_i}{\gamma_i^\star} - \sum_{j \in \mathcal{N}_i^-} \frac{w_{ji}\pi_j}{\sum_{k \in \mathcal{N}_j^+} w_{jk}\gamma_k^\star} = 0 \\ \iff \pi_i &= \sum_{j \in \mathcal{N}_i^-} \frac{w_{ji}\gamma_i^\star}{\sum_{k \in \mathcal{N}_j^+} w_{jk}\gamma_k^\star} \pi_j. \end{aligned} \tag{4.5}$$

Comparing (4.5) to (4.4), it is clear that $\boldsymbol{\gamma}^\star$ is a solution of the steady-state inversion problem. As such, the network choice model presented in this chapter can be viewed as a principled extension of the steady-state inversion problem to the finite-data case.

### 4.3.3 MLE

The log-likelihood (4.2) is not concave in $\boldsymbol{\gamma}$, but it can be made concave by using the standard reparametrization $\gamma_i = e^{\theta_i}$. Therefore, any local minimum of the likelihood is a global minimum (c.f. Section 2.1.1). Unfortunately, it turns out that the conditions guaranteeing that the ML estimate is well-defined (i.e., that it exists and is unique) are restrictive and impractical. The following definition extends the notion of comparison graph of Section 2.1.1 to the case of choices in networks.

**Definition** (comparison graph)**.** Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a directed graph and $\{a_{ij} : (i,j) \in \mathcal{E}\}$ be non-negative numbers. The *comparison graph* induced by $\{a_{ij}\}$ is the directed graph $\mathcal{G}' = (\mathcal{V}, \mathcal{E}')$, where $(i,j) \in \mathcal{E}'$ if and only if there is a node $k$ such that $i, j \in \mathcal{N}_k^+$ and $a_{kj} > 0$.

The numbers $\{a_{ij}\}$ in the definition can be loosely interpreted as transition counts (although they do not need to be integers). Intuitively, there is an edge $(i,j)$ in the comparison graph whenever there is at least one instance in which $i$ and $j$ are among the alternatives and $j$ is selected. The notion of comparison graph leads to a precise characterization of whether the ML estimate is well-defined or not, as shown by the next theorem—an extension of Theorem 2.1 to the network choice model.

**Theorem 4.2.** *Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a weighted, directed graph and $\{(c_i^-, c_i^+)\}$ be the aggregate number of transitions arriving in and originating from $i$, respectively. Let $\{a_{ij}\}$ be any set of non-negative real numbers that satisfy*

$$\sum_{j \in \mathcal{N}_i^-} a_{ji} = c_i^-, \qquad \sum_{j \in \mathcal{N}_i^+} a_{ij} = c_i^+ \quad \forall i.$$

*Then, the maximizer of the log-likelihood* (4.2) *exists and is unique (up to rescaling) if and only if the comparison graph induced by $\{a_{ij}\}$ is strongly connected.*

*Proof.* The proof borrows from Hunter [2004], in particular from the proofs of Lemmas 1 and 2. Using $\gamma_i = e^{\theta_i}$, we can rewrite the reparametrized log-likelihood using $\{a_{ij}\}$ as

$$\ell(\boldsymbol{\theta}) = \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i^+} a_{ij} \Big[ \theta_j - \log \sum_{k \in \mathcal{N}_i^+} w_{ik} e^{\theta_k} \Big],$$

and, without loss of generality, we can assume that $\sum_i \theta_i = 0$ and $\min_{ij} w_{ij} = 1$. We study the conditions under which $(a)$ super-level sets of the likelihood function $\ell(\boldsymbol{\theta})$ are bounded, and $(b)$ the likelihood function is strictly concave.

First, we prove that the super-level set $\{\boldsymbol{\theta} : \ell(\boldsymbol{\theta}) \geq c\}$ is bounded and compact for any $c$, if and only if the comparison graph is strongly connected. The compactness of all super-level sets ensures that there is at least one maximizer. Pick any unit-norm vector $\boldsymbol{u}$ such that $\sum_i u_i = 0$, and let $\boldsymbol{\theta} = s\boldsymbol{u}$ When $s \to \infty$, then $e^{\theta_i} > 0$ and $e^{\theta_j} \to 0$ for some $i$ and $j$. As the comparison graph is strongly connected, there is a path from $i$ to $j$, and along this path there must be two consecutive nodes $i', j'$ such that $e^{\theta_{i'}} > 0$ and $e^{\theta_{j'}} \to 0$. The existence of the edge $(i', j')$ in the comparison graph means that there is a $k$ such that $i', j' \in \mathcal{N}_k^+$ and $a_{kj'} > 0$. Therefore, the log-likelihood can be bounded as

$$\ell(\boldsymbol{\theta}) \leq a_{kj'} \left[ \theta_{j'} - \log \sum_{q \in \mathcal{N}_k^+} w_{kq} e^{\theta_q} \right] \leq a_{kj'} \left[ \theta_{j'} - \log(e^{\theta_{j'}} + e^{\theta_{i'}}) \right],$$

and $\lim_{s \to \infty} \ell(\boldsymbol{\theta}) = -\infty$. Conversely, suppose that the comparison graph is not strongly connected and partition the vertices into two non-empty subsets $\mathcal{S}$ and $\mathcal{T}$ such that there is no edge from $\mathcal{S}$ to $\mathcal{T}$. Let $c > 0$ be any positive constant, and take $\tilde{\theta}_i = \theta_i + c$ if $i \in \mathcal{S}$ and $\tilde{\theta}_i = \theta_i$ if $i \in \mathcal{T}$ (renormalize such that $\sum_i \tilde{\theta}_i = 0$). Clearly, $\ell(\tilde{\boldsymbol{\theta}}) \geq \ell(\boldsymbol{\theta})$, and, by repeating this procedure, $\|\boldsymbol{\theta}\|$ can be driven to infinity without decreasing the likelihood.

Second, we prove that if the comparison graph is strongly connected, the log-likelihood is strictly concave (in $\boldsymbol{\theta}$). In particular, for any $p \in (0, 1)$,

$$\ell\left[p\boldsymbol{\theta} + (1-p)\boldsymbol{\eta}\right] \geq p\ell(\boldsymbol{\theta}) + (1-p)\ell(\boldsymbol{\eta}), \tag{4.6}$$

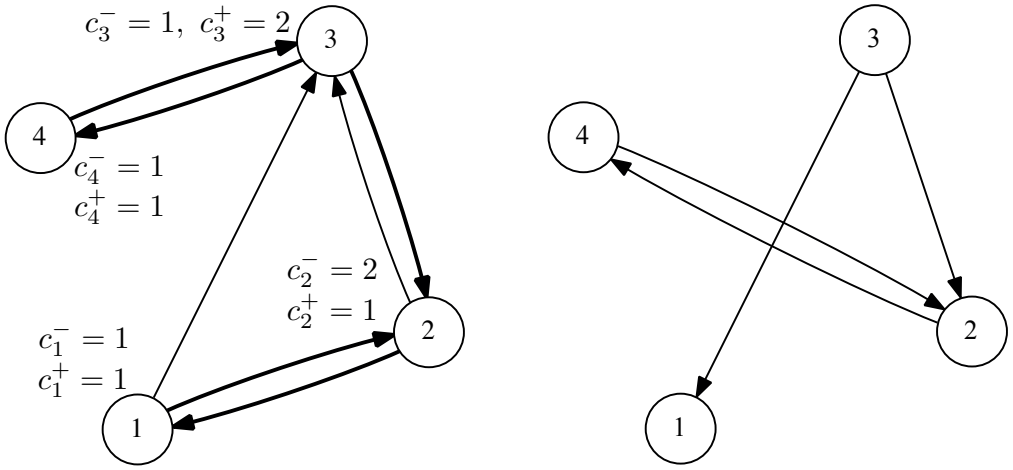
with equality if and only if $\boldsymbol{\theta} \equiv \boldsymbol{\eta}$ up to a constant shift. Strict concavity ensures that there is at most one maximizer of log-likelihood. We start with Hölder's inequality, which implies that, for positive $\{x_k\}$ and $\{y_k\}$, and $p \in (0, 1)$,

$$\log \sum_k x_k^p y_k^{1-p} \leq p \log \sum_k x_k + (1-p) \log \sum_k y_k.$$

with equality if and only $x_k = c y_k$ for some $c > 0$. Letting $x_k = w_{ik} e^{\theta_k}$ and $y_k = w_{ik} e^{\eta_k}$, we find that for all $i$

$$\log \sum_{k \in \mathcal{N}_i^+} w_{ik} e^{p\theta_k + (1-p)\eta_k} \leq p \log \sum_{k \in \mathcal{N}_i^+} w_{ik} e^{\theta_k} + (1-p) \log \sum_{k \in \mathcal{N}_i^+} w_{ik} e^{\eta_k}, \tag{4.7}$$

with equality if and only if there exists $c \in \mathbf{R}$ such that $\theta_k = \eta_k + c$ for all $k \in \mathcal{N}_i^+$. Multiplying by $a_{ij}$ and summing over $i$ and $j$ on both sides of (4.7) shows that the log-likelihood is concave in $\boldsymbol{\theta}$. Now, consider any partition of the vertices into two non-empty subsets $\mathcal{S}$ and $\mathcal{T}$. Because the comparison graph is strongly connected, there is always




(a) network structure (b) comparison graph

Figure 4.2 – An innocent-looking example where the ML estimate does not exist. The network structure, aggregate traffic data and compatible transitions are shown on the left. The comparison graph (right) is not strongly connected.

$k \in \mathcal{V}$, $i \in \mathcal{S}$ and $j \in \mathcal{T}$ such that $i, j \in \mathcal{N}_k^+$ and $a_{ki} > 0$. Therefore, the left and right side of (4.6) are equal if and only if $\boldsymbol{\theta} \equiv \boldsymbol{\eta}$ up to a constant shift. □

In order to verify the necessary and sufficient condition of Theorem 4.2 given $\{(c_i^-, c_i^+)\}$, we have to find a non-negative solution $\{a_{ij}\}$ to the system of equations

$$\sum_{j \in \mathcal{N}_i^-} a_{ji} = c_i^-, \quad \sum_{j \in \mathcal{N}_i^+} a_{ij} = c_i^+ \quad \forall i.$$

Dines [1926] presents a simple algorithm to find such a non-negative solution. Alternatively, Kumar et al. [2015] suggest recasting the problem as one of maximum flow in a network. However, the computational cost of running Dines' or max-flow algorithms is significantly greater than that of running the inference algorithm that we develop later, in Section 4.4.1.

**Example** In order to illustrate Theorem 4.2, we describe an innocuous-looking example where the MLE does not exist. Consider the network structure and traffic data depicted in Figure 4.2. The network is strongly connected and every node $i$ has positive incoming and outgoing traffic $c_i^-$ and $c_i^+$. Nevertheless, the corresponding comparison graph is *not* strongly connected, and it turns out that the likelihood can be made arbitrarily large by increasing $\gamma_1$, $\gamma_2$ and $\gamma_4$. In this simple example, we indicate the edge transitions that generated the observed marginal traffic in bold. Given this information, the comparison graph is easy to find, and the necessary and sufficient condition is easy to check. But in general, finding a set of transitions that is compatible with given marginal per-node traffic data is a nontrivial computation.

**Necessary Condition** As the conditions of Theorem 4.2 involve the observed traffic, we might ask the following question. Is there a simpler condition on the structure of $\mathcal{G}$ such that the MLE is well-defined, given sufficiently many transitions? We provide an answer in the form of a necessary condition for the uniqueness of the MLE that involves only the structure of the network. We begin with a definition that uses the notion of *hypergraph*, a generalized graph where edges can be any non-empty subset of nodes.

**Definition** (alternatives hypergraph)**.** Given a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, the *alternatives hypergraph* is defined as $\mathcal{H} = (\mathcal{V}, \mathcal{A})$, with $\mathcal{A} = \{\mathcal{N}_i^+ : i \in \mathcal{V}\}$.

Intuitively, $\mathcal{H}$ is the hypergraph induced by the sets of alternatives available at each node. Equipped with this definition, we can state the following corollary of Theorem 4.2.

**Corollary 4.3.** *If the alternatives hypergraph is not connected, then for any data $\mathcal{D}$ there are $\boldsymbol{\gamma}$ and $\boldsymbol{\lambda}$ such that $\boldsymbol{\gamma} \neq c\boldsymbol{\lambda}$ for any $c \in \mathbf{R}_{>0}$ and $\ell(\boldsymbol{\gamma}; \mathcal{D}) = \ell(\boldsymbol{\lambda}; \mathcal{D})$.*

*Proof.* If the alternatives hypergraph is disconnected, then for any data $\mathcal{D}$, the comparison graph is disconnected too. Furthermore, the connected components of the comparison graph are subsets of those of the hypergraph. Partition the vertices into two non-empty subsets $\mathcal{S}$ and $\mathcal{T}$ such that there is no hyperedge between $\mathcal{S}$ to $\mathcal{T}$ in the alternatives hypergraph. Let $\mathcal{A} = \{i : \mathcal{N}_i^+ \subset \mathcal{S}\}$ and $\mathcal{B} = \{i : \mathcal{N}_i^+ \subset \mathcal{T}\}$. By construction of the alternatives hypergraph, $\mathcal{A} \cap \mathcal{B} = \varnothing$ and $\mathcal{A} \cup \mathcal{B} = \mathcal{V}$. The log-likelihood can be rewritten as

$$\begin{aligned}\ell(\boldsymbol{\theta}) = &\sum_{i \in \mathcal{A}} \sum_{j \in \mathcal{N}_i^+} a_{ij} \left[ \log \gamma_j - \log \sum_{k \in \mathcal{N}_i^+} w_{ik} \gamma_k \right] \\ &+ \sum_{i \in \mathcal{B}} \sum_{j \in \mathcal{N}_i^+} a_{ij} \left[ \log \gamma_j - \log \sum_{k \in \mathcal{N}_i^+} w_{ik} \gamma_k \right].\end{aligned}$$

The sum over $\mathcal{A}$ involves only parameters related to nodes in $\mathcal{S}$, whereas the sum over $\mathcal{B}$ involves only parameters related to nodes in $\mathcal{T}$. Because the likelihood is invariant to a rescaling of the parameters, it is easy to see that we can arbitrarily rescale the parameters of the vertices in either $\mathcal{S}$ or $\mathcal{T}$ without affecting the likelihood. □

The network of Figure 4.1 illustrates an instance where even the necessary the condition fails: although the graph $\mathcal{G}$ is strongly connected, its associated alternatives hypergraph $\mathcal{H}$ (depicted in Figure 4.3) is disconnected, and no matter what the data $\mathcal{D}$ is, the ML estimate will never be uniquely defined. Note that this problematic situation does not affect only carefully hand-crafted networks: the alternatives hypergraph of all three real-world networks considered in Section 4.5 are disconnected as well.

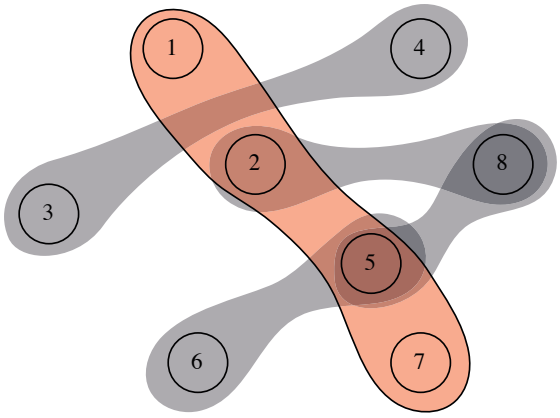



Figure 4.3 – The alternatives hypergraph associated with the network of Figure 4.1. The hyperedge associated with $\mathcal{N}_6^+$ is highlighted in red. Note that the component $\{3, 4\}$ is disconnected from the rest of the hypergraph.

## 4.4 Well-Posed Inference

The shortcomings of the MLE discussed in the previous section drive us to seek a more robust estimator. Following the ideas of Caron and Doucet [2012], we introduce an independent Gamma prior on each parameter, i.e., i.i.d. $\gamma_1, \dots, \gamma_N \sim \text{Gamma}(\alpha, \beta)$. Adding the log-prior to the log-likelihood, we can write the log-posterior as

$$\log p(\boldsymbol{\gamma} \mid \mathcal{D}) = \sum_{i=1}^{N} \left[ (c_i^- + \alpha - 1) \log \gamma_i - c_i^+ \log \left( \sum_{k \in \mathcal{N}_i^+} w_{ik} \gamma_k \right) - \beta \gamma_i \right] + \kappa. \tag{4.8}$$

The Gamma prior translates into a form of regularization that makes the inference problem well-posed, as shown by the following theorem.

**Theorem 4.4.** *If i.i.d.* $\gamma_1, \dots, \gamma_N \sim \text{Gamma}(\alpha, \beta)$ *with* $\alpha > 1$*, then the log-posterior* (4.8) *always has a unique maximizer* $\boldsymbol{\gamma}^\star \in \mathbf{R}_{>0}^N$.

*Proof.* Under the reparametrization $\gamma_i = e^{\theta_i}$, the log-prior and the log-likelihood become

$$\log p(\boldsymbol{\theta}) = \sum_{i=1}^{N} \left[ (\alpha - 1)\theta_i - \beta e^{\theta_i} \right] + \kappa$$

$$\ell(\boldsymbol{\theta}; \mathcal{D}) = \sum_{i=1}^{N} \left[ c_i^- \theta_i - c_i^+ \log \sum_{k \in \mathcal{N}_i^+} w_{ik} e^{\theta_k} \right] + \kappa.$$

It is easy to see that the log-likelihood is concave and the log-prior strictly concave in $\boldsymbol{\theta}$ (for $\alpha > 1$). As a result, the log-posterior is strictly concave in $\boldsymbol{\theta}$, which ensures that there exists at most one maximizer.

Now consider any transition counts $\{c_{ij}\}$ that satisfy $c_i^- = \sum_{j \in \mathcal{N}_i^-} c_{ji}$ and $c_i^+ = \sum_{j \in \mathcal{N}_i^+} c_{ij}$. The log-posterior can be written as

$$\begin{aligned}\log p(\boldsymbol{\theta} \mid \mathcal{D}) &= \sum_{i=1}^{N} \sum_{j \in \mathcal{N}_i^+} c_{ij} \Big[\theta_j - \log \sum_{k \in \mathcal{N}_i^+} w_{ik} e^{\theta_k}\Big] + \sum_{i=1}^{N} \Big[(\alpha - 1)\theta_i - \beta e^{\theta_i}\Big] + \kappa \\ &\leq -N^2 \cdot \max_{i,j} \log w_{ij} + \sum_{i=1}^{N} \Big[(\alpha - 1)\theta_i - \beta e^{\theta_i}\Big] + \kappa.\end{aligned}$$

For $\alpha > 1$, it follows that $\lim_{\|\boldsymbol{\theta}\| \to \infty} \log p(\boldsymbol{\theta} \mid \mathcal{D}) = -\infty$, which ensures that there is at least one maximizer. □

Note that varying the rate $\beta$ in the Gamma prior simply rescales the parameters $\boldsymbol{\gamma}$. Furthermore, it is clear from (4.1) that such a rescaling affects neither the likelihood of the observed data nor the prediction of future transitions. As a consequence, we can assume that $\beta = 1$ without loss of generality.

### 4.4.1 ChoiceRank Algorithm

The maximizer of the log-posterior does not have a closed-form solution. In the spirit of the algorithms of Hunter [2004] for variants of Luce's choice model, we develop a minorization-maximization (MM) algorithm. Simply put, the algorithm iteratively refines an estimate of the maximizer by solving a sequence of simpler optimization problems. Using the inequality $\log x \leq \log \tilde{x} + x/\tilde{x} - 1$ (with equality if and only if $x = \tilde{x}$), we can lower-bound the log-posterior (4.8) by

$$\begin{aligned}f^{(t)}(\boldsymbol{\gamma}) = \kappa + \sum_{i=1}^{N} \Bigg[&(c_i^- + \alpha - 1) \log \gamma_i - \beta \gamma_i \\ &- c_i^+ \Bigg( \log \sum_{k \in \mathcal{N}_i^+} w_{ik} \gamma_k^{(t)} + \frac{\sum_{k \in \mathcal{N}_i^+} w_{ik} \gamma_k}{\sum_{k \in \mathcal{N}_i^+} w_{ik} \gamma_k^{(t)}} - 1 \Bigg) \Bigg],\end{aligned} \tag{4.9}$$

with equality if and only if $\boldsymbol{\gamma} = \boldsymbol{\gamma}^{(t)}$. Starting with an arbitrary $\boldsymbol{\gamma}^{(0)} \in \mathbf{R}_{>0}^N$, we repeatedly solve the optimization problem

$$\boldsymbol{\gamma}^{(t+1)} = \arg\max_{\boldsymbol{\gamma}} f^{(t)}(\boldsymbol{\gamma}).$$

Unlike the maximization of the log-posterior, the surrogate optimization problem has a closed-form solution, obtained by setting $\nabla f^{(t)}$ to $\mathbf{0}$:

$$\gamma_i^{(t+1)} = \frac{c_i^- + \alpha - 1}{\sum_{j \in \mathcal{N}_i^-} w_{ji} \mu_j^{(t)} + \beta}, \quad \text{where } \mu_j^{(t)} = \frac{c_j^+}{\sum_{k \in \mathcal{N}_j^+} w_{jk} \gamma_k^{(t)}}. \tag{4.10}$$

The sequence of iterates provably converges to the maximizer of the log-posterior (4.8), as shown by the following theorem.

**Theorem 4.5.** *Let $\boldsymbol{\gamma}^\star$ be the unique maximum a-posteriori estimate. Then for any initial $\boldsymbol{\gamma}^{(0)} \in \mathbf{R}^N_{>0}$ the sequence of iterates defined by* (4.10) *converges to $\boldsymbol{\gamma}^\star$.*

*Proof.* The proof follows that of Theorem 1 in Hunter [2004]. Let $M : \mathbf{R}^N_{>0} \to \mathbf{R}^N_{>0}$ be the (continuous) map implicitly defined by one iteration of the algorithm. For conciseness, let $g(\boldsymbol{\gamma}) \doteq \log p(\boldsymbol{\gamma} \mid \mathcal{D})$. As $g$ has a unique maximizer and is concave using the reparametrization $\gamma_i = e^{\theta_i}$, it follows that $g$ has a single stationary point. First, observe that the minorization-maximization property guarantees that $g\left[M(\boldsymbol{\gamma})\right] \geq g(\boldsymbol{\gamma})$. Combined with the strict concavity of $g$, this ensures that $\lim_{t\to\infty} g(\boldsymbol{\gamma}^{(t)})$ exists and is unique for any $\boldsymbol{\gamma}^{(0)}$. Second, $g\left[M(\boldsymbol{\gamma})\right] = g(\boldsymbol{\gamma})$ if and only if $\boldsymbol{\gamma}$ is a stationary point of $g$, because the minorizing function is tangent to $g$ at the current iterate. It follows that $\lim_{t\to\infty} \boldsymbol{\gamma}^{(t)} = \boldsymbol{\gamma}^\star$. □

How fast does the sequence of iterates converge? It is known that MM algorithms exhibit geometric convergence in a neighborhood of the maximizer [Lange et al., 2000], but a thorough investigation of the convergence properties is left for future work.

The structure of the updates in (4.10) leads to an extremely simple and efficient implementation, described in Algorithm 4.1: we call it ChoiceRank. A graphical representation of an iteration from the perspective of a single node is given in Figure 4.4. Each iteration consists of two phases of message passing, with $\gamma_i$ flowing towards in-neighbors $\mathcal{N}_i^-$, then $\mu_i$ flowing towards out-neighbors $\mathcal{N}_i^+$ (each message being weighted by the edge strength $w_{ij}$). The updates to a node's state are a function of the sum of the messages. As the algorithm does two passes over the edges and two passes over the vertices, an iteration takes $O(M + N)$ time. The edges can be processed in any order, and the algorithm maintains a state over only $O(N)$ values associated with the vertices. Furthermore, the algorithm can be conveniently expressed in the well-known vertex-centric programming model [Malewicz et al., 2010]. This makes it easy to implement ChoiceRank inside scalable, optimized graph-processing systems such as Apache Spark [Gonzalez et al., 2014].

### 4.4.2 EM Viewpoint

The MM algorithm can also be interpreted from an expectation-maximization (EM) viewpoint, following the ideas of Caron and Doucet [2012]. We introduce $N$ independent random variables $\mathcal{Z} = \{z_i : i = 1, \ldots, N\}$, where

$$z_i \sim \text{Gamma}\Big(c_i^+, \sum_{j \in \mathcal{N}_i^+} w_{ij}\gamma_j\Big).$$

**Algorithm 4.1** ChoiceRank.

**Require:** graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, counts $\{(c_i^-, c_i^+)\}$, edge weights $\{w_{ij}\}$
$\boldsymbol{\gamma} \leftarrow [1 \ \cdots \ 1]^\top$
**repeat**
  $\boldsymbol{z} \leftarrow \mathbf{0}_N$ ▷ Recompute $\boldsymbol{\mu}$
  **for** $(i,j) \in \mathcal{E}$ **do** $z_i \leftarrow z_i + w_{ij}\gamma_j$
  **for** $i \in \mathcal{V}$ **do** $\mu_i \leftarrow c_i^+ / z_i$
  $\boldsymbol{z} \leftarrow \mathbf{0}_N$ ▷ Recompute $\boldsymbol{\gamma}$
  **for** $(i,j) \in \mathcal{E}$ **do** $z_j \leftarrow z_j + w_{ij}\mu_i$
  **for** $i \in \mathcal{V}$ **do** $\gamma_i \leftarrow (c_i^- + \alpha - 1)/(z_i + \beta)$
**until** $\boldsymbol{\gamma}$ has converged

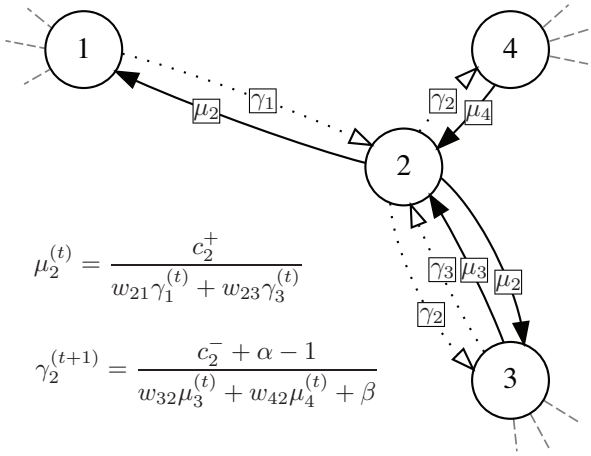



Figure 4.4 – One iteration of ChoiceRank from the perspective of node 2. Messages flow in both directions along the edges of the graph $\mathcal{G}$, first in the reverse direction (in dotted) then in the forward direction (in solid).

With the addition of these latent random variables, the complete log-likelihood becomes

$$\begin{aligned}
\ell(\boldsymbol{\gamma}; \mathcal{D}, \mathcal{Z}) &= \ell(\boldsymbol{\gamma}; \mathcal{D}) + \sum_{i=1}^{N} \log p(z_i \mid \mathcal{D}, \boldsymbol{\gamma}) \\
&= \sum_{i=1}^{N} \left[ c_i^- \log \gamma_i - c_i^+ \log \sum_{k \in \mathcal{N}_i^+} w_{ik}\gamma_k \right] \\
&\qquad + \sum_{i=1}^{N} \left[ c_i^+ \log \sum_{k \in \mathcal{N}_i^+} w_{ik}\gamma_k - z_i \sum_{k \in \mathcal{N}_i^+} w_{ik}\gamma_k \right] + \kappa \\
&= \sum_{i=1}^{N} \left[ c_i^- \log \gamma_i - z_i \sum_{k \in \mathcal{N}_i^+} w_{ik}\gamma_k \right] + \kappa.
\end{aligned}$$

Using a Gamma$(\alpha, \beta)$ prior for each parameter, the expected value of the log-posterior with respect to the conditional $\mathcal{Z} \mid \mathcal{D}$ under the estimate $\boldsymbol{\gamma}^{(t)}$ is

$$\begin{aligned} Q(\boldsymbol{\gamma}, \boldsymbol{\gamma}^{(t)}) &= \mathbf{E}_{\mathcal{Z}|\mathcal{D},\boldsymbol{\gamma}^{(t)}} \left[\ell(\boldsymbol{\gamma}; \mathcal{D}, \mathcal{Z})\right] + \log p(\boldsymbol{\gamma}) \\ &= \sum_{i=1}^{N} \left[ c_i^- \log \gamma_i - c_i^+ \frac{\sum_{k \in \mathcal{N}_i^+} w_{ik} \gamma_k}{\sum_{k \in \mathcal{N}_i^+} w_{ik} \gamma_k^{(t)}} \right] + \sum_{i=1}^{N} \Big[ (\alpha - 1) \log \gamma_i - \beta \gamma_i \Big] + \kappa. \end{aligned}$$

The EM algorithm starts with an initial $\boldsymbol{\gamma}^{(0)}$ and iteratively refines the estimate by solving the optimization problem $\boldsymbol{\gamma}^{(t+1)} = \arg\max_{\boldsymbol{\gamma}} Q(\boldsymbol{\gamma}, \boldsymbol{\gamma}^{(t)})$. It is not difficult to see that for a given $\boldsymbol{\gamma}^{(t)}$, maximizing $Q(\boldsymbol{\gamma}, \boldsymbol{\gamma}^{(t)})$ is equivalent to maximizing the minorizing function $f^{(t)}(\boldsymbol{\gamma})$ defined in (4.9). Hence, the MM and the EM viewpoint lead to the exact same sequence of iterates.

The EM formulation leads to a Gibbs sampler in a relatively straightforward way [Caron and Doucet, 2012]. We leave a systematic treatment of Bayesian inference in the network choice model for future work.

## 4.5 Experimental Evaluation

In this section, we investigate $(a)$ the ability of the network choice model to accurately recover transitions in real-world scenarios, and $(b)$ the potential of ChoiceRank to scale to very large networks.

### 4.5.1 Accuracy on Real-World Data

We evaluate the network choice model on three datasets that are representative of two distinct application domains. Each dataset can be represented as a set of transition counts $\{c_{ij}\}$ on a directed graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. We aggregate the transition counts into marginal traffic data $\{(c_i^-, c_i^+) : i \in \mathcal{V}\}$ and fit a network choice model by using ChoiceRank (for simplicity, we set $w_{ij} \equiv 1$ for all datasets). We set $\alpha = 2.0$ and $\beta = 1.0$ (these small values simply guarantee the convergence of the algorithm for any network structure) and declare convergence when $\|\boldsymbol{\gamma}^{(t)} - \boldsymbol{\gamma}^{(t-1)}\|/N < 10^{-8}$. Given $\boldsymbol{\gamma}$, we estimate transition probabilities by using $p_{ij} \propto \gamma_j$ as given by (4.1). To the best of our knowledge, there is no other published method that tackles the problem of estimating transition probabilities from marginal traffic data. Hence, we compare our method to three baselines based on simple heuristics.

**Traffic** Transitions probabilities are proportional to the traffic of the target node: $q_{ij}^T \propto c_j^-$.

**PageRank** Transition probabilities are proportional to the PageRank score of the target node: $q_{ij}^{P} \propto \mathrm{PR}_j$.

**Uniform** Any transition is equiprobable: $q_{ij}^{U} \propto 1$.

The four estimates are compared against estimates of transition probabilities derived from ground-truth edge traffic data: $p_{ij}^{\star} \propto c_{ij}$. We emphasize that although per-edge transition counts $\{c_{ij}\}$ are needed to *evaluate* the accuracy of the network choice model (and the baselines), these counts are not necessary for *learning* the model—per-node marginal counts are sufficient.

Given a node $i$, we measure the accuracy of a distribution $\boldsymbol{q}_i$ over outgoing transitions using two error metrics, the KL-divergence and the (normalized) rank displacement:

$$D_{\mathrm{KL}}(\boldsymbol{p}_i^{\star}, \boldsymbol{q}_i) = \sum_{j \in \mathcal{N}_i^{+}} p_{ij}^{\star} \log \frac{p_{ij}^{\star}}{q_{ij}},$$

$$D_{\mathrm{FR}}(\boldsymbol{p}_i^{\star}, \boldsymbol{q}_i) = \frac{1}{|\mathcal{N}_i^{+}|^2} \sum_{j \in \mathcal{N}_i^{+}} |\sigma_i^{\star}(j) - \hat{\sigma}_i(j)|,$$

where $\sigma_i^{\star}$ (respectively $\hat{\sigma}_i$) is the ranking of elements in $\mathcal{N}_i^{+}$ by decreasing order of $p_{ij}^{\star}$ (respectively $q_{ij}$). We report the distribution of errors "over choices", i.e., the error at each node $i$ is weighted by the number of outgoing transitions $c_i^{+}$.

### Clickstream Data

**Wikipedia** The Wikimedia Foundation has a long history of publicly sharing aggregate, page-level Web-traffic data[3]. Recently, it also released clickstream data from the English version of Wikipedia [Wulczyn and Taraborelli, 2016], providing us with essential ground-truth transition-level data. We consider a dataset that contains information, extracted from the server logs, about the traffic each page of the English Wikipedia received during the month of March 2016. Each page's incoming traffic is grouped by HTTP referrer, i.e., by the page visited prior to the request. We ignore the traffic generated by external websites such as search engines and keep only the internal traffic (18% of the total traffic in the dataset). In summary, we obtain counts of transitions on the hyperlink graph of English Wikipedia articles. The graph contains $N = 2\,316\,032$ nodes and $M = 13\,181\,698$ edges, and we consider slightly over 1.2 billion transitions over the edges. On this dataset, ChoiceRank converges after 795 iterations.

[3] See: `https://stats.wikimedia.org/`.

**Kosarak** We also consider a second clickstream dataset from a Hungarian online news portal[4]. The data consist of 7 029 013 transitions on a graph containing $N = 41001$ nodes and $M = 974\,560$ edges. ChoiceRank converges after 625 iterations.

The four topmost plots of Figure 4.5 show the error distributions. ChoiceRank significantly improves on the baselines, both in terms of KL-divergence and rank displacement. These results give compelling evidence that transitions do not occur proportionally with the target's page traffic: in terms of KL-divergence, ChoiceRank improves on Traffic by a factor 3× and 2×, respectively. PageRank scores, though reflecting some notion of importance of a page, are not designed to estimate transitions, and understandably the corresponding baseline performs poorly. Uniform (perhaps the simplest of our baselines) is (by design) unable to distinguish among transitions, resulting in a large displacement error. We believe that its comparatively better performance in terms of KL-divergence (for Wikipedia) is mostly an artifact of the metric, which encourages "prudent" estimates. Finally, in Figure 4.6 we observe that ChoiceRank seems to perform comparatively better as the number of possible transition increases.

#### NYC Bicycle-Sharing Data

Next, we consider trip data from Citi Bike, New York City's bicycle-sharing system[5]. For each ride on the system made during the year 2015, we extract the pick-up and drop-off stations and the duration of the ride. Because we want to focus on direct trips, we exclude rides that last more than one hour. We also exclude source-destinations pairs which have less than 1 ride per day on average (a majority of source-destination pairs appears at least once in the dataset). The resulting data consist of 3.4 million rides on a graph containing $N = 497$ nodes and $M = 5\,209$ edges. ChoiceRank converges after 7 508 iterations. We compute the error distribution in the same way as for the clickstream datasets.

The two bottommost plots of Figure 4.5 display the results. The observations made on the clickstream datasets carry over to this mobility dataset, albeit to a lesser degree. A significant difference between clicking a link and taking a bicycle trip is that, in the latter case, there is a non-uniform "cost" of a transition due to the distance between source and target. In future work, we might consider experimenting with edge weights $\{w_{ij}\}$ that capture this.

### 4.5.2 Scaling to Large Networks

To demonstrate ChoiceRank's scalability, we develop a simple implementation in the Rust programming language, based on the ideas of COST [McSherry et al., 2015]. Our

[4]The data are publicly available at `http://fimi.ua.ac.be/data/`.
[5]The data is available at `https://www.citibikenyc.com/system-data`.

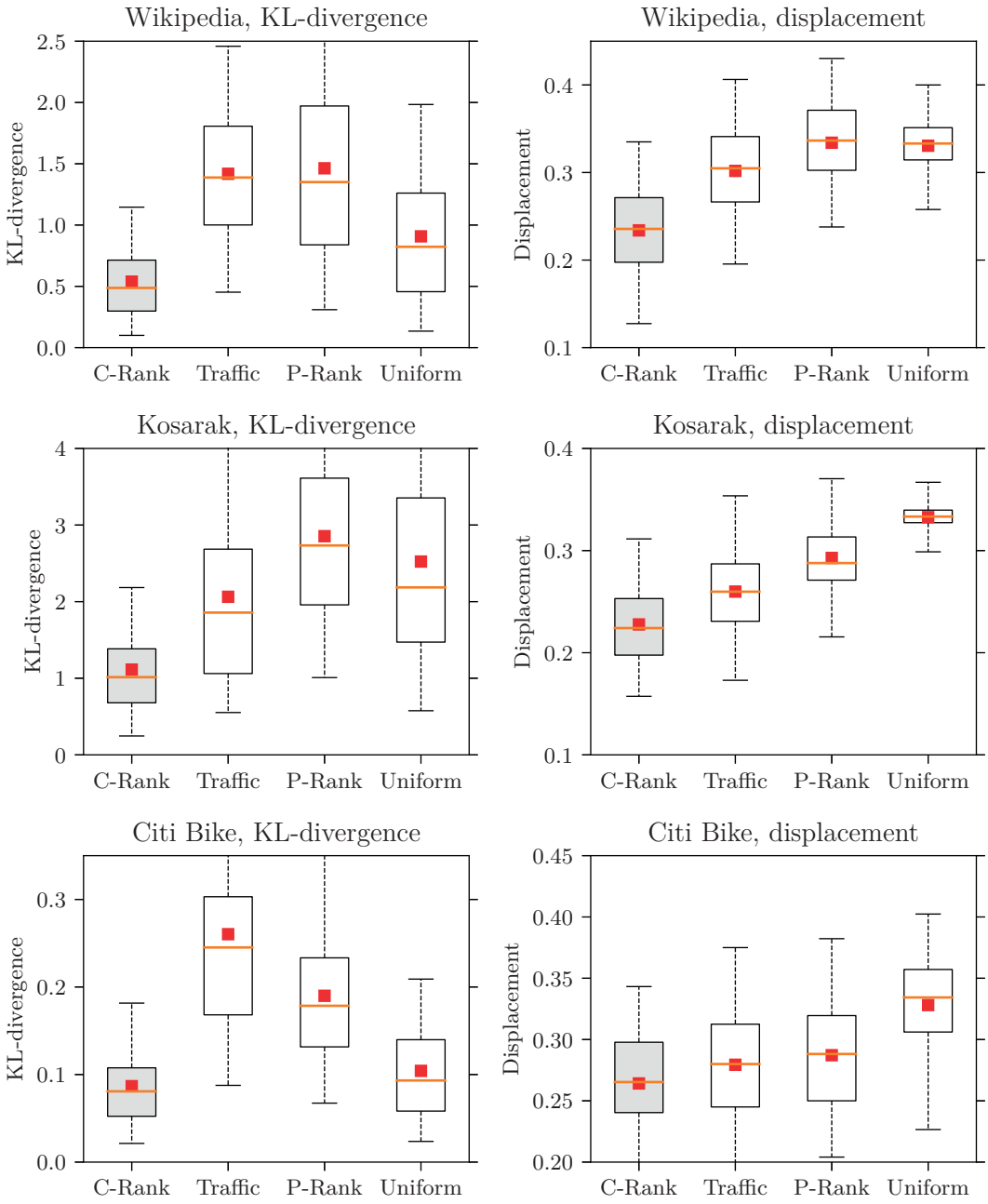



Figure 4.5 – Error distributions of the network choice model and three baselines for the Wikipedia, Kosarak and Citi Bike datasets. The boxes show the interquartile range, the whiskers show the $5^{\text{th}}$ and $95^{\text{th}}$ percentiles, the red horizontal bars show the median and the red squares show the mean.

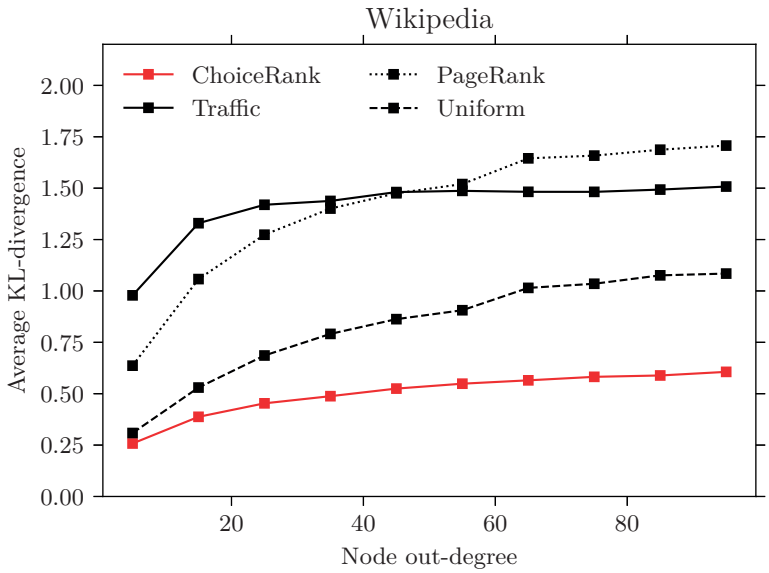



Figure 4.6 – Average KL-divergence as a function of the number of possible transitions for the Wikipedia dataset. ChoiceRank performs comparatively better in the case where a node's out-degree is large.

code is publicly available online[6]. The implementation repeatedly streams edges from disk and keeps four floating-point values per node in memory: the counts $c_i^-$ and $c_i^+$, the sum of messages $z_i$, and either $\mu_i$ or $\gamma_i$ (depending on the stage in the iteration). As edges can be processed in any order, it can be beneficial to reorder the edges in a way that accelerates the computation. For this reason, our implementation preprocesses the list of edges and reorders them in Hilbert curve order[7]. This results in better cache locality and yields a significant speedup.

We test our implementation on a hyperlink graph extracted from the 2012 Common Crawl Web corpus[8] that contains over 3.5 billion nodes and 128 billion edges [Meusel et al., 2014]. The edge list alone requires about 1 TB of uncompressed storage. There is no publicly available information on the traffic at each page, therefore we generate a value $c_i$ for every node $i$ randomly and uniformly between 100 and 500, and set both $c_i^-$ and $c_i^+$ to $c_i$. As such, this experiment does not attempt to measure the validity of the model (unlike the experiments of Section 4.5.1). Instead, it focuses on testing the algorithm's potential to scale to very large networks.

[6]See: `https://github.com/lucasmaystre/choicerank`.

[7]A Hilbert space-filling curve visits all the entries of the adjacency matrix of the graph, in a way that preserves locality of both source and destination of the edges.

[8]The data are available at `http://webdatacommons.org/hyperlinkgraph/`.

**Results** We run 20 iterations of ChoiceRank on a dual Intel Xeon E5-2680 v3 machine, with 256 GB of RAM and 6 HDDs configured in RAID 0. We arbitrarily set $\alpha = 2.0$ and $\beta = 1.0$ (but this choice has no impact on the results). Only about 65 GB of memory is used, all to store the nodes' state ($4 \times 4$ bytes per node). The algorithm takes a little less than 39 minutes per iteration on average. Collectively, these results validate the feasibility of model inference for very large datasets.

It is worth noting that, despite tackling different problems, the ChoiceRank algorithm exhibits interesting similarities with a message-passing implementation of PageRank commonly used in scalable graph-parallel systems such as Pregel [Malewicz et al., 2010] and Spark [Gonzalez et al., 2014]. For comparison, using the COST code [McSherry et al., 2015] we run 20 iterations of PageRank on the same hardware and data. PageRank uses slightly less memory (about 50 GB, or one less floating-point number per node) and takes about half of the time per iteration (a little over 20 minutes). This is consistent with the fact that ChoiceRank requires two passes over the edges per iteration, whereas PageRank requires one. The similarities between the two algorithms lead us to believe that ChoiceRank can benefit from any new system optimization developed for PageRank.

## 4.6 Summary

In this chapter, we have presented a method that tackles the problem of finding the transition probabilities along the edges of a network, given only the network's structure and aggregate node-level traffic data. This method generalizes and extends ideas recently presented by Kumar et al. [2015]. We have demonstrated that in spite of the strong model assumptions needed to learn $O(N^2)$ probabilities from $O(N)$ observations, the method still manages to recover the transition probabilities to a good level of accuracy on two clickstream datasets, and shows promise for applications beyond clickstream data. In summary, we believe that our method will be useful to practitioners interested in understanding patterns of navigation in networks from aggregate traffic data, commonly available, for example, in public datasets.

# 5 Predicting Football Matches

In this chapter[1], we shift our attention from human choices to sports outcomes. In particular, we draw attention to a connection between skill-based models of game outcomes (built on the Bradley–Terry model) and Gaussian-process classification models. The Gaussian-process perspective enables (*a*) a principled way of dealing with uncertainty and (*b*) rich models, specified through kernel functions. Using this connection, we tackle the problem of predicting outcomes of football matches between national teams. We develop a *player kernel* that relates any two football matches through the players lined up on the field. This makes it possible to share knowledge gained from observing matches between clubs (available in large quantities) and matches between national teams (available only in limited quantities). We evaluate our approach on the Euro 2008, 2012 and 2016 final tournaments.

## 5.1 Introduction

Statistical models of game outcomes have a rich and diverse history, beginning with Zermelo almost a century ago (c.f. Section 1.2.2). In this chapter, we revisit his ideas and highlight their connections to modern machine-learning techniques. In particular, we show how the Bradley–Terry model can be cast as a Gaussian-process classification model. The Gaussian-process framework provides two key advantages. First, it brings all the benefits of Bayesian inference. In particular it provides a principled way to deal with the uncertainty associated with noisy observations and with predictions. Second, it opens up new modeling perspectives through the specification of kernel functions.

Equipped with this, we study the problem of predicting outcomes of football matches between national teams. We identify two key challenges, (*a*) that of *data sparsity* (national teams usually play no more than ten matches per year), and (*b*) that of *data staleness* (the team roster is constantly evolving). Taking inspiration from the observation that

[1]This chapter is based on Maystre, Kristof, González Ferrer, and Grossglauser [2016].

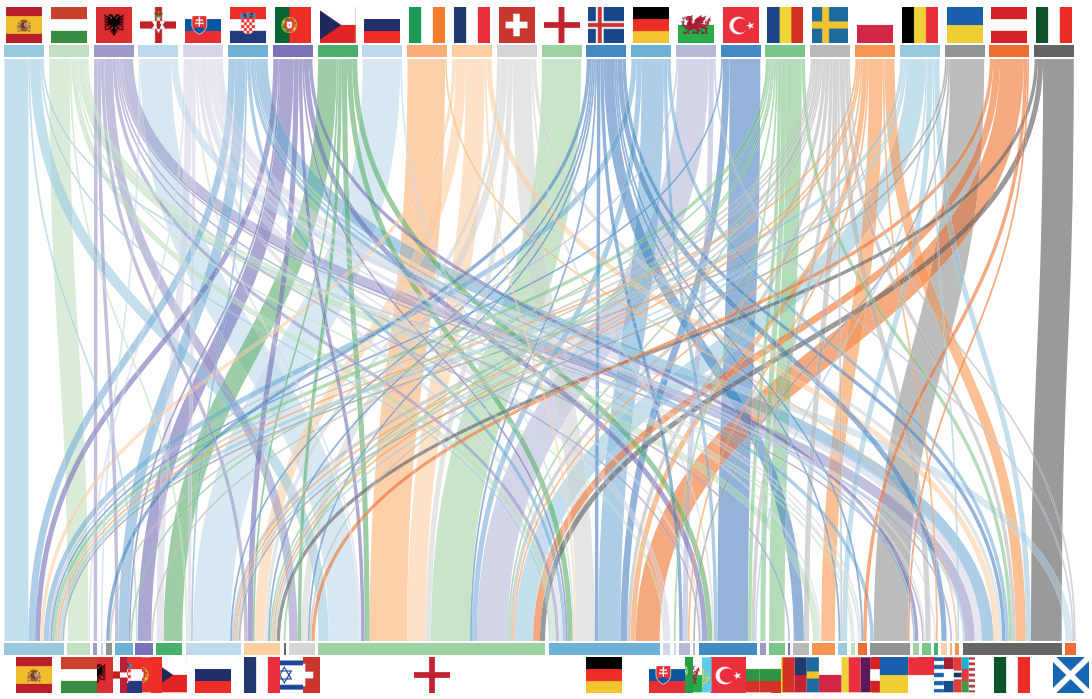


Figure 5.1 – Players of national teams qualified for the Euro 2016 (top row) are playing in clubs across Europe and beyond (bottom row). The English, German and Italian club championships contain the most selected players.

national teams' players frequently face each other in competitions between clubs (see Figure 5.1), we show that these two difficulties can be addressed by the introduction of a *player kernel*. This kernel relates any two matches through the players lined up on the field, and makes it possible to seamlessly use matches between clubs to improve a predictive model ultimately used for matches between national teams. This is beneficial because, in contrast to national teams, clubs play much more frequently, hence more data are available to train the model. This also implicitly addresses the staleness problem, as a team is defined by the set of players present at a given match.

**Outline of the Chapter** The remainder of this short chapter is organized as follows. We review related work in Section 5.2. In Section 5.3, we formalize the link between the Bradley–Terry model and Gaussian-process classification, and present the player kernel. Then, in Section 5.4, we evaluate our predictive model on the Euro 2008, 2012 and 2016 final tournaments.

## 5.2 Related Work

Zermelo's 1928 paper (discussed in Section 1.2.2) presented the first statistical model of chess game outcomes. His model, associated with a simple online stochastic gradient update rule, is known as the Elo rating system [Elo, 1978]. This rating system is currently used by the World Chess Federation (FIDE) to rank chess players[2] and by the International Federation of Football Association (FIFA) to rank women's national football teams[3], among others.

The model and related inference algorithms have been extended in various ways, e.g., by considering other types of outcomes [Rao and Kupper, 1967, Maher, 1982] or by permitting parameters to evolve over time [Glickman, 1993, Fahrmeir and Tutz, 1994, Cattelan et al., 2013]. One direction that is of particular interest in this chapter is the handling of the uncertainty of the estimated skill parameters. Glickman [1999] proposes an extension that simultaneously updates ratings and associated uncertainty values, after each observation, by using a simple closed-form update. Herbrich et al. [2006] propose TrueSkill, a comprehensive Bayesian framework for estimating player skills in various types of games based on the expectation-propagation algorithm. The models and methods described in this chapter are similar to TrueSkill, as will be discussed in Section 5.3. In the context of learning users' preferences from pairwise comparisons, Chu and Ghahramani [2005c] were the first to link the Bayesian treatment of pairwise comparisons models to Gaussian-process classification [Rasmussen and Williams, 2006].

## 5.3 Methods

In this section, we first show how the Bradley–Terry model of pairwise comparisons (the modern name of Zermelo's model), can be expressed in the Gaussian-process framework. The Gaussian-process viewpoint shifts the focus from *items* (or, in our case, contestants) to *games*: the statistical relationship between outcomes of several games is given by a covariance function. Second, we present the *player kernel*, a covariance function that relates football matches through lineups.

### 5.3.1 Gaussian-Process Classification Viewpoint

Suppose that we observe outcomes of comparisons between two items (e.g., two players or two teams) in a universe of items denoted $1, \ldots, N$. We begin by restricting ourselves to binary outcomes, i.e., we assume that one of the two items necessarily wins. The Bradley-Terry model postulates that each item $i$ can be represented by a parameter $\theta_i \in \mathbf{R}$, indicative of its relative strength against an opponent. Given these parameters,

[2] See: `https://ratings.fide.com/`.
[3] See: `http://www.fifa.com/fifa-world-ranking/procedure/women.html`.

the probability of observing the outcome $i \succ j$ is given by

$$\mathbf{P}\left[i \succ j\right] = \frac{1}{1 + \exp[-(\theta_i - \theta_j)]} = \frac{1}{1 + \exp(-\boldsymbol{\theta}^\top \boldsymbol{x})}, \tag{5.1}$$

where $\boldsymbol{\theta} = [\theta_i]$ and $\boldsymbol{x} \in \mathbf{R}^N$ is such that $x_i = 1$, $x_j = -1$ and $x_k = 0$ for $k \neq i, j$. As such, the pairwise comparison model can be seen as a special case of logistic regression [Bishop, 2006, Chapter 4], where the feature vector simply indicates the winning and losing items. Furthermore, logistic regression is itself a special case of Gaussian-process classification [Rasmussen and Williams, 2006, Chapter 3].

**Definition** (Gaussian process)**.** A *Gaussian process*

$$f(\boldsymbol{x}) \sim \mathrm{GP}[m(\boldsymbol{x}), k(\boldsymbol{x}, \boldsymbol{x}')]$$

is a stochastic process defined by a mean function $m(\boldsymbol{x}) \doteq \mathbf{E}\left[f(\boldsymbol{x})\right]$ and a positive semi-definite covariance (or kernel) function $k(\boldsymbol{x}, \boldsymbol{x}') \doteq \mathbf{Cov}\left[f(\boldsymbol{x}), f(\boldsymbol{x}')\right]$. Given any finite collection of points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_M$, the Gaussian process sampled at these points has a multivariate Gaussian distribution

$$\begin{bmatrix} f(\boldsymbol{x}_1) & \cdots & f(\boldsymbol{x}_M) \end{bmatrix} \sim \mathrm{N}(\boldsymbol{m}, \boldsymbol{K}),$$

where $m_u = m(\boldsymbol{x}_u)$ and $k_{uv} = k(\boldsymbol{x}_u, \boldsymbol{x}_v)$.

It is not hard to show that if $\boldsymbol{\theta} \sim \mathrm{N}(\mathbf{0}, \sigma^2 \boldsymbol{I})$, then $f(\boldsymbol{x}) = \boldsymbol{\theta}^\top \boldsymbol{x}$ is a Gaussian process with $m(\boldsymbol{x}) = 0$ and $k(\boldsymbol{x}, \boldsymbol{x}') = \sigma^2 \boldsymbol{x}^\top \boldsymbol{x}'$. This enables us to interpret (5.1) as the likelihood of a Gaussian-process classification model with the logit link function.

The Gaussian-process viewpoint shifts the focus from the parametric representation of the function $f(\boldsymbol{x})$ (in the case of (5.1), a linear function of items strengths) to the covariance between two function evaluations, as defined by the kernel function $k(\boldsymbol{x}, \boldsymbol{x}')$. Intuitively (and informally), the model can simply be specified by stating how similar any two match outcomes are expected to be. Furthermore, the Gaussian-process viewpoint also makes it possible to take advantage of the vast amount of literature and software related to accurate, efficient, and scalable inference.

**Handling Draws** Rao and Kupper [1967] propose an extension of the pairwise comparison model for ternary (win, draw, loss) outcomes. In this extension, the two different types of outcomes have probabilities

$$\begin{aligned} \mathbf{P}\left[i \succ j\right] &= \frac{1}{1 + \exp[f(\boldsymbol{x}) - \alpha]} \\ \mathbf{P}\left[i \equiv j\right] &= (e^{2\alpha} - 1)\mathbf{P}\left[i \succ j\right]\mathbf{P}\left[j \succ i\right], \end{aligned}$$

where $\alpha > 0$ is an additional hyperparameter controlling the frequency of draws (see also Section 2.3.5). Because a draw can be written as the product of a win and a loss, model inference can still be performed using only a *binary* Gaussian-process classification model, with the changes needed to the link function being minimal.

### 5.3.2 The Player Kernel

We now consider an application to football and propose a method to quantify how similar two match outcomes are expected to be. Let $1, \ldots, P$ denote all distinct players appearing in a dataset of matches. We define a team's *lineup* as the set consisting of the 11 players starting the match. For a given match, let $\mathcal{W}$ and $\mathcal{L}$ be the lineups of the winning and losing teams, respectively. Define $\boldsymbol{z} \in \mathbf{R}^P$ such that $z_p = 1$ if $p \in \mathcal{W}$, $z_p = -1$ if $p \in \mathcal{L}$ and $z_p = 0$ otherwise. We then define the player kernel as

$$k(\boldsymbol{z}, \boldsymbol{z}') = \sigma^2 \boldsymbol{z}^\top \boldsymbol{z}'.$$

Intuitively, the function is positive if the same players are lined up in both matches, and the same players win (respectively, lose). The function is negative when players win one match, but lose the other. Finally, the function is zero, e.g., when the lineups are completely disjoint.

This kernel implicitly projects every match into the space of players, and defines a notion of similarity in this space. In the case of national teams qualified to Euro final tournaments, we find that this approach is very useful: a significant part of national teams' players take part in one of the main European leagues and play with or against each other. International club competitions (such as the UEFA Champions League) further contribute to the "connectivity" among players. Figure 5.2 illustrates the similarity of matches across different competitions in 2011–2012.

It is interesting to note that the player kernel corresponds to a linear model over the players. That is, it is equivalent to assuming that there is one independent skill parameter per player, and that the strength of a team is the sum of its players' skills. Such a model contains a massive number of parameters (possibly much more than the number of observations), and there is little hope for a reliable estimation of every parameter. In fact, in Section 5.4 we observe that the model is "weakly" parametric: the number of distinct players usually grows with the number of matches observed. The kernel-based viewpoint that we take emphasizes the fact that estimating these parameters explicitly is *not* necessary.

**Relation to TrueSkill** Our Gaussian-process model coupled with the player kernel is very similar to TrueSkill [Herbrich et al., 2006]. The most important difference is that we take advantage of the dual representation and operate in the space of matches, instead of

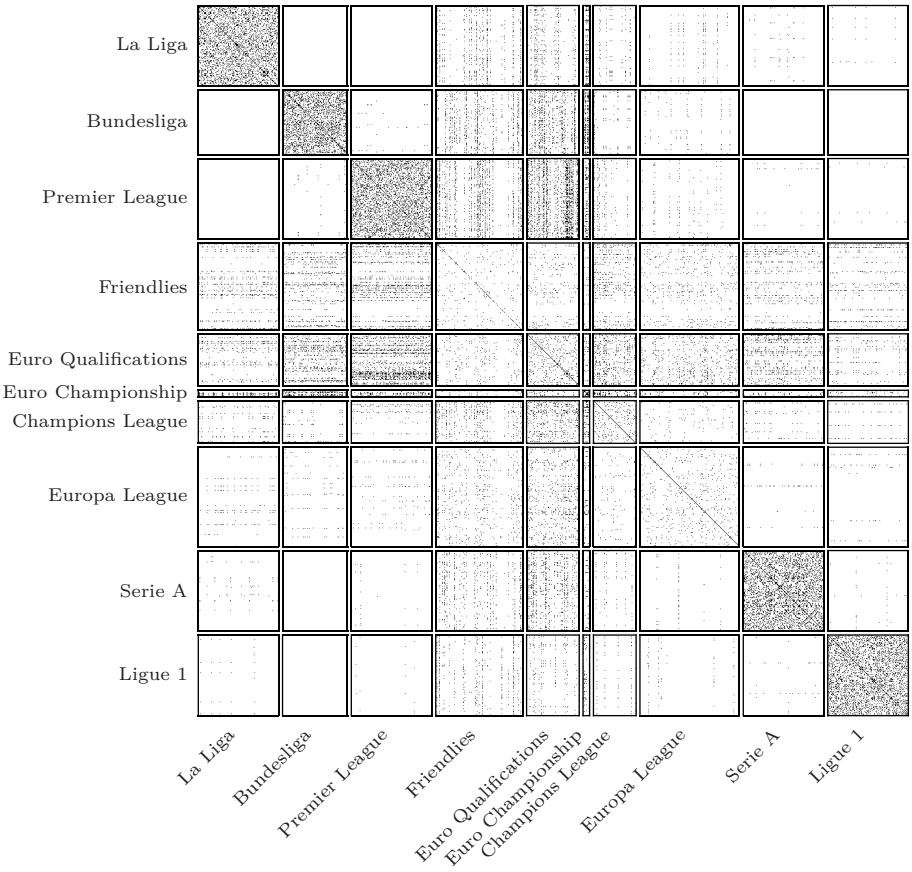



Figure 5.2 – Heat map of the magnitude of the kernel matrix for 3 184 matches played over the year preceding Euro 2012. White indicates zero correlation, black indicates non-zero correlation. Matches between national teams exhibit non-zero covariance with matches of all other competitions.

in the space of players. Beyond the conceptual reasons outlined above, the model makes inference less computationally intensive for the datasets that we consider.

## 5.4 Experimental Evaluation

In this section, we evaluate our predictive model on the matches of the Euro 2008, 2012 and 2016 final tournaments and compare it to several baselines.

We collect a dataset of matches from (*a*) official and friendly competitions involving national teams, and (*b*) the most prestigious European club competitions, starting from July 1$^{\text{st}}$, 2006. The list of competitions is displayed in Table 5.1. There are approximately 15× more matches between clubs than there are matches between national teams in our

Table 5.1 – List of competitions included in the dataset, spanning matches from 2006 to 2016. The majority of matches are played in competitions between clubs.

| Competition | Country | Involves clubs |
|---|---|---|
| Bundesliga | Germany | • |
| Confederations Cup | International | |
| EC Qualification | International | |
| European Championship | International | |
| Friendlies | International | |
| Ligue 1 | France | • |
| Premier League | England | • |
| La Liga | Spain | • |
| Serie A | Italy | • |
| UEFA Champions League | International | • |
| UEFA Europa League | International | • |
| World Cup | International | |

dataset. With respect to the model outlined in Section 5.3, our final predictive model processes one additional feature that encodes which team played at home (this feature is null for matches played on neutral ground). We train the model using a dataset $\mathcal{D}$ consisting of all $M$ matches that were played prior to the start of the competition on which we test. When computing the kernel matrix (whether on training or on test data) we use the starting lineups, usually announced shortly before the start of the match. It is interesting to note that the number of distinct players $P$ appearing in the dataset exceeds the number of training instances in each case (the values of $M$ and $P$ are shown in Table 5.2).

Starting from a Gaussian prior distribution over the $M$ matches $\boldsymbol{f} = [f_1 \ \cdots \ f_M]^\top \sim \mathrm{N}(\boldsymbol{f} \mid \boldsymbol{m}, \boldsymbol{K})$, we seek to find the posterior distribution

$$p(\boldsymbol{f} \mid \mathcal{D}) \propto \mathrm{N}(\boldsymbol{f} \mid \boldsymbol{m}, \boldsymbol{K}) \prod_{m=1}^{M} \frac{1}{1 + \exp(-f_m)}.$$

This distribution is intractable, and we use the expectation-propagation algorithm[4] to approximate it by a multivariate normal distribution [Minka, 2001]. Once the posterior is computed, we can use it to generate predictions for new matches [Rasmussen and Williams, 2006]. These predictions come in the form of probability distributions $[p^{\mathrm{W}}, p^{\mathrm{D}}, p^{\mathrm{L}}]$ over the three outcomes (win, draw, loss).

We compare our predictive distributions against three baselines. First, we consider a simple Rao-Kupper model based on national team ratings obtained from a popular

[4]We use the GPy Python library (see: `https://sheffieldml.github.io/GPy/`) to fit the model; inference takes a minute for the 2008 test set (17 minutes for 2016).

Table 5.2 – Average logarithmic loss of our predictive model (PlayerKern), a model based on national team ratings (Elo), betting odds (Odds) and a random baseline (Random) on the final tournaments of three European championships. $M$ is the number of training instances, $P$ the number of distinct players and $T$ the number of test instances.

| Competition | $M$ | $P$ | $T$ | PlayerKern | Elo | Odds | Random |
|---|---|---|---|---|---|---|---|
| Euro 2008 | 4 390 | 7 875 | 31 | 0.969 | **0.910** | 0.979 | 1.099 |
| Euro 2012 | 15 594 | 21 735 | 31 | **0.939** | 1.003 | 0.953 | 1.099 |
| Euro 2016 | 24 887 | 33 157 | 51 | 1.067 | 1.102 | **1.020** | 1.099 |

website[5]. This model is similar to ours, but (*a*) it does not relate matches through players, hence does not consider club outcomes, and (*b*) as ratings are fixed values, it does not consider uncertainty in the ratings. Second, we consider average probabilities derived from the odds given by three large betting companies. Third, we consider a random baseline which always outputs $[1/3, 1/3, 1/3]$. The predictive distributions are evaluated using the average logarithmic loss over $T$ test instances

$$-\frac{1}{T}\sum_{i=1}^{T}\left[1_{\{y_i=\mathrm{W}\}}\log p_i^{\mathrm{W}} + 1_{\{y_i=\mathrm{D}\}}\log p_i^{\mathrm{D}} + 1_{\{y_i=\mathrm{L}\}}\log p_i^{\mathrm{L}}\right].$$

The logarithmic loss penalizes more strongly predictions that are both confident and incorrect. Table 5.2 summarizes the results.

Our predictive model performs well for 2008 and 2012, but slightly less so for 2016. It is noteworthy that the 2016 final tournament was generally less predictable than earlier editions. The case of the Elo baseline is interesting, as its accuracy varies wildly. Reasons for this might include the noise due to the online gradient updates, and the lack of proper uncertainty quantification in the ratings. Our method, in contrast, seems to produce more conservative predictions, but manages to achieve a more consistent performance.

## 5.5 Summary

In this short chapter, we have exposed a connection between a well-known pairwise comparison model and Gaussian-process classification, and have proposed a kernel that is able to transfer knowledge across different types of football matches—those between clubs and those between national teams. We have shown that a predictive model built on these ideas achieves a logarithmic loss that is competitive with betting odds. In future work, we would like to investigate how to incorporate aging into the model, i.e., how to progressively downweight older data.

[5]See: `http://www.eloratings.net/`.

# 6 Conclusion

Modern technologies enable the collection of comparison data at an unprecedented scale, opening up many new opportunities for businesses and researchers. But they also raise substantial challenges. Often, the number of parameters of the models used to analyze the data grows concurrently to the amount of data. This calls for new, efficient methods for collecting comparisons and learning models. In this thesis, we propose several solutions that highlight different aspects of efficiency.

- In Chapter 2, we address the problem of parameter inference for Luce's choice model. By expressing stationary points of the likelihood function as the stationary distribution of a Markov chain, we link recently proposed spectral estimators to maximum-likelihood methods. This link enables the development of new inference algorithms that are statistically and computationally efficient.

- In Chapter 3, we consider the active-learning setting. We study theoretically and empirically the performance of Quicksort when pairwise comparison outcomes follow the Bradley–Terry model. In scenarios where it is possible to adaptively select pairs of items to compare, we show that sorting-based active-learning strategies lead to significant gains in sample efficiency. Compared to competing active-learning methods, ours is computationally cheaper.

- In Chapter 4, we focus on choices in networks. In this case, we achieve data efficiency in a different way: we find that it is not necessary to observe distinct choices among well-defined sets of alternatives in order to estimate model parameters. Marginal information about the incoming and outgoing traffic at each node is sufficient. The network structure also enables a fast algorithm that scales to very large graphs.

- Finally, in Chapter 5, we tackle a concrete problem in sports. Based on past outcomes of football matches, we seek to predict the outcome of future matches between national teams. We devise a method that uses all the available data

efficiently: it considers the outcome of all matches—including those between clubs—and obtains predictions that outperform competing models. It does so by implicitly projecting the football matches in the space of players.

The approach we take in most of this thesis consists of distilling challenges faced in modern applications of choice models into simple and fundamental problems. We then propose methods to address these problems. We have applied these methods to real use-cases; however, there remain important classes of practical applications for which our methods are not applicable directly. We discuss three directions in which our work could be extended.

**Item features** With the exception of Chapter 5, we have assumed that the item strengths $\{\gamma_i\}$ or $\{\theta_i\}$ are free parameters. However, in some applications, we might have access to features that relate items to each other. We distinguish two cases. First, suppose that item $i$ is described by a real-valued feature vector $\boldsymbol{x}_i \in \mathbf{R}^D$, where typically $D \ll N$. Then, by setting $\theta_i = \boldsymbol{x}_i^\top \boldsymbol{w}$ for some latent parameter vector $\boldsymbol{w} \in \mathbf{R}^D$, we obtain the multinomial logit model [McFadden, 1973, Train, 2009]. Inference in this model is well-studied, but the issue of effective and efficient active learning remains widely open. Second, suppose that item $i$ is described by a binary vector $\boldsymbol{x}_i \in \{0,1\}^D$ describing the presence or absence of certain features. Then, we can model comparison outcomes using the elimination-by-aspects (EBA) model [Tversky, 1972], a model closely related to that of Luce. Preliminary work shows that the algorithms developed in Chapter 2 could be extended to the EBA model in the case of pairwise comparisons.

**Context of comparisons** Sometimes, the context in which choices are made is important. For example, we might prefer to listen to a different type of music depending on whether we are spending a quiet moment or doing sports, or whether it is summer or Christmas time, etc. In cases where the context is explicit, we fall back to the problem of integrating side information in the form of feature vectors. However, if the context is not explicitly observed, the problem becomes more difficult. We make a step towards addressing this problem in Ko et al. [2016], where we study a setting in which we observe sequences of choices. We propose a model where the context at time $t$ is encoded by previous choices made by a user in $(-\infty, t)$, and where the effective utility $\theta_i^{(t)}$ of item $i$ at time $t$ varies accordingly. In general, integrating latent context into choice models remains an interesting avenue for future research.

**Personalization** In applications using recommender systems, the task is often to learn a *distinct* preference profile for each user. For example, online service providers use these systems in order to tailor their service to the specific tastes of a user. An obvious but inefficient way to achieve personalization is to learn a distinct choice model for every user. As many users share similar preferences, it is sensible to take

advantage of similar users' choices to learn a given user's preferences. One approach is to postulate that there are a small number of (global) instances of Luce's model, and that individual preferences are formed by (user-specific) mixtures of these models [Gormley and Murphy, 2008, Ammar, 2015]. The inference problem then consists of jointly learning the global models and the user-specific mixture weights. To this end, the algorithms developed in Chapter 2 could be used to carry out the $M$ step in the *EM* algorithm of Gormley and Murphy [2008].

A potential weakness of models based on Luce's axiom (as well as those based on Thurstone's ideas, see Section 1.2.1) is that they are sensitive to outliers: the probability that the outcome of a comparison between $i$ and $j$ is inconsistent decreases *exponentially* fast with $|\theta_i - \theta_j|$. Hence, a small fraction of outliers (due, e.g., to the actions of dishonest users) can affect the model significantly. In addition to the extensions outlined above, the study of robust alternatives to Luce's model is an important direction for future work. On the one hand, model inference will likely require the development of new tools, beyond those presented in this thesis. On the other hand, there is hope that the sorting-based active-learning strategies presented in Chapter 3—including the theoretical bounds on Quicksort's performance—can be extended to heavy-tailed noise.

In conclusion, we hope to have convinced the reader that the study of comparison models is of paramount importance to improve online applications, because choices are the most natural way for humans to express their opinions (whether implicitly or explicitly). This thesis hopefully brings us a step closer towards effective methods for eliciting and analyzing comparison outcomes. As several challenges remain, research on choice models has a bright future ahead of itself.

# A Python Library

In this appendix[1], we give a brief overview of `choix`, an open-source Python library that provides implementations of inference algorithms for models based on Luce's choice axiom. The library was used for most experiments presented in this thesis and its code is publicly available at `https://github.com/lucasmaystre/choix`.

## A.1 Types of Data

The library handles four different types of observations, all of which are special cases of Luce's general choice model, defined in Section 1.2.3. The specialization enables (*a*) to write programs more concisely, (*b*) to represent the data using less memory, and (*c*) to perform computations more efficiently.

**Pairwise comparisons** The specialization of Luce's model to the case of pairwise comparisons is usually referred to as the Bradley–Terry model.

**Partial rankings** If the data consists of rankings over (a subset of) the items, the model variant is referred to as the Plackett–Luce model. A $K$-way ranking is effectively equivalent to $K - 1$ successive choices over the remaining alternatives.

**Top-1 lists** Also referred to in this thesis as *multiway choices*, top-1 lists correspond to choices of one item out of several identified alternatives. This type of observation subsumes all others.

**Choices in a network** When choices arise in a network, only the marginal incoming and outgoing traffic at every node of the network is necessary for inferring model parameters (see Chapter 4). Functions handling networked choice data thus dispense us from having to specify alternatives available at every choice.

[1]This appendix is based in part on the documentation of the `choix` library, available online at `http://choix.lum.li/en/latest/`.

## A.2 Inference Algorithms

For each type of data, `choix` exposes several different algorithms for parameter inference. This makes it possible to compare algorithms, e.g., in terms of numerical stability and running time, and to choose the one that works best in the particular regime of interest.

**Luce Spectral Ranking** The library provides a reference implementation of the two algorithms developed in Chapter 2: LSR and I-LSR. Rank Centrality [Negahban et al., 2012] is also implemented.

**Minorization-Maximization** The classic MM algorithm finds the MLE using a simple iterative procedure. This algorithm is known since the seminal work of Zermelo [1928].

**Convex optimization** The choice model's likelihood function is convex when using the parametrization in $\boldsymbol{\theta}$, and off-the-shelf convex optimizers can be used for maximum-likelihood inference. `choix` offloads this task to the `scipy` library[2].

**Approximate Bayesian inference** The expectation-propagation algorithm provides an effective way for computing an approximate posterior distribution of the parameters [Minka, 2001, Chu and Ghahramani, 2005b]. It is useful in cases where a measure of the uncertainty of the parameters' values is needed (e.g., in order to implement some of the Bayesian active-learning baselines of Chapter 3).

It is interesting to note that there is not one algorithm that consistently outperforms all others in all regimes. For example, algorithms based on the convex formulation of the model are numerically more stable when the range of the parameters $\boldsymbol{\theta}$ is large. However, when the range is small, they can be orders of magnitude slower than, e.g., LSR.

[2]See: `https://www.scipy.org/scipylib/index.html`.

# Bibliography


A. Agresti. *Foundations of Linear and Generalized Linear Models*. Wiley, 2015. [Cited on page 7]

N. Ailon. Reconciling real scores with binary comparisons: A unified logistic model for ranking. In *Advances in Neural Information Processing Systems 21*, Vancouver, BC, Canada, Dec. 2008. [Cited on pages 34, 38, and 53]

N. Ailon. An active learning algorithm for ranking from pairwise preferences with an almost optimal query complexity. *Journal of Machine Learning Research*, 13(Jan): 137–164, 2012. [Cited on page 33]

N. Ailon and M. Mohri. Preference-based learning to rank. *Machine Learning*, 80(2): 189–211, 2010. [Cited on page 36]

N. Ailon, M. Charikar, and A. Newman. Aggregating inconsistent information: Ranking and clustering. *Journal of the ACM*, 55(5):23:1–23:27, 2008. [Cited on page 34]

N. Alon, B. Bollobás, G. Brightwell, and S. Janson. Linear extensions of a random partial order. *The Annals of Applied Probability*, 4(1):108–123, 1994. [Cited on page 31]

A. Ammar. *Ranked Personalized Recommendations Using Discrete Choice Models*. PhD thesis, Massachusetts Institute of Technology, 2015. [Cited on page 89]

E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen. *LAPACK Users' Guide*. SIAM, third edition, 1999. [Cited on page 15]

B. C. Arnold, N. Balakrishnan, and H. N. Nagaraja. *A First Course in Order Statistics*. SIAM, 2008. [Cited on pages 42 and 55]

D. Ashbrook and T. Starner. Using GPS to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing*, 7(5):275–286, 2003. [Cited on page 60]

H. Azari Soufiani, W. Z. Chen, D. C. Parkes, and L. Xia. Generalized method-of-moments for rank aggregation. In *Advances in Neural Information Processing Systems 26*, Lake Tahoe, CA, USA, Dec. 2013. [Cited on pages 13, 16, 17, 22, 24, and 60]

C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006. [Cited on page 82]

R. A. Bradley. Some statistical methods in taste testing and quality evaluation. *Biometrics*, 9(1):22–38, 1953. [Cited on page 7]

R. A. Bradley and M. E. Terry. Rank analysis of incomplete block designs: I. The method of paired comparisons. *Biometrika*, 39(3/4):324–345, 1952. [Cited on pages 6, 7, and 21]

M. Braverman and E. Mossel. Noisy sorting without resampling. In *Proceedings of SODA'08*, San Francisco, CA, USA, Jan. 2008. [Cited on page 33]

S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. In *Proceedings of WWW'98*, Brisbane, Australia, Apr. 1998. [Cited on pages 58 and 60]

H. Bühlmann and P. J. Huber. Pairwise comparison and ranking in tournaments. *The Annals of Mathematical Statistics*, 34(2):501–510, 1963. [Cited on page 62]

F. Caron and A. Doucet. Efficient Bayesian inference for generalized Bradley–Terry models. *Journal of Computational and Graphical Statistics*, 21(1):174–196, 2012. [Cited on pages 6, 17, 59, 68, 70, and 72]

G. Casella and R. L. Berger. *Statistical Inference*. Duxbury Press, second edition, 2002. [Cited on pages 16 and 62]

M. Cattelan, C. Varin, and D. Firth. Dynamic Bradley–Terry modelling of sports tournaments. *Journal of the Royal Statistical Society, Series C (Applied Statistics)*, 62 (1):135–150, 2013. [Cited on page 81]

X. Chen, P. N. Bennett, K. Collins-Thompson, and E. Horvitz. Pairwise ranking aggregation in a crowdsourced setting. In *Proceedings of WSDM'13*, Rome, Italy, Feb. 2013. [Cited on pages 34 and 43]

W. Chu and Z. Ghahramani. Extensions of Gaussian processes for ranking: Semi-supervised and active learning. In *Proceedings of the NIPS 2005 Workshop on Learning to Rank*, Whistler, BC, Canada, Dec. 2005a. [Cited on pages 5, 34, 43, and 44]

W. Chu and Z. Ghahramani. Gaussian processes for ordinal regression. *Journal of Machine Learning Research*, 6(Jul):1019–1041, 2005b. [Cited on page 92]

W. Chu and Z. Ghahramani. Preference learning with Gaussian processes. In *Proceedings of ICML 2005*, Bonn, Germany, Aug. 2005c. [Cited on pages 5 and 81]

A. H. Copeland. A 'reasonable' social welfare function. Notes from a seminar on applications of mathematics to the social sciences, 1951. [Cited on page 39]

H. A. David. *The Method of Paired Comparisons*. Charles Griffin & Company, second edition, 1988. [Cited on page 7]

G. Debreu. Review of individual choice behavior: A theoretical analysis. *The American Economic Review*, 50(1):186–188, 1960. [Cited on page 9]

J. W. Demmel, S. C. Eisenstat, J. R. Gilbert, X. S. Li, and J. W. H. Liu. A supernodal approach to sparse partial pivoting. *SIAM Journal on Matrix Analysis and Applications*, 20(3):720–755, 1999. [Cited on page 15]

P. Diaconis and R. L. Graham. Spearman's footrule as a measure of disarray. *Journal of the Royal Statistical Society, Series B*, 39(2):262–268, 1977. [Cited on page 33]

L. L. Dines. On positive solutions of a system of linear equations. *Annals of Mathematics*, 28(1/4):386–392, 1926. [Cited on page 66]

D. P. Dubhashi and A. Panconesi. *Concentration of Measure for the Analysis of Randomized Algorithms*. Cambridge University Press, 2009. [Cited on pages 35 and 48]

C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the web. In *Proceedings of WWW'01*, Hong Kong, China, May 2001. [Cited on pages 13 and 16]

O. Dykstra, Jr. Rank analysis of incomplete block designs: A method of paired comparisons employing unequal repetitions on pairs. *Biometrics*, 16(2):176–188, 1960. [Cited on pages 6 and 16]

A. Elo. *The Rating Of Chess Players, Past & Present*. Arco Publishing, 1978. [Cited on pages 7 and 81]

L. Fahrmeir and G. Tutz. Dynamic stochastic models for time-dependent ordered paired comparison systems. *Journal of the American Statistical Association*, 89(428): 1438–1449, 1994. [Cited on page 81]

L. Festinger. A theory of social comparison processes. *Human Relations*, 7(2):117–140, 1954. [Cited on page 2]

F. Fogel, A. d'Aspremont, and M. Vojnovic. SerialRank: Spectral ranking using seriation. In *Advances in Neural Information Processing Systems 27*, Montréal, QC, Canada, Dec. 2014. [Cited on page 16]

L. R. Ford, Jr. Solution of a ranking problem from binary comparisons. *The American Mathematical Monthly*, 64(8):28–33, 1957. [Cited on pages 6, 14, and 16]

M. E. Glickman. *Paired Comparison Models with Time-Varying Parameters*. PhD thesis, Harvard University, 1993. [Cited on page 81]

M. 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. [Cited on page 81]

M. E. Glickman. Introductory note to 1928 (= 1929). In *Ernst Zermelo - Collected Works II*, pages 616–671. Springer, 2013. [Cited on page 7]

J. E. Gonzalez, R. S. Xin, A. Dave, D. Crankshaw, M. J. Franklin, and I. Stoica. GraphX: Graph processing in a distributed dataflow framework. In *Proceedings of OSDI'14*, Broomfield, CO, USA, Oct. 2014. [Cited on pages 70 and 77]

I. C. Gormley and T. B. Murphy. Exploring voting blocs within the Irish electorate: A mixture modeling approach. *Journal of the American Statistical Association*, 103(483): 1014–1027, 2008. [Cited on page 89]

J. Guiver and E. Snelson. Bayesian inference for Plackett–Luce ranking models. In *Proceedings of ICML 2009*, Montréal, QC, Canada, June 2009. [Cited on page 17]

B. Hajek, S. Oh, and J. Xu. Minimax-optimal inference from partial rankings. In *Advances in Neural Information Processing Systems 27*, Montréal, QC, Canada, 2014. [Cited on pages 17, 24, 33, and 60]

N. Halko, P. G. Martinsson, and J. A. Tropp. Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions. *SIAM Review*, 53(2):217–288, 2011. [Cited on page 16]

T. Hastie and R. Tibshirani. Classification by pairwise coupling. *The Annals of Statistics*, 26(2):451–471, 1998. [Cited on page 6]

R. Heckel, N. B. Shah, K. Ramchandran, and M. J. Wainwright. Active ranking from pairwise comparisons and when parametric assumptions don't help. Preprint, `arXiv:1606.08842v2 [cs.LG]`, Sept. 2016. [Cited on page 34]

R. Herbrich, T. Minka, and T. Graepel. TrueSkill$^{\text{TM}}$: A Bayesian skill rating system. In *Advances in Neural Information Processing Systems 19*, Vancouver, BC, Canada, Dec. 2006. [Cited on pages 81 and 83]

C. A. R. Hoare. Quicksort. *The Computer Journal*, 5(1):10–16, 1962. [Cited on pages 10 and 34]

N. Houlsby, J. M. Hernández-Lobato, F. Huszár, and Z. Ghahramani. Collaborative Gaussian processes for preference learning. In *Advances in Neural Information Processing Systems 25*, Lake Tahoe, CA, Dec. 2012. [Cited on page 34]

D. R. Hunter. MM algorithms for generalized Bradley–Terry models. *The Annals of Statistics*, 32(1):384–406, 2004. [Cited on pages 6, 14, 16, 17, 19, 21, 25, 26, 27, 59, 64, 69, and 70]

K. G. Jamieson and R. D. Nowak. Active ranking using pairwise comparisons. In *Advances in Neural Information Processing Systems 24*, Granada, Spain, Dec. 2011. [Cited on page 33]

M. Kafsi, M. Grossglauser, and P. Thiran. Traveling salesman in reverse: Conditional Markov entropy for trajectory segmentation. In *Proceedings of ICDM'15*, Atlantic City, NJ, USA, Nov. 2015. [Cited on page 60]

T. Kamishima and S. Akaho. Efficient clustering for orders. In *Mining Complex Data*, pages 261–279. Springer, 2009. [Cited on pages 25 and 45]

J. G. Kemeny and J. L. Snell. *Finite Markov Chains*. Springer, 1976. [Cited on page 63]

H. K. Khalil. *Nonlinear Systems*. Prentice-Hall, second edition, 1996. [Cited on page 19]

D. E. Knuth. *The art of computer programming: Sorting and searching*, volume 3. Addison-Wesley, second edition, 1998. [Cited on page 44]

Y.-J. Ko, L. Maystre, and M. Grossglauser. Collaborative recurrent neural networks for dynamic recommender systems. In *Proceedings of ACML 2016*, Hamilton, New Zealand, Nov. 2016. [Cited on page 88]

R. Kumar, A. Tomkins, S. Vassilvitskii, and E. Vee. Inverting a steady-state. In *Proceedings of WSDM'15*, Shanghai, China, Feb. 2015. [Cited on pages 11, 17, 57, 59, 60, 63, 66, and 77]

H. G. Landau. On dominance relations and the structure of animal societies: III The condition for a score structure. *The Bulletin of Mathematical Biophysics*, 15(2):143–148, 1953. [Cited on page 48]

K. Lange, D. R. Hunter, and I. Yang. Optimization transfer using surrogate objective functions. *Journal of Computational and Graphical Statistics*, 9(1):1–20, 2000. [Cited on page 70]

D. A. Levin, Y. Peres, and E. L. Wilmer. *Markov Chains and Mixing Times*. American Mathematical Society, 2008. [Cited on pages 15 and 27]

Y. Liu, B. Gao, T.-Y. Liu, Y. Zhang, Z. Ma, S. He, and H. Li. BrowseRank: Letting web users vote for page importance. In *Proceedings of SIGIR'08*, Singapore, July 2008. [Cited on page 60]

R. D. Luce. *Individual Choice Behavior: A Theoretical Analysis*. Wiley, 1959. [Cited on pages 7, 8, 58, and 60]

D. J. C. MacKay. *Bayesian Methods for Adaptive Models*. PhD thesis, California Institute of Technology, 1992. [Cited on pages 34 and 43]

M. J. Maher. Modelling association football scores. *Statistica Neerlandica*, 36(3):109–118, 1982. [Cited on page 81]

G. Malewicz, M. H. Austern, A. J. C. Bik, J. C. Dehnert, I. Horn, N. Leiser, and G. Czajkowski. Pregel: A system for large-scale graph processing. In *Proceedings of SIGMOD'10*, Indianapolis, IN, USA, June 2010. [Cited on pages 70 and 77]

L. Maystre and M. Grossglauser. Fast and accurate inference of Plackett–Luce models. In *Advances in Neural Information Processing Systems 28*, Montréal, QC, Canada, Dec. 2015. [Cited on page 13]

L. Maystre and M. Grossglauser. ChoiceRank: Identifying preferences from node traffic in networks. In *Proceedings of ICML 2017*, Sydney, Australia, Aug. 2017a. [Cited on page 57]

L. Maystre and M. Grossglauser. Just sort it! a simple and effective approach to active preference learning. In *Proceedings of ICML 2017*, Sydney, Australia, Aug. 2017b. [Cited on page 31]

L. Maystre, V. Kristof, A. J. González Ferrer, and M. Grossglauser. The player kernel: Learning team strengths based on implicit player contributions. Preprint, `arXiv:1609.01176 [cs.LG]`, Sept. 2016. [Cited on page 79]

D. McFadden. Conditional logit analysis of qualitative choice behavior. In P. Zarembka, editor, *Frontiers in Econometrics*, pages 105–142. Academic Press, 1973. [Cited on page 88]

D. McFadden. Economic choices. *American Economic Review*, 91(3):351–378, 2001. [Cited on page 1]

D. McFadden, A. Talvitie, S. Cosslett, I. Hasan, M. Johnson, F. Reid, and K. Train. Demand model estimation and validation. Technical report, Institute of Transportation Studies, University of California, Berkeley, 1977. [Cited on page 1]

F. McSherry, M. Isard, and D. G. Murray. Scalability! but at what COST? In *Proceedings of HotOS XV*, Warth, Switzerland, May 2015. [Cited on pages 74 and 77]

R. Meusel, S. Vigna, O. Lehmberg, and C. Bizer. Graph structure in the Web—revisited: A trick of the heavy tail. In *Proceedings of WWW'14*, Seoul, Korea, Apr. 2014. [Cited on page 76]

T. P. Minka. *A family of algorithms for approximate Bayesian inference*. PhD thesis, Massachusetts Institute of Technology, 2001. [Cited on pages 85 and 92]

F. Mosteller. Remarks on the method of paired comparisons: I. The least squares solution assuming equal standard deviations and equal correlations. *Psychometrika*, 16(1):3–9, 1951. [Cited on page 5]

S. Negahban, S. Oh, and D. Shah. Iterative ranking from pair-wise comparisons. In *Advances in Neural Information Processing Systems 25*, Lake Tahoe, CA, Dec. 2012. [Cited on pages 10, 13, 16, 17, 21, 33, 60, and 92]

L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank citation ranking: Bringing order to the Web. Technical report, Stanford University, Jan. 1998. [Cited on page 16]

R. L. Plackett. The analysis of permutations. *Journal of the Royal Statistical Society, Series C (Applied Statistics)*, 24(2):193–202, 1975. [Cited on page 9]

A. Rajkumar and S. Agarwal. A statistical convergence perspective of algorithms for rank aggregation from pairwise data. In *Proceedings of ICML 2014*, Beijing, China, June 2014. [Cited on pages 17, 33, and 40]

P. V. Rao and L. L. Kupper. Ties in paired-comparison experiments: A generalization of the Bradley-Terry model. *Journal of the American Statistical Association*, 62(317): 194–204, 1967. [Cited on pages 22, 25, 81, and 82]

C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. MIT Press, 2006. [Cited on pages 6, 81, 82, and 85]

T. L. Saaty. *The Analytic Hierarchy Process: Planning, Priority Setting, Resource Allocation*. McGraw-Hill, 1980. [Cited on page 16]

M. J. Salganik and K. E. C. Levy. Wiki surveys: Open and quantifiable social data collection. *PLoS ONE*, 10(5):1–17, 2015. [Cited on pages 3 and 56]

T. Salimans, U. Paquet, and T. Graepel. Collaborative learning of preference rankings. In *Proceedings of RecSys'12*, Dublin, Ireland, Sept. 2012. [Cited on page 34]

A. I. Schein and L. H. Ungar. Active learning for logistic regression: an evaluation. *Machine Learning*, 68(3):235–265, 2007. [Cited on page 32]

R. Sedgewick and K. Wayne. *Algorithms*. Addison-Wesley, fourth edition, 2011. [Cited on page 36]

B. Settles. *Active Learning*. Morgan & Claypool Publishers, 2012. [Cited on pages 10, 43, and 46]

H. Stern. Are all linear paired comparison models empirically equivalent? *Mathematical Social Sciences*, 23(1):103–117, 1992. [Cited on page 7]

B. Szörényi, R. Busa-Fekete, A. Paul, and E. Hüllermeier. Online rank elicitation for Plackett–Luce: A dueling bandits approach. In *Advances in Neural Information Processing Systems 28*, Montréal, QC, Canada, Dec. 2015. [Cited on pages 34 and 39]

L. L. Thurstone. A law of comparative judgment. *Psychological Review*, 34(4):273–286, 1927a. [Cited on pages 4 and 23]

L. L. Thurstone. The method of paired comparisons for social values. *The Journal of Abnormal and Social Psychology*, 21(4):384–400, 1927b. [Cited on pages 5, 6, and 17]

J. A. Tomlin. A new paradigm for ranking pages on the World Wide Web. In *Proceedings of WWW'03*, Budapest, Hungary, May 2003. [Cited on page 60]

K. E. Train. *Discrete Choice Methods with Simulation*. Cambridge University Press, second edition, 2009. [Cited on pages 9, 58, and 88]

K. Tsukida and M. R. Gupta. How to analyze paired comparison data. Technical report, University of Washington, Seattle, WA, USA, May 2011. [Cited on page 7]

A. Tversky. Elimination by aspects: A theory of choice. *Psychological Review*, 79(4): 281–299, 1972. [Cited on page 88]

M. Vojnovic and S.-Y. Yun. Parameter estimation for generalized Thurstone choice models. In *Proceedings of ICML 2016*, New York, NY, USA, June 2016. [Cited on pages 33, 58, and 60]

J. Wang, N. Srebro, and J. Evans. Active collaborative permutation learning. In *Proceedings of KDD'14*, New York, NY, USA, Aug. 2014. [Cited on page 33]

E. Wulczyn and D. Taraborelli. Wikipedia clickstream, Apr. 2016. URL `https://figshare.com/articles/Wikipedia_Clickstream/1305770/16`. [Cited on page 73]

J. I. Yellot, Jr. The relationship between Luce's choice axiom, Thurstone's theory of comparative judgment, and the double exponential distribution. *Journal of Mathematical Psychology*, 15(2):109–144, 1977. [Cited on page 7]

Y. Yue, J. Broder, R. Kleinberg, and T. Joachims. The $K$-armed dueling bandits problem. In *Proceedings of COLT 2009*, Montréal, QC, Canada, June 2009. [Cited on page 34]

E. Zermelo. Die Berechnung der Turnier-Ergebnisse als ein Maximumproblem der Wahrscheinlichkeitsrechnung. *Mathematische Zeitschrift*, 29(1):436–460, 1928. [Cited on pages 6, 14, 16, 21, 81, and 92]

# Lucas Maystre

EPFL IC LCA 4, Station 14
CH-1015 Lausanne
☎ +41 76 482 19 63
✉ lucas.maystre@epfl.ch
http://lucas.maystre.ch

Born on July 22nd, 1988
Swiss citizen
Unmarried

## Education

2012 – 2018 **Ph.D. in Computer Science**
**École Polytechnique Fédérale de Lausanne (EPFL)**
My research focuses on statistical and algorithmic aspects of learning preferences from comparisons. I am advised by Matthias Grossglauser.

2006 – 2012 **M.Sc. in Communication Systems**
**École Polytechnique Fédérale de Lausanne (EPFL)**
Master's thesis on a music recommender system, supervised by Matthias Grossglauser.

2010 - 2011 **IVEO Graduate Exchange Program**
**University of California, Berkeley**
One-year exchange program. Hosted by Dawn Song, I worked on web application security and took graduate classes.

## Professional Experience

2016
3 months **Microsoft Research, Cambridge (UK) – Research Intern**
Collaborated with Milan Vojnovic and Brendan Murphy on a research project related to organizational analytics.

2011
3 months **Facebook, Palo Alto – Security Engineering Intern**
Participated in ongoing efforts to make Facebook's infrastructure more secure and to help engineers write safer code.

2010
4 months **Google, Zürich – Software Engineering Intern**
Launched a new version of a web application that allows to target ads on YouTube. Worked in a small, international team.

2009 – 2010
6 months **Open Systems AG, Zürich – Intern**
Collaborated with security engineers to develop network security products operated in 100+ countries worldwide.

## Honors and Awards

| | |
|---|---|
| 2016 | Distinguished Service Award, IC Department, EPFL |
| 2016 | Google PhD Fellowship in Machine Learning. |
| 2012 | EPFL I&C Departmental Fellowship. |
| 2006 | Gymnase de Chamblandes prize in Latin and French studies. |

## Selected Publications

| | |
|---|---|
| 2017 | **A Simple and Effective Approach to Active Preference Learning**<br>L. Maystre, M. Grossglauser, *ICML 2017*. |
| 2017 | **ChoiceRank: Identifying Preferences from Node Traffic in Networks**<br>L. Maystre, M. Grossglauser, *ICML 2017*. |
| 2016 | **Collaborative RNNs for Dynamic Recommender Systems**<br>Y.-J. Ko, L. Maystre, M. Grossglauser, *ACML 2016*. |
| 2015 | **Fast and Accurate Inference of Plackett–Luce Models**<br>L. Maystre, M. Grossglauser, *NIPS 2015*. |

## Open-Source Contributions

| | |
|---|---|
| 2015 – | **choix**<br>Python library for research on statistical choice models. |
| 2012 | **CPython**<br>Fixed issue #11175. My code shipped with Python 3.4 |
| 2012 – | I have many other public repositiories on GitHub, ranging from a fork of a popular SVD library to cooking recipes. |

## Languages & Miscellanea

| | |
|---|---|
| Languages | **French :** native<br>**English :** fluent, written and spoken<br>**German :** fluent, written and spoken |
| Service | Since 2013, I have proposed and supervised 11 semester-long student projects at the bachelor and master levels. |
| Hobbies | Football – I am a referee, currently in 3rd division; playing piano, photography. |