DATA DRIVEN EXPERT ASSIGNMENT

A Dissertation Presented

by

JUSTIN PAYAN

Submitted to the Graduate School of the
University of Massachusetts Amherst in partial fulfillment
of the requirements for the degree of

DOCTOR OF PHILOSOPHY

February 2025

Manning College of Information and Computer Sciences

DATA DRIVEN EXPERT ASSIGNMENT

A Dissertation Presented

by

JUSTIN PAYAN

Approved as to style and content by:

Yair Zick, Chair

Andrew McCallum, Member

Andrew McGregor, Member

Nihar B. Shah, Member

Ramesh K. Sitaraman, Associate Dean for Educational Programs and Teaching
Manning College of Information and Computer Sciences

DEDICATION

To Hushang M. Payan, the original Dr. Payan.

ACKNOWLEDGMENTS

I am incredibly grateful to Yair Zick for teaching me about market design and how to do good research in general. I greatly enjoyed working in Yair's lab, and I am so fortunate to have spent my Ph.D. with someone who is not only a great advisor but also a fun person.

I would also like to thank the members of my thesis committee, Andrew McCallum, Andrew McGregor, and Nihar Shah. They provided fantastic ideas and comments on each piece of this thesis, and supported me throughout my Ph.D. Thank you in particular to Andrew McCallum for supporting me as a research assistant and mentoring me through my first three semesters at UMass.

I would like to thank all of my collaborators, especially Cyrus Cousins, Elita Lobo, Amirmahdi Mirfakhar, Rik Sengupta, and Vignesh Viswanathan, who have been close collaborators and great friends through my Ph.D. Thank you to Nick Perello and Paula Navarrete for being great friends and providing important feedback on my work, even though we did not get to formally collaborate. I also learned a lot from and enjoyed working with (in alphabetical order) Arpita Biswas, Rasika Chakravarthy, Ananya Gupta, Hadi Hosseini, Eric Johnson, Ari Kobren, Yuval Merhav, Swaroop Mishra, Sheshera Mysore, Neha Nayak Kennard, Carina Negreanu, Elnaz Nouri, Sweta Panda, Christian Poelitz, Kartik Sridhar, Chris Tensmeyer, and Rohit Vaish, among others. Special thanks goes to my mentees Vedant Bhatia, Hassan Gudal, and Vishisht Rao; I greatly enjoyed working with you and seeing your skills grow. I would particularly like to thank Nick Monath, who took so much time to mentor me in my first year. I also want to thank Fred Maier and Don Potter at the University of Georgia, who provided me with support and guidance early in my career. Thanks to the

Massachusetts and U.S. federal taxpayers who funded my work directly via the National Science Foundation and indirectly through support for the Manning College of Information and Computer Sciences. I am grateful to UMass CICS in particular for supporting me via the generous Thesis Proposal Writing Fellowship in Spring 2023 and the Summer Dissertation Writing Fellowship in Summer 2024.

Finally, I would never have made it through the Ph.D. without the support of my family; a heartfelt thank you goes to my parents John and Jana Payan, my brother Jackson Payan, and my fiancée Melissa Shinfuku.

ABSTRACT

DATA DRIVEN EXPERT ASSIGNMENT

FEBRUARY 2025

JUSTIN PAYAN
B.A., B.Sc., UNIVERSITY OF GEORGIA
M.Sc., UNIVERSITY OF GEORGIA
Ph.D., UNIVERSITY OF MASSACHUSETTS AMHERST

Directed by: Professor Yair Zick

The modern knowledge economy relies on expertise. In important technocratic tasks such as scientific peer review and community question answering, knowledge workers can only fulfill requests they have the expertise, interest, and availability to complete. We develop multiple novel approaches to assign experts to requests, addressing questions of fairness, scalability, assignment quality, and robustness to uncertainty. We use peer review as the primary case study, though Chapter 3 highlights the domain of community question answering. Our algorithms can be applied to other domains where resource-constrained experts are assigned to complete complex requests, such as crowd-sourced editing of knowledge repositories or corporate staff assignment.

Expert assignments must be both fair and welfare efficient, so that all requests receive a reasonably well-qualified set of experts. We first present a set of simple mechanisms that fairly distribute expertise across requests, with welfare guarantees. Our algorithms, Greedy Expert Round Robin and FairSequence, assign experts in such a way that no request “envies” another request’s assigned experts.

Although fairness and welfare criteria ensure evenly-distributed, high quality expertise, they both depend on the method of quantifying expert performance. In automated reviewer assignment systems, existing methods for estimating the benefits of assigning each reviewer to each paper can be noisy and ineffective. We take a data-driven perspective on the expert assignment problem, demonstrating how to more accurately estimate the benefits of assigning experts to requests. We train a variety of models to predict answer quality on StackExchange, then compare the results when using these models to produce constrained assignments of users to questions. This study demonstrates the benefits of fully predictive expert assignment.

No matter how accurate our predictive model, we always are uncertain when we assign experts to requests. Distribution shift can cause our models to make errors, or experts may be unable to perform due to unforeseen circumstances. We discuss two main solutions to hedge against the worst outcomes. The *robust optimization* framework optimizes over a region containing the true matching scores with high probability. The *stochastic optimization* framework assigns experts using a percentile criterion over the assignment objective. We study both the robust and stochastic approaches for utilitarian and egalitarian welfare objectives, and we detail applications in reviewer assignment and community question answering.

Expert assignment is a rich problem, which needs to be addressed from both a data analysis and algorithmic lens. Our work improves the end-to-end expert assignment pipeline, which will result in less wasted time and greater productivity for knowledge workers.

TABLE OF CONTENTS

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

CHAPTER 1
INTRODUCTION

Peer review is a fundamental institution for evaluating scientific knowledge. Over the 20^{th} century, the scientific profession has grown significantly, and the institution of peer review has struggled with the increased scale. Modern computer science conferences receive thousands of submissions, matched to committees of similar size. At this scale, it is challenging to ensure reviewers possess the proper expertise for their assigned papers. In large conferences such as NeurIPS/ICML/AAAI/IJCAI, reviewer assignment is largely automated through systems such as the Toronto Paper Matching System (TPMS) [38], Microsoft CMT [137], or OpenReview [119]. Inappropriately assigned reviewers may lead to failures: misinformed decisions, reviewer disinterest, and a general mistrust of the peer-review process.

Constrained Expert Assignment. Reviewer assignment is just one example of the *constrained expert assignment problem.* In this problem, experts with limited resources are assigned to complete requests requiring specific technical knowledge or skills. Experts without the requisite skills for a request may fail to produce a satisfactory response, so we aim to assign qualified experts to each request. We cannot route all requests to the top experts, since there may be a large number of requests and each expert can complete a limited number of them. In many expert assignment domains, a massive number of requests needs to be solved in a short time period, necessitating algorithmic routing of experts to requests. Other examples of large-scale constrained expert assignment include question routing for community question answering [175, 181], employee shift or project assignment [94, 130], recommending editors in collaborative wiki editing [89, 136, 172], peer grading in MOOCs [63, 149], assigning teachers to courses [166], and selecting software engineers to fix bugs [178].

Desiderata for Expert Assignment. We study the accuracy, fairness, robustness, and scalability of expert assignment algorithms. Overall assignment accuracy maintains quality standards for the conferences, question-answering sites, community-written encyclopedias, etc that rely on pools of expert contributors. However, it is imperative that we do not sacrifice quality on some requests to obtain higher overall matching scores. In peer review, papers which receive poorly matched reviewers may be unfairly rejected or receive unhelpful feedback, causing the authors real harm [32, 152, 153]. Similar problems can occur in any task requiring expertise; when the experts recommended for or assigned to a request are ill-fitting, they are liable to produce incorrect or unhelpful responses. We thus desire algorithms which are globally accurate and fair. Uncertainty in affinity score computation is another major source of error in assignment [100]. When we assign an expert to a request, we are interested in ensuring the *quality of the future response*, which is fundamentally noisy. In peer review, conferences often must assign reviewers to papers within a timespan of several days [41]. To enable interactive workflows at that timescale, we require algorithms that can scale to thousands of experts and requests with minimal computational overhead. We present algorithms and analyses that address all of these concerns.

Speed, Fairness, and Welfare-Efficiency. In Chapter 2, we present a variant on the round-robin procedure, *Expert Round Robin* (`ERR`), that satisfies *approximate envy-freeness* [27, 103] for expert assignment. `ERR` assigns the same number of experts to each request, but some domains require variable-sized assignments. We therefore also study a family of weighted picking sequences which satisfy a commonly-used approximate *weighted* envy-freeness constraint [35]. This constraint generalizes envy-freeness to settings where agents have unequal item demands, ensuring that agents are approximately envy-free after normalizing by their demands. Finally, we present a weighted picking sequence mechanism called `FairSequence` that targets the weighted envy-freeness criterion to offer fairness in a more general setting. Using data from three conferences, we show that `FairSequence` runs an order of magnitude faster and provides approximate envy-freeness guarantees that are violated by existing approaches. Its simple design makes it flexible to new assignment constraints. `FairSequence` is available in the OpenReview conference management platform [119], and has been used by at least 14 venues to date (source: personal correspondence with OpenReview).

Computing Valuations. The remainder of the thesis studies *valuations*, which are the basis of the objectives in expert assignment. In automated reviewer assignment, conferences typically construct affinity scores that reflect reviewer expertise and interest via four main sources of information. These sources are (a) *subject-area matching* (SAM) scores or keyword-based matching, where reviewer-provided areas of expertise are compared against keywords submitted by paper authors, (b) *textual similarity scores*, often implemented by the well-known Toronto Paper Matching System (TPMS) [38] or ACL scores [117], (c) *bidding*, where reviewers express their explicit ability and desire to review papers, and finally (d) *recommendations*, through which program committee members may suggest reviewers for papers. The overall affinity scores are typically computed as a linear combination of these four scores. CMT implements their affinity scores this way [138], as does OpenReview [121]. Recent conferences such as AAAI 2021 took a similar approach, linearly combining TPMS scores, ACL scores, and SAM scores, and raising the sum to some power based on the reviewer bids [100].

Uncertainty in Valuations. Each of these common affinity score components can be missing or inaccurate. State-of-the-art document similarity measures disagree with expert judgments up to 43% of the time [159], and nearly 40% of TPMS scores were completely missing in AAAI 2021 [100]. Although the AAAI 2021 organizers do not explain why so many TPMS scores are missing, missing scores occur for several reasons, including reviewers opting out of the system or providing insufficient or empty publication records. Between 5% and 15% of papers in major AI conferences receive fewer than three positive bids, but there is evidence that many missing bids would be positive if collected [60, 112, 141]. Although no systematic study has been performed on keyword-based similarity scores, keyword matching accuracy depends on authors and reviewers using consistent terminology, and subtleties are invariably lost in the process. Even reviewers directly suggested by knowledgeable editors or the paper authors have been shown to perform surprisingly poorly on average, as measured by third-party annotators via the Review Quality Index [144, 171], showing that recommendations can be noisy as well.

Improving Valuation Predictions. In Chapter 3, we investigate more comprehensive sets of features for building predictive models of expert performance. Without access to comprehensive data sources for peer review, we take the community question answering site StackExchange as our primary case study. In constrained expert assignment tasks, we would ideally assign experts to tasks so as to optimize overall task performance, but expert performance is unknown prior to making the assignment. We propose predicting these performance metrics and assigning using the predictions. Using an expert-assignment task derived from StackExchange, we show that explicitly predicting expert performance has a large impact on assignment decisions and can improve overall welfare. We demonstrate this claim using both theoretical bounds on statistical generalization guarantees and automated metrics of assignment quality. This work highlights the effectiveness of predictive assignment, and the need to collect high quality datasets linking pre- and post-allocation measures in other important expert assignment tasks such as peer review.

Optimizing over Uncertain Valuations. In Chapter 4, we address the uncertainty inherent in expert assignment. To our knowledge, every reviewer assignment system still relies on affinity score estimates, but does not directly account for the fact that these scores are noisy estimates of assignment quality. Our work addresses this fundamental gap. We investigate a generalized notion of affinity score, where system designers can implement affinity using any measure of fit between experts and requests. These measures may or may not be fully observable; for example, conference organizers may estimate unknown bids as part of affinity computation, or a community question answering site could use a predictive model like the one described in Chapter 3. Using these predictors, we apply robust and stochastic optimization methods to either maximize the minimum welfare objective over an uncertainty set, or to maximize statistics of the distribution over welfare.

1.1 Related Work

The generalized expert assignment problem has deep roots in the general *assignment problem* [30]. A few papers have investigated the expert assignment problem as a general construct, originally formulating it as a linear assignment problem (estimate match scores, then maximize the sum of scores subject to constraints) [166]. Follow-up work incorporates other concerns like online matching settings [75], team dynamics [4], or incorporating predictions of task completion rates [134].

Although data-driven expert assignment has multiple variants and many applications, we focus on two primary applications in this thesis – automated reviewer assignment for peer reviewed conferences, and identifying expert users on community question answering sites. We highlight papers in each of these areas. The communities working on these problems are typically distinct. However, there are many similarities between the problems, and insights often translate across domains.

We also provide brief discussions of solutions we apply to solve the expert assignment problem; fair allocation of indivisible items, and robust and stochastic optimization algorithms.

1.1.1 Automated Reviewer Assignment

Computer scientists have been automating conference reviewer assignment for over three decades [58]. Aksoy et al. [2], Zhao and Zhang [184] provide comprehensive overviews of the problem; we focus on the most recent and relevant approaches here.

Most recent approaches model the problem as a mixed-integer linear program maximizing affinity between reviewers and papers; the Toronto Paper Matching System (`TPMS`) is the most notable work with this formulation [38, 39], though there are others [101, 167]. The affinity typically models alignment between reviewer expertise and paper topics, but can incorporate other relevant notions like reviewer bids, conflicts of interest, and author suggestions [100, 115, 141]. Affinities are generally considered a good proxy for value at both an individual and collective level, since higher-affinity reviewers will typically be more qualified for and interested in a paper, resulting in more detailed and accurate reviews. Affinity scores are universally available in systems like `TPMS`, Microsoft CMT, or OpenReview, and it is standard practice to use these affinity scores to compute welfare and fairness measures [44, 80, 91, 100, 148, 157].

Chapter 2 focuses heavily on fairness in reviewer assignment. A number of prior works consider fairness objectives in peer review, though none of them consider envy-freeness up to one item. Hartvigsen et al. [74] ensure that at least one qualified reviewer is assigned to each paper. A few recent approaches approximately maximize the minimum paper score, or maximize the sum of scores subject to a minimum individual score threshold [91, 118, 157]. Aziz et al. [9] present a reviewer assignment algorithm that satisfies the *core*, ensuring that no group of papers can deviate without reducing their total welfare. A number of works study fair assignment of papers to reviewers, allowing reviewers to express preferences over papers by bidding [8, 65, 102, 165]. This setting aims to be fair to the reviewers rather than the papers. Other works target reviewer assignments with properties besides fairness or efficiency; Long et al. [106] avoid conflicts of interest, while Kou et al. [92] and Ahmed et al. [1] assign sets of reviewers with diverse interests and full coverage of the papers' topics.

Chapters 3 and 4 focus on how to assign experts to requests when the affinities between experts and requests are not known. This work is echoed in prior efforts for reviewer assignment. Some prior work attempts to predict missing bids for reviewer assignment, and to use these predicted bids to compute and evaluate assignments [42, 139]. Charlin et al. [40] also evaluate reviewer assignments under a predictive model imputing missing relevance scores. Although these works study the quality of assignment under a predictive model, none assigns using predictions of final task completion quality (we discuss this strategy in Chapter 3, and show in detail how to optimize under these predictive scores in Chapter 4). Saveski et al. [143] develop a model for counterfactual evaluation of alternative reviewer assignments on past conferences. They measure review quality by the reviewers' self-reported expertise and confidence measures, and demonstrate that textual similarity measures (like the Toronto Paper Matching System [38, 39]) are more directly relevant to assigning confident and (self-reported) expert reviewers than bids and keywords. However, existing similarity score computation methods still make many errors [159]. Taken together, the current findings in the literature demonstrate that we cannot fully trust existing affinity score computation methods as predictors of match quality. These works have similar motivation to our own; our work proposes a holistic, end-to-end solution to the problem of noisy affinity scores by predicting and optimizing for metrics of review quality. Other works use modern NLP techniques to improve document-based similarity scores [117], encourage reviewers to bid on underbid papers [60, 112, 141], or disincentivize strategic bidding behavior [80, 81, 82]. Although these approaches reduce uncertainty, they do not explicitly account for uncertainty.

1.1.2 Community Question Answering

Many existing works learn to recommend users on community question answering (CQA) forums like StackExchange, Quora, and Yahoo Answers. Sun et al. [163, 164] predict the future user-voted score when assigning a user to a given question. In Chapter 3 we target the same outcome variable, but we additionally contribute rigorous feature importance analysis, updated NLP techniques, and examination through the lens of constrained expert assignment (rather than providing isolated, per-question user recommendation). Liu et al. [104], Tondulkar et al. [169] predict which user's answer will be marked as accepted by the original question poster. Yang et al. [180] recommend users to questions with topics that interest them and in which they have expertise, irrespective of the user's competence. Qian et al. [134] study the setting where experts are sent requests for work, and they maximize the topical similarity of recommended experts as well as the acceptance rate of the invitations. While much attention has been paid to expert recommendation and prediction of answer quality in CQA forums, none of these works have addressed the constraints of the experts being assigned, or trade-offs among the many valid measures of answer quality in CQA. Users can only answer a small number of questions at a time, so we must consider their limited time when recommending questions if we want all questions answered.

A particularly thorny aspect of predictive expert assignment is identifying metrics for answer quality. Zhu et al. [186] asked both users and subject-matter experts to give a list of important criteria for evaluating answer quality, identifying 13 major criteria for answer quality measurement. They also asked experts to label answers as satisfying or not satisfying each criterion, and finally to rate each answer as good or bad overall. These and other similar criteria have been used or rediscovered in other important CQA studies [62, 147]. We employ them as features in our predictive model and use them for evaluation in Chapter 3.

1.1.3 Fair Allocation of Indivisible Items

The theory of *fair allocation of indivisible items* is quite relevant to expert assignment problems. We use techniques from this literature in Chapter 2.

Aziz et al. [8] present an algorithm which attempts to output a W-satisfying EF1 allocation for constraint(s) W. When W includes a minimum threshold for welfare, their approach is somewhat similar to our GERR and FairSequence algorithms. However, rather than greedily maximizing welfare by letting the locally optimal agent pick, they let any agent pick as long as W can still be achieved. Biswas and Barman [21] present a modification of the round-robin mechanism that assigns a complete EF1 allocation when items are partitioned into categories and agents can receive a limited number of items from each category, but the overall number of items per agent is unlimited. Dror et al. [56] study fair allocation under matroid constraints, but only for identical or binary valuations, less than four agents, or a single uniform matroid constraint. Caragiannis and Rathi [33] show that greedily selecting agents is a 2-approximation for the maximum welfare picking sequence when agents can choose at most one item.

Weighted envy-freeness up to one item, or WEF1, is the main fairness notion considered in Chapter 2. It was one of the first fairness notions studied for agents with unequal entitlements [16]. Much recent work in fair allocation has focused on this and other guarantees to agents with unequal entitlements [11, 37, 114, 161, 161, 162]. Much of this work focuses on the setting of binary valuations or binary submodular valuations. Of particular relevance is recent work recommending usage of picking sequences for weighted fair division [36].

Our application of submodular optimization to optimizing orders for round-robin is inspired by previous work on fair allocation with submodular *valuations* [10, 12, 16, 174]. Prior work has also studied maximization of approximately submodular functions, though none has combined matroid constraints with a definition of approximate submodularity similar to ours [49, 69].

Existing work shows the hardness of maximizing welfare for EF1 and picking sequence allocations. Aziz et al. [8] show maximizing welfare subject to EF1 is NP-hard, and Barman et al. [13] show the same problem is not even polynomial-time approximable. Aziz et al. [7] show that the problem of determining if a given welfare is possible under a picking sequence of a certain class is NP-complete for some classes of picking sequences (but not round-robin).

Finally, our `FairSequence` algorithm uses a subroutine inspired by the *exchange graph* routine from the Yankee Swap algorithm [45, 46, 173, 174].

1.1.4 Robust and Stochastic Optimization

In Chapter 4, we study robust and stochastic optimization of various welfare functions for expert assignment problems.

Gorissen et al. [70] provide an excellent overview of optimization under uncertainty, including techniques used in this work, while Ben-Tal et al. [15], Bertsimas et al. [18] offer additional background on robust optimization. A standard approach in this regime is analyzing the dual of the uncertainty, we take this approach frequently in Chapter 4. Stochastic optimization has a wide literature; the books by Birge and Louveaux [20], Levy et al. [99], Prékopa [131], Ruszczyński and Shapiro [142] present wide-ranging introductions to the topic. One of the primary concepts in stochastic optimization is conditional value at risk (CVaR), which can often be approximately optimized by sampling and optimizing over an objective composing the different samples [95, 105, 140]. We take this approach in Chapter 4 as well.

Chapter 4 introduces a framework to solve expert assignment problems. We discuss a naive projected sub-gradient ascent method which solves a quadratic program over a large number of iterations, making it somewhat inefficient. This algorithm is based on an iterative supergradient-ascent approach; similar techniques have been applied to supervised learning with unknown labels [110] and fair learning with unknown group identities [55]. Later in Chapter 4 we demonstrate the inefficiency of this method and present more efficient algorithms, though the method applies to more general settings than the more efficient algorithms. Fair machine learning algorithms [43, 55, 57, 116, 183] often employ similar adversarial optimization techniques over an uncertainty set in a machine learning context.

Some existing work applies stochastic or robust optimization to fair division problems. A line of work studies the minimax regret objective in combinatorial optimization problems, such as constrained resource allocation [5, 22, 23, 93]. This work does not explicitly consider multi-matching problems like the expert assignment problem, nor does it address the robust egalitarian welfare problem. Pujol et al. [133] study fair division problems with parameters noised for differential privacy, showing that the noise can cause unfair allocations; they propose a Monte Carlo approach to mitigate the unfairness with high probability. Peters et al. [129] study envy-free rent division under probabilistic uncertainty. A central mechanism divides rooms and sets room prices to minimize envy. We study a setting without money, both utilitarian and egalitarian objectives, and robust optimization in addition to stochastic optimization. Other fair allocation research has studied the case where agent demand or item availability are uncertain but preferences are known [3, 29, 54, 71]. In our case demand and availability are known but preferences are not. Devic et al. [52] consider fair two-sided matching where the fairness constraint is defined with respect to unknown parameters; we assume knowledge of the parameters that define the fairness constraint (i.e., group identities).

1.2 Preliminaries

We first define the notation that will be used throughout the paper. We will use non-bold capital letters to denote sets, bold capital letters to denote matrices, and bold lowercase letters to denote vectors. For any two sets X and Y, let Y^X denote the set of all functions $f\colon X \mapsto Y$. Given a matrix represented by a bold uppercase letter, e.g., $\mathbf{X} \in \mathbb{R}^{n\times m}$, we will use the convention that the same letter in bold lowercase, e.g., $\boldsymbol{x} \in \mathbb{R}^{nm}$, denotes the row-major vectorization of the matrix such that $\mathbf{X}_{i,j} = \boldsymbol{x}_{im+j}$. Given a matrix $\mathbf{X} \in \mathbb{R}^{n\times m}$ and a set $S \subseteq \{1, \ldots, n\}$, let $\mathbf{X}|_S \in \mathbb{R}^{|S|\times m}$ denote the matrix derived by taking the rows corresponding to S from $\mathbf{X}$ in sorted order, and let $\boldsymbol{x}|_S \in \mathbb{R}^{|S|m}$ denote the row-major vectorization of $\mathbf{X}|_S$.

Example 1.2.1. Given the set $S = \{1, 3\}$, and the matrix $\mathbf{X}$ and its vectorization $\boldsymbol{x}$

$$\mathbf{X} = \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix} \Rightarrow \boldsymbol{x} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \end{pmatrix},$$

we have that

$$\mathbf{X}|_S = \begin{pmatrix} 1 & 2 \\ 5 & 6 \end{pmatrix} \Rightarrow \boldsymbol{x}|_S = \begin{pmatrix} 1 & 2 & 5 & 6 \end{pmatrix}.$$

To simplify notation, given a set X and an element y, we often write $X + y$ and $X - y$ instead of $X \cup \{y\}$ and $X \setminus \{y\}$. Given vectors $\boldsymbol{x}, \boldsymbol{y} \subseteq \mathbb{R}^n$ and real number $c \in \mathbb{R}$, let $\boldsymbol{x} \succeq c$ denote that $\boldsymbol{x}_j \geq c$ for all $j \in \{1, \ldots, n\}$, and let $\boldsymbol{x} \succeq \boldsymbol{y}$ denote that $\boldsymbol{x} - \boldsymbol{y} \succeq 0$. Given matrices $\mathbf{X}, \mathbf{Y} \subseteq \mathbb{R}^{n \times m}$ and real number $c \in \mathbb{R}$, let $\mathbf{X} \succeq c$ denote that $\mathbf{X}_{i,j} \geq c$ for all $i \in \{1, \ldots, n\}$ and $j \in \{1, \ldots, m\}$, and let $\mathbf{X} \succeq \mathbf{Y}$ denote that $\mathbf{X} - \mathbf{Y} \succeq 0$. The $\preceq$ operator is defined analogously. We will also use $\mathbb{I}^{(n \times m)}(i, g) \in \{0, 1\}^{n \times m}$ to denote the indicator matrix with $\mathbb{I}^{(n \times m)}(i, g)_{i,g} = 1$ and $\mathbb{I}^{(n \times m)}(i, g)_{i',g'} = 0$ if $i' \neq i$ or $g' \neq g$. We will omit the $(n \times m)$ and just write $\mathbb{I}(i, g)$ when n and m are obvious. For matrices $\mathbf{X}, \mathbf{Y} \in \mathbb{R}^{n \times m}$, let $\|\mathbf{X}\|_{\mathrm{F}} \doteq \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{m} \mathbf{X}_{i,j}^2}$ denote the *Frobenius norm* of $\mathbf{X}$ and $\langle \mathbf{X}, \mathbf{Y} \rangle_{\mathrm{F}} \doteq \sum_{i=1}^{n} \sum_{j=1}^{m} \mathbf{X}_{i,j} \mathbf{Y}_{i,j}$ denote the *Frobenius matrix product* of $\mathbf{X}$ and $\mathbf{Y}$. Let $\mathbf{X} \odot \mathbf{Y}$ denote the *Hadamard product* of $\mathbf{X}$ and $\mathbf{Y}$, where $(\mathbf{X} \odot \mathbf{Y})_{i,j} = \mathbf{X}_{i,j} \mathbf{Y}_{i,j}$.

1.2.1 Expert Assignment Problem

We assign *experts* $M = \{g_1, \ldots g_m\}$ to *requests* $N = \{1, \ldots, n\}$, optimizing welfare and fairness measures for the requests. We assume that N and M come from the *universes* $\mathcal{N}$ and

$\mathcal{M}$, such that $N \subseteq \mathcal{N}$ and $M \subseteq \mathcal{M}$. Some of the theoretical guarantees in this dissertation require us to model probability distributions over $\mathcal{N}$, $\mathcal{M}$, or the joint space $\mathcal{N} \times \mathcal{M}$.

Let M^n denote the set of all ordered lists of subsets of M. Given an *allocation* $A = (A_1, A_2, \dots A_n) \in M^n$, each $A_i \subseteq M$ is the *bundle* of experts assigned to request i. Not all assignments are allowed; for instance, in reviewer assignment we typically forbid reviewers from reviewing their own papers. We therefore often enforce constraints through some *constraint set* $\mathcal{Z} \subseteq M^n$; we require $A \in \mathcal{Z}$. Typical constraints include lower and upper limits for the number of experts per request and the number of requests per expert, as well as individual constraints limiting the assignment of specific pairs. Formally, define vectors $\underline{\mathbf{k}}^N, \overline{\mathbf{k}}^N \in \mathbb{N}^n$ and $\underline{\mathbf{k}}^M, \overline{\mathbf{k}}^M \in \mathbb{N}^m$ and a *constraint matrix* $\mathbf{C} \in \mathbb{N}^{n \times m}$. Typically, $\mathbf{C}_{i,j} \le 1$ for all $i \in N$ and $j \in \{1, \dots m\}$, and $\mathbf{C}_{i,j} = 0$ when there is a conflict of interest between i and j. Under these basic constraints, the space of all allocations is defined as $\mathcal{Z} \doteq \{A \in M^n \mid \forall i; \underline{\mathbf{k}}^N_i \le |A_i| \le \overline{\mathbf{k}}^N_i, \forall j; \underline{\mathbf{k}}^M_j \le \sum_{i=1}^n |A_i \cap \{g_j\}| \le \overline{\mathbf{k}}^M_j, \forall i, j; |A_i \cap \{g_j\}| \le \mathbf{C}_{i,j}\}$. Other constraints may sometimes be incorporated into $\mathcal{Z}$. For example, we may use pairwise constraints on certain pairs of experts g, g' requiring $\forall i; |A_i \cap \{g\}| + |A_i \cap \{g'\}| \le 1$; in reviewer assignment, these pairwise constraints can forbid assignment of reviewers who bid on each others' papers (possible evidence of collusion [80]) or implement geographic and institutional diversity [100]. Each request i has a *valuation function* over sets of experts $v_i \colon 2^M \mapsto \mathbb{R}$, which defines the suitability of the set of experts for the request. For simplicity, we will typically write $v_i(g)$ instead of $v_i(\{g\})$ when referring to the valuation function applied to a singleton set. We assume valuations are *additive*, i.e., for any set $X \subseteq M$, $v_i(X) = \sum_{g \in X} v_i(g)$.

An allocation A can also be represented as a matrix $\mathbf{A} \in \{0,1\}^{n \times m}$, where $\mathbf{A}_{i,j} = 1$ if and only if $g_j \in A_i$. We will use $\mathbf{A}_i$ to denote the i^{th} row of $\mathbf{A}$. This formalization is isomorphic to the set-based formalization given earlier. Because valuations are additive, we can write the valuations using a valuation matrix $\mathbf{V} \in \mathbb{R}^{n \times m}$, where $\mathbf{V}_{i,j} = v_i(g_j)$.

For deterministic allocations, the set-based notation and the matrix-based notation represent structures that have natural isomorphisms. However, the matrix notation allows us to more easily represent *fractional allocations*. Using the matrix notation, we also define the space of all fractional allocations $\tilde{\mathcal{Z}} \subseteq [0,1]^{n \times m}$. Given an allocation $\tilde{\mathbf{A}} \in \tilde{\mathcal{Z}}$, $\tilde{\mathbf{A}}_{i,j}$ indicates the probability of assigning expert g_j to request i. $\tilde{\mathcal{Z}}$ is defined by the same constraints as $\mathcal{Z}$.

The theory and algorithms of Chapter 2 are more naturally expressed using the set-based notation, while Chapters 3 and 4 are more naturally expressed using matrix notation. Despite the change in notation, we refer to the same underlying objects throughout the thesis.

Example 1.2.2. Consider an expert assignment instance with requests $N = \{1, 2\}$, and $M = \{g_1, g_2, g_3\}$. Suppose the requests and experts have lower and upper limits

$$\underline{\mathbf{k}}^N = \overline{\mathbf{k}}^N = \begin{pmatrix} 2 & 1 \end{pmatrix} \qquad \underline{\mathbf{k}}^M = \overline{\mathbf{k}}^M = \begin{pmatrix} 1 & 1 & 1 \end{pmatrix}.$$

Consider allocation $\mathbf{A}$ and valuation $\mathbf{V}$ as

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix} \qquad \mathbf{V} = \begin{pmatrix} 0 & 2 & 3 \\ 0 & 1 & 2 \end{pmatrix}.$$

Then we have $\mathbf{A} \in \mathcal{Z}$, $A_1 = \{g_2, g_3\}$, and $v_1(A_1) = 5$.

1.2.2 Models of Valuations

In the previous section, we assume valuations are known with certainty. Although this is a simplifying assumption commonly made in the literature (and we make it in Chapter 2), much of this thesis concerns settings where valuations are estimated with some uncertainty. In Chapters 3 and 4 we consider both probabilistic and robust models of $\mathbf{V}$. Under the *probabilistic model*, we assume we have access to a *probability distribution* $\mathcal{D}_{\mathbf{V}}$ over the random variable $\mathbf{V} \in \mathbb{R}^{n \times m}$. In the *robust model*, we instead assume access to an (ε, δ) *uncertainty set* $\mathcal{V}$, defined as follows:

Definition 1.2.3 ((ε, δ) Uncertainty Set)**.** Suppose $\mathbf{V}^* \in \mathbb{R}^{n \times m}$ is the true valuation matrix. An (ε, δ) uncertainty set $\mathcal{V}$ obeys $\Pr\left(\inf_{\mathbf{V} \in \mathcal{V}} \|\mathbf{V} - \mathbf{V}^*\|_1 > \varepsilon\right) < \delta$, i.e., it probably contains some $\mathbf{V}$ that is ε-close to $\mathbf{V}^*$.

These two models are sometimes, but not always, connected. We will further explore these in Chapter 4, but for now consider a simple example for the sake of illustration.

Example 1.2.4. Consider an expert assignment instance with n requests and m experts. Suppose we have a subset of valuations already labeled, and estimate the remaining valuations using Gaussian process matrix factorization [97]. This approach could be used in reviewer assignment, where reviewers often bid on a subset of papers [60, 112]. This model outputs a mean vector $\boldsymbol{\mu} \in \mathbb{R}^{nm}$ and covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{nm \times nm}$. If we assume the model generalizes perfectly, and there is no model misspecification, then we can take $\text{Normal}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ to be our probability distribution $\mathcal{D}_{\mathbf{V}}$. In addition,

$$\mathcal{V} = \{\boldsymbol{v} \mid (\boldsymbol{v} - \boldsymbol{\mu})^\intercal \boldsymbol{\Sigma}^{-1} (\boldsymbol{v} - \boldsymbol{\mu}) \leq \chi^2_{mn}(1 - \delta)\}$$

is a $(0, \delta)$ uncertainty set.

1.2.3 Allocation Desiderata

The *utilitarian social welfare* (also called utilitarian welfare or USW) of an allocation $\mathcal{Z}$ under valuation $\mathbf{V}$ is the average of the requests' valuations under that allocation.

$$\text{USW}(\mathbf{A}, \mathbf{V}) = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{m} \mathbf{V}_{i,j} \mathbf{A}_{i,j} = \frac{1}{n} \sum_{i=1}^{n} v_i(A_i).$$

USW is a natural objective in the context of reviewer assignment, and has been used in many prior works on this topic [38, 42, 44, 91, 124, 157]. Optimizing for USW alone can lead to unfair outcomes. This unfairness can be mitigated by applying fairness constraints or by modifying the objective.

In Chapter 2 we discuss a fairness constraint based on the notion of *envy-freeness*. An allocation A is considered envy-free if for all pairs of requests i and i', $v_i(A_i) \geq v_i(A_{i'})$ This criterion is not achievable in general (consider the example of two requests and one expert g_j whose upper bound is $\overline{\mathbf{k}}_j^M = 1$), so we relax the criterion. An allocation A is *envy-free up to one item* (EF1) if for all pairs of requests i and i', either $v_i(A_i) \geq v_i(A_{i'})$ or there is some $g_{j'}$ such that $g_{j'} \in A_{i'}$ and $v_i(A_i) \geq v_i(A_{i'} - g_{j'})$. EF1 should only be used when all requests have the same upper bounds, $\overline{\mathbf{k}}_i^N = \overline{\mathbf{k}}_{i'}^N$ for all $i, i' \in N$. When requests have distinct upper bounds, we apply the fairness notion of *weighted envy-freeness* [35]. An allocation A is considered weighted envy-free if for all pairs of requests i and i', $v_i(A_i)/\overline{\mathbf{k}}_i^N \geq v_i(A_{i'})/\overline{\mathbf{k}}_{i'}^N$. Analogously, an allocation A is *weighted envy-free up to one item* (WEF1) if for all pairs of requests i and i', either $v_i(A_i)/\overline{\mathbf{k}}_i^N \geq v_i(A_{i'})/\overline{\mathbf{k}}_{i'}^N$, or there is an expert $g_{j'}$ such that $g_{j'} \in A_{i'}$ and $v_i(A_i)/\overline{\mathbf{k}}_i^N \geq v_i(A_{i'} - g_{j'})/\overline{\mathbf{k}}_{i'}^N$.

We may also consider the sum of envy as a metric for evaluation, defined by

$$\sum_{i \in N} \sum_{i' \in N} \max\{(v_i(A_{i'}) - v_i(A_i)), 0\}.$$

Example 1.2.5. Consider an expert assignment instance with requests $N = \{1, 2\}$ and experts $M = \{g_1, g_2, g_3, g_4\}$, where $\overline{\mathbf{k}}_1^N = 2$ and $\overline{\mathbf{k}}_2^N = 3$. Given the allocation $\mathbf{A}$ and valuation $\mathbf{V}$ as

$$\mathbf{A} = \begin{pmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \end{pmatrix} \qquad \mathbf{V} = \begin{pmatrix} 5 & 4 & 8 & 8 \\ 9 & 6 & 1 & 8 \end{pmatrix},$$

$v_1(\mathbf{A}_1) = 9, v_1(\mathbf{A}_2) = 21, v_2(\mathbf{A}_2) = 18$, and $v_2(\mathbf{A}_1) = 15$. The sum of envy is 12. As $v_1(\mathbf{A}_1) < v_1(\mathbf{A}_2)$, request 1 envies request 2. This envy cannot be alleviated by removing one item from $\mathbf{A}_2$, so $\mathbf{A}$ does not satisfy EF1. However, since $v_1(A_1)/2 > v_1(A_2 - g_3)/3$, $\mathbf{A}$ satisfies the WEF1 criterion.

Another way to treat fairness is to modify the objective function. The canonical fair objective function is the *egalitarian welfare* or ESW,

$$\mathrm{ESW}(\mathbf{A}, \mathbf{V}) = \min_{i \in N} \sum_{j=1}^{m} \mathbf{V}_{i,j} \mathbf{A}_{i,j} = \min_{i \in N} v_i(A_i).$$

This objective has some history in reviewer assignment; algorithms approximately optimizing egalitarian welfare have been implemented in OpenReview [91], and used for large conferences like ICML [157].

In settings where we cannot access the exact valuation scores $\mathbf{V}$ but instead have a probability distribution $\mathcal{D}_{\mathbf{V}}$ or an uncertainty set $\mathcal{V}$, ESW is too difficult to achieve. In Chapter 4, we instead consider fairness through the lens of *group* egalitarian welfare. Given a partition of the agents into groups $\mathcal{G} = \{G_1, \ldots, G_\kappa\}$, the *group egalitarian welfare* or GESW is defined as

$$\mathrm{GESW}(\mathbf{A}, \mathbf{V}) = \min_{G \in \mathcal{G}} \mathrm{USW}(\mathbf{A}|_G, \mathbf{V}|_G).$$

Another commonly-studied objective function is the *Gini coefficient*, which computes the normalized mean absolute difference in outcomes for all pairs of requests [66]. In expert assignment, the Gini coefficient is computed as

$$\mathrm{Gini}(\mathbf{A},\mathbf{V}) = \big(2n^2\,\mathrm{USW}(\mathbf{A},\mathbf{V})\big)^{-1}\sum_{i\in N}\sum_{i'\in N}|v_i(A_i) - v_{i'}(A_{i'})|\,.$$

The Gini coefficient is used to capture overall inequality. Although we will not optimize for it, we will use it as an additional metric in some of our experiments.

Example 1.2.6. Consider an expert assignment instance with four requests and four experts. Suppose the groups are $G_1 = \{1,2\}$ and $G_2 = \{3,4\}$. If we select an assignment A with $v_1(A_1) = 2, v_2(A_2) = 6, v_3(A_3) = 9, v_4(A_4) = 7$, then we obtain the following welfare metrics:

1. The utilitarian social welfare is $\mathrm{USW}(\mathbf{A},\mathbf{V}) = \frac{1}{4}\sum_{i=1}^{4} v_i(A_i) = 6$.
2. The egalitarian social welfare is the utility of the least well-off request, which is 2.
3. The utilitarian social welfare of group 1 is $\mathrm{USW}(\mathbf{A}|_{G_1}, \mathbf{V}|_{G_1}) = 4$.
4. The utilitarian social welfare of group 2 is $\mathrm{USW}(\mathbf{A}|_{G_2}, \mathbf{V}|_{G_2}) = 8$.
5. The group egalitarian social welfare, or $\mathrm{GESW}(\mathbf{A},\mathbf{V})$, is the utilitarian social welfare of the least well-off group, which is 4.
6. The Gini coefficient is $\mathrm{Gini}(\mathbf{A},\mathbf{V}) = \frac{2}{2\cdot 16\cdot 6}(4+7+5+3+1+2) = \frac{11}{48}$.

In all tables in this thesis, we will use the up and down arrows (↑, ↓) to indicate when higher values or lower values of a metric are preferable, *ceteris paribus*. For example, USW will be denoted as USW(↑), and Gini will be denoted as Gini(↓).

CHAPTER 2

FAST AND FAIR EXPERT ASSIGNMENT WITH FAIRSEQUENCE

In this chapter, we consider the expert assignment problem through the lens of *fair allocation.* We present fast, fair, flexible, and welfare efficient algorithms for assigning experts to requests. Our approaches extend picking sequence mechanisms, standard tools from the fair allocation literature to ensure (weighted) envy-freeness up to one item, or (W)EF1. These mechanisms impose a picking order on the requests, and then assign experts to requests in that imposed order. The order is chosen to satisfy EF1 or WEF1. However, fairness can come at the cost of decreased welfare. To overcome this challenge, we carefully select approximately optimal picking sequence orders. Applying the notion of γ-weak submodularity, we show our Greedy Expert Round Robin (`GERR`) approach is EF1 and yields a $(1+\gamma)$-approximation to the maximum welfare attainable by a round-robin picking sequence mechanism under any order. We present a weighted picking sequence mechanism called `FairSequence` that targets the WEF1 criterion to offer fairness in a more general setting. Using data from three conferences, we show that `FairSequence` runs an order of magnitude faster and provides approximate envy-freeness guarantees that are violated by existing reviewer assignment approaches. `FairSequence` is available in the OpenReview conference management platform [119], giving conference organizers access to faster reviewer assignment with high welfare and envy-freeness guarantees.

2.1 Achieving (W)EF1 with High Welfare Picking Sequences

Maximizing welfare subject to EF1 is NP-hard and is not approximable in polynomial time [13]. Thus, we explore methods that produce EF1 allocations with good empirical performance and partial welfare guarantees using *picking sequence mechanisms.* In standard fair allocation settings, the well-known *round-robin* (`RR`) mechanism produces EF1 allocations by setting an order of agents, and letting them select one item at a time. Due to the constraints of the expert assignment problem, round-robin is not EF1 for expert assignment. We thus present a variation on classic `RR`, which we term *Expert Round Robin* (`ERR`).

Round-robin mechanisms assign the same number of experts to each request, but sometimes requests require variable-sized assignments. For instance, two-phase conference reviewing processes often need to assign a variable numbers of reviewers in the second phase when some reviewers failed to respond in the first phase. We therefore also study a family of weighted picking sequences which satisfy the Weighted EF1 constraint (WEF1) [35]. Our algorithms, Weighted Expert Picking (`WEP`) and `FairSequence`, follow `ERR` by applying a standard weighted picking sequence with some added steps to accommodate the generalized constraints of expert assignment.

While picking sequence mechanisms are known to satisfy fairness constraints, their welfare guarantees are highly dependent on the order in which players pick items. For example, consider a stylized setting where there are two requests (i and j) and two experts (g_1 and g_2): request i has an affinity score of 5 for both experts, while request j has a score of 10 for g_1 and 0 for g_2. A round-robin mechanism that assigns to i first might assign g_1 to i, leaving j with g_2. Assigning to j first results in a much better outcome, without compromising on fairness. It is generally difficult to identify optimal picking sequences [6, 7, 24, 84]. We ask the question: Can we identify *approximately optimal* orders?

2.1.1 Chapter Overview

We run a combinatorial search for orders of requests that yield high efficiency allocations for picking-sequence mechanisms like ERR and WEP. To this end, we examine the problem of finding an optimal assignment order via the lens of *submodular optimization.* We optimize a function on partial picking sequences, which varies according to the welfare of the allocation resulting from the picking sequence. This function is not submodular in general, but we can capture its distance from submodularity via a variable γ. Our main theoretical result (Theorem 2.4.5), which is of independent interest to the fair division community, shows that a simple greedy approach maximizes this function up to a factor of $1+\gamma$. We call this approach Greedy Expert Round Robin (GERR). The approach can also be applied to weighted picking sequences like WEP, by optimizing over the order in which ties in priority are broken.

Though we do not offer theoretical welfare guarantees for the FairSequence algorithm, we present a heuristic approach that optimizes for a high welfare weighted picking sequence. FairSequence breaks ties in priority order adaptively when they occur, by assigning an expert to any request with top priority which can receive the highest increase in welfare. This approach is fast and straightforward to implement. FairSequence thus achieves our four goals: it is a fast, fair, flexible, and welfare efficient expert assignment mechanism.

We compare our GERR and FairSequence algorithms with three state-of-the-art reviewer assignment frameworks on three computer science conference reviewer-assignment datasets. Not only are GERR and FairSequence the only provably (W)EF1 approaches, FairSequence is an order of magnitude faster than the other methods on all datasets. Most importantly, we show that FairSequence is often significantly more fair than FairFlow, a fair allocation protocol [91], on many fairness metrics.

2.1.2 Additional Preliminaries

Orders. Picking sequence mechanisms assign an order to the requests, and allocate experts to requests in that order. We define an *order* on requests $i \in N$ as a tuple $O = (S, o)$, where $S \subseteq N$ is the set of papers in the order and $o \colon S \mapsto \{1, 2, \ldots, |S|\}$ is a permutation on S mapping papers to positions. Let $\Psi(N)$ denote the set of all orders over subsets of N. We will also refer to $\Psi(M)$, the set of all orders over subsets of M, defined analogously to $\Psi(N)$. We slightly abuse notation and say that a request $i \in O$ if $i \in S$. For any $i, i' \in O$, we say that $i \prec_O i'$ if and only if $o(i) < o(i')$. We write $i \prec i'$ when O is clear from context. We can think of an order $O = (S, o)$ as an ordered list $[o_1, o_2, \ldots o_{|S|}]$ such that $o_l = o^{-1}(l)$ for all positions l. We use the notation $O + i$ to indicate the order (S', o') that appends i to the end of O. Formally, $S' = S + i$, $o'(i') = o(i')$ for $i' \in S$, and $o'(i) = |S'|$. We write the empty order as $O_\emptyset = (\emptyset, \emptyset)$.

Example 2.1.1. Suppose we have a set of requests $N = \{1, 2, 3, 4\}$, and order $O = (S, o)$ with $S = \{1, 3\}$, $o(3) = 1$, and $o(1) = 2$. O can also be written as $[3, 1]$. The order $O + 4$ is represented using S' and function o' with $S' = \{1, 3, 4\}$, $o'(3) = 1$, $o'(1) = 2$, and $o'(4) = 3$. We can also write it as $[3, 1, 4]$.

Matroids. A *matroid* [122] is a pair (E, I) with *ground set* E and *independent sets* I, which must satisfy $\emptyset \in I$. Independent sets must satisfy the *inclusion property*: $\forall X \subseteq Y \in I$, $X \in I$, and the *exchange property*: $\forall X, Y \in I$ with $|X| < |Y|$, $\exists e \in Y \setminus X$ such that $X \cup \{e\} \in I$. A *partition matroid* is defined using *categories* $X_1, X_2, \ldots X_b$ such that $X_i \cap X_j = \emptyset$ for all i, j and $\bigcup_{1 \leq i \leq b} X_i = E$, and *capacities* $d_1, d_2, \ldots d_b$; the independent sets are $I = \{Y \subseteq E \mid \forall i, |Y \cap X_i| \leq d_i\}$. Given two matroids over the same ground set (E, I_1) and (E, I_2), the *intersection* of the two matroids is the pair $(E, \{Y \mid Y \in (I_1 \cap I_2)\})$. The intersection of two matroids may not be a matroid [122].

Example 2.1.2. Consider the set $S = \{1, 2, 3\}$. Define the ground set as $E = S \times S$. Define categories X_1, X_2, X_3 such that $X_i = \{(i, 1), (i, 2), (i, 3)\}$, and define capacities $d_i = 1$ for $i \in \{1, 2, 3\}$. These sets and capacities define the independent sets of a partition matroid over E. Any set that duplicates the first element of a tuple, such as $\{(1, 1), (1, 2)\}$ is not an independent set, while any set is independent if it contains tuples with unique first elements, such as $\{(1, 1), (3, 2), (2, 2)\}$. This matroid satisfies the inclusion property, as we cannot introduce a duplicate first element by deleting a tuple from a set. It also satisfies the exchange property; for any pair of sets with different sizes, since both sets have tuples with unique first elements, the larger set must contain a tuple (i, j) such that no tuple in the smaller set has the form $(i, \cdot)$.

Submodularity. We also use the notion of a *submodular set function*; submodular functions formalize the notion of diminishing marginal gains. Given a ground set E, a set function $f \colon 2^E \mapsto \mathbb{R}$, a set $X \subseteq E$, and an element $e \in (E \setminus X)$, we can write the *marginal gain* of adding e to X under f as $\rho_e^f(X) = f(X + e) - f(X)$ or simply $\rho_e(X)$ if f is understood from context. Given a set E, a function $f \colon 2^E \mapsto \mathbb{R}$ is *submodular* if for all $X \subseteq Y \subseteq E$ and $e \in E \setminus Y$, $\rho_e^f(X) \geq \rho_e^f(Y)$. A set function is *monotonically non-decreasing* if for all $X \subseteq Y \subseteq E$, $f(X) \leq f(Y)$. We define the notion of γ-weak submodularity for monotonically non-decreasing, non-negative functions. Given a monotonically non-decreasing, non-negative function $f \colon 2^E \mapsto \mathbb{R}_{\geq 0}$, we say that f is *γ-weakly submodular* if for all $X \subseteq Y \subseteq E$ and $e \in E \setminus Y$, $\gamma \rho_e^f(X) \geq \rho_e^f(Y)$. When $\gamma = 1$ we recover submodularity, and $\gamma \geq 1$ always.

Example 2.1.3. Given a set E, a *budget additive* submodular function f has a set of *weights* w_e assigned to each element $e \in E$, along with a *budget* B. The value of a subset $S \subseteq E$ is defined as $f(S) \doteq \min\{\sum_{e \in E} w_e, B\}$. This function is monotonically increasing, and for any two sets $X \subseteq Y \subseteq E$ and $e \in E \setminus Y$, $\rho_e(X) = w_e - \min\{f(X) + w_e, B\} \geq w_e - \min\{f(Y) + w_e, B\} = \rho_e(Y)$. Consider the specific set $E = \{a, b, c, d\}$, with $w_a = w_b = w_c = w_d = 1$, and budget $B = 2$. We can define a budget additive function f_{BA} using these parameters, as well as a modified function f_{MOD} such that $f_{\text{MOD}}(E) \doteq f_{\text{BA}}(E)|E|$. Notice that $f_{\text{MOD}}(\{a\}) = 1$, $f_{\text{MOD}}(\{a, b\}) = 4$, $f_{\text{MOD}}(\{a, b, c\}) = 6$, $f_{\text{MOD}}(\{a, b, c, d\}) = 8$, and all sets of the same size have the same value. So $\rho_b(\{a\}) = 3$ is the maximum marginal gain for a single element b, but $\rho_b(\emptyset) = 1$. Thus f_{MOD} is γ-weakly submodular with $\gamma = 3$.

Scale Invariance. Different methods of computing expert assignment affinity scores may yield scores on different scales, even when the relative ordering of expert-request pairs remains consistent. Therefore, we analyze the invariance of EF1 and WEF1 under affine scaling of affinity scores. Both criteria are invariant to any affine shift with non-negative coefficients.

Definition 2.1.4. Given any valuation function $v : 2^M \to \mathbb{R}$, the valuation function $v' : 2^M \to \mathbb{R}$ is a *non-negative affine transformation* of v if $v'(g) = av(g) + b$ for all $g \in M$ and $a, b \in \mathbb{R}_{\geq 0}$.

It is natural to assume that $a \geq 0$, since experts that provide positive utility under v should not be estimated to provide negative utility (and vice-versa) under any reasonable affine shift. If all affinities are assumed to be non-negative, the assumption that $b \geq 0$ is reasonable if the lowest affinity score is already set at 0. We can extend this setting to settings where there is some obvious *lowest possible* affinity score. As long as some expert-request pair has the lowest conceivable affinity score, we can assume that $b \geq 0$ for any affine transformation.

Proposition 2.1.5. *Given two requests $i, j \in N$, suppose that $\underline{\mathbf{k}}_i^N = \overline{\mathbf{k}}_i^N = k$ and $\underline{\mathbf{k}}_j^N = \overline{\mathbf{k}}_j^N = l$. Consider the valuation function $v : 2^M \to \mathbb{R}$, and a non-negative affine transformation of v, labelled v'. Given some allocation $A \in \mathcal{Z}$, for any $g \in A_j$, if $v_i(A_i)/k \geq v_i(A_j - g)/(l)$, then*

$v'_i(A_i)/k \geq v'_i(A_j - g)/l$. In other words, the WEF1 criterion is invariant under non-negative affine transformation. This implies that the EF1 criterion is invariant under non-negative affine transformation when all bundles have equal size.

Proof.

$$\begin{aligned} v'_i(A_i)/k - v'_i(A_j - g)/l &= av_i(A_i)/k + b - (av_i(A_j - g)/l + b - b/l)) \\ &= a(v_i(A_i)/k - v_i(A_j - g)/l) + b/l \geq 0. \end{aligned}$$

The implication for EF1 holds by setting $k = l$ and multiplying all (in)equalities by k. □

2.2 Fair and Efficient Assignment with Expert Round Robin

We first show how to obtain EF1 expert assignments when all requests have equal demands ($\underline{\mathbf{k}}_i^N = \overline{\mathbf{k}}_i^N = k$ for some fixed $k \in \mathbb{N}$), and experts do not have lower bounds ($\underline{\mathbf{k}}_g^M = 0$ for all $g \in M$); we handle the more general case in Section 2.3. Our algorithm draws upon the simple and well-known *round-robin* mechanism. Given an ordered list of requests, round-robin proceeds in rounds. In each round, we iterate over the requests in the provided order, assigning each request its highest valued remaining expert. This allocation is EF1 for additive valuations without constraints by a simple argument [34]. For any request i, we divide the assignments into rounds. Request i prefers its own experts to the experts of any request $i' \succ i$, and it prefers its own bundle to that of any request $i' \prec i$ if we ignore the expert assigned to i' in the first round.

The constraint that requests must be assigned k *distinct* experts, $\mathbf{C} \preceq 1$, poses a challenge. A trivial modification of round-robin allows us to satisfy the cardinality constraint — proceed for exactly k rounds, then stop. We might naively update round-robin to satisfy the distinctness constraint as well, by assigning each request the best expert they do not already have. However, this modification can break the EF1 constraint. To see why, suppose a request i is assigned an expert g in one round. In the next round, i may still prefer g over any other expert, but we cannot assign it. We will be forced to assign i a worse expert, giving another request the "second copy" of g. We consider a detailed counterexample below.

Example 2.2.1. Consider an expert assignment instance with 4 requests and 6 experts, $\underline{\mathbf{k}}_i^N = \overline{\mathbf{k}}_i^N = 3$ for all requests i, $\underline{\mathbf{k}}_g^M = 0$ and $\overline{\mathbf{k}}_g^M = 2$ for all experts g, and

$$\mathbf{V} = \begin{pmatrix} 2 & 0 & 0 & 1 & 0.5 & \epsilon \\ 3 & 1 & 2 & 10 & 0 & 0 \\ 0 & \epsilon & 0 & 10 & 1 & 0 \\ 2 & 1 & 3 & 10 & 0 & \epsilon \end{pmatrix}.$$

The naive constrained round-robin, where each request is assigned the best expert they do not already have, can fail to satisfy EF1. If we apply naive constrained round-robin with the requests in increasing numerical order, we obtain the allocation $A_1 = \{g_1, g_5, g_6\}, A_2 = \{g_4, g_1, g_3\}, A_3 = \{g_4, g_5, g_2\}, A_4 = \{g_3, g_2, g_6\}$. However, $v_4(A_2 - g) \geq 5$ for all $g \in A_2$, while $v_4(A_4) = 4 + \epsilon$. In contrast, the allocation $A_1 = \{g_1, g_5, g_6\}, A_2 = \{g_4, g_1, g_2\}, A_3 = \{g_4, g_5, g_3\}, A_4 = \{g_3, g_2, g_6\}$ is EF1.

We present a modification of round-robin that takes an order O and assigns experts to requests in order O such that all constraints are satisfied and the allocation is EF1. Expert Round Robin or ERR (Algorithm 1) forbids any assignment that violates a crucial invariant for proving EF1. This invariant derives from the proof of EF1 in the additive case. Any time we would assign an expert such that EF1 would be violated, we forbid the assignment and instead assign a different expert. EF1 violations can only arise when another request

Algorithm 1 Expert Round Robin (`ERR`)

```
Require: Requests $N$, experts $M$, expert upper limits $\overline{\mathbf{k}}^M$, order $O$, affinity functions $v_i$,
    bundle size limit $k$, constraints $\mathbf{C}$
Initialize allocation $A$ as $A_i \leftarrow \emptyset$ for all requests $i \in O$
Initialize the attempted set $S_i \leftarrow \emptyset$ for all $i$
for Round $t \in \{1, \ldots, k\}$ do
    for $i \in O$ in order do
        for Expert $g$ in decreasing order of $v_i(g)$ (break ties lexicographically) do
            Attempt to assign $g$ to $i$ ($S_i \leftarrow S_i + g$)
            if $\sum_{j \in N} |A_j \cap \{g\}| < \overline{\mathbf{k}}_g^M$ and $\mathbf{A} + \mathbb{I}(i, g) \preceq \mathbf{C}$
                if No $i'$ with $g \in S_{i'}$ envies $A_i + g$ more than one expert
                    $A_i \leftarrow (A_i + g)$
                    Move to the next request in $O$
        If no new expert is assigned to $i$, return $A$
return $A$
```

preferred that expert but the assignment was forbidden, either because it had been assigned already, or because it would have caused an EF1 violation for that request as well. We always attempt to assign experts in preference order. Thus when we attempt to assign an expert g to request i, we only need to check for EF1 violations against other requests to which we have attempted to assign g in the past. Theorem 2.2.2 asserts the correctness of `ERR`.

Theorem 2.2.2. *`ERR` terminates with an EF1 allocation* $\mathbf{A}$ *where requests receive at most* k *distinct experts, no expert* g *is assigned to more than* $\overline{\mathbf{k}}_g^M$ *requests, and* $\mathbf{A} \preceq \mathbf{C}$.

Proof. The algorithm assigns at most one expert to each request in each round for k rounds, so the constraint that all requests receive at most k experts is satisfied. In addition, the algorithm always checks that $g \notin A_i$, the number of requests which already have g is no more than $\overline{\mathbf{k}}_g^M$, and $\mathbf{A} + \mathbb{I}(i, g) \preceq \mathbf{C}$ is satisfied before assigning g to i. Thus no request receives duplicate experts, expert upper bounds are satisfied, and constraints $\mathbf{C}$ are never violated.

We now prove that the returned allocation is EF1. Consider some arbitrary request i'; we show that i' does not envy any other request by more than 1 expert. As in the original round-robin argument, we divide the assignments of experts to requests into rounds $0, 1, \ldots, s$, where $s \leq k$ ($s < k$ only when the algorithm terminates early). Round 0 contains the assignments made during the first iteration of Algorithm 1 to all requests preceding i' in O. Rounds 1 through $s-1$ begin with the assignment of a new expert to i' and end with the assignment of a new expert to the request immediately preceding i' in O, while round s begins with assignment to i' and ends with assignment to some request after i'.

Consider the bundle $A_i^{(t)}$ assigned to some request $i \neq i'$ after the end of some round $t \in \{0, 1, \dots s\}$. We will define modified bundles B_i^t for all i, and prove by induction that $v_{i'}\left(B_i^{(t)}\right) \leq v_{i'}\left(A_{i'}^{(t)}\right)$. For all t, let $B_i^{(t)} = A_i^{(t)}$ if $i' \prec i$. If $i' \succ i$, $B_i^{(t)} = A_i^{(t)} - \arg\max_{g \in A_i^{(t)}} v_{i'}(g)$.

For the base case, we see that after round 0, $|B_i^{(0)}| = 0$ for all i and $|A_{i'}^{(0)}| = 0$, so $v_{i'}\left(A_{i'}^{(0)}\right) = v_{i'}\left(B_i^{(0)}\right) = 0$.

Now suppose that after round $t-1$, we have $v_{i'}\left(A_{i'}^{(t-1)}\right) \geq v_{i'}\left(B_i^{(t-1)}\right)$ for all i. Suppose there is some i such that after round t, $v_{i'}\left(B_i^{(t)}\right) > v_{i'}\left(A_{i'}^{(t)}\right)$. i' is assigned an expert in all rounds except 0, and because affinities are non-negative i must be assigned an expert in round t to obtain $v_{i'}\left(B_i^{(t)}\right) > v_{i'}\left(A_{i'}^{(t)}\right)$. By the inductive hypothesis and the fact that affinities are additive, i' must prefer the expert i was assigned in t, g_i, to the expert i' was assigned in t, $g_{i'}$. Because i' went first in t, this means that we attempted to assign g_i to i' either in t or earlier and i must have checked for envy against i'. This is a contradiction, since $v_{i'}\left(B_i^{(t)}\right) > v_{i'}\left(A_{i'}^{(t)}\right)$ violates EF1. □

It is possible for ERR to return an incomplete allocation when a complete one exists. This is tight in some sense; we can easily show an instance of the reviewer assignment problem where no valid, complete allocation is EF1.

Example 2.2.3. Consider an expert assignment instance with two requests with $\underline{\mathbf{k}}_i^N = \overline{\mathbf{k}}_i^N = k = 2$ for all $i \in N$ and four experts $M = \{g_1, g_2, g_3, g_4\}$ with $\underline{\mathbf{k}}_g^M = 0$ and

$\overline{\mathbf{k}}^M_g = 1$ for all $g \in M$. Suppose that $\mathbf{V}_1 = \begin{pmatrix} 1 & 1 & 0 & 0 \end{pmatrix}$, and $\mathbf{C}_1 = \begin{pmatrix} 0 & 0 & 1 & 1 \end{pmatrix}$. Then the only valid complete allocation is $A = \{\{g_3, g_4\}, \{g_1, g_2\}\}$, which is not EF1.

It is straightforward to show that `ERR` always returns a complete, EF1 allocation when the number of experts is large and there are no additional constraints $\mathbf{C}$.

Proposition 2.2.4. *Given an expert assignment instance with m experts, n requests, $\mathbf{C} = \mathbf{1}$, and $\underline{\mathbf{k}}^N_i = \overline{\mathbf{k}}^N_i = k$ for all $i \in N$, if $m \geq kn$, `ERR` returns a complete, EF1 allocation.*

Proof. Algorithm 1 only refuses to assign an expert g to a request i when g is assigned to too many requests, g has already been assigned to i, or some other request (to which we have previously attempted to assign g) "objects" to the assignment. Thus if we have assigned less than kn distinct experts under Algorithm 1, it must be the case that there is an expert g that we have not considered for any request. Because there are at least kn distinct experts, we can see that during any round of the algorithm, there will be such an unconsidered expert. Thus in any round, a request can always be assigned some expert that has never been considered for any request, and the selection will not be refused. This proves that the allocation returned by Algorithm 1 is complete, and we have EF1 from Theorem 2.2.2. □

2.3 Non-Uniform Demands and Minimum Reviewer Supply

Requests may sometimes require different numbers of experts. As a motivating example, conference organizers often run reviewer assignment multiple times to account for late reviews, borderline papers, and other mitigating circumstances [100]. In addition, conference organizers might wish to require that each reviewer receives a minimum number of papers to review. These reviewer lower bounds ensure more balanced workloads for the reviewers. To satisfy these additional real-world constraints, we introduce variants of `ERR` that allow for variable demands $\overline{\mathbf{k}}^N$ (here, we assume $\underline{\mathbf{k}}^N = \overline{\mathbf{k}}^N$) and expert load lower bounds $\underline{\mathbf{k}}^M$.

Algorithm 2 Weighted Expert Picking (`WEP`)

```
Require: Requests $N$, experts $M$, expert upper limits $\overline{\mathbf{k}}^M$, order $O$, affinity functions $v_i$,
    bundle size limits $\overline{\mathbf{k}}^N$, constraints $\mathbf{C}$
Initialize allocation $A$ as $A_i \leftarrow \emptyset$ for all requests $i \in O$
Initialize the attempted set $S_i \leftarrow \emptyset$ for all $i \in O$
while $\exists i \in N$ such that $|A_i| < \overline{\mathbf{k}}^N_i$ do
    $i^* \leftarrow \arg\min_{i \in N} |A_i|/\overline{\mathbf{k}}^N_i$, breaking ties using $O$
    for Expert $g$ in decreasing order of $v_{i^*}(g)$ (break ties lexicographically) do
        Attempt to assign $g$ to $i^*$ ($S_{i^*} \leftarrow S_{i^*} + g$)
        if $\sum_{i \in N} |A_i \cap \{g\}| < \overline{\mathbf{k}}^M_g$ and $\mathbf{A} + \mathbb{I}(i^*, g) \preceq \mathbf{C}$
            if All $i' \neq i^*$ with $|S_{i^*} \cap S_{i'}| > 0$ satisfy WEF1 with respect to $i^*$
                $A_{i^*} \leftarrow (A_{i^*} + g)$
                Move to the next request in $\arg\min_{i \in N} |A_i|/\overline{\mathbf{k}}^N_i$
    If no new reviewer is assigned to $i^*$, return $A$
return $A$
```

To accommodate variable request demands $\underline{\mathbf{k}}^N$ and $\overline{\mathbf{k}}^N$, we present a weighted analogue of `ERR` called Weighted Expert Picking (`WEP`). We replace round-robin with the picking sequence introduced by Chakraborty et al. [35], which guarantees WEF1. Requests no longer receive assignments in a fixed order; at each iteration the request $i \in N$ which has reached the smallest fraction of its bundle size limit $\overline{\mathbf{k}}^N_i$ is chosen to receive the next expert. We break ties in this "fraction of satisfied demand" criterion by consulting a fixed tie-breaking order O. Algorithm 2 shows the complete approach.

Parallel to Theorem 2.2.2, we state that Algorithm 2 yields a WEF1 allocation satisfying all constraints.

Theorem 2.3.1. *Algorithm 2 returns a WEF1 allocation* $\mathbf{A}$ *where each request* $i \in N$ *receives at most* $\overline{\mathbf{k}}^N_i$ *distinct experts, no expert* g *is assigned to more than* $\overline{\mathbf{k}}^M_g$ *requests, and* $\mathbf{A} \preceq \mathbf{C}$*.*

Proof. As long as there is some request i with $|A_i| < \overline{\mathbf{k}}^N_i$, we will never pick a request i' with $|A_{i'}| = \overline{\mathbf{k}}^N_{i'}$ in the picking sequence, since $|A_i|/\overline{\mathbf{k}}^N_i < 1 = |A_{i'}|/\overline{\mathbf{k}}^N_{i'}$. This proves that when we terminate, no request $i \in N$ has more than $\overline{\mathbf{k}}^N_i$ experts.

In addition, the algorithm always checks that $\sum_{j \in N} |A_j \cap \{g\}| < \overline{\mathbf{k}}^M_g$ and $\mathbf{A} + \mathbb{I}(i,g) \preceq \mathbf{C}$ before assigning g to i. Thus no request receives duplicate experts, expert upper bounds are satisfied, and $\mathbf{A} \preceq \mathbf{C}$.

We now show that the allocation is WEF1, by showing that after a request $i \in N$ is assigned an expert $g \in M$, all requests are WEF1 with respect to the request i. Assuming $\overline{\mathbf{k}}^N_i \geq 1$ for all $i \in N$, the first n iterations of the algorithm assign each request a single expert, in order of $\mathcal{O}$. The allocation is clearly WEF1 at each of those iterations. Suppose that now request i has been assigned an expert g, after all requests have at least one expert. For any request $i' \in N$ which we have attempted to give an expert that we have also attempted to give to request i ($|S_i \cap S_{i'}| > 0$), we have already checked that request i' does not have weighted envy for request i over one expert.

It remains to analyze the case when $|S_i \cap S_{i'}| = 0$. We have not attempted to give request i' any experts that we have also attempted to give to request i, including expert g. Thus, for any expert $g_{i'}$ given to request i', $v_{i'}(g_{i'}) > v_{i'}(g_i)$ for any g_i that i was given after i' was given $g_{i'}$. This criterion is sufficient for the proof from [35] to go through. □

2.3.1 Generalized Expert Picking Sequences

Both ERR and WEP can be viewed as instantiations of a broader meta-algorithm, depicted in Algorithm 3. This algorithm applies a request selection criterion θ^N, and an expert selection criterion θ^M. The request selection criterion $\theta^N(A, \underline{\mathbf{k}}^N, \overline{\mathbf{k}}^N, \underline{\mathbf{k}}^M, \overline{\mathbf{k}}^M) \in \Psi(N)$ computes, given a partial allocation A, an ordered set of requests that can be assigned an expert. θ^N may also take as an argument a fixed order over requests, O. If it does, then the request selection function will simply select the next request in the order O. Given a partial allocation A and a request $i \in N$, the expert selection criterion $\theta^M(i, A, \mathbf{C}, \underline{\mathbf{k}}^N, \overline{\mathbf{k}}^N, \underline{\mathbf{k}}^M, \overline{\mathbf{k}}^M) \in \Psi(M)$ computes an ordered set of experts that request i can be assigned. Since most of the arguments of these criteria are clear from context, we will write simply $\theta^N(A(, O))$ and $\theta^M(A, i)$.

Algorithm 3 Picking Sequence Expert Assignment

```
Require: Requests $N$, experts $M$, request upper limits $\overline{\mathbf{k}}^N$, expert upper limits $\overline{\mathbf{k}}^M$, affinity functions $v_i$, constraints $\mathbf{C}$, request selection criterion $\theta^N$, expert selection criterion $\theta^M$
Initialize allocation $A$ as $A_i \leftarrow \emptyset$ for all requests $i \in N$
while $A$ is not complete do
    if $\theta^N(A) \neq O_\emptyset$ and $\theta^M(A, i) \neq O_\emptyset$ for some $i \in \theta^N(A)$
        $i \leftarrow$ first request in $\theta^N(A)$ with $\theta^M(A, i) \neq O_\emptyset$
        $g \leftarrow$ first expert in $\theta^M(A, i)$
        $A_i \leftarrow A_i + g$
    else
        return $A$
return $A$
```

Concretely, `ERR` can be implemented by requiring $\theta^N(A, O)$ to select the singleton set containing the next request in the order O, and requiring $\theta^M(A, i)$ to select experts $g \in M$ such that $\sum_{j\in N}|A_j \cap \{g\}| < \overline{\mathbf{k}}_g^M$, $\mathbf{A} + \mathbb{I}(i, g) \preceq \mathbf{C}$, and no other request i' would envy i more than one expert after adding g to A_i. Naturally, $\theta^M(A, i)$ is ordered in decreasing order of $v_i(g)$. `WEP` is implemented with $\theta^N(A) = \arg\min_{i\in N} |A_i|/\overline{\mathbf{k}}_i^N$, ordered by O, and $\theta^M(A, i)$ defined analogously to before. $\theta^M(A, i)$ now requires WEF1 to be satisfied, rather than EF1.

2.3.2 Minimum Expert Supply

We can also easily introduce minimum expert supply constraints. We will utilize a simple trick to satisfy these constraints, which can be applied to any instantiation of Algorithm 3. When the remaining demand equals the number of assignments required to meet expert minima $\underline{\mathbf{k}}^M$, we restrict the available set of experts to those who need to be assigned to meet minimum requirements. More formally, when we have

$$\sum_{g\in M} \max\left(\underline{\mathbf{k}}_g^M - \sum_{i\in N}|A_i \cap \{g\}|, 0\right) = \sum_{i\in N} \overline{\mathbf{k}}_i^N - |A_i|,$$

Algorithm 4 Picking Sequence Expert Assignment with Expert Lower Bounds

```
Require: Requests $N$, experts $M$, request bounds $\underline{\mathbf{k}}^N, \overline{\mathbf{k}}^N$, expert bounds $\underline{\mathbf{k}}^M, \overline{\mathbf{k}}^M$, affinity
    functions $v_i$, constraints $\mathbf{C}$, request selection criterion $\theta^N$, expert selection criterion $\theta^M$,
    bound modification function $h$
Initialize allocation $A$ as $A_i \leftarrow \emptyset$ for all requests $i \in N$
while $A$ is not complete do
    $\underline{\mathbf{k}}^M, \overline{\mathbf{k}}^M, \underline{\mathbf{k}}^N, \overline{\mathbf{k}}^N \leftarrow h(A, \underline{\mathbf{k}}^M, \overline{\mathbf{k}}^M, \underline{\mathbf{k}}^N, \overline{\mathbf{k}}^N)$
    if $\theta^N(A) \neq O_\emptyset$ and $\theta^M(A, i) \neq O_\emptyset$ for some $i \in \theta^N(A)$
        $i \leftarrow$ first request in $\theta^N(A)$ with $\theta^M(A, i) \neq O_\emptyset$
        $g \leftarrow$ first expert in $\theta^M(A, i)$
        $A_i \leftarrow A_i + g$
    else
        return $A$
return $A$
```

we set $\overline{\mathbf{k}}_g^M = \max\left(\underline{\mathbf{k}}_g^M - \sum_{i\in N}|A_i \cap \{g\}|, 0\right)$ for all experts $g \in M$. Thus, we spend the final $\sum_{i\in N} \overline{\mathbf{k}}_i^N - |A_i|$ steps assigning exactly the experts needed to meet expert minima. If the algorithm terminates with a complete allocation, it will satisfy the expert lower bounds.

A more general version of this modification takes as input a bound modification function h, such that $h(A, \underline{\mathbf{k}}^M, \overline{\mathbf{k}}^M, \underline{\mathbf{k}}^N, \overline{\mathbf{k}}^N)$ takes the current allocation A and expert and request bounds and determines if the bounds need to be modified to satisfy a constraint. The modified meta-algorithm is displayed in Algorithm 4.

Theorems 2.3.2 and 2.3.3 parallel Theorems 2.2.2 and 2.3.1, showing that when we modify `ERR` and `WEP` to include the bound modification function h, they return EF1 (WEF1) allocations satisfying all constraints.

Theorem 2.3.2. *Algorithm 4, implemented using the θ^N and θ^M from `ERR`, terminates with an EF1 allocation where requests receive at most k distinct experts, all constraints $\mathbf{C}$ are satisfied, and no expert g is assigned to more than $\overline{\mathbf{k}}^M$ requests. If the algorithm assigns k experts to each request, then all experts will be assigned to at least $\underline{\mathbf{k}}^M$ requests.*

Proof. Proof of the distinctness of experts per request, expert upper bounds $\overline{\mathbf{k}}^M$, satisfaction of $\mathbf{C}$, and upper limit on k experts per request follow that of Theorem 2.2.2.

We show that when the algorithm terminates with a complete allocation, the expert minima have been satisfied. We assume that the assignment problem is feasible (i.e., $\sum_{i \in N} \underline{\mathbf{k}}_i^N \leq \sum_{g \in M} \overline{\mathbf{k}}_g^M$ and $\sum_{i \in N} \overline{\mathbf{k}}_i^N \geq \sum_{g \in M} \underline{\mathbf{k}}_g^M$), so $\sum_{i \in N} \overline{\mathbf{k}}_i^N - |A_i| \geq \sum_{g \in M} \max(\underline{\mathbf{k}}_g^M - \sum_{i \in N} |A_i \cap \{g\}|, 0)$ at the beginning of the assignment process. $\sum_{i \in N} \overline{\mathbf{k}}_i^N - |A_i|$ decreases by 1 each time a request is assigned an expert and reaches 0 by the end of the assignment process, and $\sum_{g \in M} \max(\underline{\mathbf{k}}_g^M - \sum_{i \in N} |A_i \cap \{g\}|, 0)$ decreases by either 1 or 0 each iteration. At some point, either $\sum_{g \in M} \max(\underline{\mathbf{k}}_g^M - \sum_{i \in N} |A_i \cap \{g\}|, 0) = 0$ and thus all lower bounds are satisfied, or we have $\sum_{g \in M} \max(\underline{\mathbf{k}}_g^M - \sum_{i \in N} |A_i \cap \{g\}|, 0) = \sum_{i \in N} \overline{\mathbf{k}}_i^N - |A_i|$. Setting $\overline{\mathbf{k}}^M$ to $\max(\underline{\mathbf{k}}_g^M - \sum_{i \in N} |A_i \cap \{g\}|, 0)$ for all $g \in M$ ensures that every remaining choice of reviewer decreases $\sum_{g \in M} \max(\underline{\mathbf{k}}_g^M - \sum_{i \in N} |A_i \cap \{g\}|, 0)$ by exactly 1 for each of the remaining request choices. If the algorithm terminates with a complete allocation, there will be $\sum_{i \in N} \overline{\mathbf{k}}_i^N - |A_i|$ more choices, so $\sum_{g \in M} \max(\underline{\mathbf{k}}_g^M - \sum_{i \in N} |A_i \cap \{g\}|, 0)$ will be 0.

Finally, we must prove that the allocation remains EF1. The proof from Theorem 2.2.2 applies in this case as well. In that proof, we consider two requests i' and i. Then we show that if we assume i' envies i more than one item after some round t (where each round begins with an assignment to i'), we can derive a contradiction. We still have that i' must prefer the expert i got in t, g_i, to the expert i' got in t, $g_{i'}$. It is now possible that we performed the restriction of remaining experts between the assignment of $g_{i'}$ and the assignment of g_i. But because the set of experts available after the restriction is a subset of the set of experts available before the restriction, we still would have attempted to assign g_i to i' in t or earlier. The algorithm would therefore have checked for the EF1 violation when assigning g_i to i, and we can derive the same contradiction as in the proof of Theorem 2.2.2. □

Theorem 2.3.3. *Algorithm 4, implemented using the θ^N and θ^M for* `WEP`*, terminates with a WEF1 allocation where each request $i \in N$ receives at most $\overline{\mathbf{k}}_i^N$ distinct experts, all constraints*

$\mathbf{C}$ *are satisfied, and no expert* $g \in M$ *is assigned to more than* $\overline{\mathbf{k}}_g^M$ *requests. If the algorithm assigns exactly* $\overline{\mathbf{k}}_i^N$ *experts to each request, then all experts are assigned to at least* $\underline{\mathbf{k}}_g^M$ *requests.*

Proof. The distinctness of experts per request, expert upper bounds $\overline{\mathbf{k}}_g^M$, satisfaction of $\mathbf{C}$, and upper limit on $\overline{\mathbf{k}}_i^N$ experts per request $i \in N$ are satisfied for the same reasons described in Theorem 2.3.1. The lower limits are satisfied for complete assignments for the same reason described in the proof of Theorem 2.3.2.

We must prove the allocation remains WEF1. Again, we consider the moment when a request i is assigned an expert g. For all i' to which we have attempted to assign g, we check that i' does not have weighted envy for i over one item if we assign g to i. This check will still occur even if the expert restriction occurs between i''s attempt at g and i's assignment of g.

When we have $|S_{i'} \cap S_i| = 0$, we have not attempted to assign any experts to i' that we have attempted to assign to i, including g. We stated in the proof of Theorem 2.3.1 that for any expert $g_{i'}$ assigned to i', $v_{i'}(g_{i'}) > v_{i'}(g_i)$ for any g_i that i received after i' received $g_{i'}$. This claim is sufficient for the proof from [35] to go through. If the expert restriction occurs before i' received $g_{i'}$, clearly we still have $v_{i'}(g_{i'}) > v_{i'}(g_i)$ for any g_i that i received after i' received $g_{i'}$ (it is the same argument, but in the restricted problem setting). If the restriction happens after i' received $g_{i'}$, we see that i' is being assigned experts from a superset of experts compared to the previous case. Thus we still have $v_{i'}(g_{i'}) > v_{i'}(g_i)$ for any g_i that i received after i' received $g_{i'}$. □

2.4 Optimizing Orders for Picking Sequences

We showed how to obtain (W)EF1 allocations of experts to requests, but have not offered any welfare guarantees so far. We first state that maximizing welfare under `ERR` or `WEP` is NP-hard, which we prove following the techniques of Aziz et al. [6] and Aziz et al. [7].

Aziz et al. [7] present the decision problem PossibleUtilitarianWelfare: given a fair allocation instance with n agents, m goods, additive valuations functions v_i for all agents i, a class of picking sequence mechanisms $\mathcal{C}$, and an integer t, is it possible to run a picking sequence in $\mathcal{C}$ and obtain welfare t? To obtain hardness results, they use a problem top-k PossibleSet [6]: given a fair allocation instance with n agents, m goods, additive valuations functions v_i for all agents i, a class of picking sequence mechanisms $\mathcal{C}$, an agent i, and an integer k, is it possible to run a picking sequence in $\mathcal{C}$ such that i receives its top-k goods?

Proposition 2.4.1. *Maximizing welfare subject to round-robin (and `ERR`) is NP-hard.*

Proof. Aziz et al. [6] show that for $k \geq 3$, top-k PossibleSet is NP-complete for round-robin orders (they refer to round-robin orders as *strict alternation policies*). We use that fact to show that PossibleUtilitarianWelfare is NP-complete over the set of round-robin orders. Given an instance of top-k PossibleSet with $k \geq 3$ over round-robin orderings, we construct an instance of PossibleUtilitarianWelfare. One agent has utility of mk^2 for its k most preferred items, and 0 for the rest. The other agents have utility at most k for all items. Top-k PossibleSet returns true if and only if mk^3 utility is achievable.

Reducing an instance of PossibleUtilitarianWelfare with welfare threshold t under round-robin to maximizing welfare under `ERR` is simple. We can construct an instance where all experts complete at most one request. If the round-robin would run for k rounds, then we require each request to have k experts. The additional envy checks in `ERR` are only required when some request cannot be assigned its highest valued expert even though that expert has remaining capacity. Since each expert cannot be selected more than once total and there are no conflicts of interest, we never invoke that check, and `ERR` becomes equivalent to standard round-robin. Therefore, the maximum welfare under this instance of `ERR` is at least t if and only if the original PossibleUtilitarianWelfare problem evaluates to true. □

A similar proof extends to show that optimizing welfare over the tie-breaking order O is NP-hard for WEP.

Proposition 2.4.2. *Maximizing welfare subject to weighted picking (and WEP) is NP-hard.*

Proof. Aziz et al. [7] show that PossibleUtilitarianWelfare is NP-complete for recursively balanced orders (orders where each agent picks once in each round). Given an instance of PossibleUtilitarianWelfare for recursively balanced orders with welfare threshold t, we can reduce to the problem of maximizing welfare over WEP. If there are k rounds in the recursively balanced picking sequence, then assume there are k rounds in WEP. Again all experts complete at most one request. When that is the case, WEP is exactly equivalent to the weighted picking sequence from [35] (that is, no allocations are rejected because any attempted assignment will satisfy WEF1). This means that the sequence of assignments in WEP could have been achieved by a recursively balanced picking sequence in the original problem, and therefore we achieve a maximum welfare of at least t in WEP if and only if the original PossibleUtilitarianWelfare problem evaluates to true. □

In the remainder of this section, we present a simple greedy approach to approximately maximize the USW of our picking sequence by optimizing over the *ordering* of the requests. We present results using ERR (Algorithm 1), but all results apply equally well to Algorithms 2 to 4. Theorem 2.4.5 can be used to show that greedy selection of an order O is approximately welfare maximizing for any algorithm implementing Algorithm 4. In particular, when applied to WEP, this implies that greedily selecting a *tie-breaking* order for weighted picking sequences is approximately welfare optimal. Tie-breaking in weighted picking is quite powerful; for example, the tie-breaking order directly determines the first n picks, since all requests start with 0 experts.

We define a function $\text{USW}_{\texttt{ERR}}(\mathcal{O}, M, k, \overline{\mathbf{k}}^M, \{v_i\}_{i \in N}, \mathbf{C})$, which represents the USW from running ERR on agents in the order $\mathcal{O}$ with experts M, request bundle size limits $\underline{\mathbf{k}}_i^N = \overline{\mathbf{k}}_i^N = k$

Algorithm 5 Greedy Expert Round Robin (GERR)

```
Require: Requests $N$, experts $M$, expert upper limits $\overline{\mathbf{k}}^M$, affinity functions $\{v_i\}_{i \in N}$, request bundle size limit $k$, constraint matrix $\mathbf{C}$
$\mathcal{O} \leftarrow O_\emptyset$
for $t \in \{1, \ldots, n\}$ do
    $\mathcal{O} \leftarrow \mathcal{O} + i$ where $i$ maximizes $\mathrm{USW}_{\mathtt{ERR}}(\mathcal{O} + i, M, k, \overline{\mathbf{k}}^M, \{v_i\}_{i \in N}, \mathbf{C})$ over all $i \in N \setminus \mathcal{O}$
return $\mathcal{O}$
```

for all $i \in N$, expert upper limits $\overline{\mathbf{k}}^M$, affinity functions v_i for all requests $i \in N$, and constraint matrix $\mathbf{C}$. When clear from context, we will drop most of the arguments, writing $\mathrm{USW}_{\mathtt{ERR}}(\mathcal{O})$ to indicate that we run ERR with the order $\mathcal{O}$ and all other parameters defined by the current problem instance. Our algorithm, called Greedy Expert Round Robin (GERR), maintains an order $\mathcal{O}$, always adding the request i which maximizes $\mathrm{USW}_{\mathtt{ERR}}(\mathcal{O} + i)$.

The pseudocode of GERR is presented as Algorithm 5. It returns an order on requests, which can be directly input to ERR to obtain an EF1 allocation of experts. This algorithm is simple and flexible. It admits trivial parallelization, as the function $\mathrm{USW}_{\mathtt{ERR}}$ can be independently computed for each request. One can also reduce runtime by subsampling the remaining requests at each step. Subsampling weakens the approximation guarantee in theory; while we do not attempt to analyze the approximation ratio of the subsampling approach in this work, we run our largest experiments with this variant, and still obtain high-welfare allocations. Let us now establish the welfare guarantees of GERR. For this section, we assume that the valuation functions v_i have non-negative range.

We show that GERR is equivalent to greedily maximizing a γ-weakly submodular function over the intersection of two partition matroids. Consider tuples of the form (i, p) where i is a request and p represents a position in an order. We define a mapping from sets of tuples to orders. Consider the set $E = \{(i, p) : i \in N, p \in \{1, \ldots, n\}\}$. Define two partition matroids (E, I_1) and (E, I_2), such that I_1 forbids duplicating requests, and I_2 forbids duplicating positions. Define I_1 using a category for each request i, where $X_i^1 = \{(i, p) : p \in \{1, \ldots, n\}\}$, and $I_1 = \{I \subseteq E : \forall i \in N, |I \cap X_i^1| \leq 1\}$. Likewise, I_2 is defined using a category for each position p, where $X_p^2 = \{(i,p) : i \in N\}$, and $I_2 = \{I \subseteq E : \forall p \in \{1, 2, \ldots n\}, |I \cap X_p^2| \leq 1\}$. Any set Q in the intersection of these two matroids can be converted into an order $\mathcal{O}_Q$ by sorting Q on the position elements and outputting the request elements in that order. Formally, given any set $Q \in (I_1 \cap I_2)$, we construct an order $\mathcal{O}_Q = (S_Q, o_Q)$ by taking $S_Q = \{i \in N : \exists p, (i,p) \in Q\}$. For all $(i,p) \in Q$, let $p' = |\{(i', q) \in Q : q \leq p\}|$ and set $o_Q(i) = p'$. We extend this mapping to all subsets of E by sorting on the position elements as a primary key and request elements as a secondary key, then deleting all but the first tuple for each request.

Example 2.4.3. Example showing how sets $Q \subseteq E$ map to orders, and the resulting allocations from executing round-robin. Consider an instance of the expert assignment problem with 4 experts and 3 requests, in which each request receives $k = 2$ experts, and the expert upper bounds and valuations are given as

$$\overline{\mathbf{k}}^M = \begin{pmatrix} 2 & 1 & 2 & 1 \end{pmatrix} \qquad \mathbf{V} = \begin{pmatrix} 2 & 5 & 2 & 7 \\ 1 & 2 & 0 & 9 \\ 4 & 3 & 6 & 3 \end{pmatrix}.$$

The table below indicates sets $Q \subseteq E$, how they map to the order O_Q, the allocation $\texttt{ERR}(O_Q)$ resulting from executing $\texttt{ERR}$ with order O_Q, and the welfare of that allocation $\text{USW}_{\texttt{ERR}}(\mathcal{O}_Q)$. The greedy choices of $\texttt{GERR}$ are indicated with asterisks. $\texttt{GERR}$ finishes with an EF1 allocation with welfare 21, although there is an ordering which achieves 27. Order [2, 1, 3] results in an incomplete allocation.

Q	O_Q	$\texttt{ERR}(O_Q)$	$\text{USW}_{\texttt{ERR}}(O_Q)$
$\emptyset$	$[]$	$\{\},\{\},\{\}$	0
$^*\{(1,1)\}$	$[1]$	$\{g_4,g_2\},\{\},\{\}$	12
$\{(2,1)\}$	$[2]$	$\{\},\{g_4,g_2\},\{\}$	11
$\{(3,1)\}$	$[3]$	$\{\},\{\},\{g_3,g_1\}$	10
$\{(1,1),(2,2)\}$	$[1,2]$	$\{g_4,g_1\},\{g_2,g_1\},\{\}$	12
$^*\{(1,1),(3,2)\}$	$[1,3]$	$\{g_4,g_2\},\{\},\{g_3,g_1\}$	22
$^*\{(1,1),(3,2),(2,3)\}$	$[1,3,2]$	$\{g_4,g_1\},\{g_2,g_3\},\{g_3,g_1\}$	21
$\{(3,1),(2,2),(1,3)\}$	$[3,2,1]$	$\{g_2,g_3\},\{g_4,g_1\},\{g_3,g_1\}$	27
$\{(2,1),(1,2),(3,3)\}$	$[2,1,3]$	$\{g_2,g_1\},\{g_4,g_1\},\{g_3\}$	23

With these constructions defined, we observe that maximizing USW for ERR over a fixed number of rounds k is equivalent to the problem $\max_{Q\in(I_1\cap I_2):|Q|=n} \text{USW}_{\texttt{ERR}}(O_Q)$ for the matroids defined above. We will show that GERR greedily maximizes a monotonically non-decreasing version of our function over our two partition matroids. Next, we show that when our function is γ-weakly submodular, we provide a γ-dependent approximation ratio.

To make $\text{USW}_{\texttt{ERR}}(O_Q)$ monotonically non-decreasing, we will multiply by a factor of $|Q|^\alpha$, where α is defined as the smallest positive number such that $f(Q) = \text{USW}_{\texttt{ERR}}(O_Q)|Q|^\alpha$ is monotonically non-decreasing. We first prove that GERR greedily maximizes $f(Q)$. Formally, Lemma 2.4.4 states that GERR selects the request i maximizing $f(Q + (i,p))$ at each iteration.

Lemma 2.4.4. *Let $f(Q) = \text{USW}_{\texttt{ERR}}(O_Q)|Q|^\alpha$ for some α such that f is monotonically non-decreasing. Suppose GERR selects request i_t at each round t, resulting in a set of tuples Q_t. Then for all t, (i_t, t) maximizes $f(Q_{t-1} + (i,p))$ over all (i,p) such that $Q_{t-1} + (i,p) \in (I_1 \cap I_2)$.*

Proof. We first show that any greedy maximizer for $\mathrm{USW}_{\texttt{ERR}}$ is also a greedy maximizer of f. Suppose that $\mathrm{USW}_{\texttt{ERR}}(O_{Q+(i,p)}) \geq \mathrm{USW}_{\texttt{ERR}}(O_{Q+(i',p')})$. Because $|Q+(i,p)|^\alpha = (|Q|+1)^\alpha = |Q+(i',p')|^\alpha$, we have

$$\underset{\texttt{ERR}}{\mathrm{USW}}(O_{Q+(i,p)})|Q+(i,p)|^\alpha \geq \underset{\texttt{ERR}}{\mathrm{USW}}(O_{Q+(i',p')})|Q+(i',p')|^\alpha.$$

We also must prove that we can always simply append to the end of the current ordering (rather than perhaps selecting an arbitrary tuple (i,p)). Formally, we want to show that at any point in the algorithm, there is a tuple $(i,|O|+1)$ that maximizes $\mathrm{USW}_{\texttt{ERR}}(O+(i,p))$. This is shown via a strong induction argument. For the base case, if some tuple (i,p) maximizes $\mathrm{USW}_{\texttt{ERR}}(O_\emptyset+(i,p))$, then it is easy to see that $O_\emptyset+(i,p) = O_\emptyset+(i,1)$ and so we can use $(i,1)$ without loss of generality. Inductively, assume that we have a set $Q = \{(i_1,1),(i_2,2),\ldots(i_{|Q|},|Q|)\}$ and some tuple (i,p) maximizes $f(Q+(i,p))$ such that $Q+(i,p) \in (I_1 \cap I_2)$. Necessarily, $p > |Q|$, since all other positions $p' \leq |Q|$ have been filled in Q. Therefore, for any available position k, we have that $O_{Q+(i,k)} = [i_1,i_2,\ldots i_{|Q|},i]$ and thus $f(Q+(i,k))$ is the same for all allowed k. So without loss of generality, we can assume that we can select a request for the next available position (as is done in `GERR`). □

The greedy algorithm for maximizing $f(Q)$ terminates when $|Q| = n$, so we must also ensure `GERR` terminates with an order on all n requests. Although `GERR` only considers $\mathrm{USW}_{\texttt{ERR}}(O)$, which may not be monotonically increasing, by construction it runs until reaching a full order over all requests. Thus `GERR` is equivalent to greedily maximizing f.

We are now ready to prove the $1+\gamma$ approximation ratio for `GERR` (Theorem 2.4.5). Our proof is inspired by the proof in [61] that a similar greedy algorithm gives a $\frac{1}{p+1}$-approximation for maximizing a monotone submodular function over the intersection of p matroids. However, the introduction of γ-weak submodularity changes the proof, and we can simplify elements of the proof for our setting.

Theorem 2.4.5. *Suppose that f is the monotonically non-decreasing, γ-weakly submodular function $f(Q) = \mathrm{USW}_{\texttt{ERR}}(O_Q)|Q|^\alpha$. The set Q^{alg} returned by* **`GERR`** *satisfies $f(Q^{alg}) \geq \frac{1}{1+\gamma} f(Q^*)$, where O_{Q^*} is the optimal request order for* **`ERR`**. *Because $|Q^{alg}| = |Q^*|$, this implies that Q approximates the maximum welfare with constant $\frac{1}{1+\gamma}$.*

Proof. Let Q_t represent the subset of Q^{alg} after step t of `GERR`, where we add the element (i_t, t) to Q_{t-1}. Let (i^*_t, t) denote the pair in Q^* which places request i^*_t in position t. Denote $L = |Q^* \setminus Q^{\text{alg}}|$. Consider the elements of $Q^* \setminus Q^{\text{alg}} = \{(i^*_{t_1}, t_1), \dots (i^*_{t_L}, t_L)\}$, ordered so that $t_1 < t_2 < \dots t_L$. Let $Q^{\text{alg}} \cup \{(i^*_{t_1}, t_1), \dots (i^*_{t_l}, t_l)\}$ be denoted as Q^{alg}_l (with $Q^{\text{alg}}_0 = Q^{\text{alg}}$). By monotonicity of f, $f(Q^*)$ is bounded from above by:

$$f(Q^{\text{alg}} \cup Q^*) = f(Q^{\text{alg}}) + \sum_{l=1}^{L} \rho_{(i^*_{t_l}, t_l)}(Q^{\text{alg}}_{l-1}). \tag{2.1}$$

By γ-weak submodularity of f, we have that

$$\rho_{(i^*_{t_l}, t_l)}(Q^{\text{alg}}_{l-1}) \leq \gamma \rho_{(i^*_{t_l}, t_l)}(Q_{t_l - 1}). \tag{2.2}$$

Equality 2.1 and inequality 2.2 imply that

$$f(Q^*) \leq f(Q^{\text{alg}}) + \gamma \sum_{l=1}^{L} \rho_{(i^*_{t_l}, t_l)}(Q_{t_l - 1}),$$

which (again by monotonicity of f) is bounded by

$$f(Q^{\text{alg}}) + \gamma \sum_{t=1}^{n} \rho_{(i^*_t, t)}(Q_{t-1}). \tag{2.3}$$

Next, we claim that for all t,

$$\rho_{(i^*_t, t)}(Q_{t-1}) \leq \rho_{(i_t, t)}(Q_{t-1}). \tag{2.4}$$

At step t, the greedy algorithm chose to add (i_t, t) to Q_{t-1}, with i_t maximizing $f(Q_{t-1}+(i_t,t))$. If (i^*_t, p) is not present in Q_{t-1} for any p, then the greedy algorithm would have considered adding (i^*_t, t) and determined that i_t was better. Suppose that $(i^*_t, p) \in Q_{t-1}$ for some p. The greedy algorithm proceeds by filling positions from left to right, so $p \leq t-1$. By the definition of our mapping from sets to orders, i^*_t will take position p and ignore (i^*_t, t). Thus $\rho_{(i^*_t,t)}(Q_{t-1}) = 0 \leq \rho_{(i_t,t)}(Q_{t-1})$. In either case, inequality (2.4) holds. Combining (2.3) with (2.4) yields

$$f(Q^*) \leq f(Q^{\text{alg}}) + \gamma \sum_{t=1}^{n} \rho_{(i_t,t)}(Q_{t-1}) = (1+\gamma) f(Q^{\text{alg}}).$$

□

When $\gamma = 1$ (f is submodular), Theorem 2.4.5 yields a $\frac{1}{2}$-approximation guarantee, which beats the $\frac{1}{3}$-approximation guarantee provided by Fisher et al. [61]. The greedy algorithm is a tight $\frac{1}{2}$-approximation for submodular maximization in the *unconstrained* regime [26], which our result matches even though we operate in a constrained (but less general) space.

2.5 `FairSequence`: a Fast and Fair Expert Assignment Algorithm

We have shown extensively how to obtain fair and approximately welfare efficient expert assignments. However, `GERR` can be prohibitively slow.

Proposition 2.5.1. *The runtime of `GERR` is $O(kmn^4)$, where k is the number of experts assigned per request, m is the total number of experts, and n is the total number of requests.*

Proof. There are n positions to fill in the round-robin order. At each position, we have to check at most n requests to determine which is the greedy choice. For each choice of request, we run `ERR` on at most n requests over at most k rounds. When a request gets assigned an expert during `ERR`, we may have to attempt to assign at most m experts, and each time we attempt to assign, we have to check against at most n other requests for EF1 violations. Thus, each individual assignment during ERR takes $O(nm)$ time, and there are $O(nk)$ iterations of this. ERR takes $O(kmn^2)$ time, and we have to run it $O(n)$ times to select the greedy maximizer for each position, over n positions. Thus, total runtime is $O(kmn^4)$. □

In initial experiments of conference reviewer assignment we found that GERR can finish in a reasonable amount of time (one or two days) on smaller conferences ($m, n < 1000$). Even this runtime is not ideal, as conference organizers often must determine reviewer assignments over the course of several days and typically try multiple assignments using different formulas for affinity scores. In this section, we describe an approach that has improved speed and empirical welfare. Our algorithm, FairSequence, uses the weighted picking sequence described by Chakraborty et al. [35], similarly to WEP described in Section 2.3. However, rather than breaking ties using an order that is fixed ahead of time, we will break ties greedily *for each tie-break*. Thus, rather than having to run the picking sequence multiple times to determine the greedy choice in the order, we can run a single picking sequence with the determination of the greedy choice consisting of a simple maximum operation over a matrix. Because FairSequence uses the picking sequence derived from Chakraborty et al. [35], it still maintains the WEF1 criterion. FairSequence is described in Algorithm 6. FairSequence has a provably faster runtime than GERR (Proposition 2.5.2), and in practice it is significantly faster than three baseline reviewer assignment algorithms (Section 2.6).

The proof of Theorem 2.4.5 also applies to FairSequence. Rather than constructing a set of tuples of requests and positions (i, p), we construct a set of triples of requests, positions, and times (i, p, t). These triples determine how to break ties in priority at each time step. Each request-time pair must be assigned at most one position, and each position-time pair must be assigned at most one request.

Algorithm 6 `FairSequence`

```
Require: Requests $N$, experts $M$, request bounds $\underline{\mathbf{k}}^N, \overline{\mathbf{k}}^N$, expert bounds $\underline{\mathbf{k}}^M, \overline{\mathbf{k}}^M$, affinity
    functions $v_i$, constraints $\mathbf{C}$, bound modification function $h$
 1: Initialize allocation $A$ as $A_i \leftarrow \emptyset$ for all requests $i \in N$
 2: while $\exists i : |A_i| < \overline{\mathbf{k}}^N_i$ do
 3:     $\underline{\mathbf{k}}^M, \overline{\mathbf{k}}^M, \underline{\mathbf{k}}^N, \overline{\mathbf{k}}^N \leftarrow h(A, \underline{\mathbf{k}}^M, \overline{\mathbf{k}}^M, \underline{\mathbf{k}}^N, \overline{\mathbf{k}}^N)$
 4:     $S^* \leftarrow \arg\min_{i \in N} |A_i|/\overline{\mathbf{k}}^N_i$
 5:     for $i \in S^*$ do
 6:         Select $g_i \in M$ as the expert maximizing $v_i(g_i)$ such that $\sum_{i \in N} |A_i \cap \{g_i\}| < \overline{\mathbf{k}}^M_{g_i}$,
            $g_i \notin A_i$, $\mathbf{A} + \mathbb{I}(i, g_i) \preceq \mathbf{C}$, and all $i' \neq i$ satisfy WEF1 with respect to $A_i + g_i$
 7:     if No valid pair $i, g_i$ exists
 8:         return $A$
 9:     else
10:         Select $(i^*, g_{i^*})$ that maximizes $v_{i^*}(g_{i^*})$
11:         $A_{i^*} \leftarrow A_{i^*} + g_{i^*}$
12: return $A$
```

Proposition 2.5.2. *The runtime of* `FairSequence` *is $O(kmn^3)$, where k is the maximum number of experts assigned per request, m is the total number of experts, and n is the total number of requests.*

Proof. `FairSequence` fills each of the nk positions in the picking sequence sequentially. At each position, we select from at most n requests, and each request selects from m experts. For each request-expert pair, we may have to check n other requests to avoid WEF1 violations. □

In addition to the improved worst-case runtime, `FairSequence` can drastically improve on the $O(kmn^3)$ upper bound in many cases. In contrast to all algorithms presented up to this point, once we find a single valid request-expert pair to assign, we can automatically rule out any request-expert pairs with lower affinity score. We can also apply another shortcut; when checking that the WEF1 criterion is met on line 6, it is sufficient to only compare against the requests i' that have $v_{i'}(g) > v_{i'}(g')$ for some $g \in A_i \cup \{g_i\}$ and $g' \in A_{i'}$. When $v_{i'}(g) \leq v_{i'}(g')$ for all $g \in A_i$ and $g' \in A_{i'}$, $\frac{v_{i'}(A_i)}{|A_i|} \leq \frac{v_{i'}(A_{i'})}{|A_{i'}|}$. This is equivalent to WEF1 when $|A_i| = \overline{\mathbf{k}}^N_i$ and $|A_{i'}| = \overline{\mathbf{k}}^N_{i'}$, as is the case in a complete allocation. In expert assignment settings with subject areas (such that experts will largely be assigned to requests within a small number of subjects), this can rule out most comparisons across major subject areas.

`FairSequence` maintains a WEF1 guarantee if it terminates with a complete allocation.

Proposition 2.5.3. *If `FairSequence` returns a complete allocation, that allocation is WEF1.*

Proof. This holds by the final condition on line 6 of Algorithm 6. This condition implies that after every assignment of some expert g to some request i, we have checked for all other requests i' if WEF1 is met. □

Despite its benefits, `FairSequence` may not always terminate with a complete allocation. In fact, OpenReview reported a large AI conference on which `FairSequence` fails to return a complete allocation (the team was not authorized to share the data with us, nor to release the details of the conference). Therefore, we introduce a second algorithm `FairSequenceUnchecked` that will always terminate with a complete allocation. Although the second algorithm is not guaranteed to be WEF1, the fact that the algorithm is based on a weighted picking sequence implies that the allocation will still be roughly fair. `FairSequenceUnchecked` operates similarly to `FairSequence`. However, we no longer perform the additional checks used to ensure `FairSequence` is WEF1. In addition, because the picking sequence assigns experts irrevocably, when an assignment problem instance is highly constrained it may be possible for some request to have no feasible assignments later in the process. When this occurs, we have to revoke (and replace) some of the earlier assignments in order to free up a feasible expert to assign. We use a heuristic approach, where we run the algorithm with multiple different β values, indicating the amount by which we allow welfare to drop during any of these swaps. In order to allow i to swap g' for g, we require that $v_i(g) \geq \beta v_i(g')$. Setting $\beta = 1$ requires that no swaps can reduce the welfare of i, and provides little flexibility in assignments. Setting $\beta = 0$ allows any swap, potentially impacting welfare and fairness quite a bit, but ensuring termination with a complete and constraint-satisfying allocation. Although these swaps might again cause WEF1 violations, by progressively decreasing β we can limit the amount of welfare lost during this process. FairSequenceUnchecked is presented in Algorithm 7.

Algorithm 7 FairSequenceUnchecked

```
Require: Requests $N$, experts $M$, request bounds $\underline{\mathbf{k}}^N, \overline{\mathbf{k}}^N$, expert bounds $\underline{\mathbf{k}}^M, \overline{\mathbf{k}}^M$, affinity functions $v_i$, constraints $\mathbf{C}$, bound modification function $h$, sorted beta values $\boldsymbol{\beta}$
$v_{\texttt{max}} \leftarrow \max_{i \in N, g \in M} v_i(g)$
for $\beta \in \boldsymbol{\beta}$ do
    Initialize allocation $A$ as $A_i \leftarrow \emptyset$ for all requests $i \in N$
    while $\exists i : |A_i| < \overline{\mathbf{k}}^N$ do
        $\underline{\mathbf{k}}^M, \overline{\mathbf{k}}^M, \underline{\mathbf{k}}^N, \overline{\mathbf{k}}^N \leftarrow h(A, \underline{\mathbf{k}}^M, \overline{\mathbf{k}}^M, \underline{\mathbf{k}}^N, \overline{\mathbf{k}}^N)$
        $S^* \leftarrow \arg\min_{i \in N} |A_i| / \overline{\mathbf{k}}^N_i$
        for $i \in S^*$ do
            Select $g_i \in M$ as the expert maximizing $v_i(g_i)$ such that $\sum_{i \in N} |A_i \cap \{g_i\}| < \overline{\mathbf{k}}^M_{g_i}$, $g_i \notin A_i$, $\mathbf{A} + \mathbb{I}(i, g_i) \preceq \mathbf{C}$
        if No valid pair $i, g_i$ exists
            $S_{avail} \leftarrow \left\{g \in M : \sum_{i \in N} |A_i \cap \{g_i\}| < \overline{\mathbf{k}}^M_{g_i}\right\}$
            $S \leftarrow \{(g, -1) : g \in S_{avail}\}$
            $S \leftarrow S + \{(g, i) : i \in N, g \in A_i\}$
            $E \leftarrow \{((g', i'), (g, i), v_{\texttt{max}} + v_i(g) - v_i(g')) : g \in A_i, g' \notin A_i, \mathbf{A} + \mathbb{I}(i, g') - \mathbb{I}(i, g) \preceq \mathbf{C}, v_i(g') \geq \beta v_i(g)\}$
            Set graph $G \leftarrow (S, E)$
            $Z = (g_0, -1), (g_1, i_1), \ldots (g_z, i_z) \leftarrow$ shortest weighted path in $G$ from $S_{avail}$ to a request $i_z$ with expert $g_z \in A_{i_z}$ s.t. $\exists i^* \in S^*$ with $g_z \notin A_{i^*}$ and $\mathbf{A} + \mathbb{I}(i^*, g_z) \preceq \mathbf{C}$
            Swap experts along $Z$ ($A_{i_j} \leftarrow A_{i_j} - g_j + g_{j-1}$ for $j \in \{1, \ldots z\}$
            $i^* = \arg\max_{\{i \in S^* : g_z \notin A_i, \mathbf{A} + \mathbb{I}(i, g_z) \preceq \mathbf{C}\}} v_i(g_z)$
            $g_{i^*} \leftarrow g_z$
            if No valid $i^*$ exists
                Restart with lower value of $\beta$
        else
            Select $(i^*, g_{i^*})$ that maximizes $v_{i^*}(g_{i^*})$
        $A_{i^*} \leftarrow A_{i^*} + g_{i^*}$
    return $A$
```

Theorem 2.5.4. *FairSequenceUnchecked returns a complete allocation satisfying all assignment constraints, if such an allocation exists.*

Proof. It suffices to consider the case when $\beta = 0$. Suppose the picking sequence reaches a point where there are no valid assignments possible to the requests $S^* = \arg\min_{i \in N} |A_i|/\overline{\mathbf{k}}_i^N$, but some experts have not yet reached their maximum load. Denote the partial allocation up to this point as A. Consider any request $i \in S^*$. There is a complete allocation A^c satisfying all assignment constraints, where $A_i^c = \{g_1, g_2, \ldots g_{\overline{\mathbf{k}}_i^N}\}$. $|A_i| < |A_i^c|$, so there is some expert $g \in A_i^c \setminus A_i$. If g has not reached its load upper bound in A, we could add g to A_i, since g does not have a conflict of interest with i. So g must have reached its load upper bound under A. Consider any request that has been assigned g in A. One such request, i', must have $g \notin A_{i'}^c$, and instead there is some $g' \notin A_{i'}$ with $g' \in A_{i'}^c$. Again, if g' is available, we can simply assign g' to i' and g to i. If not, the argument repeats for g' – there must be some request currently assigned g' that does not receive g' in A^c. This sequence must eventually terminate, since there are a finite number of experts and requests. Thus, eventually we will find a path to the set of experts with remaining load, and we will be able to make transfers along this path to our request p, allowing us to increase the total number of assignments by 1 and move on with the rest of the picking sequence. □

`FairSequenceUnchecked` may have a very long runtime. However, `FairSequence` often terminates with a complete allocation in practice, making `FairSequenceUnchecked` unnecessary. In addition, we expect that the picking sequence will run almost to completion before needing to make any of the swaps proscribed by `FairSequenceUnchecked`. This means that the additional runtime of finding and executing these swaps will typically not have a strong impact on the overall runtime.

Proposition 2.5.5. *The runtime of `FairSequenceUnchecked` is $O(|\boldsymbol{\beta}|kn(nm + m)^2)$, where $|\boldsymbol{\beta}|$ is the number of values of β in $\boldsymbol{\beta}$, k is the maximum number of experts assigned per request, m is the total number of experts, and n is the total number of requests.*

Proof. Like FairSequence, FairSequenceUnchecked fills each of the nk positions in the picking sequence sequentially. At each position, we may have to visit all nodes and edges of the exchange graph. Since there are $O\left((n+1)m\right)$ nodes, this operation costs $O((nm+m)^2)$ time. In the worst case, we have to run the main loop of the algorithm $|\boldsymbol{\beta}|$ times. □

2.6 Empirical Fairness, Efficiency, and Runtime Analysis

We compare GERR and FairSequence against baselines on three conference datasets. We find FairSequence is over an order of magnitude faster than all baselines and much fairer (in terms of WEF1 and Gini) than all baselines except PeerReview4All (PR4A) [157].

2.6.1 Experimental Design

We run experiments on three conference datasets: Medical Imaging with Deep Learning (MIDL), Conference on Computer Vision and Pattern Recognition (CVPR), and the 2018 iteration of CVPR. According to Shah [148], the "CVPR" dataset is from 2017, though the paper introducing the dataset does not list the year [91]. We assign the experts (reviewers) to the requests (papers).

MIDL, CVPR, and CVPR'18 are standard datasets in the reviewer assignment literature, provided as pre-computed affinity score matrices, reviewer load upper bounds, and paper demands. MIDL is an order of magnitude smaller than CVPR and CVPR'18, and CVPR'18 is less challenging than CVPR due to a higher ratio of reviewer availability to paper demand.

A summary of the data statistics appears in Table 2.1. For CVPR'18, while affinities are between 0 and 11, most are between 0 and 1. In addition, reviewer load upper bounds vary by reviewer but range between 2 and 9. All conferences have $\underline{\mathbf{k}}_g^M = 0$ for all $g \in M$.

We compare our methods to the FairFlow algorithm [91], the Toronto Paper Matching System (TPMS) [38], and PeerReview4All (PR4A) [157]. FairFlow is currently implemented in OpenReview, and it is in widespread use. TPMS (also in widespread use) provides an

Table 2.1: Data summary for the three reviewer assignment datasets.

Name:	n	m	val. range	k	$\overline{\mathbf{k}}_g^M$
MIDL:	118	177	$[-1, 1]$	3	4
CVPR:	2623	1373	$[0, 1]$	3	6
CVPR'18:	5062	2840	$[0, 11]$	3	$2 - 9$

upper bound on welfare without fairness guarantees. PR4A was used by ICML 2020 [156]. All algorithms are publicly available on Github [119, 125, 160].

Following Kobren et al. [91] and Stelmakh et al. [157], we only run one iteration of PR4A on CVPR and CVPR'18. On those two conferences, PR4A maximizes the minimum paper score, but stops before maximizing the next smallest score.

We also implemented the Constrained Round Robin (CRR) algorithm [8]. CRR is approximately 40 times slower than GRRR on MIDL, taking 400 seconds instead of 10. GRRR takes about 18 hours to run on CVPR. Extrapolating these results, we can expect CRR to require a month of computation time or longer on CVPR (it did not terminate in our experiments). Given its infeasible runtime, we did not continue to compare against CRR as a baseline. Due to the size of the CVPR'18 dataset, we subsample 100 papers at each iteration of GRRR rather than testing every available paper. We recorded means and standard deviations over 5 runs, but found little variation in solution quality from run to run.

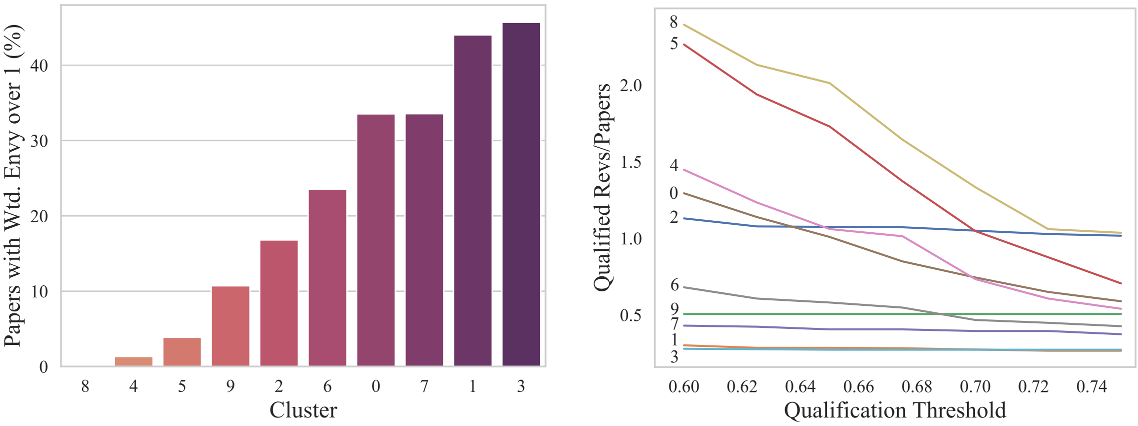
2.6.2 Fairness and Efficiency

Fairness and efficiency results for all conferences are included in Table 2.2. We report the USW, minimum paper score, and number of EF1 violations for each algorithm. For each setting with at least one violation of the EF1 criterion, we report the total number of papers that envy some other paper more than one reviewer, and the total number of papers that are envied by some paper by more than one reviewer. We report the USW as the percentage of the optimal value (given by TPMS). For an allocation A, the number of EF1 violations is the number of ordered pairs of papers $(i, i') \in (N \times N)$ failing EF1. There are $n^2 - n$ total potential violations, since an agent cannot envy itself.

Table 2.2: High-level fairness and welfare statistics for all algorithms on all three conferences.

	Alg.	USW (% OPT) (↑)	Min Score (↑)	EF1 Viol. (↓)	Num. Envious (↓)	Num. Envied (↓)
MIDL	FairFlow	**100%**	**0.94**	**0**	**0**	**0**
	TPMS	**100%**	0.90	**0**	**0**	**0**
	PR4A	98%	0.92	**0**	**0**	**0**
	GRRR	98%	0.83	**0**	**0**	**0**
	FairSeq	99%	0.87	**0**	**0**	**0**
CVPR	FairFlow	96%	**0.77**	23244	688	1058
	TPMS	**100%**	0.00	471256	717	2097
	PR4A	94%	**0.77**	83	14	75
	GRRR	88%	0.00	**0**	**0**	**0**
	FairSeq	92%	0.00	**0**	**0**	**0**
CVPR'18	FairFlow	97%	9.79	23	21	23
	TPMS	**100%**	1.37	134	65	108
	PR4A	97%	**12.68**	2	1	2
	GRRR	94%	1.78	**0**	**0**	**0**
	FairSeq	96%	1.74	**0**	**0**	**0**

FairFlow and TPMS have very high levels of EF1 violations on CVPR. Although some EF1 violations may be permissible, a large number of violations implies that many papers received unnecessarily imbalanced assignments relative to other papers. We also show the number of papers that have envy for another paper, that cannot be rectified by dropping one reviewer. This analysis shows that the EF1 violations are generally spread out across roughly 10% of the papers, but those papers typically envy a much larger number of papers. In other words, there is a small proportion of papers that received an unduly low quality of reviewer assignments, and would have strongly preferred many other papers' assignments.



(a) Number of papers with envy over one reviewer for any other paper, by paper cluster.

(b) The number of qualified reviewers per cluster, divided by the cluster size.

Figure 2.1: Envy violations and reviewer quality in CVPR, using `TPMS` assignment.

To further understand the potential sources and impacts of EF1 violations, we analyze the distribution of papers that have envy over one reviewer for any other paper in the `TPMS` assignment for CVPR. Although the paper identities are not available in the dataset, we can simulate a subject area classification by clustering papers. We represent each paper $i \in N$ as the vector of affinity scores between i and all reviewers, $\mathbf{V}_i$. We then cluster these vectors into 10 clusters using agglomerative clustering under a Euclidean metric and Ward linkage function [145]. Figure 2.1a shows the percentage of papers in each cluster that envy at least one other paper more than one reviewer. We can see that some of the clusters have very high percentages of strongly envious papers (up to 40%), while others have almost no envy.

What drives these WEF1 violations? In Figure 2.1b, we plot the number of “highly qualified” reviewers for a subject area, divided by the number of papers in the area. We define a reviewer as highly qualified for a subject if the average affinity over the top four

Table 2.3: Inequality statistics for `GRRR`, `FairSequence`, `FairFlow`, and `PR4A`.

	Alg.	Lowest 10% (↑)	Lowest 25% (↑)	Gini (↓)	Envy (↓)
MIDL	`FairFlow`	1.051 ± .072	1.186 ± .131	.146	.501
	`PR4A`	**1.069 ± .082**	**1.211 ± .135**	**.127**	.448
	`GRRR`	.995 ± .095	1.164 ± .157	.145	.834
	`FairSeq`	1.040 ± .085	1.191 ± .139	.140	**.146**
CVPR	`FairFlow`	.838 ± .032	.908 ± .068	.233	64462
	`PR4A`	**1.065 ± .150**	**1.324 ± .247**	**.145**	**9287**
	`GRRR`	.898 ± .176	1.110 ± .217	.183	22400
	`FairSeq`	.978 ± .139	1.197 ± .218	.169	10602
CVPR’18	`FairFlow`	11.053 ± .536	12.519 ± 1.805	.151	6940
	`PR4A`	**15.280 ± .952**	**16.668 ± 1.348**	**.103**	**2480**
	`GRRR`	8.923 ± 2.890	12.220 ± 3.528	.168	28840
	`FairSeq`	10.084 ± 2.540	12.950 ± 3.182	.154	17419

most similar papers in the subject exceeds some threshold. Figure 2.1b demonstrates the number of “highly qualified” reviewers per subject area as we vary the qualification threshold. The subject areas with the highest fraction of submissions violating EF1 also tend to have lower ratios of qualified reviewers to papers, indicating that EF1 violations accumulate in more heavily resource-constrained subject areas. It is possible that these subject areas suffer from a dearth of qualified reviewers; thus, any papers that receive qualified reviewers are inevitably envied by other papers in that subject area.

For additional measures of inequality, we compute the mean and standard deviation of paper scores for the bottom decile and quartile of papers per allocation. We consider allocations to be more fair if they allocate higher scores to these disadvantaged papers. We also calculate the Gini coefficient, and the sum of the total envy over all ordered paper pairs $(i, i') \in (N \times N)$. The results are summarized in Table 2.3. `FairSequence` significantly outperforms `FairFlow` in the scores given to lower decile and quartile papers, as well as in the Gini index. These metrics are closer on MIDL and CVPR'18, which we have already seen are less constrained settings.

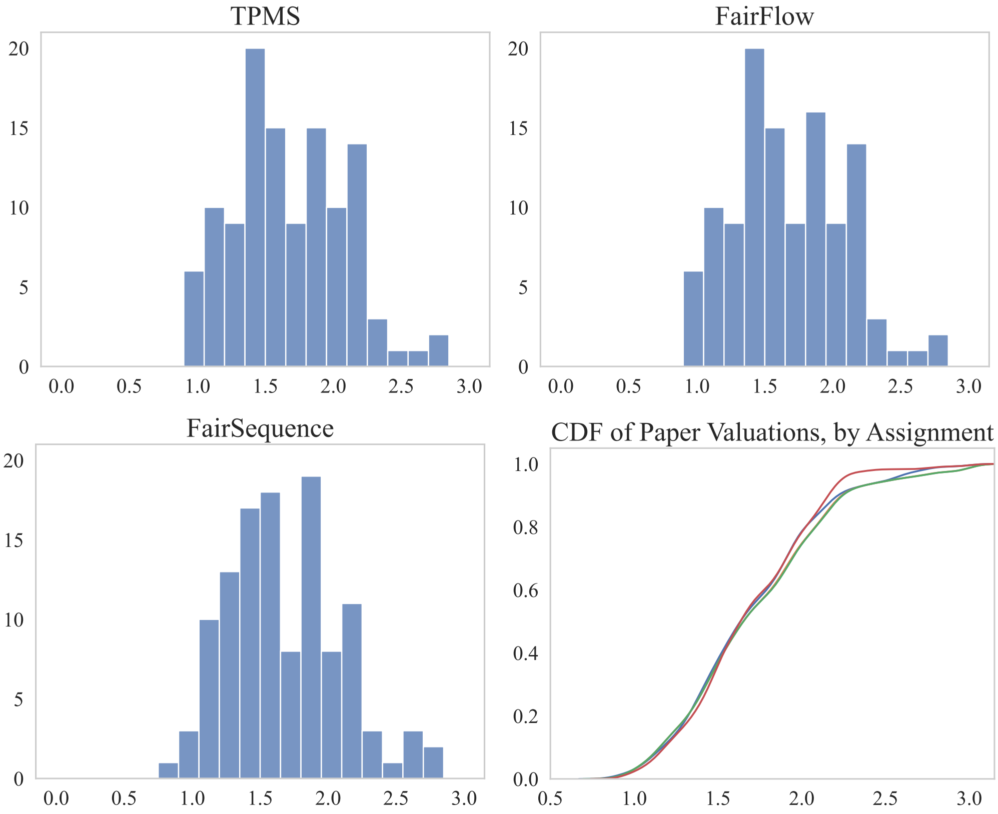


Figure 2.2: Distribution of paper valuations for MIDL under `TPMS`, `FairFlow`, and `FairSequence`.

Figures 2.2 to 2.4 show the full distribution of paper scores under `TPMS`, `FairFlow`, and `FairSequence` for MIDL, CVPR, and CVPR'18. MIDL shows almost no variation across algorithms, while CVPR and CVPR'18 show more nuanced tradeoffs across algorithms. For CVPR (Figure 2.3), `FairFlow` maximizes the minimum paper score, but it leaves many papers clustered near the minimum paper score. `FairSequence` smoothly shifts the entire distribution of paper scores rightward. `PR4A` is much fairer than both algorithms, but it does not handle variable paper demands, and has a higher computational overhead, as we will see in Section 2.6.3. For CVPR'18 (Figure 2.4), `FairSequence` outperforms `FairFlow` on

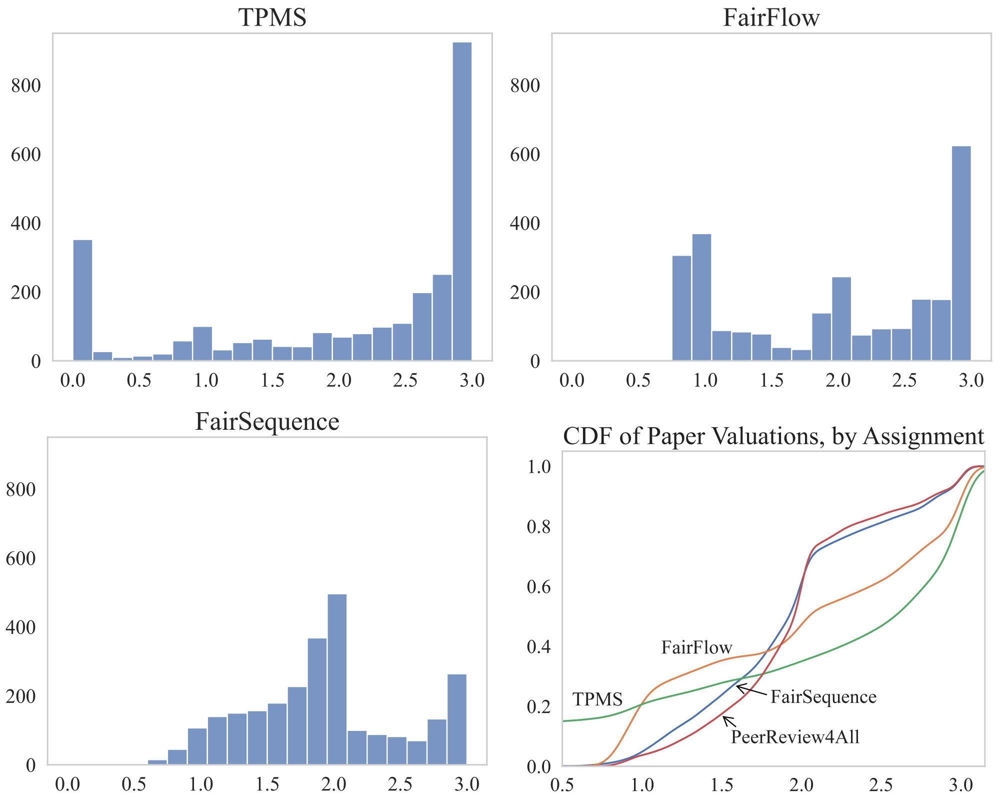


Figure 2.3: Distribution of paper valuations for CVPR under `TPMS`, `FairFlow`, and `FairSequence`.

low percentiles, but `TPMS` shows stronger bottom quartile scores than either. However, `TPMS` creates a cluster of very low scoring papers that are fully or partially mitigated by the other approaches. Once again, `PR4A` seems to maintain strong fairness guarantees relative to all baselines.

2.6.3 Runtime Analysis

Perhaps the biggest benefit of `FairSequence` is its greatly improved runtime. We display the runtimes (in seconds) of `PR4A`, `FairFlow`, `TPMS`, and `FairSequence` in Figure 2.5. Runtimes are not reported for MIDL since all algorithms take $<$ 10 seconds to run. `GRRR` takes longer than a day to run on CVPR and CVPR'18, so we also do not include it in

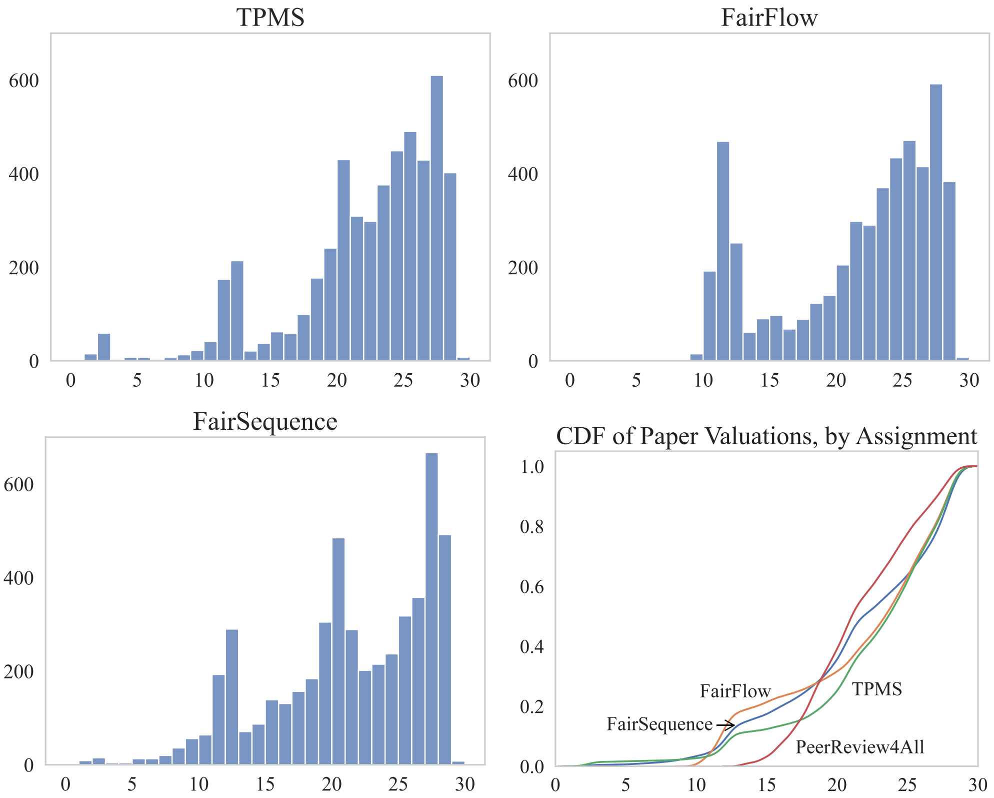


Figure 2.4: Distribution of paper valuations for CVPR'18 under `TPMS`, `FairFlow`, and `FairSequence`.

the analysis. We find that `FairSequence` is faster than even `TPMS`, which has no fairness guarantees. Further, `FairSequence` has been implemented in pure Python while the other three rely primarily on highly optimized optimization tools (`PR4A` and `TPMS` were implemented using Gurobi, and much of the computation in `FairFlow` was done using Google's OR-Tools). Despite these implementation differences, `FairSequence` is consistently at least 3 times faster than `TPMS`, 10 times faster than `FairFlow`, and 50 times faster than `PR4A`.

2.6.4 Estimation of Empirical Guarantees

We estimate α and γ for `GERR` on MIDL, CVPR, and CVPR'18. The results are displayed in Table 2.4. For any order $\mathcal{O}$ and any paper $i \notin \mathcal{O}$, we must have that $\text{USW}_{\texttt{ERR}}(\mathcal{O}+i)|\mathcal{O}+i|^{\alpha} \geq$

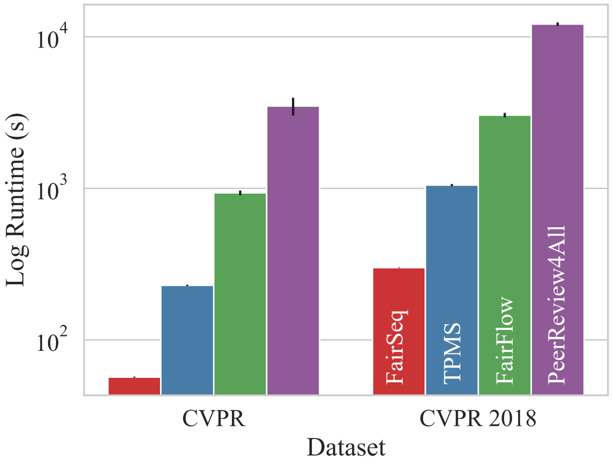


Figure 2.5: Runtimes of `FairSequence`, `TPMS`, `FairFlow`, and `PR4A` on CVPR and CVPR 2018.

$\text{USW}_{\texttt{ERR}}(\mathcal{O})|\mathcal{O}|^{\alpha}$. When $\text{USW}_{\texttt{ERR}}(\mathcal{O}+i) > \text{USW}_{\texttt{ERR}}(\mathcal{O})$, any positive α will satisfy this inequality. We estimate α by sampling orders $\mathcal{O}$ and papers $i \notin \mathcal{O}$, and we take our estimate to be slightly greater than the maximum α found for any $\mathcal{O}$ and paper $i \in N$. For MIDL, we found that no sampled $\mathcal{O}$ and i violate monotonicity, so we set $\alpha = 0.01$. Using our estimated α values, we then estimate γ. Here, we sample X and Y so that $X \subseteq Y$, and some pair $(i,p) \in (N \times \{1, \ldots n\})$ with $(i,p) \notin Y$. We need $\gamma \geq \frac{\rho_{(i,p)}(Y)}{\rho_{(i,p)}(X)}$ for all samples. Similarly to our α estimate, we compute $\frac{\rho_{(i,p)}(Y)}{\rho_{(i,p)}(X)}$ for all samples and then estimate γ to be slightly greater than the maximum value. We found in all experiments that our chosen α parameter led to all positive marginal gains during the γ estimation, improving our confidence in the α estimates. γ is rather large for CVPR and CVPR'18. It is possible that other conferences or other application areas would yield welfare functions that are closer to monotonically increasing and submodular, leading to lower values of γ.

Table 2.4: Estimated α and γ parameters for all three conference datasets.

	α	γ
MIDL	0.01	1.21
CVPR	1.03	50.62
CVPR'18	0.51	17.41

2.7 Fairness to Reviewers

We take the position throughout this thesis that whenever fairness is relevant, fairness is considered with respect to requests rather than with respect to experts. In most applications, completion of a request is a chore that the experts have volunteered or are being paid to complete, while requests (and society more broadly) benefit substantially from assignment of appropriate experts. Still, we verify that `GERR` and `FairSequence` are at least as fair to reviewers as the baselines. For each conference, we compute the distribution of reviewing loads for all algorithms. Our method is relatively consistent with the baselines, and does not introduce a large unfairness in reviewing load. Applying Algorithm 4, we also test `ERR` with reviewer load lower bounds of $\underline{\mathbf{k}}_g^M = 2$ for all $g \in M$ (the value used in [91]). On all three conferences, the algorithm terminates with complete allocations satisfying reviewer lower and upper bounds, while maintaining EF1 guarantees and competitive USW.

For each algorithm and conference, we compute the number of reviewers receiving each possible reviewing load. These results are displayed in Figure 2.6. This figure also include results from `FairIR`, another algorithm introduced in Kobren et al. [91] that has not been applied in a real conference setting to date. Reviewer $g \in M$ can receive a load in the interval $[0, \overline{\mathbf{k}}_g^M]$. In general, our approaches are about as fair as the other algorithms in terms of reviewer load, though on CVPR there are about 100 reviewers receiving one or two extra papers compared to `FairFlow` and `TPMS`.

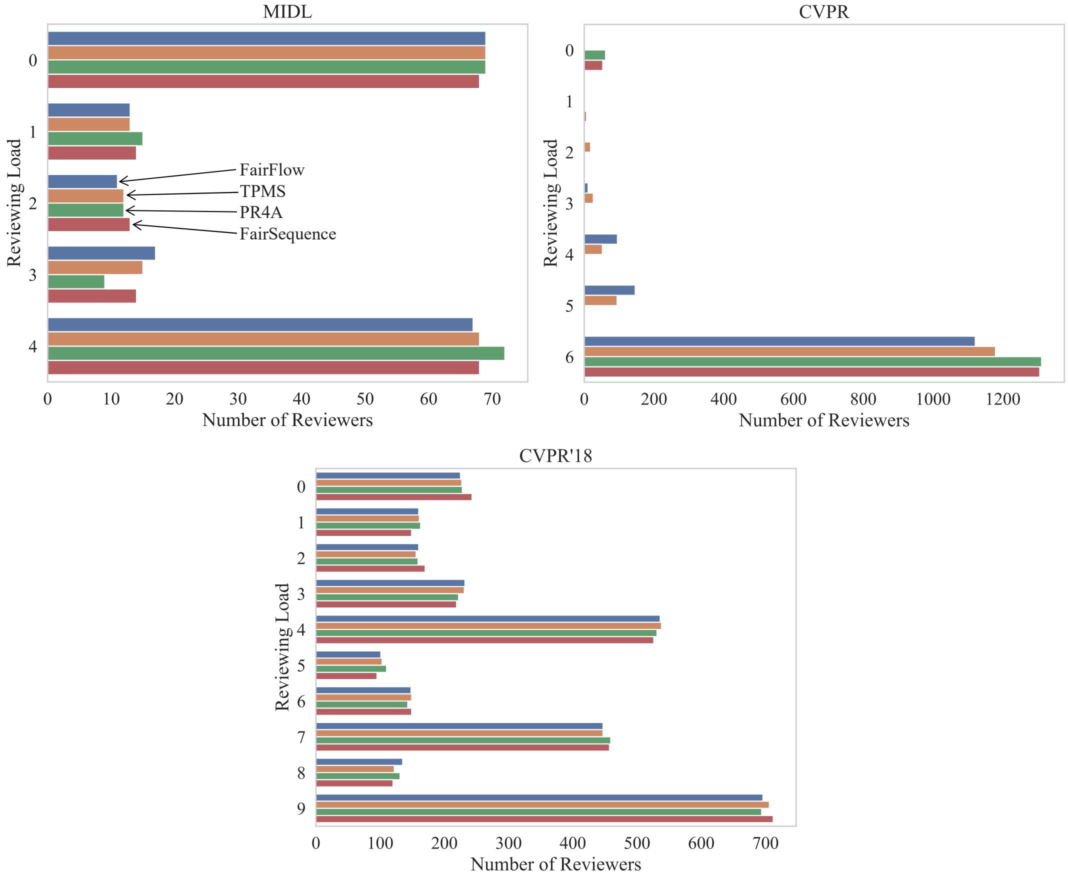


Figure 2.6: Distribution of reviewer loads for all algorithms on each dataset.

We also compare multiple statistics of reviewer welfare and fairness in Table 2.5. Because reviewers consider papers to be chores, we first convert the affinity scores by subtracting the maximum affinity score (per conference) from $v_i(g)$ for all papers $i \in N$ and reviewers $g \in M$. We can then specify $v_g(i)$ using these new values (and bundles are valued additively, as before). The maximum score for any $v_g(i)$ is 0 and the minimum score is the negative of the original maximum $v_i(g)$. In addition, higher scores are better, with the best score being 0 either because no papers were assigned or because all assigned papers had maximum affinity.

Table 2.5: Statistics for reviewer-centric scores on all three conferences.

	Alg.	USW (↑)	Min Score (↑)	EF1 Viol. (↓)	Lowest 10% (↑)	Lowest 25% (↑)
MIDL	FairFlow	**-.86**	**-2.57**	6611	**-2.34 ± .16**	**-2.04 ± .27**
	TPMS	**-.86**	**-2.57**	**6560**	**-2.34 ± .15**	**-2.04 ± .27**
	PR4A	-.88	-2.60	6632	-2.36 ± .15	-2.10 ± .24
	GRRR	-.88	-2.86	6667	-2.40 ± .16	-2.12 ± .27
	FairSeq	-.88	-2.77	6659	-2.38 ± .18	-2.07 ± .29
CVPR	FairFlow	**-1.92**	**-6.00**	$\mathbf{1.645 \times 10^4}$	**-5.50 ± .50**	**-4.85 ± .71**
	TPMS	-1.77	-6.00	2.778×10^4	-5.92 ± .27	-4.91 ± 1.00
	PR4A	-1.99	-6.00	7.921×10^4	-6.00 ± 0.00	-5.38 ± 1.12
	GRRR	-2.26	-6.00	7.865×10^4	-6.00 ± 0.00	-5.73 ± .54
	FairSeq	-2.05	-6.00	7.468×10^4	-6.00 ± 0.00	-5.53 ± .83
CVPR ’18	FairFlow	-21.00	-95.50	1.119×10^6	-65.50 ± 11.53	-48.84 ± 16.07
	TPMS	**-19.72**	-94.00	$\mathbf{1.094 \times 10^6}$	**-63.90 ± 11.82**	**-46.93 ± 16.37**
	PR4A	-21.08	**-93.87**	1.134×10^6	-64.01 ± 11.73	-47.44 ± 16.08
	GRRR	-22.21	-94.45	1.219×10^6	-69.77 ± 10.26	-53.28 ± 15.86
	FairSeq	-21.28	-95.22	1.216×10^6	-70.19 ± 10.48	-52.89 ± 16.58

Note that the EF1 criterion is different for chores as well. An allocation A is envy-free up to one chore (EF1) for experts (reviewers) if for all pairs of experts $(g, g') \in M^2$, either $v_g(A_g) \geq v_g(A_{i'})$ or $\exists i \in A_g$ such that $v_g(A_g \setminus \{i\}) \geq v_g(A_{i'})$. Interestingly, it appears that TPMS has the best efficiency and fairness properties for reviewers, despite the fact that TPMS especially shows poor fairness properties for papers. No algorithm performs particularly well in terms of reviewer fairness, since all algorithms have fairly large numbers of EF1 violations. These results highlight the tradeoff between fairness for papers and fairness for reviewers. Nothing about these results indicates that obtaining fair outcomes for both papers and reviewers is impossible, but they do highlight the need for an algorithm explicitly designed to be fair to both sides.

2.8 Extending to Submodular Valuations

It might be natural to try to extend `FairSequence` to valuations beyond additive. Indeed, existing literature has already explored the problem of utilitarian welfare maximization for reviewer assignment under some submodular valuations. One line of work identifies multiple topics per paper, and assigns reviewers to obtain strong coverage over each paper's topic [85, 86]. Ahmed et al. [1], Kou et al. [92] work in a similar setting, providing approximation guarantees for the problem of maximizing utilitarian welfare. These works do not consider fairness across papers.

The picture becomes more interesting once we turn our attention to fairness constraints or fair welfare objectives. When allowing submodular valuations, Chakraborty et al. [37] identify an example where a complete weighted EF1 allocation does not exist. In their setting, requests' weights for computing the WEF1 constraint do not necessarily correspond to limits on the requests' bundle sizes (as they do in this thesis).

Example 2.8.1 (Chakraborty et al. [37])**.** Suppose we have two requests and six or more experts (all with unit capacity). Both requests can receive between zero and six experts. Request 1 has weight 1 and request 2 has weight 2. Request 1 values all bundles $A_1 \subseteq M$ with value $|A_1|$. Request 2 values all bundles except the empty bundle with value 1. Request 2 will have weighted envy over one item for request 1 whenever request 1 is assigned more than one expert. But assigning five items to request 2 will cause request 1 to have weighted envy over one item.

Although this example discounts the ability to compute complete, WEF1 allocations under submodular valuations with general weights, it may not exclude the ability to do so in our setting. Perhaps it is precisely the mismatch between bundle sizes and requests' weights that causes difficulty. We leave a proof or counterexample for this setting for future work.

It is rather straightforward to extend `FairSequence` in other ways. Montanari et al. [114] show that to extend the picking sequence of Chakraborty et al. [37] to submodular valuations, one can simply let requests select the expert with the highest *marginal* gain. They prove that this adapted algorithm satisfies the fairness notion of WMEF$(x,\, 1-x)$ for all $x \in [0,1]$. WMEF$(x,\, 1-x)$ requires that for all pairs of requests i, i' with weights $w_i, w_{i'}$, either $A_{i'} = \emptyset$ or there is some $g \in A_{i'}$ such that

$$\frac{v_i(A_i) + x(v_i(A_i + g) - v_i(A_i))}{w_i} \geq \frac{v_i(A_i \cup A_{i'}) - v_i(A_i) - (1-x)(v_i(A_i \cup A_{i'}) - v_i(A_i \cup A_{i'} - g))}{w_{i'}}.$$

We can adapt their algorithm to the expert assignment setting in much the same way we adapted the weighted picking sequence of Chakraborty et al. [37].

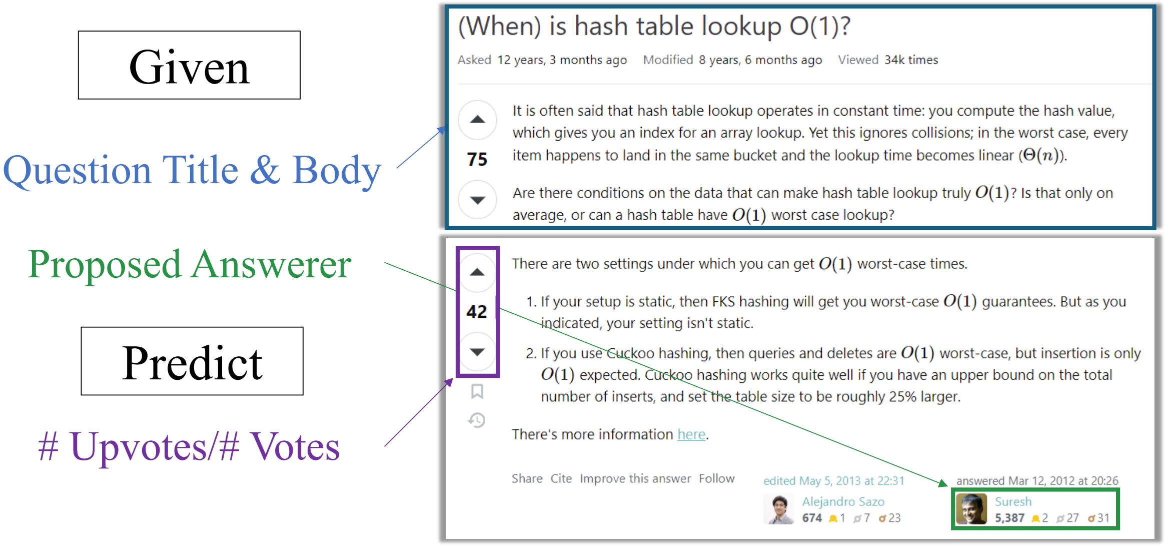
In this chapter, we have shown how to quickly obtain fair and welfare-efficient allocations of experts to requests. In the next section, we will start to investigate the question of how to compute the valuation matrix $\mathbf{V}$.

CHAPTER 3

BUILDING PREDICTIVE MODELS FOR EXPERT ASSIGNMENT

In this section, we explore how to compute the valuation matrix $\mathbf{V} \in \mathbb{R}^{n \times m}$. Ideally, we would assign experts to requests so as to optimize overall task performance, but expert performance is unknown prior to making the assignment. We propose predicting these performance metrics and assigning using the predictions. Using an expert-assignment task derived from StackExchange, we show that explicitly predicting expert performance has a large impact on assignment decisions and can improve overall welfare. We demonstrate this claim using both theoretical bounds on statistical generalization guarantees and automated metrics of assignment quality. This work highlights the effectiveness of predictive assignment, and the need to collect high quality datasets linking pre- and post-allocation measures in other important expert assignment tasks.

We frame the expert assignment task as the combination of performance prediction and optimal assignment. We select a single target outcome of interest, which measures expert task performance. These outcomes are labeled from among a set of outcome labels. We then train a model to predict the outcome when assigning different experts to a new problem. By estimating the probabilities of each outcome, we can easily trade-off between different experts for a given task. More importantly, these probabilities help us assign many experts to many tasks at scale.



(When) is hash table lookup O(1)?

Asked 12 years, 3 months ago Modified 8 years, 6 months ago Viewed 34k times

75

It is often said that hash table lookup operates in constant time: you compute the hash value, which gives you an index for an array lookup. Yet this ignores collisions; in the worst case, every item happens to land in the same bucket and the lookup time becomes linear ($\Theta(n)$).

Are there conditions on the data that can make hash table lookup truly $O(1)$? Is that only on average, or can a hash table have $O(1)$ worst case lookup?

42

There are two settings under which you can get $O(1)$ worst-case times.

1. If your setup is static, then FKS hashing will get you worst-case $O(1)$ guarantees. But as you indicated, your setting isn't static.
2. If you use Cuckoo hashing, then queries and deletes are $O(1)$ worst-case, but insertion is only $O(1)$ expected. Cuckoo hashing works quite well if you have an upper bound on the total number of inserts, and set the table size to be roughly 25% larger.

There's more information here.

Share Cite Improve this answer Follow

edited May 5, 2013 at 22:31
Alejandro Sazo
674 1 7 23

answered Mar 12, 2012 at 20:26
Suresh
5,387 2 27 31

Figure 3.1: Visual display of the answer quality prediction task.

3.1 Predicting Expert Performance on StackExchange

We build a model to predict the conditional probability that a vote on a user's answer will be an upvote. We use features derived from the user's past answers, as well as textual features of the question, to build our predictive model. A visual of the prediction task is shown in Figure 3.1.

Once we have these predictive scores, we simulate an assignment of users to questions using the predicted scores as $\mathbf{V}$.

3.1.1 Contributions

The contributions of this chapter are:

• We rigorously investigate feature importance of a logistic regression model trained to predict probability of upvotes, showing the strength of historical performance measures in particular (Section 3.3.2).

- We derive theoretical high-probability bounds for assignment quality under our predictive model (Theorem 3.3.3).
- We demonstrate experimentally that assignments made under the predictive model have higher worst-case predicted quality and assign users with a stronger historical track record than baselines (Section 3.4).
- We study the stability of and correlations between 11 different automated metrics of assignment quality, enabling a nuanced understanding of the interactions between expert assignment measures and outcomes (Section 3.4.1).

3.2 Data Availability and Causal Inference

Our ultimate goal in this chapter is to determine which features and assignment methods improve outcomes in resource-constrained, expertise-driven tasks. Ideally, we want answers that generalize across tasks (i.e., apply to peer review, community question answering, etc) as well as across instances within tasks (i.e., we can expect the answers to hold for all CQA expert-assignment instances). To address the first point, we would need to have access to data from each of the domains in question. To address the second point, we would need to apply methods that allow us to infer *causal* relationships between input and output variables. We discuss both of these limitations here.

3.2.1 Data Availability

The purpose of this chapter is to identify the relationships between inputs and outputs in expert assignment. However, it is difficult to obtain data comparing inputs and outputs in our primary motivating example of reviewer assignment. Saveski et al. [143] analyzed the connections between inputs (such as bids, TPMS/ACL scores, and keyphrase matching scores) and outputs (reviewer self-reported expertise and confidence scores). To study the connections between inputs and outputs, it is necessary to have access to all reviewers' identities, as well as the outcomes of the review process including the review text and numerical scores. The authors obtained access to this data by volunteering as workflow chair for multiple years of AAAI. However, this data is not typically available to most researchers. This high level of privacy makes it very difficult to study the impact of reviewer assignment policies. This chapter gives a sense of the insights available to those who can fully analyze reviewer assignment data from conferences.

3.2.2 Causal Inference

One other limitation of this chapter is the causal-explanatory power of the analyses.

In causal analysis, A/B tests are the gold standard, but are very costly. Some recent conferences have performed A/B testing to determine the consistency of the review process or to measure the impact of reviewer/author blinding [19, 96, 132, 158, 168]. These tests can demonstrate causal effects, but require doubling the number of solicited reviews.

Saveski et al. [143] avoid such costly experiments by applying an off-policy evaluation technique. They "harvest" the randomness of randomized reviewer assignment policies used by the large AI conference, AAAI. Parts of their analysis require access to the true distribution over assignments, which is only known to the workflow chairs of the conference. Later in this chapter, we will apply their analysis to deterministic allocations, treating a deterministic allocation as a trivial distribution where the deterministic allocation has probability 1 and everything else has probability 0. Since a large part of the analysis in Saveski et al. [143] concerns imputing bounds for the values of reviewer-paper pairs with 0 probability under the conference's randomized allocation, we still derive useful bounds using the imputation alone.

We develop a similar idea in Section 3.3.1, where we compute bounds on the true average answer quality by estimating generalization error bounds on our predictor of answer quality. This model requires us to have bounds on the ratio of the probability of seeing a request-expert pair in the training data vs. the probability of seeing that same pair among the assigned pairs. This analysis is similar to that introduced by Cousins et al. [44].

In addition to the off-policy evaluation techniques used by Saveski et al. [143], we also evaluate our assignments under multiple different proxies. In Section 3.4, we analyze different assignment policies by computing the average of the assigned users' past answer qualities, the average similarity of the users' answers with the questions, and multiple other proxies for match quality. Although these metrics do not directly correspond to downstream answer quality (indeed, we argue in this chapter that a fully predictive model of answer quality is necessary), they give a more complete picture of the trade-offs between assignment approaches.

Other approaches can give us some approximation of causal effects without the costliness of a full A/B test in a live conference, but they all have their own limitations. For example, we might choose to hand-annotate request-expert pairs that are assigned by a proposed assignment but were not observed in the data. However, hand-annotation can only tell us whether we think the pair is a good match a priori; it cannot tell us the downstream quality of the answer that the expert would provide for the request. In this chapter, we intend to measure the effects of assignment policies on the answer quality (in reviewer assignment, this would mean evaluating the impacts of assignments on review quality). Another option would be to employ a simulation of the entire review process using large language models [83]. Although intriguing, the validity of such simulations is still an open research question.

3.3 Predicting Response Quality

In addition to the expert assignment task description laid out in Section 1.2, we also define additional task components related to the modeling of $\mathbf{V}$.

Every expert $g \in M$ who is assigned to the request $i \in N$ responds to the request. These responses are labeled with a *quality label* by one or multiple individuals, perhaps users of our community question answering platform or meta-reviewers in the context of reviewer assignment. To determine a final *quality score* for the expert's response to the request, we map the labels to real values and take the expected value of the labels.

Denote the set of quality labels as $L \doteq \{l_1 \ldots l_k\}$. For every request-expert pair (i, g), there is a ground truth probability distribution over labels $p(l_j|(i, g))$; $p(l_j \mid (i, g))$ denotes the likelihood that an annotator will assign the label l_j to expert g's response to request i. We write $l(i, g)$ to denote the random variable that is the label for (i, g). $l(i, g)$ thus represents a single draw from $p(l_j|(i, g))$, or the label provided by a random annotator for g's response to i. Let function $f : L \to \mathbb{R}$ denote the numerical value of each label. The quality score of expert g's answer to task i is computed as $\mathbf{V}_{(i,g)} = \underset{l \sim p(l_j|(i,g))}{\mathbb{E}}[f(l)]$.

The true label distribution $p^*(l_j|(i, g))$ is often unknown prior to making our assignment of experts to requests. We must estimate a distribution $\hat{p}(l_j|(i, g))$ on existing data, and then apply that learned estimate to make our assignments. We can then optimize for any welfare objective over $\hat{\mathbf{V}} \doteq \underset{l \sim \hat{p}(l_j|(i,g))}{\mathbb{E}}[f(l)]$. In this chapter, we will mainly focus on USW as the welfare function, turning to other objectives in Chapter 4. Recall that the space of all constraint-satisfying allocations is denoted by $\mathcal{Z}$. For this chapter, we will assume that all requests receive the same number of experts; $\underline{\mathbf{k}}^N = \overline{\mathbf{k}}^N = k$. We select the assignment $\mathbf{A} \in \mathcal{Z}$ that maximizes $\text{USW}(\mathbf{A}, \hat{\mathbf{V}})$ over $\mathcal{Z}$. This problem is a totally unimodular linear program, and thus the optimal binary solution can be found in polynomial time by relaxing the space of A from $\mathcal{Z}$ to $\tilde{\mathcal{Z}}$ and solving the corresponding continuous linear program [167]. The optimal continuous solution corresponds to the optimal binary solution.

To learn $\hat{p}$, we estimate the model in our hypothesis class with the minimum *cross-entropy loss*. The cross entropy loss of $\hat{p}$ with respect to a distribution p for a single request-expert pair (i, g) is computed as $\mathbb{H}(p(l|(i, g)), \hat{p}(l|(i, g))) = -\sum_{l \in L} p(l|(i, g)) \log \hat{p}(l|(i, g))$. Given a set of t request-expert pairs T, we can compute the cross-entropy loss of $\hat{p}$ with respect to p over all T as $\frac{1}{t} \sum_{(i,g) \in T} \mathbb{H}(p(l|(i, g)), \hat{p}(l|(i, g)))$.

We will train our model $\hat{p}$ by minimizing the cross-entropy loss on a training set, against the ground truth distribution p^* constructed from known labellings. In the context of expertise assignment, the training data is often not sampled i.i.d. As we will see in Section 3.3.2, in StackExchange we typically construct features for (i, g) pairs from user g's past answers and the text of the question i. Therefore, when learning the model $\hat{p}(l|(i, g))$ for a question i, we use the previously-seen labelled pairs to construct the features for (i, g). This precludes some methods for estimating the generalization error of $\hat{p}$, which rely on the assumption of an i.i.d. data distribution. However, we show that we can still obtain high quality lower bounds on the welfare of our allocation.

For analysis purposes, we will assume there exists a value $\gamma \in \mathbb{R}$ such that the cross-entropy loss cannot exceed γ for any g and i. This can be achieved by considering a smoothed labelling, such that each label l has a minimum probability γ_l under both the true and predicted probability distributions.

Example 3.3.1. In StackExchange, users upvote and downvote the answer provided by user g to question i. Answers receive multiple labels from the set $L = \{\text{Upvote}, \text{Downvote}\}$. We assume that $p(\text{Upvote}|(i, g))$ can be accurately estimated using the empirical conditional distribution over votes. Under this estimate, $p(\text{Upvote}|\text{Vote}, (i, g))$ is computed for g's answer to i as $\frac{\#\text{Upvotes}}{\#\text{Votes}}$. Likewise, $p(\text{Downvote}|\text{Vote}, (i, g))$ can be estimated as $\frac{\#\text{Downvotes}}{\#\text{Votes}}$. One reasonable label value model is to set $f(\text{Upvote}) = 1$ and $f(\text{Downvote}) = -1$, similarly to how answer scores are computed for display on the site. However, upvotes are free and only require 15 reputation to cast, while downvotes are only available to users with at least 125 reputation and cost 1 reputation to cast.[a] The decision-maker could therefore decide to weight downvotes more than upvotes, using $f(\text{Upvote}) = 1$ and $f(\text{Downvote}) = -5$.

Under this second model, if g leaves an answer for i receiving 11 upvotes and 1 downvote, $\mathbf{V}_{(i,g)} = \underset{l \sim p(l_j|(i,g))}{\mathbb{E}}[f(l)] = \frac{11}{12}(1) + \frac{1}{12}(-5) = .5$.

[a] https://stackexchange.com/tour

Example 3.3.2. In reviewer assignment, we might ask meta-reviewers to label the quality of the reviews, using the label set $L = \{$Meets Expectations, Exceeds Expectations, Fails to Meet Expectations$\}$. When there is only one meta-reviewer per paper, the true probability of a label is estimated as 1 if assigned and 0 otherwise. A conservative conference organizer may set $f(\text{Fails to Meet Expectations}) = -10$, $f(\text{Meets Expectations}) = .8$, and $f(\text{Exceeds Expectations}) = 1$. Under this model, $\mathbf{V}_{i,g} \in \{-10, .8, 1\}$ for all $i \in N$ and $g \in M$, since the distribution over labels is deterministic for each review.

3.3.1 Bounding Assignment Quality

Once we have a predictive model for $\hat{p}$, we must output a decision using our predicted values. We solve the problem using the theory of total unimodularity as previously described.

We obtain high probability bounds on the approximation error introduced from optimizing with $\hat{\mathbf{V}}$ instead of $\mathbf{V}^*$. We first evaluate our model $\hat{p}$'s performance on a test set T_{TEST} sampled from distribution $\mathcal{D}_{\text{TEST}}$. Consider a test set T_{TEST} containing t request-expert pairs (i, g). The cross-entropy loss of $\hat{p}$ on the test set is $\xi \doteq \frac{1}{t} \sum_{(i,g) \in T_{\text{TEST}}} \mathbb{H}(p^*(l|(i,g)), \hat{p}(l|(i,g)))$. Using the empirical loss ξ, we construct generalization bounds for any assignment $\mathbf{A}$ using McDiarmid's bounded differences inequality [111] and likelihood weighting [90]. We cannot apply traditional generalization error bounds based on training set loss because our training set often does not consist of i.i.d. samples. Given an assignment $\mathbf{A}$, let $T_{\mathbf{A}}$ denote the set of (i, g) pairs such that $\mathbf{A}_{i,g} = 1$, $T_{\mathbf{A}} \doteq \{(i, g) \in M \times N \mid \mathbf{A}_{i,g} = 1\}$. We assume that these pairs are also random variables drawn from some distribution $\mathcal{D}_{\mathbf{A}}$, where the support of $\mathcal{D}_{\mathbf{A}}$ is $N \times M$. We define a matrix $\Lambda \in \mathbb{R}^{n\times m}$ such that

$$\Lambda_{i,g} \doteq \frac{\Pr_{(\mathbf{i},\mathbf{g})\sim\mathcal{D}_{\text{TEST}}}((\mathbf{i},\mathbf{g}) = (i,g))}{\Pr_{(\mathbf{i},\mathbf{g})\sim\mathcal{D}_{\mathbf{A}}}((\mathbf{i},\mathbf{g}) = (i,g))}.$$

For any distribution p, let $H(p,\hat{p}) \in \mathbb{R}^{n\times m}$ be a matrix such that

$$H(p,\hat{p})_{i,g} = \mathbb{H}(p(l|(i,g)), \hat{p}(l|(i,g)).$$

We are now ready to state the theorem governing the generalization error of our predictor $\hat{p}$. Our generalization error will be in terms of the empirical loss ξ plus an additive error term depending on a confidence parameter δ.

Theorem 3.3.3. *For any $\delta \in (0,1)$, with probability at least $1-\delta$ the true probability distribution p^* satisfies*

$$\frac{1}{kn}\langle \mathbf{A}\odot\Lambda, H(p^*,\hat{p})\rangle_F \leq \xi + \sqrt{\frac{\gamma^2 \ln\frac{1}{\delta}}{2}\left(\frac{1}{t} + \frac{1}{k^2n^2}\sum_{(i,g)\in T_{\mathbf{A}}} \Lambda_{i,g}^2\right)}.$$

Proof. We bound

$$\begin{aligned} X &\doteq \frac{1}{kn}\langle \mathbf{A}\odot\Lambda, H(p^*,\hat{p})\rangle_{\mathrm{F}} - \xi \\ &= \frac{1}{kn}\sum_{(i',g')\in T_{\mathbf{A}}} \Lambda_{i',g'}\mathbb{H}(p^*(l|(i',g')), \hat{p}(l|(i',g'))) - \frac{1}{t}\sum_{(i,g)\in T_{\mathrm{TEST}}} \mathbb{H}(p^*(l|(i,g)), \hat{p}(l|(i,g))). \end{aligned}$$

First, we show that $\mathbb{E}[X] = 0$. Recall that for all $(i',g') \in T_{\mathbf{A}}$, $(i',g') \sim \mathcal{D}_{\mathbf{A}}$, and that for all $(i,g) \in T_{\mathrm{TEST}}$, $(i,g) \sim \mathcal{D}_{\text{TEST}}$. Thus,

$$\mathop{\mathbb{E}}_{(i,g)\sim\mathcal{D}_{\text{TEST}},(i',g')\sim\mathcal{D}_{\mathbf{A}}}[X] = \mathop{\mathbb{E}}_{T_{\mathbf{A}}\sim\mathcal{D}_{\mathbf{A}}}\left[\frac{1}{kn}\sum_{(i',g')\in T_{\mathbf{A}}}\Lambda_{i',g'}\mathbb{H}(p^*(l|(i',g')),\hat{p}(l|(i',g')))\right] \\ - \mathop{\mathbb{E}}_{T_{\text{TEST}}\sim\mathcal{D}_{\text{TEST}}}\left[\frac{1}{t}\sum_{(i,g)\in T_{\text{TEST}}}\mathbb{H}(p^*(l|(i,g)),\hat{p}(l|(i,g)))\right].$$

Let

$$Y \doteq \mathop{\mathbb{E}}_{T_{\mathbf{A}}\sim\mathcal{D}_{\mathbf{A}}}\left[\frac{1}{kn}\sum_{(i',g')\in T_{\mathbf{A}}}\Lambda_{i',g'}\mathbb{H}(p^*(l|(i',g')),\hat{p}(l|(i',g')))\right].$$

We have that,

$$\begin{aligned} Y &= \frac{1}{kn}\sum_{(i',g')\in T_{\mathbf{A}}}\mathop{\mathbb{E}}_{(i',g')\sim\mathcal{D}_{\mathbf{A}}}[\Lambda_{i',g'}\mathbb{H}(p^*(l|(i',g')),\hat{p}(l|(i',g')))] \\ &= \frac{1}{kn}\sum_{(i',g')\in T_{\mathbf{A}}}\mathop{\mathbb{E}}_{(i,g)\sim\mathcal{D}_{\text{TEST}}}[\mathbb{H}(p^*(l|(i,g)),\hat{p}(l|(i,g)))] \\ &= \frac{1}{t}\sum_{(i,g)\in T}\mathop{\mathbb{E}}_{(i,g)\sim\mathcal{D}_{\text{TEST}}}[\mathbb{H}(p^*(l|(i,g)),\hat{p}(l|(i,g)))] \\ &= \mathop{\mathbb{E}}_{T_{\text{TEST}}\sim\mathcal{D}_{\text{TEST}}}\left[\frac{1}{t}\sum_{(i,g)\in T}\mathbb{H}(p^*(l|(i,g)),\hat{p}(l|(i,g)))\right]. \end{aligned}$$

Now that we have shown $\mathbb{E}[X] = 0$, we apply McDiarmid's method of bounded differences to obtain a tail bound on X [111], as X is a function of negatively-dependent random variables. To use this method, we bound the impact of changing any of the random variables in X. We assume that the cross-entropy loss cannot exceed γ for a single sample, so t terms in X contribute at most $\frac{\gamma}{t}$ each, and each of the first kn terms contribute at most $\frac{\gamma\Lambda_{i,g}}{kn}$ for each $(i,g)\in T_{\mathbf{A}}$. The sum of the squared bounded differences is

$$\gamma^2\left(\frac{1}{t}+\frac{1}{k^2n^2}\sum_{(i,g)\in T_{\mathbf{A}}}\Lambda_{i,g}^2\right),$$

which implies finally that

$$\Pr\left(X \geq \sqrt{\frac{\gamma^2 \ln \frac{1}{\delta}}{2}\left(\frac{1}{t} + \frac{1}{k^2 n^2} \sum_{(i,g) \in T_{\mathbf{A}}} \Lambda_{i,g}^2\right)}\right) \leq \delta.$$

□

Once we have a candidate assignment $\mathbf{A}$ that maximizes $\mathrm{USW}(\mathbf{A}, \hat{\mathbf{V}})$, we can apply Theorem 3.3.3 to give high probability lower and upper bounds on $\mathrm{USW}(\mathbf{A}, \mathbf{V}^*)$ by maximizing or minimizing $\mathrm{USW}(\mathbf{A}, \mathbf{V})$ over the high probability region defined for $\mathbf{V}$ by Theorem 3.3.3. This problem is a linear program and can be solved using off-the-shelf solvers.

3.3.2 Empirical Details of Response Quality Model

We experiment on publicly available StackExchange data [154], using the `cs`, `biology`, `chemistry`, and `academia` sites.

We represent users using their profiles and their past answers. For each question, we compute features representing each user's topical affinity for the question and overall answering proficiency. Using these features, we predict labels of answer quality. We construct a train and test set for building our predictive model of answer quality. We then evaluate the model's performance for labeling the test set. We also construct an assignment problem using the set of test questions, and use our predicted distribution over labels to assign a set of users to these questions. We evaluate the assignments using the theoretical bounds developed in Section 3.3.1 and by comparing multiple automated metrics of assignment.

We first filter the questions and answers for quality. We select all questions that have an accepted answer and have a score of at least 3, where the score is computed by adding 1 for each upvote and subtracting 1 for each downvote. We sort the questions by creation time. The first 10% of questions are used to initialize user representations. The next 70% of

Table 3.1: Dates and sizes for all StackExchanges.

			Train			Test		
Topic	First Post	Last Post	Start	# Q's	# A's	Start	# Q's	# A's
`cs`	11/25/08	12/02/23	11/30/12	4831	7045	02/05/19	1182	1666
`biology`	05/26/11	11/30/23	09/30/12	4067	5426	04/24/18	821	1004
`chemistry`	02/17/12	12/01/23	03/16/14	5864	8155	08/06/19	1403	1931
`academia`	10/12/11	03/31/24	10/29/23	8258	20459	07/14/20	2161	5705

questions are used as the training set, and the remaining 20% as the test set. The sizes of all 4 StackExchanges, along with the dates used for train and test, are shown in Table 3.1.

We target the community votes on answers as a measure of answer quality. Because users lose 1 reputation point for every downvote cast [155], we view downvotes as a stronger signal of answer quality than upvotes. We set $f(\text{Upvote}) = 1$ and $f(\text{Downvote}) = -5$.

Altogether, we have 22 features employed by the model. Because we represent users using their previous answers, the features are constructed for each question-answer pair sequentially. Each training point consists of a question i, and a user g that responded to that question at any point in time and has answered at least 1 previous question. Our model incorporates user reputation, number of total views of user's profile, number of upvotes the user has issued, number of downvotes the user has issued, the average time taken to answer questions prior to the current one, mean reciprocal rank for answers posted on previous questions, average view count for questions previously answered, average absolute score for previous answers (upvotes minus downvotes), the number of accepted answers/(total number of answers + c), the average usefulness, relevance, and informativeness of past answers as annotated by the Vicuna-7B large language model [170, 185], and the $0, 5, 10, 25$, and 50^{th} percentiles for $\frac{\text{\# Upvotes}}{\text{\# Votes}}$ on all previous answers. We refer to the latter five features as the *past* $p(\text{Upvote})$ *distribution* as a shorthand, though for each answer $\frac{\text{\# Upvotes}}{\text{\# Votes}}$ is only an estimate of the true conditional distribution of $p(\text{Upvote}|\text{Vote}, (i, g))$.

We also score all question-answer pairs for usefulness, informativeness, and relevance using Vicuna, released August 2023. These three criteria were found to be the most highly correlating criteria with overall answer quality in a survey of experts on community question answering [186]. Using these annotations, for each question-answer pair we compute the mean usefulness, relevance, and informativeness scores on the user's past answers. These scores are used as input features for our predictive model, but we also apply them as part of our automated assignment evaluation.

For all question-answer pairs, we set the system prompt for the Vicuna model as (text written inside square brackets is filled in programmatically):

```
I am going to provide you with a question—answer pair from the [
    Topic Name] StackExchange. Please annotate the informativeness,
     relevance, and usefulness of the answer. Your response should
    rate each of these three aspects on a scale from 1—5, with 1
    being the least and 5 being the most. Please structure your
    response by outputting the informativeness, then the relevance,
     and then the usefulness, one per line. Please add an
    additional explanation of your ratings. Informativeness asks
    Does this answer provide enough information for the question?
    Relevance asks Is this answer relevant to the question?
    Usefulness asks Is this answer useful or helpful to address the
     question? Use this template for your output:
     Informativeness: <Rating>
     Relevance: <Rating>
     Usefulness: <Rating>
     Explanation: <Additional Explanation>
```

We then prompt the model as the user:

```
Question:
Title: [Question Title]

Body: [Question Body]

Answer:
[Answer Body]
```

We then collect and parse the response from the model to obtain the LLM annotations of the usefulness, relevance, and informativeness of the answer.

We also include pairwise features of the question and the user, to identify content-based similarity. We collect the set of all of the user's previous answers. We represent each user as a weighted bag of keyword tags on the questions they have previously answered, and we represent the current question as a bag of tags. We can then compute the keyword similarity score as the product of the number of matching tags times the total count of the matching tags (following [169]). We also embed all question titles, question bodies, and answer bodies using the SentenceTransformer `multi-qa-mpnet-base-cos-v1` model [79, 135]. We then include the mean and maximum cosine similarity between the current question title and the titles of questions the user previously answered, as well as the mean and maximum cosine similarity between the current question's body and the bodies of the user's previous answers.

We train a logistic regression model [146] on the training set to minimize cross-entropy loss against the true distribution over labels for each user-question pair. For each pair (i, g), we compute the empirical conditional distribution $p(\text{Upvote}|\text{Vote}, (i, g)) \doteq \frac{\text{\# Upvotes}}{\text{\# Votes}}$ and $p(\text{Downvote}|\text{Vote}, (i, g)) \doteq \frac{\text{\# Downvotes}}{\text{\# Votes}}$ as the target distribution.

We train two baseline models that use less information about the users' history compared to the full logistic regression model. In some domains, like reviewer assignment, the user's history may be considered private. Ideally, we could obtain strong predictive performance even without storing this private information. The *Logistic Regression (Badges)* model uses a count vector of the badges awarded to the user instead of all features except for the user-answer textual cosine similarity measures, and the tag-based similarity measure. The *User Embeddings* model uses randomly-initialized user embeddings in place of all features except for the user-answer textual similarity measures, and the tag-based similarity measure. This model is trained using a 2-layer feed-forward neural network with sigmoid activations. It is trained over 400 epochs using the Adam optimizer [88].

Label Classification Performance After training the logistic regression model, we evaluate the cross-entropy loss on the test set. We also compare against three baseline approaches. The *Constant* model predicts the average empirical $p(\text{Upvote}|\text{Vote}, (i, g))$ over the training set for each sample in the test set. The *Constant (per User)* model predicts the user's empirical average $p(\text{Upvote}|\text{Vote}, (i, g))$ up to the point in time of the question. The *Similarity and Reputation* model predicts a mixture of the user's reputation and the cosine similarity between the user's past answers and the question text. We construct this baseline to mirror standard expert assignment approaches, where a linear combination of topical match scores and user characteristics are used for assignment (as in reviewer assignment, which typically employs keyword match scores, bids, and document-based similarity scores [100]). We set the value for each (i, g) pair to be $\mathbf{V}_{i,g} = \lambda\frac{x_1(g)}{\max_{g\in M} x_1(g)} + (1-\lambda)\frac{x_2(i,g)}{\max_{(i,g)\in(N\times M)} x_2(i,g)}$, where $x_1(g)$ is the reputation of g and $x_2(i, g)$ is the cosine similarity score between the answer bodies for g and i. We set $\lambda = .5$ based on initial experiments varying λ over $[0, 1]$ in increments of 0.1. Note that order statistics of the user's empirical conditional probability of receiving an upvote are also included as features in our logistic regression model.

Table 3.2: Predictive performance of each model on the test set for `cs`.

Model	XE (↓)	τ (↑)	ρ (↑)	P@100 (↑)	AUROC (↑)
Constant	.0890	–	–	.94	.5000
Constant (per User)	.1028	.0850	.0866	.94	.5308
Similarity and Reputation	–	.0828	.1029	**.98**	.6221
Logistic Regression (Badges)	.0922	.1001	.1245	.95	.6461
User Embeddings	.1054	.1331	.1653	**.98**	.6964
Logistic Regression (All)	**.0802**	**.1472**	**.1824**	.97	**.7155**

Table 3.3: Predictive performance of each model on the test set for `biology`.

Model	XE (↓)	τ (↑)	ρ (↑)	P@100 (↑)	AUROC (↑)
Constant	.1470	–	–	.88	.5000
Constant (per User)	.1688	.0381	.0399	.87	.5150
Similarity and Reputation	–	.0759	.0962	.85	.5771
Logistic Regression (Badges)	.1531	.0987	.1253	**.9**	.6036
User Embeddings	.1613	.1046	.1323	**.9**	.6070
Logistic Regression (All)	**.1363**	**.1225**	**.1547**	**.9**	**.6221**

Tables 3.2 to 3.5 report, for all 4 StackExchanges, the cross-entropy loss, precision at 100 (P@100), and the area under the receiver operating characteristic curve (AUROC) of the predictive models. P@100 and AUROC are computed assuming that a positive example is one with no downvotes, and a negative example is one with at least one downvote. We also compute Kendall's τ and Spearman's ρ statistics between the ranking produced by each model and the true ranking over all test examples by $p(\text{Upvote}|\text{Vote}, (i, g))$. All τ and ρ statistics are statistically significant with a p-value of less than .001. Our trained model outperforms all the baselines, but the predictive models that use only badges or user embeddings are close to the performance of the full predictive model.

Table 3.4: Predictive performance of each model on the test set for `chemistry`.

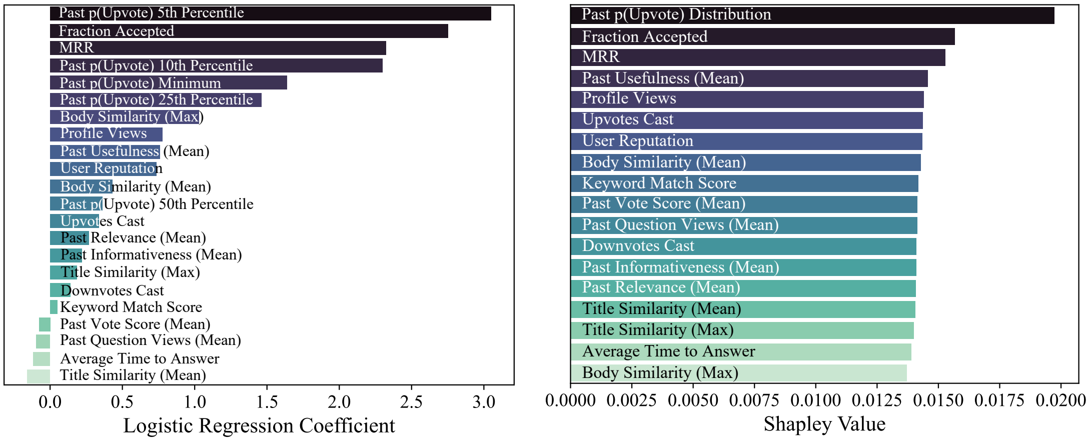
Model	XE (↓)	τ (↑)	ρ (↑)	P@100 (↑)	AUROC (↑)
Constant	.1408	–	–	.86	.5000
Constant (per User)	.1582	.1319	.1364	.88	.5545
Similarity and Reputation	–	.1246	.1563	**.97**	.6455
Logistic Regression (Badges)	.1624	.0906	.1136	.94	.6025
User Embeddings	.1381	.1078	.1351	**.97**	.6250
Logistic Regression (All)	**.1177**	**.1767**	**.2206**	.96	**.7056**

Table 3.5: Predictive performance of each model on the test set for `academia`.

Model	XE (↓)	τ (↑)	ρ (↑)	P@100 (↑)	AUROC (↑)
Constant	**.2403**	–	–	.77	.5000
Constant (per User)	.2848	.0400	.0437	.77	.5089
Similarity and Reputation	–	-.0004	-.0003	.69	.4878
Logistic Regression (Badges)	.3255	.0480	.0640	.67	.5292
User Embeddings	.2860	**.1199**	**.1546**	.8	**.5969**
Logistic Regression (All)	**.2403**	.1060	.1384	**.83**	.5708

Feature Importance In this section, we study the importance of different features for predicting $p(\text{Upvote}|\text{Vote}, (i, g))$ using our full-featured logistic regression model. We only study feature importance on the `cs` StackExchange, since Shapley value computation is very expensive. We investigate the coefficients of the logistic regression model in Figure 3.2a. Figure 3.2b shows the feature importance for each feature in our model using the exact computation of the Shapley value [150]. The Shapley value of a feature x measures the average marginal decrease in test-set cross entropy loss when training the logistic regression model using a set of features $S \cup \{x\}$ compared to using the set of features $S \setminus \{x\}$.

The top three features by both measures are measures of quality for the user's previous answers. Distributional measures of $p(\text{Upvote}|\text{Vote}, (i, g))$ over the user's past answers are



(a) Logistic regression model coefficients.

(b) Shapley values for features, measured as the average marginal contribution of the feature to decreasing cross entropy loss on the test set.

Figure 3.2: Logistic regression model coefficients and Shapley values on the test set for `cs`.

incredibly important for predicting future $p(\text{Upvote}|\text{Vote}, (i, g))$. In addition, the ratio of accepted answers written by the user to total answers written by the user, as well as the mean reciprocal rank of answers written by the user are both very important.

All features contribute meaningfully to reducing cross-entropy loss on the test set. This finding should encourage decision makers to collect and leverage as much information as is available in constructing predictive models of performance.

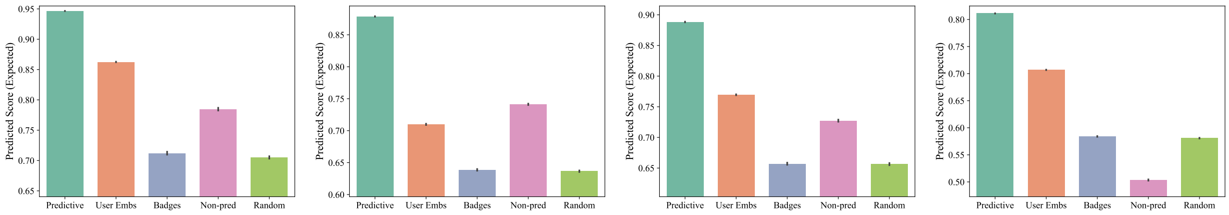
One likely explanation for the relative unimportance of topic-based features is the (extreme) sampling bias in StackExchange data. Our (i, g) pairs consist only of answers that were actually submitted on the website. If a user g does not have the expertise or interest to answer question i, he or she simply will not answer that question. Topical similarity is likely more useful for predicting *whether* a user g would naturally answer question i. These measures are less useful in determining the probability that user g provides a high quality answer to question i, conditioned on the fact that they chose to answer the question.

3.4 Making and Evaluating Assignments using the Predictive Model

We also use the test set to evaluate the overall assignment quality when assigning for predicted answer quality. We assume that all questions in the test set are received simultaneously, immediately following the end of the time period of the training set. Thus, each user's past answers (used in computing user and pairwise user-task features) are limited to only the answers to the first 80% of questions by time of posting. This difference from the training set allows us to simulate testing on multiple questions without interdependence between questions, and shows the trade-offs required when assigning users with limited resources to many questions.

Overall, to evaluate assignments we include $n = 1{,}402$ questions (our set of requests N) and the $m = 220$ users who answered any of those questions and have answered at least one question before (our set of experts M). For robustness, we take 1,000 samples of 60% of the questions and 60% of the users, and report the distributions of all evaluation metrics across all 1,000 runs. For each assignment, we evaluate 12 metrics: $\hat{p}(\text{Upvote}|\text{Vote}, (i, g))$ under our predictive model, worst-case $p(\text{Upvote}|\text{Vote}, (i, g))$ according to Theorem 3.3.3 with $\delta = .1$, 5^{th} percentile and median of user's historical $p(\text{Upvote}|\text{Vote}, (i, g))$, number of user-question pairs that are observed in reality, average cosine similarity between user's past answers and question body, average keyword matching score, average assigned reputation score for assigned users, true $p(\text{Upvote}|\text{Vote}, (i, g))$ for recovered pairs, and the average usefulness, relevance, and informativeness for assigned users' past answers. These metrics capture a wide range of automated measures for assignment quality, including topical similarity between the users and questions, and the users' propensities to leave satisfying answers.

We compare the assignment made using our predictive model to baseline assignments made using the scores from the *Similarity and Reputation* baseline. In addition, we also compute 100 random values for each (i, g) pair drawn from the uniform distribution $\mathcal{U}_{[0,1]}$.



(a) CS (b) Biology (c) Chemistry (d) Academia

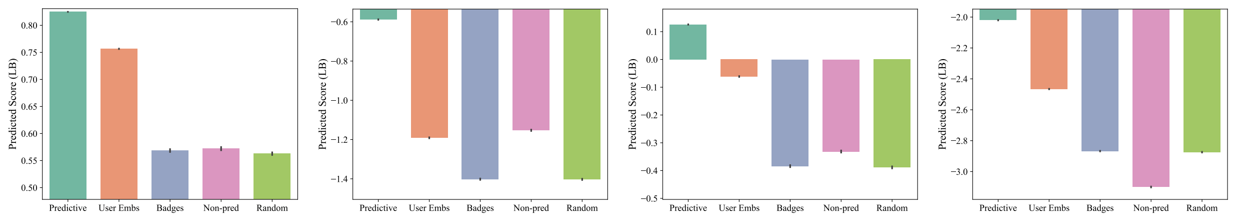
Figure 3.3: Comparison of all models on $\hat{p}(\text{Upvote}|\text{Vote})$, on all 4 StackExchange websites.

We set the number of experts per request i to $\underline{\mathbf{k}}_i^N = \overline{\mathbf{k}}_i^N = k = 2$, and the number of requests per expert g to $\underline{\mathbf{k}}_g^M = 0$ and $\overline{\mathbf{k}}_g^M = 26$ (twice the smallest integer such that $m\overline{\mathbf{k}}_g^M \geq n\underline{\mathbf{k}}_i^N$).

To compute the worst-case bounds for the predictive assignment model (according to Theorem 3.3.3), we set $\delta = .1$. We estimate the cross entropy loss on the 1,666 labeled question-answer pairs in the test set. We estimate the distribution $\mathcal{D}_{\text{TEST}}$ over the test set by fitting a 2-component principal component analysis on the test data and then applying kernel density estimation on the transformed features. We also estimate the distribution $\mathcal{D}_{\mathbf{A}}$ over the pairs in $T_{\mathbf{A}}$ for each assignment A using the same procedure.

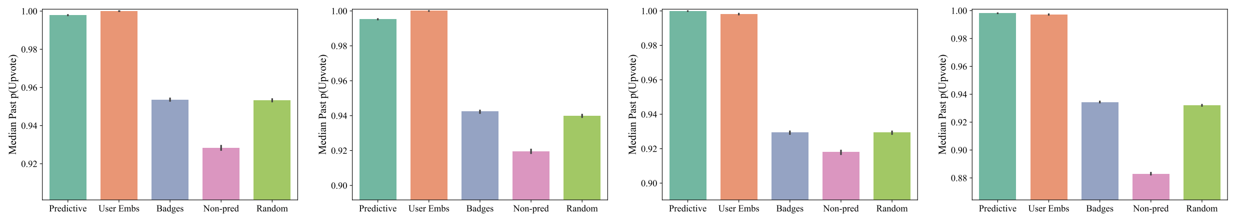
In Figures 3.3 to 3.9, we demonstrate the value of 6 metrics of interest over 1,000 repeated experiments for the predictive assignments, the *Similarity and Reputation* baseline with $\lambda = .5$, and random assignment. Each figure includes results for a single metric for all 4 StackExchange websites. For each of the $1,000$ experiments, for each assignment method, we compute the average of the value of the metric over all pairs assigned by that method. We then report the average value and error bars over a 95% confidence interval for the average (over runs) of the average metric value (over assigned pairs).

Figure 3.3 shows the average value of the predicted score, estimated using our logistic regression model with all the features. It is not surprising that the assignment using the predictive model is optimal on this measure, but the degree of suboptimality of the other



(a) CS (b) Biology (c) Chemistry (d) Academia

Figure 3.4: Comparison of all models on the statistical lower bound on the true $p(\text{Upvote}|\text{Vote})$ at 90% confidence, on all 4 StackExchange websites.

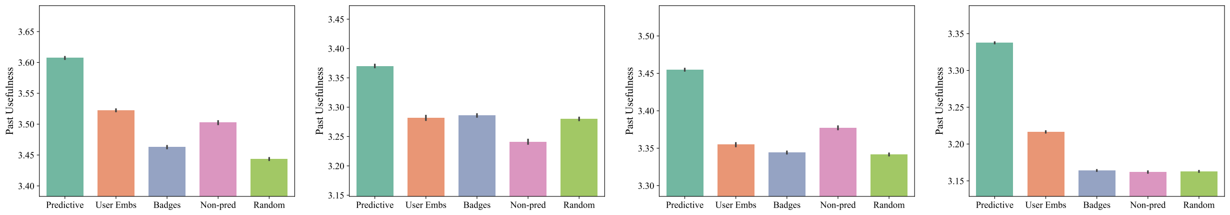


(a) CS (b) Biology (c) Chemistry (d) Academia

Figure 3.5: Comparison of all models on median of users' past $p(\text{Upvote}|\text{Vote})$, on all 4 StackExchange websites.

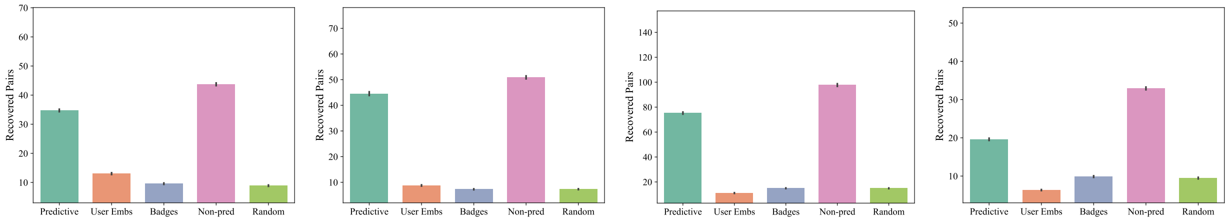
assignment methods is quite surprising, with the non-predictive baseline showing between $15 - 30\%$ decrease in predicted performance. In Figure 3.4, we show the the lower bound computed using Theorem 3.3.3. The patterns in these results largely reflect the patterns of Figure 3.3. Notably, the non-predictive model underperforms random assignment on the `academia` StackExchange.

Figures 3.5 and 3.6 show the performance of each assignment method over two important metrics of historical user performance. Figure 3.5 shows that the full predictive model and the trained user embedding model strongly outperform the other approaches on the average of the user's median past $p(\text{Upvote}|\text{Vote})$. In addition, the non-predictive baseline actually under-performs random assignment on this metric. Figure 3.6 shows the performance



(a) CS (b) Biology (c) Chemistry (d) Academia

Figure 3.6: Comparison of all models on average value of users' past "Usefulness" score, on all 4 StackExchange websites.

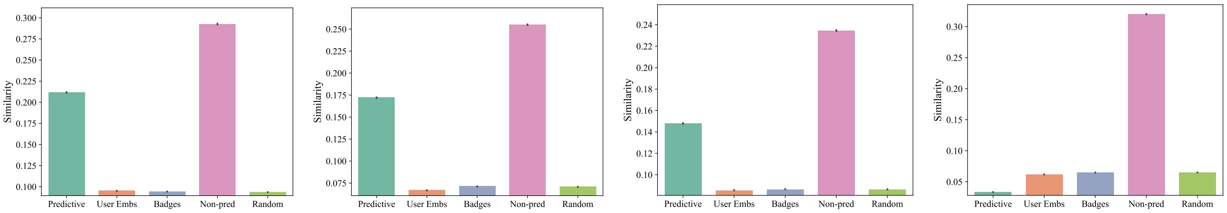


(a) CS (b) Biology (c) Chemistry (d) Academia

Figure 3.7: Comparison of all models on number of assigned pairs that were observed in reality, on all 4 StackExchange websites.

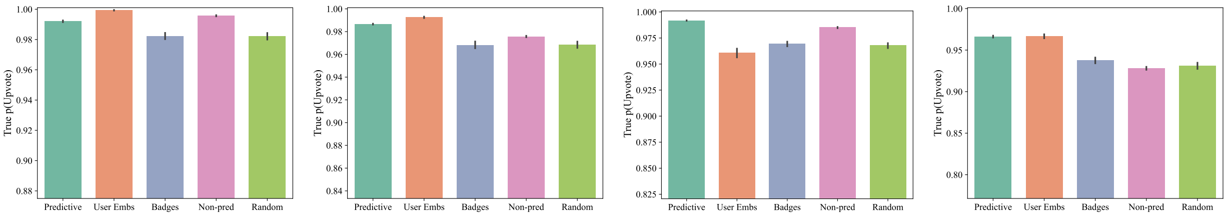
measured as the average of the LLM-annotated "Usefulness" scores over each assigned user's past answers. This metric is also a measure of historical performance of the assigned users. Again, we see that the full predictive model outperforms all models, and the user embedding model does second-best, except in the `chemistry` StackExchange. Figure 3.2 showed that under both feature importance measures we considered, distributional measures of past user performance were the most important features for our predictive model. The results in Figures 3.5 and 3.6 further corroborate this claim; they indicate that the predictive model assigns users with significantly higher historical performance measures.

The final set of plots we analyze, Figures 3.7 and 3.9, demonstrate the two "ground truth" metrics: number of observed pairs recovered (Figure 3.7), and observed $p(\text{Upvote}|\text{Vote})$ for



(a) CS (b) Biology (c) Chemistry (d) Academia

Figure 3.8: Comparison of all models on average cosine similarity of user's past answers to the question body, on all 4 StackExchange websites.



(a) CS (b) Biology (c) Chemistry (d) Academia

Figure 3.9: Comparison of all models on the true $p(\text{Upvote}|\text{Vote})$ estimated on recovered pairs, on all 4 StackExchange websites.

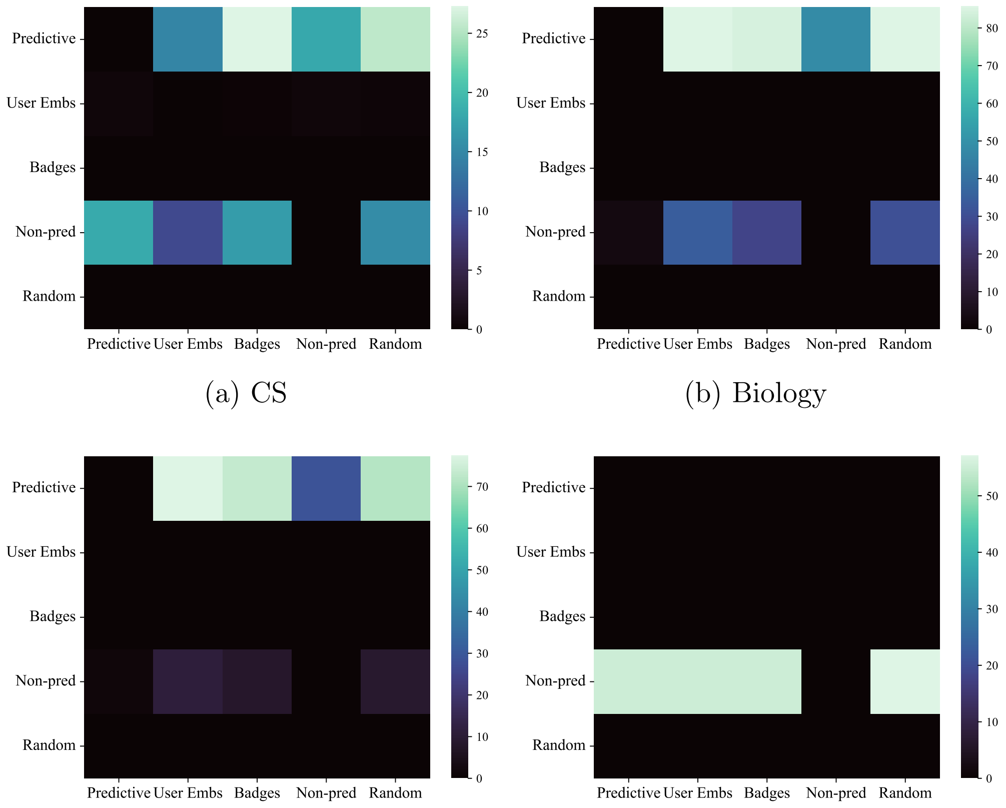
those pairs. Both metrics should be considered somewhat skeptically; although the users who actually answer a question are often qualified to do so, observing a user-question pair is neither necessary nor sufficient to indicate a good match. In addition, the constraints imposed on assignments limit the number of observed pairs we can recover, since a small number of users answer a large number of questions. The $p(\text{Upvote}|\text{Vote})$ over observed pairs is a useful measure, but the number of observed pairs is only a small fraction of the number of assigned pairs. Nonetheless, we see some interesting trends in these plots. Figure 3.7 shows that the non-predictive baseline outperforms the predictive model on recovering observed pairs, while the predictive model strongly outperforms the other 3 models. This likely derives from the distribution shift between the training and test sets. Our models are trained on the observed user-question pairs, which generally have higher content-based similarity than independently selected users and questions. Thus our predictive models likely do not place as much importance on content-based similarity as the baseline, though clearly some content-based similarity is important in making assignments. The plots of average cosine similarity between the assigned user's past answers and the question body (Figure 3.8) shows a similar trend, further supporting this theory. Figure 3.9 shows the true value of the $p(\text{Upvote}|\text{Vote})$ over the pairs which have annotations. Our predictive model slightly underperforms on this metric on the `cs` StackExchange, but shows consistently high performance across all datasets.

Across all metrics of interest, the `academia` StackExchange shows different trends than the other 3 datasets. Specifically, the non-predictive baseline is much worse relative to the predictive model on this dataset than others, when measured by $\hat{p}(\text{Upvote}|\text{Vote})$, true $p(\text{Upvote}|\text{Vote})$ on observed pairs, and historical user performance. We also see the predictive model has a much lower average cosine similarity score on this dataset. These results can have a huge bearing on expert assignment systems; moving forward, our predictive models of fit should be designed to reweight feature importances differently depending on the subject matter. Conferences that have many different types of contributions may be most impacted; a position paper, a paper introducing a new benchmark, a theoretical paper, and a paper introducing a novel machine learning model may all require different feature weights when assigning reviewers. Although our predictive model does not account for this variation in feature importance across subject areas, we suggest this as a major direction for future work in expertise modeling.

We also apply the analysis tools presented by Khan et al. [87], Saveski et al. [143]. Specifically, for each assignment, we compute the upper and lower bounds for average $p(\text{Upvote}|\text{Vote})$ over all pairs (observed and unobserved) under the assumption that $p(\text{Upvote}|\text{Vote})$ is a monotonically non-decreasing function of the answer-question similarity score, the keyphrase matching score, and user reputation. Saveski et al. [143] use probabilistic reviewer assignments and reweight quality scores for observed and imputed pairs based on the marginal probability of assignment. In our case, both the observed and proposed assignments are deterministic, so we apply their bounds under the trivial deterministic distributions. After computing the upper and lower bounds, we determine for all 1000 runs when the lower bound for some assignment algorithm exceeds the upper bound for another assignment algorithm. In Figure 3.10, we show the percentage of runs in which each algorithm dominates each other algorithm in this way. Overall, the full predictive model and the baseline model with $\lambda = .5$ are the only two models to dominate other approaches a non-trivial number of times. We see that the predictive model dominates the other approaches more frequently than the baselines on the `cs`, `biology`, and `chemistry` StackExchanges. However, the baseline with $\lambda = .5$ outperforms the predictive model on the `academia` StackExchange according to this metric. We performed similar experiments using the Lipschitz continuity assumption from Khan et al. [87], Saveski et al. [143], with multiple choices for the Lipschitz constant selected using the aforementioned works' methodology. Although these bounds were often tighter than the monotonicity-based bound, we found that the bound rarely distinguished between any two approaches.

Appendix A contains the analysis of several additional metrics, including the 5^{th} percentile of historical $p(\text{Upvote}|\text{Vote})$ for assigned users, keyword matching score, user reputation, and LLM-annotated "Relevance" and "Informativeness" of assigned users' past answers.

Overall, we find that the assignments made using the predicted answer scores strongly and robustly outperform most baselines on expected and worst-case predictive score, order statistics of user's past probability of upvote, and assigned users' average usefulness, informativeness, and relevance over past answers. However, the baseline using trained user embeddings performs very well on the objectives that measure user behavior without considering content similarity.



(a) CS

(b) Biology

(c) Chemistry

(d) Academia

Figure 3.10: Percentage of time that the assignment approach on the y-axis has a higher lower bound than the upper bound for the assignment approach on the x-axis.

3.4.1 Metric Correlation Analysis

To further understand the trade-offs between evaluation metrics, we rank the assignment approaches using each metric for each of our 1,000 experiment repetitions. We then study the correlation between rankings *over assignments* produced by each metric. Correlations and p-values for all four StackExchanges are displayed in Appendix A.1, along with a more detailed explanation of the procedure for calculating these correlations.

The rankings produced under $\hat{p}$ are generally highly correlated with the rankings produced by order statistics of the user's past $p(\text{Upvote})$, user reputation, and the average LLM-annotated usefulness, relevance, and informativeness. This further bolsters our finding in the previous section that these historical features are highly informative for predicting the future probability of an upvote. The metric producing rankings most correlated with the rankings produced by the *Recovered Pairs* metric (the number of observed user-question pairs recovered by the assignment) is the *Keyword Match Score* metric, suggesting again that textual similarity is predictive of whether a user would naturally answer a question (but this does not mean that this feature is predictive of their response quality given that they chose to answer this question). Finally, we see that the similarity score produces rankings that are generally negatively correlated wth rankings produced by measures of user competence (reputation with correlation $-.7$, and order statistics of past probability of upvote with correlation $-.7$ and -1.0). This dynamic suggests that perhaps assignments yielding content similarity may sacrifice user competence. StackExchange uses tags to recommend questions to users, and conference management platforms like Microsoft CMT compute keyword-based similarity measures, so this dynamic is incredibly worthy of further study.

3.5 Discussion

An ideal evaluation setting would make multiple assignments, have the assigned people write a response, and see how many upvotes they receive. It is not currently possible to run this experiment through StackExchange, and such an experiment would be very expensive in reviewer assignment. However, our 11 automated metrics demonstrate that the predictive model incorporates multiple elements of user suitability in a satisfying way.

Our approach requires access to historic behavior for every user. While this is routine for StackExchange, this information is harder to access in privacy-sensitive areas like peer review. Reviewers may not be comfortable with sharing review performance history with systems like CMT or OpenReview without additional assurances, and this persistent state would require additional consent. However, many conferences have persistent organizing teams that retain access to review data from prior years, and with proper consent these conferences could simply make use of this already accessible information. Multiple research studies have expressed concerns that high-quality data about peer review is incredibly difficult to obtain [42, 113]. We hope that the current work encourages further study of the connections between input and output measures in other areas of expert assignment, and that conferences in particular will be encouraged to cooperate further with researchers to understand how decisions made during reviewer assignment impact downstream metrics of review quality.

3.6 Conclusion

We find defining metrics of interest and then optimizing for the predicted values of those metrics is more effective than optimizing for variables that are available prior to assignment. We give probabilistic bounds on the quality of such assignments, ensuring that the metric is optimized with high probability. Historic measures of answer quality prove to be the most important predictors of future answer quality; this finding can have important implications for expert recommendation in StackExchange as well as reviewer assignment for peer review. Although different measures of assignment quality can result in different results, the predictive model-based assignments outperform baseline approaches across a wide range of metrics. The correlations between the rankings produced by different metrics reveal important connections between these competing or complementary objectives; notably, we find some indication that assigning for content similarity may conflict with assigning for user competency. These trade-offs can be incorporated directly into a predictive model, such as the one proposed in the current study.

CHAPTER 4

ROBUST EXPERT ASSIGNMENT UNDER UNCERTAINTY

In Chapters 2 and 3, we have largely treated the valuation matrix $\mathbf{V} \in \mathbb{R}^{n \times m}$ as a single, fixed value. However, in the real world most methods for estimating $\mathbf{V}$ are noisy. Chapter 3 shows that predictive models of answer quality in StackExchange sometimes outperform linear combinations of simple features, but neither model can predict the future perfectly. In reviewer assignment, reviewers can only bid on a small number of papers, and textual similarity models and subject-area matching are inherently noisy estimators.

In the remainder of this thesis, we develop principled methods of handling uncertainty around the valuations $\mathbf{V}$. The methods laid out in this chapter optimize utilitarian (USW) and egalitarian welfare (GESW). Valuations $\mathbf{V}$ are not known in advance, but are instead observed after realizing the allocation. Optimizing over estimates of $\mathbf{V}$ requires trading-off between mean valuations and their predictive variances. We discuss these trade-offs under two paradigms for preference modeling – in the *stochastic optimization* regime, the optimizer has access to a probability distribution $\mathcal{D}_{\mathbf{V}}$ over $\mathbf{V}$, and in the *robust optimization* regime they have access to an *uncertainty set* $\mathcal{V}$ containing the true valuations with high probability. We discuss utilitarian and egalitarian based objectives, and we explore how to optimize for them under stochastic and robust paradigms. The approaches enable scalable constrained resource allocation under uncertainty for many different objectives and preference models.

4.1 The Importance of Considering Uncertainty

We briefly outline the case for robustly incorporating uncertainty during optimization, using our primary expert assignment problem of interest, reviewer assignment for peer-reviewed conferences. Uncertainty in affinity score computation is a major source of error in reviewer assignment [100]. When we assign a reviewer to a paper, we are interested in ensuring the *quality of the future review*, which is fundamentally noisy. Because of this unpredictability, conferences typically construct affinity scores that reflect reviewer expertise and interest via four main sources of information. These sources include (a) *subject-area matching* (SAM) scores or keyword-based matching, where reviewer-provided areas of expertise are compared against keywords submitted by paper authors, (b) *textual similarity scores*, often implemented by the well-known Toronto Paper Matching System (TPMS) [38] or ACL scores [117], (c) *bidding*, where reviewers express their explicit ability and desire to review papers, and finally (d) *recommendations*, through which program committee members may suggest reviewers for papers. The overall affinity scores are typically computed as a linear combination of these four scores [121, 138]. Recent conferences such as AAAI 2021 took a similar approach, linearly combining TPMS scores, ACL scores, and SAM scores, and raising the sum to some power based on the reviewer bids [100].

Each of these common affinity score components can be missing or inaccurate. State-of-the-art document similarity measures disagree with expert judgments 9% of the time for easy examples and 45% of the time for hard examples [120, 159], and nearly 40% of TPMS scores were completely missing in AAAI 2021 [100]. Although the AAAI 2021 organizers do not explain why so many TPMS scores are missing, missing scores occur for several reasons, including reviewers opting out of the system or providing insufficient or empty publication records. Between 5% and 15% of papers in major AI conferences receive fewer than three positive bids, but there is evidence that many missing bids would be positive if collected [60, 112, 141]. Out of $2,425$ papers submitted to NeurIPS 2016, 816 or about

1/3 of all papers had fewer than 6 positive bids. Since papers required 6 reviewers at this conference, this is a shockingly high rate. Although no systematic study has been performed on keyword-based similarity scores, keyword matching accuracy depends on authors and reviewers using consistent terminology, and subtleties are invariably lost in the process. Even reviewers directly suggested by knowledgeable editors or the paper authors have been shown to perform surprisingly poorly on average, as measured by third-party annotators via the Review Quality Index [144, 171], showing that recommendations can be noisy as well. Finally, experts often disagree when evaluating the quality of peer reviews, showing that in addition to the multiple sources of input uncertainty there is also uncertainty in the evaluation of outcomes [68].

The above discussion helps us understand why we might consider optimizing with uncertainty in mind. The following example demonstrates numerically how much we lose when we do not account for uncertainty.

Example 4.1.1 (The Importance of Considering Uncertainty)**.** Consider a simple two-request, two-expert instance, where each request needs exactly one expert, and either likes (value 1) or dislikes (value 0) each expert. Request values over experts are Bernoulli random variables, where $\Pr[\mathbf{V}_{1,1} = 1] = 0.8, \Pr[\mathbf{V}_{1,2} = 1] = 0.9, \Pr[\mathbf{V}_{2,1} = 1] = 0.5$, and $\Pr[\mathbf{V}_{2,2} = 1] = 0.8$. If we maximize the sum of values over the expected value of each variable, we would assign g_1 to i_1 and g_2 to i_2, for a total expected value of 1.6. However, consider instead the objective of *Conditional Value at Risk*, which is the conditional expectation over the left tail of the distribution at a certain percentile. When we make the expectation-maximizing assignment, then $\Pr[\text{USW}(\mathbf{A}, \mathbf{V}) = 0] = 0.04$ and $\Pr[\text{USW}(\mathbf{A}, \mathbf{V}) = 1] = 0.32$. However, if we assign g_2 to agent i_1 and item g_1 to agent i_2, we have that $\Pr[\text{USW}(\mathbf{A}, \mathbf{V}) = 0] = 0.05$ and $\Pr[\text{USW}(\mathbf{A}, \mathbf{V}) = 1] = 0.5$. This means that the conditional expectation of welfare at the 30th percentile is higher if we assign g_2 to i_1 and g_1 to i_2 (it is .32 in the first case and .5 in the second case). If we want to retain welfare in the face of uncertainty, we might well choose to maximize this quantity rather than the expectation of the welfare.

4.2 Our Contributions

We study the broad problem of fair and efficient expert assignment under preference uncertainty. Specifically, we develop methods to efficiently optimize utilitarian and egalitarian objectives using the robust approach [15, 18, 70] and CVaR approach [140].

For robust optimization, we construct an uncertainty set containing the true preferences with high probability, then maximize the minimum welfare over the uncertainty set. This model is appropriate when building a predictor with statistical error bounds, but without making any assumptions on the full probability distribution over preferences. Uncertainty sets generalize probability distributions — while it is possible to construct an uncertainty set from a probability distribution, non-Bayesian models will frequently *not* specify full probability distributions. In these cases, worst-case guarantees over an uncertainty set are quite natural.

For utilitarian and egalitarian welfare functions, we robustly maximize welfare over such uncertainty sets. We provide numerous examples of uncertainty sets, starting with axis-aligned, hyperrectangular uncertainty sets, and spherical and ellipsoidal uncertainty sets. Theorems 4.4.7 and 4.4.8 offer detailed end-to-end examples of construction of ellipsoidal uncertainty sets using bounds on the square error of an estimator from partially-observed or historical data. We also present a calculus of uncertainty sets, enabling construction of complex and highly informative uncertainty sets from simple components. Our results are agnostic to the valuation model; decision-makers can define valuations arbitrarily, so long as they can be estimated for all request-expert pairs, and sampled for some pairs.

We show that maximizing the minimum USW is NP-hard over convex uncertainty sets (Theorem 4.4.11). We also present an approximation algorithm that applies to any convex uncertainty set (under any welfare function), as long as worst-case welfare can be efficiently computed (Section 4.4.2). The algorithm, *adversarial projected sub-gradient ascent*, applies randomized rounding methods to a *convex relaxation* of the discrete optimization problem, and we analyze both the *optimization error* due to convex programming methods and randomized rounding, and the *maximin error* due to operating with an uncertainty set, rather than known valuation $\mathbf{V}$. We give bounds on the true welfare relative to the maximin welfare solution (Proposition 4.4.1), and explore the integrality gap under USW (Proposition 4.4.2).

In addition to our general approach, we discuss specific cases where a more efficient solution exists. When the uncertainty sets are linear we efficiently compute the exact optimal maximin allocations for both utilitarian and egalitarian welfare in polynomial time (Corollary 4.4.16 and Corollary 4.4.20). Under ellipsoidal uncertainty sets, we apply an *iterated quadratic programming* approach for utilitarian and egalitarian welfare (Corollary 4.4.17 and Corollary 4.4.21).

When the market designer can construct a full probability distribution over preferences, we consider *stochastic optimization* using the robustness concept of *Conditional Value at Risk*, or CVaR [140]. This approach selects an allocation that maximizes the conditional expectation of welfare over the left tail of the distribution. We largely approach CVaR objectives using sampling, then solving the resulting linear program (LP). Section 4.5 deals with CVaR of welfare.

Our results are summarized in Table 4.1. Robust optimization under a wide range of convex uncertainty sets and welfare functions can be solved using adversarial projected sub-gradient ascent. Table 4.1 outlines more efficient approaches that work in certain settings.

We complement our theoretical results with an empirical exploration of uncertainty set/probability distribution construction and robust/stochastic optimization. We first show

Table 4.1: Summary of optimization algorithms for efficiently computing utilitarian and egalitarian welfare under different robustness concepts. Green highlights indicate problems which require solving a single linear program (low difficulty). Yellow highlights indicate solving a small number of linear or quadratic programs (medium difficulty). Red highlights indicate solving numerous quadratic programs or arbitrary concave programs.

	Robustness Concept				
	Robust			**CVaR**	
	Linear	**One Ellipsoid**	**ℓ Ellipsoids**	**Any (approx.)**	**Gaussian**
Utilitarian	LP Reduction (Coro. 4.4.16)	Iterated QP (Coro. 4.4.17)	Projected SGA (Prop. 4.4.15)	Sampling+LP (Prop. 4.5.1)	Projected GA (Prop. 4.5.6)
Egalitarian	LP Reduction (Coro. 4.4.20)	Iterated QP (Coro. 4.4.21)	Projected SGA (Prop. 4.4.19)	Sampling+LP (Prop. 4.5.7)	
Monotone, Concave	Adversarial Projected SGA (Sec. 4.4.2)			Sampling + Concave Program (Sec. 4.5)	

a simulation demonstrating the negative consequences faced by an uncertainty-unaware approach. In this simulation, we create a scenario where we know the true $\mathbf{V}$, and demonstrate that the adversarial projected sub-gradient ascent algorithm can maintain high true welfare, while the uncertainty-unaware optimization rapidly loses true welfare when noise increases.

Finally, we investigate multiple types of uncertainty sets and distributions for two reviewer assignment settings. Using publicly available data from five recent iterations of ICLR, we show that the adversarial projected sub-gradient ascent algorithm can maintain high worst-case USW. We also study bid data from AAMAS 2015, 2016, and 2021, where we estimate missing bids using both logistic and Gaussian process matrix factorization, and optimize under both the robust and stochastic regimes.

4.3 Optimizing Allocations under Uncertainty

We consider two main approaches to dealing with uncertainty: the *robust optimization* approach and the *Conditional Value at Risk* approach.

In the robust approach, we obtain an *uncertainty set* $\mathcal{V}$ that contains the true agent valuations $\boldsymbol{v}^*$ (recall that $\boldsymbol{v}^*$ denotes the row-major vectorization of $\mathbf{V}^*$) with probability at least $1-\delta$ for some confidence parameter $\delta \in [0,1)$. We then optimize the welfare (for any welfare function W) corresponding to the *worst* valuation matrix in the uncertainty set, i.e.,

$$\max_{\boldsymbol{a}\in\mathcal{Z}} \min_{\boldsymbol{v}\in\mathcal{V}} W(\boldsymbol{a}, \boldsymbol{v}). \tag{4.1}$$

This approach is appropriate when we do not have access to a full distribution $\mathcal{D}_{\boldsymbol{v}}$ but have error bounds on $\boldsymbol{v}^*$, which define the uncertainty set $\mathcal{V}$.

When we have access to a full distribution $\mathcal{D}_{\boldsymbol{v}}$ over a random variable $\boldsymbol{v} \in [0,1]^{nm}$, we apply a stochastic approach instead. We compute the welfare distribution and optimize the conditional expectation over an α-percentile of the welfare or *Conditional Value at Risk at* α (CVaR$_\alpha$), where the confidence parameter α is determined by the market-maker. This approach is also referred to as the *soft-robust* approach. Suppose that $\mathcal{D}_{\boldsymbol{v}}$ represents the probability distribution of the random valuation matrix $\mathbf{V}$ and α denotes the percentile of the welfare we wish to optimize. For any $\alpha \in (0, 0.5)$, CVaR$_\alpha$ is defined as $\mathbb{E}_{\boldsymbol{v}\sim\mathcal{D}_{\boldsymbol{v}}}[X \mid X \leq \nu_\alpha(W; \boldsymbol{a}, \boldsymbol{v})]$ where $\nu_\alpha(W; \boldsymbol{a}, \boldsymbol{v})$ denotes the α-percentile of welfare. We thus seek to solve

$$\max_{\boldsymbol{a}\in\mathcal{Z}} \mathop{\mathbb{E}}_{\boldsymbol{v}\sim\mathcal{D}_{\boldsymbol{v}}} [X \mid X \leq \nu_\alpha(W; \boldsymbol{a}, \boldsymbol{v})] \tag{4.2}$$

This approach is only appropriate when $\mathcal{D}_{\boldsymbol{v}}$ is fully known, and $\mathbb{E}_{\boldsymbol{v}\sim\mathcal{D}_{\boldsymbol{v}}}[X \mid X \leq \nu_\alpha(W; \boldsymbol{a}, \boldsymbol{v})]$ can be efficiently computed and optimized.

4.4 Robust Optimization

In this section, we investigate various methods of computing uncertainty sets. We then study how to optimize USW and GESW over multiple uncertainty set structures.

We begin with a simple observation relating the worst-case welfare over $\mathcal{V}$ for any assignment $\mathbf{A}$ to the true welfare of $\mathbf{A}$ with $\mathbf{V}^*$. Proposition 4.4.1 shows the minimum welfare of an assignment $\mathbf{A}$ over an (ε, δ) uncertainty set $\mathcal{V}$ relates additively to the true welfare of $\mathbf{A}$. The tightness of the approximation depends on the $\mathcal{L}_1$ diameter of the set $\mathcal{V}$ and the additive $\mathcal{L}_1$ error ε allowed for the uncertainty set.

Proposition 4.4.1 (Relating True and Worst-Case Welfare over $\mathcal{V}$)**.** *Suppose $\mathcal{V}$ is a (ε, δ) uncertainty set with $\|\mathbf{V} - \mathbf{V}'\|_1 \leq L$ for all $\mathbf{V}, \mathbf{V}' \in \mathcal{V}$, and the true affinity score matrix is labeled $\mathbf{V}^*$. Consider any assignment $\mathbf{A} \in \mathcal{Z}$. Then with probability at least $1 - \delta$,*

$$\mathrm{USW}(\mathbf{A}, \mathbf{V}^*) - \tfrac{L+\varepsilon}{n} \leq \inf_{\mathbf{V} \in \mathcal{V}} \mathrm{USW}(\mathbf{A}, \mathbf{V}) \leq \mathrm{USW}(\mathbf{A}, \mathbf{V}^*) + \tfrac{\varepsilon}{n}.$$

Proof. For the right hand side, note that with probability at least $1-\delta$ there exists some $\mathbf{V}^*_\varepsilon \in \mathcal{V}$ with $\|\mathbf{V}^*_\varepsilon - \mathbf{V}^*\|_1 \leq \varepsilon$. Let $\mathbf{V}' = \arg\min_{\mathbf{V} \in \mathcal{V}} \mathrm{USW}(\mathbf{A}, \mathbf{V})$. By definition, $\mathrm{USW}(\mathbf{A}, \mathbf{V}') \leq \mathrm{USW}(\mathbf{A}, \mathbf{V}^*_\varepsilon)$. If $\mathrm{USW}(\mathbf{A}, \mathbf{V}^*_\varepsilon) \leq \mathrm{USW}(\mathbf{A}, \mathbf{V}^*)$, we have the desired inequality. Otherwise, we can apply the fact that $\mathrm{USW}(\mathbf{A}, \mathbf{V}^*_\varepsilon) - \mathrm{USW}(\mathbf{A}, \mathbf{V}^*) = \mathrm{USW}(\mathbf{A}, \mathbf{V}^*_\varepsilon - \mathbf{V}^*) \leq \frac{1}{n}\|\mathbf{V}^*_\varepsilon - \mathbf{V}^*\|_1 \leq \frac{1}{n}\varepsilon$, where the second-to-last inequality holds since every entry of $\mathbf{A}$ is in $\{0, 1\}$.

To derive the left hand side, we will aim to bound $\mathrm{USW}(\mathbf{A}, \mathbf{V}^*) - \mathrm{USW}(\mathbf{A}, \mathbf{V}')$ where $\mathbf{V}' = \arg\min_{\mathbf{V} \in \mathcal{V}} \mathrm{USW}(\mathbf{A}, \mathbf{V})$. Again with probability at least $1 - \delta$ there exists some $\mathbf{V}^*_\varepsilon \in \mathcal{V}$ with $\|\mathbf{V}^*_\varepsilon - \mathbf{V}^*\|_1 \leq \varepsilon$. So $\mathrm{USW}(\mathbf{A}, \mathbf{V}^*) - \mathrm{USW}(\mathbf{A}, \mathbf{V}^*_\varepsilon) = \mathrm{USW}(\mathbf{A}, \mathbf{V}^* - \mathbf{V}^*_\varepsilon) \leq \frac{1}{n}\|\mathbf{V}^* - \mathbf{V}^*_\varepsilon\|_1 \leq \frac{\varepsilon}{n}$. Applying similar logic, we can bound $\mathrm{USW}(\mathbf{A}, \mathbf{V}^*_\varepsilon) - \mathrm{USW}(\mathbf{A}, \mathbf{V}') = \mathrm{USW}(\mathbf{A}, \mathbf{V}^*_\varepsilon - \mathbf{V}') \leq \frac{1}{n}\|\mathbf{V}^*_\varepsilon - \mathbf{V}'\|_1 \leq \frac{L}{n}$. □

Proposition 4.4.1 implies that if we aim for low $\mathcal{L}_1$ diameter uncertainty sets $\mathcal{V}$, we can approximately optimize true utilitarian welfare using the robust objective. We show a stronger result later (Theorem 4.4.3); the *true welfare* of the robust optimization result is also close to the maximum welfare we could have achieved had we known $\mathbf{V}^*$.

In general, we will solve Equation (4.1) by relaxing the binary decision space $\mathcal{Z}$ to $\tilde{\mathcal{Z}}$, solving for some continuous assignment $\tilde{\mathbf{A}} \in \tilde{\mathcal{Z}}$, and then rounding to a selected integral allocation $\mathbf{A}_{\text{ROUND}}$. That is, we first solve

$$\max_{\tilde{\mathbf{A}} \in \tilde{\mathcal{Z}}} \min_{\mathbf{V} \in \mathcal{V}} W(\tilde{\mathbf{A}}, \mathbf{V}), \tag{4.3}$$

then round $\tilde{\mathbf{A}}$ to $\mathbf{A}_{\text{ROUND}}$ in such a way that $\mathbb{E}[\mathbf{A}_{\text{ROUND}}] = \tilde{\mathbf{A}}$ and $\mathbf{A}_{\text{ROUND}} \in \mathcal{Z}$. Aside from the optimization error of any algorithm used to obtain $\tilde{\mathbf{A}}$, there are two more error sources: *maximin error* for working under uncertainty, and *rounding error*.

The integrality gap of Equation (4.1) can be quite large; similarly, the $\mathcal{L}_1$ difference between a rounded assignment $\mathbf{A}_{\text{ROUND}} \in \mathcal{Z}$ and a continuous assignment $\tilde{\mathbf{A}} \in \tilde{\mathcal{Z}}$ may be quite large as well. Surprisingly, we show that when the welfare function is USW, although the integrality gap is large, with high probability this does not translate to a large amount of suboptimality in the *true* USW of the rounded solution $\mathbf{A}_{\text{ROUND}}$. Intuitively, whenever the maximin optimal continuous solution has a high $\mathcal{L}_1$ distance from any valid binary integer assignment, the decisions made during rounding cancel out on average, and have relatively little impact on the true welfare of the assignment.

Proposition 4.4.2 ($\mathcal{L}_1$ Distance to Integral Solution)**.** *Suppose an unrounded assignment $\tilde{\mathbf{A}}$ and a randomized rounding $\mathbf{A}_{\text{ROUND}}$ of $\tilde{\mathbf{A}}$ such that $\mathbb{E}[\mathbf{A}_{\text{ROUND}}] = \mathbf{A}$. Then the expected $\mathcal{L}_1$ deviation of the assignment due to rounding obeys*

$$\mathbb{E}_{\mathbf{A}_{\text{ROUND}}}\left[\|\mathbf{A}_{\text{ROUND}} - \tilde{\mathbf{A}}\|_1\right] = 2\left(\|\tilde{\mathbf{A}}\|_1 - \|\tilde{\mathbf{A}}\|_2^2\right) \leq nm - 2\left\|\tfrac{1}{2} - \tilde{\mathbf{A}}\right\|_1.$$

Although the assignments may need to be rounded quite significantly, Theorem 4.4.3 shows that the rounded assignment has near-optimal true welfare (in expectation). Theorem 4.4.3 also allows for ε error in the discrete/continuous maximin assignments. We define $\mathbf{A}^{\varepsilon} \in \mathcal{Z}$, representing any ε-optimal discrete solution to Equation (4.1) and $\tilde{\mathbf{A}}^\varepsilon \in \tilde{\mathcal{Z}}$, any ε-optimal continuous solution to Equation (4.3). $\mathbf{A}^\varepsilon$ and $\tilde{\mathbf{A}}^\varepsilon$ are formally defined by the properties

$$\max_{\mathbf{A}\in\mathcal{Z}} \inf_{\mathbf{V}\in\mathcal{V}} \mathrm{USW}(\mathbf{A},\mathbf{V}) - \inf_{\mathbf{V}\in\mathcal{V}} \mathrm{USW}(\mathbf{A}^\varepsilon,\mathbf{V}) \leq \varepsilon \ \ \& \ \ \max_{\tilde{\mathbf{A}}\in\tilde{\mathcal{Z}}} \inf_{\mathbf{V}\in\mathcal{V}} \mathrm{USW}(\tilde{\mathbf{A}},\mathbf{V}) - \inf_{\mathbf{V}\in\mathcal{V}} \mathrm{USW}(\tilde{\mathbf{A}}^\varepsilon,\mathbf{V}) \leq \varepsilon.$$

Theorem 4.4.3 (Maximin and Integrality Gaps in Welfare)**.** *Suppose $\mathcal{V}$ is an (ε,δ) uncertainty set with $\mathcal{L}_1$ diameter L. Let $\mathbf{A}^\varepsilon$ denote an ε-optimal discrete* RAU *solution, and $\tilde{\mathbf{A}}^\varepsilon$ denote an ε-optimal continuous* RAU *solution. Let $\mathbf{A}_{\mathrm{ROUND}}$ denote the random variable that arises from applying the randomized rounding procedure* ROUND *to $\tilde{\mathbf{A}}^\varepsilon$, and assume that* ROUND *preserves expectation, i.e., $\mathbb{E}[\mathbf{A}_{\mathrm{ROUND}}] = \tilde{\mathbf{A}}^\varepsilon$. Suppose also that the true valuations are $\mathbf{V}^*$, and denote the optimal solution $\mathbf{A}^* \doteq \arg\max_{\mathbf{A}\in\mathcal{Z}} \mathrm{USW}(\mathbf{A},\mathbf{V}^*)$. The following then hold.*

1. Maximin Gap: $\Pr\bigl(\mathrm{USW}(\mathbf{A}^*,\mathbf{V}^*) - \mathrm{USW}(\mathbf{A}^\varepsilon,\mathbf{V}^*) > \varepsilon + \frac{2\varepsilon+L}{n}\bigr) < \delta$.

2. Expected Regret: $\Pr\bigl(\mathrm{USW}(\mathbf{A}^*,\mathbf{V}^*) - \mathbb{E}_{\mathbf{A}_{\mathrm{ROUND}}}[\mathrm{USW}(\mathbf{A}_{\mathrm{ROUND}},\mathbf{V}^*)] > \varepsilon + \frac{2\varepsilon+L}{n}\bigr) < \delta$.

3. Probabilistic Regret: $\Pr\bigl(\mathrm{USW}(\mathbf{A}^*,\mathbf{V}^*) - \mathrm{USW}(\mathbf{A}_{\mathrm{ROUND}},\mathbf{V}^*) > \frac{\varepsilon+(2\varepsilon+L)/n}{\delta'}\bigr) < \delta' + \delta$.

Proof. We begin with the maximin gap bound. With probability at least $1-\delta$, there is some $\mathbf{V}^*_\varepsilon \in \mathcal{V}$ such that $\|\mathbf{V}^*_\varepsilon - \mathbf{V}^*\|_1 \leq \varepsilon$, and thus $\mathrm{USW}(\mathbf{A}^*,\mathbf{V}^*) \leq \mathrm{USW}(\mathbf{A}^*,\mathbf{V}^*_\varepsilon) + \frac{\varepsilon}{n}$ (by Proposition 4.4.1). Likewise, $\mathrm{USW}(\mathbf{A}^\varepsilon,\mathbf{V}^*_\varepsilon) \leq \mathrm{USW}(\mathbf{A}^\varepsilon,\mathbf{V}^*) + \frac{\varepsilon}{n}$. Together, these imply

$$\begin{aligned}\mathrm{USW}(\mathbf{A}^*,\mathbf{V}^*) - \mathrm{USW}(\mathbf{A}^\varepsilon,\mathbf{V}^*) &\leq \mathrm{USW}(\mathbf{A}^*,\mathbf{V}^*_\varepsilon) - \mathrm{USW}(\mathbf{A}^\varepsilon,\mathbf{V}^*) + \tfrac{\varepsilon}{n}\\ &\leq \mathrm{USW}(\mathbf{A}^*,\mathbf{V}^*_\varepsilon) - \mathrm{USW}(\mathbf{A}^\varepsilon,\mathbf{V}^*_\varepsilon) + 2\tfrac{\varepsilon}{n}.\end{aligned}$$

Now, $\mathrm{USW}(\mathbf{A}^*,\mathbf{V}^*_\varepsilon) \leq \max\limits_{\mathbf{A}\in\mathcal{Z}} \inf\limits_{\mathbf{V}\in\mathcal{V}} \mathrm{USW}(\mathbf{A},\mathbf{V}) + \frac{L}{n}$, and by definition

$$\mathrm{USW}(\mathbf{A}^\varepsilon,\mathbf{V}^*_\varepsilon) \geq \inf_{\mathbf{V}\in\mathcal{V}} \mathrm{USW}(\mathbf{A}^\varepsilon,\mathbf{V}),$$

thus

$$\mathrm{USW}(\mathbf{A}^*, \mathbf{V}_\varepsilon^*) - \mathrm{USW}(\mathbf{A}^\varepsilon, \mathbf{V}_\varepsilon^*) + 2\frac{\varepsilon}{n} \leq \max_{A \in \mathcal{Z}} \inf_{\mathbf{V} \in \mathcal{V}} \mathrm{USW}(\mathbf{A}, \mathbf{V}) - \inf_{\mathbf{V} \in \mathcal{V}} \mathrm{USW}(\mathbf{A}^\varepsilon, \mathbf{V}) + \tfrac{2\varepsilon + L}{n}.$$

The definition of $\mathbf{A}^\varepsilon$ implies $\max\limits_{\mathbf{A} \in \mathcal{Z}} \inf\limits_{\mathbf{V} \in \mathcal{V}} \mathrm{USW}(\mathbf{A}, \mathbf{V}) - \inf\limits_{\mathbf{V} \in \mathcal{V}} \mathrm{USW}(\mathbf{A}^\varepsilon, \mathbf{V}) \leq \varepsilon$.

To obtain the expected regret over $\mathbf{A}_{\mathrm{ROUND}}$, we first apply the two facts that $\mathbb{E}[\mathbf{A}_{\mathrm{ROUND}}] = \tilde{\mathbf{A}}^\varepsilon$ and USW is a linear objective function. We now must bound $\mathrm{USW}(\mathbf{A}^*, \mathbf{V}^*) - \mathrm{USW}(\tilde{\mathbf{A}}^\varepsilon, \mathbf{V}^*)$, which we can do using the same proof we used to obtain the maximin gap above (replacing $\mathbf{A}^\varepsilon$ and $\mathcal{Z}$ with $\tilde{\mathbf{A}}^\varepsilon$ and $\tilde{\mathcal{Z}}$).

Finally, the probabilistic worst-case regret bound follows from Markov's inequality. □

Note that the distribution for the probabilistic regret is over the randomness of the rounding procedure, while all bounds in Theorem 4.4.3 are probabilistic with respect to δ of the uncertainty set $\mathcal{V}$.

4.4.1 Building Uncertainty Sets

Now that we have explored the structure of the robust optimization problem, we discuss some theoretical results about the structure of uncertainty sets.

We start by analyzing simple uncertainty sets, namely the case where $\mathbf{V}$ is known, as well as hyperrectangular, spherical, and ellipsoidal uncertainty sets. We show a compositionality rule that allows market designers to combine multiple uncertainty sets into a single robust optimization problem. We will also prove two general theorems that aid in constructing uncertainty sets from data, one inductive and one transductive.

The case where $\mathbf{V}$ is known, $\mathcal{V} = \{\mathbf{V}\}$, is quite straightforward. As discussed in Chapter 3, this problem is known to be polynomial-time solvable, as it is a linear program with totally unimodular constraints. Let us now consider hyperrectangular, spherical, and ellipsoidal uncertainty sets. Many simple and intuitive models for uncertainty sets take the form of axis-aligned hyperrectangles. A naïve uncertainty set might estimate confidence intervals for each $\mathbf{V}_{i,g}$ independently and use a union bound to give a high-probability region for the affinity scores. Market designers might also make assumptions about intervals bounding affinity scores with certainty, taking the intersection of multiple such interval bounds. For example, they might start with the global constraints of the unit hypercube. To take the example of reviewer assignment, lower and upper bounds can then be given for pairs based on whether they receive certain bids, whether the program committee recommends the assignment, or whether a threshold on document similarity score is met. This model is ad-hoc and simple, but may be suitable in practice. Furthermore, if we assume that with probability at least $1 - \delta$, only a small constant fraction ε of these bounds can be violated, we can establish a (ε, δ) confidence interval under more realistic assumptions.

If we take all the lower bounds on request-expert scores, we obtain a lower bound valuation matrix $\underline{\mathbf{V}}$. Similarly, taking all the maximal possible values for request-expert scores yields an upper bound valuation matrix $\overline{\mathbf{V}}$. Our uncertainty set is thus $\mathcal{V}_\square \doteq \{\mathbf{V} \in \mathbb{R}^{n\times m} \mid \underline{\mathbf{V}}_{i,g} \leq \mathbf{V}_{i,g} \leq \overline{\mathbf{V}}_{i,g}\ \forall i, g\}$. Our first result is that Equation (4.1) can be solved for USW in polynomial time for axis-aligned, hyperrectangular uncertainty sets.

Theorem 4.4.4 (Hyperrectangular Uncertainty)**.** *When the uncertainty set is an axis-aligned hyperrectangular region $\mathcal{V}_\square$, then*

$$\underset{\mathbf{A}\in\mathcal{Z}}{\arg\max} \inf_{\mathbf{V}\in\mathcal{S}_\square} \mathrm{USW}(\mathbf{A}, \mathbf{V}) = \underset{\mathbf{A}\in\mathcal{Z}}{\arg\max}\, \mathrm{USW}(\mathbf{A}, \underline{\mathbf{V}}).$$

This requires polynomial time via LP reduction.

Proof. For any assignment $\mathbf{A} \in \mathcal{Z}$, $\inf_{\mathbf{V}\in\mathcal{V}_\square} \mathrm{USW}(\mathbf{A}, \mathbf{V}) = \mathrm{USW}(\mathbf{A}, \underline{\mathbf{V}})$, i.e., the lowest welfare is achieved if we assume that all scores are the worst they can be under the rectangular constraints. To see why, consider any matrix $\mathbf{V} \in \mathcal{V}_\square$ with $\mathbf{V}_{i,g} > \underline{\mathbf{V}}_{i,g}$ for some i and g.

Consider $\mathbf{V}'$ where $\mathbf{V}'_{i,g} = \underline{\mathbf{V}}_{i,g}$ and $\mathbf{V}'_{i,j} = \mathbf{V}_{i,j}$ for $i \neq i$ or $j \neq g$. Either $\mathbf{A}_{i,g} = 0$ and hence $\mathrm{USW}(\mathbf{A}, \mathbf{V}) = \mathrm{USW}(\mathbf{A}, \mathbf{V}')$ or $\mathbf{A}_{i,g} = 1$ and thus $\mathrm{USW}(\mathbf{A}, \mathbf{V}) > \mathrm{USW}(\mathbf{A}, \mathbf{V}')$. □

Axis-aligned, hyperrectangular uncertainty sets correspond to the case where uncertainty is bounded independently across request-expert pairs, hence their relative simplicity. Although hyperrectangular uncertainty sets are easy to work with, they are unnecessarily pessimistic, since it is very unlikely that all affinities take extreme values at once (i.e., $\underline{\mathbf{V}}$ is actually a very unlikely outcome). We can improve our estimates using uncertainty set models that account for the low probability of many simultaneous extreme values.

Many standard models directly bound the $\mathcal{L}_1$ or $\mathcal{L}_2$ error of their predictions, which implies uncertainty sets that are more optimistic than hyperrectangular $\mathcal{V}$ (and hence have tighter guarantees for Proposition 4.4.1).

We first analyze the case of symmetric uncertainty sets with $\mathcal{L}_2$ error guarantees. For example, in reviewer assignment, we might solicit bids uniformly at random and then predict unsampled bids using collaborative filtering with $\mathcal{L}_2$ error guarantees [31, 59, 98]. $\mathcal{V}$ could then be constructed as a linear combination of values known with certainty (document-based similarity scores and keyword-based matching scores) and the real and estimated bids, yielding a spherical uncertainty set $\mathcal{V}$.

We can consider a spherical $(0, \delta)$ uncertainty set to consist of a point estimate $\mathbf{V}^0$ and a radius ε limiting the $\mathcal{L}_2$ error from the point estimate $\mathbf{V}^0$. Formally, we aim to solve the problem $\arg\max_{\mathbf{A} \in \mathcal{Z}} \inf_{\mathbf{V} \in B_\varepsilon(\mathbf{V}^0)} \mathrm{USW}(\mathbf{A}, \mathbf{V})$, where $B_\varepsilon(\mathbf{V}^0) \doteq \{\mathbf{X} \in \mathbb{R}^{n \times m} \mid \|\mathbf{X} - \mathbf{V}^0\|_{\mathrm{F}} \leq \varepsilon\}$ denotes the ε Frobenius-norm ball around $\mathbf{V}^0$.

Theorem 4.4.5 (RAU under Spherical Uncertainty). *When $\mathcal{V}$ is a sphere,*

$$\operatorname*{arg\,max}_{\mathbf{A} \in \mathcal{Z}} \inf_{\mathbf{V} \in B_\varepsilon(\mathbf{V}^0)} \mathrm{USW}(\mathbf{A}, \mathbf{V}) = \operatorname*{arg\,max}_{\mathbf{A} \in \mathcal{Z}} \mathrm{USW}(\mathbf{A}, \mathbf{V}^0),$$

which can be computed in polynomial time. Furthermore, for all $\mathbf{A} \in \mathcal{Z}$*,*

$$|\mathrm{USW}(\mathbf{A}, \mathbf{V}^0) - \mathrm{USW}(\mathbf{A}, \mathbf{V}^*)| \leq \tfrac{\varepsilon\sqrt{\|\mathbf{A}\|_1}}{n}.$$

Proof. Let $\mathbf{0}$ denote the $n \times m$ matrix of all 0 values. For any $\mathbf{A} \in \mathcal{Z}$, it holds that

$$\begin{aligned}
\inf_{\mathbf{V} \in B_\varepsilon(\mathbf{V}^0)} \mathrm{USW}(\mathbf{A}, \mathbf{V}) &= \min_{\mathbf{X} \in B_\varepsilon(\mathbf{0})} \mathrm{USW}(\mathbf{A}, \mathbf{V}^0 + \mathbf{X}) \\
&= \min_{\mathbf{X} \in B_\varepsilon(\mathbf{0})} \mathrm{USW}(\mathbf{A}, \mathbf{X}) + \mathrm{USW}(\mathbf{A}, \mathbf{V}^0) && \textsc{Linearity of USW} \\
&= \mathrm{USW}\left(\mathbf{A}, \mathbf{V}^0 - \mathbf{A}\frac{\varepsilon}{\|\mathbf{A}\|_\mathrm{F}}\right) && \textsc{See Below} \\
&= \mathrm{USW}(\mathbf{A}, \mathbf{V}^0) - \mathrm{USW}\left(\mathbf{A}, \mathbf{A}\frac{\varepsilon}{\|\mathbf{A}\|_\mathrm{F}}\right) && \textsc{Linearity of USW} \\
&= \mathrm{USW}(\mathbf{A}, \mathbf{V}^0) - \tfrac{1}{n}\|\mathbf{A}\|_\mathrm{F}\varepsilon \\
&= \mathrm{USW}(\mathbf{A}, \mathbf{V}^0) - \tfrac{1}{n}\sqrt{\|\mathbf{A}\|_1}\varepsilon.
\end{aligned}$$

Because $\mathrm{USW}(\mathbf{A}, \mathbf{V}^0)$ does not depend on X, we just need to find $\arg\min_{\mathbf{X} \in B_\varepsilon(\mathbf{0})} \mathrm{USW}(\mathbf{A}, \mathbf{X})$. Because $\|\mathbf{A}\|_\mathrm{F}$ is fixed (and the argmin occurs at $\|\mathbf{X}\|_\mathrm{F} = \varepsilon$) the argmin must be $\mathbf{X} = -\mathbf{A}\frac{\varepsilon}{\|\mathbf{A}\|_\mathrm{F}}$.

Again, $\sqrt{\|\mathbf{A}\|_1}$, ε, and n are all fixed. So we see that our function has the same argmax as $\mathrm{USW}(\mathbf{A}, \mathbf{V}^0)$. The problem $\arg\max_{\mathbf{A} \in \mathcal{Z}} \mathrm{USW}(\mathbf{A}, \mathbf{V}^0)$ is an instance of expert assignment with known $\mathbf{V}$, which is solvable in polynomial time.

We can quantify the worst-case welfare loss due to spherical uncertainty with radius ε as

$$\mathrm{USW}(\mathbf{A}, \mathbf{V}^0) - \mathrm{USW}(\mathbf{A}, \mathbf{V}^*) \leq \tfrac{1}{n}\sqrt{\|\mathbf{A}\|_1}\varepsilon.$$

A similar argument to the above shows that

$$\max_{\mathbf{V} \in B_\varepsilon(\mathbf{V}^0)} \mathrm{USW}(\mathbf{A}, \mathbf{V}) = \mathrm{USW}(\mathbf{A}, \mathbf{V}^0) + \tfrac{1}{n}\sqrt{\|\mathbf{A}\|_1}\varepsilon,$$

and thus we also have

$$\mathrm{USW}(\mathbf{A}, \mathbf{V}^*) - \mathrm{USW}(\mathbf{A}, \mathbf{V}^0) \leq \tfrac{1}{n}\sqrt{\|\mathbf{A}\|_1}\varepsilon.$$

□

One way of looking at Theorem 4.4.5 is that solving Equation (4.1) for USW over spherical uncertainty sets provides no additional robustness guarantees over $\arg\max_{\mathbf{A}\in\mathcal{Z}} \mathrm{USW}(\mathbf{A}, \mathbf{V}^0)$. Thus, in order to obtain meaningful robustness guarantees, we require both a limit to the total amount of variation in affinity scores (Theorem 4.4.4) as well as asymmetry between the noise on affinity scores (Theorem 4.4.5). We can explain these results intuitively; to decide how best to assign experts, we need to be able to make tradeoffs between assigning pairs with potentially high (but also potentially low) value or assigning pairs that have an average amount of value with higher certainty. Those tradeoffs are only meaningful if uncertainty varies across request-expert pairs, and if there is a limited total amount of uncertainty.

With that intuition in mind, we generalize to the case of ellipsoidal uncertainty sets. In a simple model, we might model affinity scores as multivariate Gaussians. In this case, we obtain a mean vector $\boldsymbol{\mu} \in [0,1]^{nm}$ and a positive semi-definite covariance matrix $\Sigma \in \mathbb{R}^{nm\times nm}$. Given a confidence level $1-\delta$, we create an uncertainty set

$$\mathcal{V} \doteq \left\{ \boldsymbol{v} \in \mathbb{R}^{nm} \mid (\boldsymbol{v}-\boldsymbol{\mu})^T \Sigma^{-1} (\boldsymbol{v}-\boldsymbol{\mu}) \leq \chi^2_{nm}(1-\delta) \right\}, \tag{4.4}$$

where χ^2_k is the inverse CDF of the χ^2 distribution with k degrees of freedom. If we assume no model error (which is not a safe assumption generally), then the true affinity scores are contained within $\mathcal{V}$ with probability at least $1-\delta$. We know also that $\chi^2_{nm}(1-\delta) \leq nm + 2\sqrt{nm \ln\frac{1}{\delta}} + 2\ln\frac{1}{\delta}$. The size of the uncertainty set depends only logarithmically on $\frac{1}{\delta}$, and thus we can trade off between δ and the $\mathcal{L}_1$ diameter of the uncertainty set.

While the Gaussian model employed in (4.4) makes a strong modeling assumption, we can use a validation set and a predictive model with provable tail bounds to obtain uncertainty sets that do not require any such assumptions. To accomplish this, we require a predictive model of valuations. Market makers can obtain $\mathcal{L}_2$ error bounds using validation data, and this will yield an ellipsoid due to sampling bias (only certain request-expert pairs will be observed). In this setting, the uncertainty set is not derived as a confidence interval of a probability distribution, but rather directly comes from a tail bound on total generalization error. Optimizing in this setting *requires* using a robust approach, and cannot be done with average case analysis. This approach is similar to the bounds we derived in Theorem 3.3.3, but that theorem derives a tighter bound using the fact that $\mathbf{A}$ is fixed prior to computing the bound. We derive a general bound that instead allows us to select $\mathbf{A}$ robustly.

Suppose the true value of some request-expert pair is $f^*(i, g)$, and we have access to a predictive model $\hat{f}(i, g)$, perhaps learned on historical data. In practice, $\hat{f}$ predicts the value of assigning expert $g \in M$ to request $i \in N$ based on any information available prior to assignment, and the specific definition of value is left to the market designer. For example, a reviewer assignment venue may decide that affinity is best measured via reviewer bids, and they may use historical data to train a predictor $\hat{f}$ to predict missing bids from document-based similarity scores and keywords. Alternatively, venue organizers may decide that the ground truth affinity $f^*(i, g)$ should correspond to a meta-reviewer's judgment of review quality, and $\hat{f}$ can then be trained on historical data to predict these judgments. We will take $\hat{\mathbf{V}}_{i,g} \doteq \hat{f}(i, g)$ and $\mathbf{V}^*_{i,g} \doteq f^*(i, g)$. We assume that we can evaluate $\hat{f}$ on all request-expert pairs in the current assignment problem, and potentially on pairs from historical data as well. We may be able to sample $f^*(i, g)$ for some, but not all, pairs in the current problem and historical data (the validation data). We then probabilistically bound a weighted average of the square error between $\hat{f}$ and f^* in terms of an estimate of expected square error computed on the validation set.

The details of the validation set vary by application, but the overall strategy will be to estimate the square error $\mathbb{E}[(\mathbf{V}^* - \hat{\mathbf{V}})^2]$, where $\mathbf{V}^*$ and $\hat{\mathbf{V}}$ are given by f^* and $\hat{f}$ on a random request-expert pair. We show two such approaches, the first inductive, using historic or auxilliary data to form the validation set, and the second transductive, assuming a small random sample of true valuations can be queried within the current set of requests and experts. We will need to define the notion of sampling a sequence of random variables *conditionally independently without replacement.*

Definition 4.4.6. A sequence of variables $x_1, x_2, \dots x_t$, where all $x_i \in X$, is sampled conditionally independently without replacement from a distribution $\mathcal{D}$ with support X if the variables x_i are sampled in order from x_1 to x_t, and for any $i \in \{1, \dots t\}$, $\Pr_{\mathcal{D}}(x_i = x | x_1, \dots x_{i-1}) = \frac{\Pr_{\mathcal{D}}(x_i = x)}{\int_{X \setminus \{x_1, \dots x_{i-1}\}} \Pr_{\mathcal{D}}(x_i = y) dy}$ for all $x \in X \setminus \{x_1, \dots x_{i-1}\}$ and $\Pr_{\mathcal{D}}(x_i = x | x_1, \dots x_{i-1}) = 0$ for $x \in \{x_1, \dots x_{i-1}\}$.

Theorem 4.4.7 (Ellipsoidal Uncertainty Sets from Inductive Predictors)**.** *Let $\mathcal{D}'$ be a probability distribution over request-expert pairs, and let $\mathcal{D}^{\mathcal{N}}$ and $\mathcal{D}^{\mathcal{M}}$ be distributions over requests and experts, respectively. Assume that N and M were drawn conditionally independently without replacement from $\mathcal{D}^{\mathcal{N}}$ and $\mathcal{D}^{\mathcal{M}}$, respectively. Suppose we sample t request-expert pairs $\{(i_j, g_j)\}_{j=1}^t$ conditionally independently without replacement from $\mathcal{D}'$, and these request-expert pairs have true and estimated affinity scores $\{f^*(i_j, g_j)\}_{j=1}^t$ and $\{\hat{f}(i_j, g_j)\}_{j=1}^t$, respectively.*

Let

$$\alpha(i, g) \doteq \frac{\Pr_{I \sim \mathcal{D}^{\mathcal{N}}}(I = i) \Pr_{E \sim \mathcal{D}^{\mathcal{M}}}(E = g)}{\Pr_{(I,E) \sim \mathcal{D}'}((I, E) = (i, g))}$$

denote the probability ratio *of sampling i from $\mathcal{D}^{\mathcal{N}}$ and g from $\mathcal{D}^{\mathcal{M}}$ to sampling (i, g) from $\mathcal{D}'$, and let*

$$\alpha_{\min} \doteq \inf_{i \in \mathcal{N}, g \in \mathcal{M}} \alpha(i, g)$$

denote the infimum probability ratio*. Now construct the* ellipsoid matrix $\Sigma \in \mathbb{R}^{nm \times nm}$ *as the diagonal matrix such that $\Sigma_{im+j, im+j} = \alpha(i, g_j)$ for all $i \in N, g_j \in M$.*

Then for any $\delta \in (0,1)$, the ellipsoid

$$\mathcal{V} \doteq \left\{ \boldsymbol{v} \in \mathbb{R}^{nm} \,\middle|\, \frac{1}{nm}(\boldsymbol{v}-\hat{\boldsymbol{v}})^\intercal \Sigma^{-1}(\boldsymbol{v}-\hat{\boldsymbol{v}}) \le \underbrace{\frac{1}{t}\sum_{j=1}^{t}(f^*(i_j,g_j)-\hat{f}(i_j,g_j))^2}_{=\hat{\xi}} + \underbrace{\sqrt{\left(\frac{1}{t}+\frac{n+m}{nm\alpha^2_{\min}}\right)\frac{\ln\frac{1}{\delta}}{2}}}_{=\eta} \right\}$$

is a $(0,\delta)$ uncertainty set, where $\hat{\xi}$ denotes the empirical square error *of our estimated scores, and η denotes the* excess error bound *due to sampling.*

Proof. We aim to show the ellipsoid $\mathcal{V}$ is a $(0,\delta)$ uncertainty set. In other words, it must hold that with probability $1-\delta$, the true affinity scores $\boldsymbol{v}^*$ satisfy

$$\frac{1}{nm}(\boldsymbol{v}^*-\hat{\boldsymbol{v}})\Sigma^{-1}(\boldsymbol{v}^*-\hat{\boldsymbol{v}}) \le \underbrace{\frac{1}{t}\sum_{j=1}^{t}(f^*(i_j,g_j)-\hat{f}(i_j,g_j))^2}_{=\hat{\xi}} + \underbrace{\sqrt{\left(\frac{1}{t}+\frac{n+m}{nm\alpha^2_{\min}}\right)\frac{\ln\frac{1}{\delta}}{2}}}_{=\eta}.$$

We will give a probabilistic bound on the difference

$$X \doteq \frac{1}{nm}(\boldsymbol{v}^*-\hat{\boldsymbol{v}})^\intercal\Sigma^{-1}(\boldsymbol{v}^*-\hat{\boldsymbol{v}}) - \hat{\xi} = \frac{1}{nm}\sum_{i=1}^{n}\sum_{g=1}^{m}\frac{(\mathbf{V}^*_{i,g}-\hat{\mathbf{V}}_{i,g})^2}{\alpha(i,g)} - \frac{1}{t}\sum_{j=1}^{t}(f^*(i_j,g_j)-\hat{f}(i_j,g_j))^2.$$

We first note that $\mathbb{E}[X]=0$. We assume that our requests $i \sim \mathcal{D}^{\mathcal{N}}$, our experts $g \sim \mathcal{D}^{\mathcal{M}}$, and our samples (which are request-expert pairs) are drawn as $(i_j,g_j) \sim \mathcal{D}'$. For simplicity of notation, we will write $\{(i_j,g_j)\}_{j=1}^t \sim \mathcal{D}'$ to denote the set of all request-expert pairs $(i_j,g_j) \sim \mathcal{D}'$. Also recall that $\alpha(i,g) = \frac{\Pr_{I\sim\mathcal{D}^{\mathcal{N}}}(I=i)\Pr_{E\sim\mathcal{D}^{\mathcal{M}}}(E=g)}{\Pr_{(I,E)\sim\mathcal{D}'}((I,E)=(i,g))}$. We have that

$$\begin{aligned}
\mathop{\mathbb{E}}_{\substack{N\sim\mathcal{D}^{\mathcal{N}},M\sim\mathcal{D}^{\mathcal{M}}\\ \{(i_j,g_j)\}_{j=1}^t\sim\mathcal{D}'}}[X] &= \mathop{\mathbb{E}}_{\substack{N\sim\mathcal{D}^{\mathcal{N}},M\sim\mathcal{D}^{\mathcal{M}}\\ \{(i_j,g_j)\}_{j=1}^t\sim\mathcal{D}'}}\left[\frac{1}{nm}\sum_{i=1}^{n}\sum_{g=1}^{m}\frac{(\mathbf{V}^*_{i,g}-\hat{\mathbf{V}}_{i,g})^2}{\alpha(i,g)} - \frac{1}{t}\sum_{j=1}^{t}(f^*(i_j,g_j)-\hat{f}(i_j,g_j))^2\right] \\
&= \mathop{\mathbb{E}}_{N\sim\mathcal{D}^{\mathcal{N}},M\sim\mathcal{D}^{\mathcal{M}}}\left[\frac{1}{nm}\sum_{i=1}^{n}\sum_{g=1}^{m}\frac{(\mathbf{V}^*_{i,g}-\hat{\mathbf{V}}_{i,g})^2}{\alpha(i,g)}\right] \\
&\quad - \mathop{\mathbb{E}}_{\{(i_j,g_j)\}_{j=1}^t\sim\mathcal{D}'}\left[\frac{1}{t}\sum_{j=1}^{t}(f^*(i_j,g_j)-\hat{f}(i_j,g_j))^2\right].
\end{aligned}$$

Following the standard argument for importance sampling, it is also clear that

$$\begin{aligned}
\mathop{\mathbb{E}}_{N\sim\mathcal{D}^{\mathcal{N}},M\sim\mathcal{D}^{\mathcal{M}}}\left[\frac{1}{nm}\sum_{i=1}^{n}\sum_{g=1}^{m}\frac{(\mathbf{V}^*_{i,g}-\hat{\mathbf{V}}_{i,g})^2}{\alpha(i,g)}\right] &= \frac{1}{nm}\sum_{i=1}^{n}\sum_{g=1}^{m}\mathop{\mathbb{E}}_{N\sim\mathcal{D}^{\mathcal{N}},M\sim\mathcal{D}^{\mathcal{M}}}\left[\frac{(\mathbf{V}^*_{i,g}-\hat{\mathbf{V}}_{i,g})^2}{\alpha(i,g)}\right] \\
&= \frac{1}{nm}\sum_{j=1}^{nm}\mathop{\mathbb{E}}_{(i_j,g_j)\sim\mathcal{D}'}\left[(f^*(i_j,g_j)-\hat{f}(i_j,g_j))^2\right] \\
&= \frac{1}{t}\sum_{j=1}^{t}\mathop{\mathbb{E}}_{(i_j,g_j)\sim\mathcal{D}'}\left[(f^*(i_j,g_j)-\hat{f}(i_j,g_j))^2\right] \\
&= \mathop{\mathbb{E}}_{\{(i_j,g_j)\}_{j=1}^{t}\sim\mathcal{D}'}\left[\frac{1}{t}\sum_{j=1}^{t}(f^*(i_j,g_j)-\hat{f}(i_j,g_j))^2\right].
\end{aligned}$$

In addition, X is a function of a system of negatively-dependent (due to sampling without replacement) random variables comprising t auxiliary terms, m experts, and n requests. Modifying any of the t auxiliary terms can result in at most $\frac{1}{t}$ change in X, each of the n requests has impact at most $\frac{1}{n\alpha_{\min}}$, and each of m experts has impact at most $\frac{1}{m\alpha_{\min}}$. The sum of square bounded differences is thus $\frac{1}{t}+\frac{1}{n\alpha^2_{\min}}+\frac{1}{m\alpha^2_{\min}}$. Consequently, by McDiarmid's inequality [111], it holds

$$\Pr\left(X \geq \sqrt{\left(\frac{1}{t}+\frac{1}{n\alpha^2_{\min}}+\frac{1}{m\alpha^2_{\min}}\right)\frac{\ln\frac{1}{\delta}}{2}}\right) \leq \delta.$$

□

Theorem 4.4.7 assumes that both the historic data and the current sets of requests and experts are drawn at random. In particular, historic request-expert pairs are sampled from $\mathcal{D}'$ (modeling the historic data generation process), and requests and experts for the current venue are sampled from $\mathcal{D}^{\mathcal{N}}$ and $\mathcal{D}^{\mathcal{M}}$. Let us consider reviewer assignment as the example. We might let $\mathcal{D}^{\mathcal{N}}$ and $\mathcal{D}^{\mathcal{M}}$ model the processes by which papers are submitted and reviewers volunteer. We then construct $\alpha(i,g)$ to reweight square error on the current venue to match expected square error on $\mathcal{D}'$ (i.e., we use importance sampling to calibrate expectations over $\mathcal{D}'$ versus those over $\mathcal{D}^{\mathcal{N}}$ and $\mathcal{D}^{\mathcal{M}}$). For example, $\mathcal{D}'$ reflects all elements of historic data generation, most importantly the availability of historic data from multiple venues with different focuses. We might then use the (relatively stable) topic areas of papers and reviewers to model $\mathcal{D}^{\mathcal{N}}$ and $\mathcal{D}^{\mathcal{M}}$, and thus $\alpha(i,g)$ reflects the ratio of the popularity of i and g's topic areas in the current venue to historic venues.

We show a similar result in the transductive setting. Instead of constructing a predictive function from historical data, we generalize a small set of known affinities for the current problem instance to the unknown valuations for the same instance. Note that $\mathcal{D}'$ now reflects the process by which we obtain samples for (i_j, g_j) pairs from the current problem instance, rather than from historical data sources.

Theorem 4.4.8 (Ellipsoidal Uncertainty Sets from Transductive Predictors)**.** *Suppose we sample t request-expert pairs $\{(i_j, g_j)\}_{j=1}^t$ conditionally independently without replacement from $\mathcal{D}'$, and these request-expert pairs have true and estimated affinity scores $\{f^*(i_j, g_j)\}_{j=1}^t$ and $\{\hat{f}(i_j, g_j)\}_{j=1}^t$, respectively. Let*

$$\alpha(i,g) \doteq \frac{(nm)^{-1}}{\Pr_{(I,E)\sim\mathcal{D}'}((I,E) = (i,g))}$$

denote the probability ratio *between sampling (i,g) uniformly at random and sampling (i,g) from $\mathcal{D}'$, and construct the* ellipsoid matrix $\Sigma \in \mathbb{R}^{nm\times nm}$ *as the diagonal matrix such that $\Sigma_{im+j,im+j} = \alpha(i, g_j)$ for all $i \in N, g_j \in M$.*

Then for any $\delta \in (0,1)$, the ellipsoid

$$\mathcal{V} \doteq \left\{ \boldsymbol{v} \in \mathbb{R}^{nm} \,\middle|\, \frac{1}{nm}(\boldsymbol{v} - \hat{\boldsymbol{v}})^\intercal \Sigma^{-1} (\boldsymbol{v} - \hat{\boldsymbol{v}}) \leq \underbrace{\frac{1}{t}\sum_{j=1}^{t}(f^*(i_j,g_j) - \hat{f}(i_j,g_j))^2}_{=\hat{\xi}} + \underbrace{\sqrt{\frac{\ln\frac{1}{\delta}}{2t}}}_{=\eta}, \right\}$$

is a $(0, \delta)$ *uncertainty set, where* $\hat{\xi}$ *denotes the* empirical square error *of our estimated scores, and* η *denotes the* excess error bound *due to sampling.*

Proof. Proof of result follows similarly to that of Theorem 4.4.7. Here, reviewers and papers are fixed. We will bound the deviation from the mean for the random variable

$$X \doteq \frac{1}{nm}\sum_{i=1}^{n}\sum_{g=1}^{m}\frac{(\mathbf{V}_{i,g}-\hat{\mathbf{V}}_{i,g})^2}{\alpha(p,r)} - \frac{1}{t}\sum_{j=1}^{t}(f^*(i_j,g_j)-\hat{f}(i_j,g_j))^2 .$$

We show that $\mathbb{E}[X] = 0$, as

$$\begin{aligned}
\mathop{\mathbb{E}}_{\{(i_j,g_j)\}_{j=1}^t\sim\mathcal{D}'}[X] &= \mathop{\mathbb{E}}_{\{(i_j,g_j)\}_{j=1}^t\sim\mathcal{D}'}\left[\frac{1}{nm}\sum_{i=1}^{n}\sum_{g=1}^{m}\frac{(\mathbf{V}_{i,g}-\hat{\mathbf{V}}_{i,g})^2}{\alpha(p,r)} - \frac{1}{t}\sum_{j=1}^{t}(f^*(i_j,g_j)-\hat{f}(i_j,g_j))^2\right] \\
&= \frac{1}{nm}\sum_{i=1}^{n}\sum_{g=1}^{m}\frac{(\mathbf{V}_{i,g}-\hat{\mathbf{V}}_{i,g})^2}{\alpha(p,r)} - \frac{1}{t}\sum_{j=1}^{t}\mathop{\mathbb{E}}_{\{(i_j,g_j)\}_{j=1}^t\sim\mathcal{D}'}\left[(f^*(i_j,g_j)-\hat{f}(i_j,g_j))^2\right] \\
&= \frac{1}{nm}\sum_{i=1}^{n}\sum_{g=1}^{m}\frac{(\mathbf{V}_{i,g}-\hat{\mathbf{V}}_{i,g})^2}{\alpha(p,r)} - \frac{1}{nm}\sum_{i=1}^{nm}\mathop{\mathbb{E}}_{\{(i_j,g_j)\}_{i=1}^{nm}\sim\mathrm{Unif}(N\times M)}\left[\frac{(f^*(i_j,g_j)-\hat{f}(i_j,g_j))^2}{\alpha(i_j,g_j)}\right] \\
&= \frac{1}{nm}\sum_{i=1}^{n}\sum_{g=1}^{m}\frac{(\mathbf{V}_{i,g}-\hat{\mathbf{V}}_{i,g})^2}{\alpha(p,r)} - \frac{1}{nm}\sum_{i=1}^{n}\sum_{g=1}^{m}\frac{(\mathbf{V}_{i,g}-\hat{\mathbf{V}}_{i,g})^2}{\alpha(p,r)} .
\end{aligned}$$

To apply the final bound, we need only consider the t independent auxiliary terms, each of bounded difference $\frac{1}{t}$. In this case, we have the bounded difference term $\frac{1}{t}$ for each of t auxiliary terms. This yields, via the Hoeffding [76] or McDiarmid [111] inequalities, the final result

$$\Pr\left(X \geq \sqrt{\frac{\ln\frac{1}{\delta}}{2t}}\right) \leq \delta .$$

□

The transductive result can be straightforwardly applied to many different contexts in which market designers can solicit samples of f^* on (i_j, g_j) pairs from the current assignment problem, rather than historical data. This information must be obtained prior to assigning the majority of experts. For example, conference organizers could define f^* as a reviewer's hypothetical bid and Theorem 4.4.8 then requires soliciting a small number of bids to estimate the error of $\hat{f}$. Similarly, f^* could correspond to meta-reviewer judgments of review quality, accomplished by opting for a two-stage reviewing process, in which the reviews and feedback generated in the first stage are used to estimate the error of $\hat{f}$. These definitions of f^* are costly to sample, but organizers can still efficiently target sophisticated affinity models by solving Equation (4.1) over the uncertainty sets of Theorem 4.4.8.

We naturally ask the question, "How many samples are sufficient to obtain a sharp confidence bound?" Observe that, by Proposition 4.4.1, the gap between adversarial and true welfare is $\frac{L}{n}$, where L denotes the $\mathcal{L}_1$ diameter of $\mathcal{V}$. For the ellipsoidal uncertainty set of Theorem 4.4.8, $\frac{L}{n} \leq 2m\sqrt{\alpha_{\max}(\hat{\xi} + \eta)}$, where $\alpha_{\max} \doteq \sup_{i \in \mathcal{N}, g \in \mathcal{M}} \alpha(p, r)$. Furthermore, the empirical square error $\hat{\xi}$ converges to some ξ as the number of samples t increases, thus we need only select $t \in \Omega\big(\frac{\log \frac{1}{\delta}}{\xi^2}\big)$ samples to ensure that the uncertainty set is constant-factor optimal, at which point the welfare gap is $O(m\sqrt{\alpha_{\max}\xi})$, which is also optimal to within constant factors. Notably, the sufficient sample size t is *independent of the problem size* (i.e., n and m), thus the added burden of soliciting these extra samples is negligible. We also see that the fundamental limitation of this method is the average square error ξ, which depends on the predictor, the problem instance, and the sampling distribution $\mathcal{D}'$. It is thus paramount to use predictors for which this quantity will be small. Fortunately, this is often the case, as many predictive models are explicitly trained to minimize $\mathcal{L}_2$ error on some task, which motivates the choice of our ellipsoidal uncertainty sets. Note that while this argument pertains to Theorem 4.4.8, one can argue similarly for the necessary size of the validation set to ensure $\eta = O(\xi)$ in Theorem 4.4.7.

Finally, we note that it is possible to extend either result under less favorable (more realistic) assumptions about the sampling process using the ε parameter ($\mathcal{L}_1$ error) of our uncertainty set construction. In particular, either result produces a $(t\varepsilon, \delta)$ uncertainty set if $\hat{\mathbf{V}}$, $\mathbf{V}^*$, and the associated (i, g) pairs are subject to *adversarial corruption* of $t\varepsilon$ of the validation set samples drawn from $\mathcal{D}'$, which has immediate applications in privacy, adversarial robustness, and various notions of strategy-proofness. Furthermore, to model more complicated and potentially not fully understood distribution shift, we obtain via Bennett's inequality [17] a $\left(t\varepsilon + \frac{1}{3}\ln\frac{1}{\delta'} + \sqrt{2t\varepsilon(1-\varepsilon)\ln\frac{1}{\delta'}}, \delta + \delta'\right)$ uncertainty set if the validation set is instead drawn from some $\mathcal{D}''$ such that $\mathrm{TVD}(\mathcal{D}', \mathcal{D}'') \leq \varepsilon$.

We may often have more complicated uncertainty sets than the simple geometries described above. For example, we can intersect the constraints of the unit hypercube with an ellipsoidal uncertainty set. This produces a *truncated ellipsoid*, a common construction that we will see again in Section 4.6.

Lemma 4.4.9 (Uncertainty Set Intersection). *Suppose each $\mathcal{V}_i$ for $i \in [k]$ is a $(0, \delta_i)$ uncertainty set. Then $\mathcal{V}_\cap \doteq \bigcap_{i=1}^k \mathcal{V}_i$ is a $(0, \|\boldsymbol{\delta}\|_1)$ uncertainty set.*

Proof. Suppose some true valuation matrix $\mathbf{V}^*$. We know that with probability at most δ_i, $\mathbf{V}^*$ is not in $\mathcal{V}_i$ (or more formally, there is no $\mathbf{V} \in \mathcal{V}_i$ such that $\|\mathbf{V}^* - \mathbf{V}\|_1 = 0$). By a union bound, the probability that at least one of these conditions is violated is at most $\sum_{i=1}^k \delta_i$, and hence the probability that none is violated is at least $1 - \sum_{i=1}^k \delta_i$. □

We use Lemma 4.4.10 to convert (ε, δ) uncertainty sets to larger $(0, \delta)$ uncertainty sets.

Lemma 4.4.10 ($\mathcal{L}_1$ Error Terms). *If $\mathcal{V}$ is a (ε, δ) uncertainty set, then for any $\eta \in [0, \varepsilon]$, it holds that the* Minkowski sum

$$\mathcal{V}' \doteq \mathcal{V} + \{\boldsymbol{x} \in \mathbb{R}^{nm} \mid \|\boldsymbol{x}\|_1 \leq \eta\} = \{\boldsymbol{v} + \boldsymbol{x} \mid \boldsymbol{v} \in \mathcal{V}, \|\boldsymbol{x}\|_1 \leq \eta\}$$

is a $(\varepsilon - \eta, \delta)$ uncertainty set.

Proof. Denote the true valuations as $\mathbf{V}^*$. If $\mathbf{V}^* \in \mathcal{V}'$, then we have $\|\mathbf{V} - \mathbf{V}^*\|_1 = 0 \leq \varepsilon - \eta$. We argue the case when $\mathbf{V}^* \notin \mathcal{V}'$. By Definition 1.2.3, with probability at least $1 - \delta$, there is some $\mathbf{V} \in \mathcal{V}$ such that $\|\mathbf{V} - \mathbf{V}^*\|_1 \leq \varepsilon$. We can assume that $\eta \leq \|\mathbf{V}^* - \mathbf{V}\|_1$, since if not, then $\mathbf{V}^* \in \mathcal{V}'$. Consider $\mathbf{V}' = \mathbf{V} + \eta \frac{\mathbf{V}^* - \mathbf{V}}{\|\mathbf{V}^* - \mathbf{V}\|_1}$. To show $\mathbf{V}' \in \mathcal{V}'$, we must show that $\left\|\eta \frac{\mathbf{V}^* - \mathbf{V}}{\|\mathbf{V}^* - \mathbf{V}\|_1}\right\|_1 \leq \eta$. This can be shown easily, as $\left\|\eta \frac{\mathbf{V}^* - \mathbf{V}}{\|\mathbf{V}^* - \mathbf{V}\|_1}\right\|_1 = \frac{\eta}{\|\mathbf{V}^* - \mathbf{V}\|_1} \|\mathbf{V}^* - \mathbf{V}\|_1 = \eta$. In addition,

$$\begin{aligned}
\|\mathbf{V}^* - \mathbf{V}'\|_1 &= \left\|\mathbf{V}^* - \left(\mathbf{V} + \eta \frac{\mathbf{V}^* - \mathbf{V}}{\|\mathbf{V}^* - \mathbf{V}\|_1}\right)\right\|_1 \\
&= \left(1 - \frac{\eta}{\|\mathbf{V}^* - \mathbf{V}\|_1}\right) \|\mathbf{V}^* - \mathbf{V}\|_1 \\
&= \|\mathbf{V}^* - \mathbf{V}\|_1 - \eta \\
&\leq \varepsilon - \eta,
\end{aligned}$$

where the final inequality holds with probability at least $1 - \delta$. Therefore, with probability at least $1 - \delta$, $\mathcal{V}'$ contains a point $\mathbf{V}'$ within $\varepsilon - \eta$ $\mathcal{L}_1$ distance from $\mathbf{V}^*$, and $\mathcal{V}'$ is a $(\varepsilon - \eta, \delta)$ uncertainty set.

□

We can apply Lemmas 4.4.9 and 4.4.10 sequentially to intersect arbitrary (ε, δ) uncertainty sets. We first expand them via Lemma 4.4.10 to obtain larger $(0, \delta_i)$ uncertainty sets, and we then apply Lemma 4.4.9 to obtain their intersection.

We can also apply these results to the uncertainty sets previously described. For example, the intersection of multiple axis-aligned, hyperrectangular constraints produces an axis-aligned hyperrectangle. This may occur when structural constraints defined by the market maker (e.g., in reviewer assignment, hard upper and lower bounds defined based on topic overlap) intersect with per-pair error bounds. Similarly, we might consider cases with two intersecting ellipsoidal error bounds derived from two different estimators using Theorems 4.4.7 and 4.4.8. This intersection is uninteresting if the two ellipsoids have the same centroid and one is strictly smaller than the other, but if these ellipsoids have different centroids (as when the estimators have different biases) their intersection can be quite beneficial.

We finish this section by demonstrating more concretely how to construct an uncertainty set using a logistic regression estimator, when we have κ groups of requests. Logistic regression models with bounded cross-entropy loss result in polyhedral uncertainty sets.

Assume we have a discrete set of c values $L \subseteq \mathbb{R}$, with $L = \{l_1, \dots l_c\}$. For each request $i \in N$ and expert $g \in M$ we denote the true distribution over values $p^*(l|(i,g))$ and the distribution predicted by the logistic regression model is $\hat{p}(l|(i,g))$.

We estimate the cross-entropy loss of the model on a test set T_{TEST}, where $|T_{\text{TEST}}| = t$. This test set can be segmented by the group identity of the request, such that we have $T_{\text{TEST}}^{G_1}, T_{\text{TEST}}^{G_2}, \dots T_{\text{TEST}}^{G_\kappa}$ for each of the κ groups (with sizes $t^{G_1}, \dots t^{G_\kappa}$). We assume that the test set comes from the same distribution as the agent-item pairs of the assignment problem; this can be achieved either during dataset construction or by limiting the assignments (through the $\mathbf{C}$ constraints) to better reflect the test distribution. We can also apply likelihood reweighting in our uncertainty set construction.

For a request i and expert g, the cross-entropy loss of the distribution $\hat{p}$ with respect to the distribution p is defined as

$$\mathbb{H}(p(l|(i,g)), \hat{p}(l|(i,g))) \doteq -\sum_{l \in L} p(l|(i,g)) \log \hat{p}(l|(i,g)).$$

For each T_{TEST}^{G}, we compute the mean of the cross-entropy loss,

$$\hat{\xi}_G = \frac{1}{t^G} \sum_{(i,g) \in T_{\text{TEST}}^{G}} \mathbb{H}(p(l|(i,g)), \hat{p}(l|(i,g))),$$

as well as the standard error of the mean,

$$\hat{\eta}_G = \left(\frac{1}{t^G} \sum_{(i,g) \in T^G_{\text{TEST}}} \left(\mathbb{H}(p(l|(i,g)), \hat{p}(l|(i,g))) - \hat{\xi}_G \right)^2 \right)^{\frac{1}{2}} .$$

We model the distribution over cross-entropy losses for group G as Normal$(\hat{\xi}_G, \hat{\eta}_G)$. We want an uncertainty set $\mathcal{V}$ such that the true values lie outside $\mathcal{V}$ with probability at most δ. Thus, using a union bound, we require each uncertainty set $\mathcal{V}_G$ for individual groups to contain the true valuations with probability at least $1 - \frac{\delta}{\kappa}$. We can thus give the bound that the cross entropy loss is at most $\Phi^{-1}(1 - \frac{\delta}{\kappa}, \hat{\xi}_G, \hat{\eta}_G)$, where $\Phi^{-1}(p, \mu, \sigma)$ denotes the p percentile of a normal distribution with mean μ and standard deviation σ.

For each group G with agents N_G we obtain the uncertainty set

$$\frac{1}{t^G m} \sum_{i \in N_G, g \in M} \mathbb{H}(p(l|(i,g)), \hat{p}(l|(i,g))) \leq \Phi^{-1}(1 - \frac{\delta}{\kappa}, \hat{\xi}_G, \hat{\eta}_G).$$

The bound can be made tighter if we restrict some pairs using $\mathbf{C}$, in which case the cross-entropy term on the left side is only averaged over the pairs which are not restricted.

4.4.2 Adversarial Projected Sub-Gradient Ascent Algorithm

We now present a general purpose algorithm for approximately solving Equation (4.1) for any welfare objective W over convex uncertainty sets, as long as the adversarial (worst-case) welfare can be computed in polynomial time. Theorem 4.4.11 shows that solving Equation (4.1) is NP-hard for convex uncertainty regions of this type, even for USW.

Theorem 4.4.11 (Hardness of Robust Optimization)**.** *Solving Equation* (4.1) *with* USW *is NP-hard over a convex uncertainty set $\mathcal{V}$, even for $\mathcal{V}$ with a polynomial-time adversary. In particular, robust optimization remains NP-hard even when $\mathcal{V}$ is restricted to bounded polytopes formed by intersections of polynomially many halfspaces.*

Proof. We reduce from the problem of maximizing egalitarian welfare under the reviewer assignment constraints, which is known to be NP-hard [65, 157]. In the maximal egalitarian welfare problem for reviewer assignment, we have the same feasible set of assignments $\mathcal{Z}$, as well as a fixed score matrix $\mathbf{V}$. The goal is to find an assignment $\mathbf{A}$ maximizing the minimum total score of any paper, or equivalently $\min_{i \in N} \sum_{g \in M} \mathbf{A}_{i,g} \mathbf{V}_{i,g}$.

Given an instance of max egalitarian welfare $(N, M, \mathcal{Z}, \mathbf{V})$, we now construct an instance of the general problem Equation (4.1) over a convex uncertainty set $\mathcal{V}$. The set of n requests N, m experts M, and valid assignments $\mathcal{Z}$ remain the same. To construct an uncertainty set $\mathcal{V}$, first consider the set $\mathcal{V}' = \{\mathbf{V}^{(1)}, \ldots, \mathbf{V}^{(n)}\}$, where $\mathbf{V}^{(p)}$ is defined so that $\mathbf{V}^{(p)}_{i,g} = \mathbf{V}_{i,g}$ when $i = p$ and $\mathbf{V}^{(p)}_{i,g} = 0$ when $i \neq p$. Let $\mathcal{V}$ be the convex hull of $\mathcal{V}'$. In other words, $\mathcal{V}$ is the set

$$\mathcal{V} \doteq \left\{ \mathbf{X} \in [0,1]^{n \times m} \;\middle|\; \mathbf{X} = \sum_{i=1}^{n} \alpha_i \mathbf{V}^{(i)}, \sum_{i=1}^{n} \alpha_i = 1 \right\}.$$

Since $\mathcal{V}$ is a convex set, $(N, M, \mathcal{Z}, \mathcal{V})$ is an instance of the problem in Equation (4.1) over a convex uncertainty set. Furthermore, $\mathcal{V}$ is a bounded polytope formed by the intersection of polynomially many halfspaces.

For any $\mathbf{A} \in \mathcal{Z}$, we have that

$$\mathbf{V}_{\min} \doteq \operatorname*{arg\,min}_{\mathbf{V} \in \mathcal{V}} n \,\mathrm{USW}(\mathbf{A}, \mathbf{V}) = \operatorname*{arg\,min}_{\mathbf{V} \in \mathcal{V}} \mathrm{USW}(\mathbf{A}, \mathbf{V}).$$

We will show that $\mathbf{V}_{\min} \in \mathcal{V}'$, completing the proof. Fix $\mathbf{A}$ and consider some element $\mathbf{X} \in \mathcal{V}$, where $\mathbf{X} = \sum_{i=1}^{n} \alpha_i \mathbf{V}^{(i)}$ and (w.l.o.g.) $\alpha_1, \alpha_2 > 0$. Assume also w.l.o.g. that $\sum_{g \in M} \mathbf{A}_{1,g} \mathbf{V}_{1,g} \leq \sum_{g \in M} \mathbf{A}_{2,g} \mathbf{V}_{2,g}$. Consider the alternative $\mathbf{V}' \in \mathcal{V}$ with $\mathbf{V}' = (\alpha_1 + \alpha_2)\mathbf{V}^{(1)} + \sum_{i=3}^{n} \alpha_i \mathbf{V}^{(i)}$. Then

$$
\begin{aligned}
n\,\mathrm{USW}(\mathbf{A},\mathbf{X}) &= \alpha_1 \sum_{g\in M} \mathbf{A}_{1,g}\mathbf{V}_{1,g} + \alpha_2 \sum_{g\in M} \mathbf{A}_{2,g}\mathbf{V}_{2,g} + \sum_{i=3}^{n} \alpha_i \sum_{g\in M} \mathbf{A}_{i,g}\mathbf{V}_{i,g} \\
&\geq (\alpha_1+\alpha_2) \sum_{g\in M} \mathbf{A}_{1,g}\mathbf{V}_{1,g} + \sum_{i=3}^{n} \alpha_i \sum_{g\in M} \mathbf{A}_{i,g}\mathbf{V}_{i,g} \\
&= n\,\mathrm{USW}(\mathbf{A},\mathbf{V}').
\end{aligned}
$$

Since we can always decrease $\mathrm{USW}(\mathbf{A},\mathbf{V})$ by restricting the support of the score matrices (i.e., setting some $\alpha_i \neq 0$ to be 0), the minimal USW is reached for some $\mathbf{V} \in \mathcal{V}'$.

Therefore, since for any given $\mathbf{A}$, we have that

$$
\min_{\mathbf{V}\in\mathcal{V}'} n\,\mathrm{USW}(\mathbf{A},\mathbf{V}) = \min_{i\in N} \sum_{g\in M} \mathbf{A}_{i,g}\mathbf{V}_{i,g},
$$

the maximizer for the maximin USW problem and the maximizer for the egalitarian welfare problem are equivalent. In other words, if we can efficiently compute a robust solution to Equation (4.1) with USW, we can solve the egalitarian welfare problem. □

Due to this hardness result, we outline an approach to approximately solve Equation (4.1) efficiently for convex uncertainty sets with polynomial-time adversaries. We start by allowing *fractional* (rather than binary) assignments. We then apply *sub-gradient ascent* to approximate $\arg\max_{\tilde{\mathbf{A}}\in\tilde{\mathcal{Z}}} \inf_{\mathbf{V}\in\mathcal{V}} \mathrm{USW}(\tilde{\mathbf{A}},\mathbf{V})$, where $\tilde{\mathcal{Z}}$ is the convex closure of the feasible set of discrete allocations $\mathcal{Z}$. When the sub-gradient ascent algorithm terminates, we randomly round the assignment to a binary assignment.

In particular, we present Algorithm 8, termed adversarial projected sub-gradient ascent. Adversarial projected sub-gradient ascent applies an iterative adversarial optimization strategy to the objective. In each iteration j, we take an *adversary step*, which identifies the pessimal $\mathbf{V}^{(j)}$ given assignment $\mathbf{A}^{(j-1)}$. We then take a *gradient ascent step* from $\mathbf{A}^{(j-1)}$ to $\mathbf{A}^{(j)}$ assuming the valuation matrix remains fixed at $\mathbf{V}^{(j)}$, followed by a *projection step*, which

Algorithm 8 Adversarial Projected Sub-gradient Ascent

```
Require: Error tolerance $\varepsilon$, sub-gradient norm bound $\lambda$, uncertainty set $\mathcal{V}$, constrained allocation space $\tilde{\mathcal{Z}}$, total
    demands $K = \sum_{i \in N} \underline{\mathbf{k}}_i^N$ (i.e., the total number of assignments required)
Initialize $\mathbf{V}^{(0)} \in \mathcal{V}$ arbitrarily
$\mathbf{A}^{(0)} \leftarrow \operatorname*{arg\,max}_{\mathbf{A} \in \mathcal{Z}} \mathrm{USW}(\mathbf{A}, \mathbf{V}^{(0)})$        // Initialize $\mathbf{A}^{(0)}$ to optimize $\mathbf{V}^{(0)}$ (via LP reduction)
$\hat{\mathbf{A}} \leftarrow \mathbf{A}^{(0)}; \hat{w} \leftarrow -\infty$        // Maintain best allocation $\hat{\mathbf{A}}$ and adversarial welfare $\hat{w}$
$t \leftarrow \lceil 2K(\frac{\lambda}{\varepsilon})^2 \rceil; \alpha \leftarrow \frac{\varepsilon}{\lambda^2}$        // Compute sufficient step count $t$ and step size $\alpha$
for $j \in \{1, 2, \dots t\}$ do
    $\mathbf{V}^{(j)} \leftarrow \operatorname*{arg\,inf}_{\mathbf{V} \in \mathcal{V}} \mathrm{USW}(\mathbf{A}^{(j-1)}, \mathbf{V})$        // Adversary selects $\mathbf{V}^{(j)}$ from $\mathcal{V}$
    if $\mathrm{USW}(\mathbf{A}^{(j-1)}, \mathbf{V}^{(j)}) > \hat{w}$        // Update $\hat{\mathbf{A}}$ if adversarial welfare beats previous best
        $\hat{\mathbf{A}} \leftarrow \mathbf{A}^{(j-1)}; \hat{w} \leftarrow \mathrm{USW}(\mathbf{A}^{(j-1)}, \mathbf{V}^{(j)})$
    $\mathbf{A}^{(j)} \leftarrow \mathbf{A}^{(j-1)} + \alpha \nabla_{\mathbf{A}^{(j-1)}} \mathrm{USW}(\mathbf{A}^{(j-1)}, \mathbf{V}^{(j)})$        // Update allocation with a sub-gradient step
    $\mathbf{A}^{(j)} \leftarrow \operatorname*{arg\,min}_{\tilde{\mathbf{A}} \in \tilde{\mathcal{Z}}} \|\tilde{\mathbf{A}} - \mathbf{A}^{(j)}\|_2$        // $\mathcal{L}_2$ projection onto feasible allocation set $\tilde{\mathcal{Z}}$
return ROUND($\hat{\mathbf{A}}$)        // Sample integral assignment
```

ensures $\mathbf{A}^{(j)}$ remains feasible (i.e., does not violate any constraints on assignments). The gradient ascent step is valid since the gradient $\nabla_{\mathbf{A}} \mathrm{USW}(\mathbf{A}, \operatorname{arg\,inf}_{\mathbf{V} \in \mathcal{V}} \mathrm{USW}(\mathbf{A}, \mathbf{V}))$ is an element of the sub-gradient $\nabla_{\mathbf{A}} \inf_{\mathbf{V} \in \mathcal{V}} \mathrm{USW}(\mathbf{A}, \mathbf{V})$.

The number of iterations required for convergence depends on the *gradient norm bound* λ, which is the smallest term such that $\|\nabla_{\mathbf{A}} \mathrm{USW}(\mathbf{A}, \mathbf{V})\|_2 \leq \lambda$ for all $\mathbf{A}$ and $\mathbf{V}$. We approximate the maximin optimal continuous matrix within an error of ε in number of iterations polynomial in λ, $\frac{1}{\varepsilon}$, and the total demand for experts K. The time complexity of adversarial projected sub-gradient ascent also depends on the *adversarial minimization* and *projection* steps, but when these take polynomial time, so does the full algorithm. We state the convergence results in Proposition 4.4.12. The proof applies standard convergence results for sub-gradient descent [151].

Proposition 4.4.12 (Sub-gradient Ascent Efficiency)**.** *Let λ denote an upper bound on the $\mathcal{L}_2$ norm of the sub-gradient elements $\nabla_{\mathbf{A}} \mathrm{USW}(\mathbf{A}, \mathbf{V})$ used during the algorithm. The sub-gradient ascent component converges to within ε of the maximin optimal continuous assignment in $\lceil 2K(\frac{\lambda}{\varepsilon})^2 \rceil$ iterations. The algorithm runs in time $O\big(2KC(\frac{\lambda}{\varepsilon})^2\big)$, where C is the time cost of one adversary and projection step.*

We use the following result from [151].

Theorem 4.4.13. *Suppose that $w\colon \mathbb{R}^n \mapsto \mathbb{R}$ is a convex function with minimizer w^* and optimal set $X^* = \{\boldsymbol{x} \in \mathbb{R}^n : w(\boldsymbol{x}) = w^*\}$. Let $\boldsymbol{x}_0$ denote the starting point for sub-gradient descent. Suppose we have an upper bound $\rho \geq \max_{\boldsymbol{x} \in X^*} \|\boldsymbol{x}_0 - \boldsymbol{x}\|_2$. Suppose we also have an upper bound λ for the Euclidean norm of the sub-gradient at each step of the sub-gradent descent algorithm. After t steps of sub-gradient descent with constant step size α, if $\hat{w}$ represents the best function value found after t steps, then*

$$\hat{w} - w^* \leq \frac{\rho^2 + \lambda^2 \alpha^2 t}{2\alpha t}.$$

Proof. First, we show that $\rho \leq \sqrt{2K}$ (recall ρ is defined as the upper bound $\rho \geq \max_{\boldsymbol{x} \in X^*} \|\boldsymbol{x}_0 - \boldsymbol{x}\|_2$). Because the initial assignment $\mathbf{A}^{(0)}$ and the optimal assignment $\mathbf{A}^*$ both lie in $\tilde{\mathcal{Z}}$, they both must satisfy the constraint that for all requests $i \in N$, $\sum_{g \in M} \mathbf{A}_{i,g} = \underline{\mathbf{k}}_i^N$. This implies that even if there is no overlap in the two assignments, they will differ on at most $2K$ entries. Taking the Euclidean norm over the difference of the two assignments in the worst case gives the bound on ρ.

Our function is concave, and we are maximizing over the input space, so we can apply Theorem 4.4.13 to minimize its negative. From Theorem 4.4.13 we know that after t iterations of sub-gradient ascent with step size α, the error $\varepsilon = \inf_{\mathbf{V} \in \mathcal{V}} \mathrm{USW}(\mathbf{A}^*, \mathbf{V}) - \inf_{\mathbf{V} \in \mathcal{V}} \mathrm{USW}(\hat{\mathbf{A}}, \mathbf{V})$ satisfies

$$\varepsilon \leq \frac{\rho^2 + \lambda^2 \alpha^2 t}{2\alpha t}.$$

The right-hand side is minimized at $\alpha = \dfrac{\rho}{\lambda \sqrt{t}}$. If we substitute this value of α, we get

$$\varepsilon \leq \frac{\rho \lambda}{\sqrt{t}}.$$

Solving for t then yields

$$t \geq \left(\frac{\rho\lambda}{\varepsilon}\right)^2 .$$

Since $\rho \geq \sqrt{2K}$, the larger number of iterations $t \geq 2K(\frac{\lambda}{\varepsilon})^2$ will suffice. Substituting this value of T into the equation given for α above yields $\alpha = \frac{\varepsilon}{\lambda^2}$. □

Although the bound on the number of iterations can be quite large, it proves the convex relaxation of Equation (4.1) is solvable in polynomial time, as long as the adversary and projection steps can be solved in polynomial time and λ is bounded. In addition, the required number of iterations until convergence will typically be much smaller in practice.

The complexity result improves in the case of (truncated) ellipsoidal uncertainty sets. The *adversarial minimization* step requires polynomial time under truncated ellipsoidal uncertainty sets, as it is a *linear objective* under *convex quadratic constraints* (and box constraints), which is a *second-order conic program*, and the *projection step* always requires polynomial time, as it is a *convex quadratic* objective under *linear* constraints (i.e., the assignment constraints $\tilde{\mathcal{Z}}$). The bound λ can be difficult to compute in the general case, but we show λ is typically well-bounded in the case of truncated ellipsoidal uncertainty sets.

Corollary 4.4.14 (Sub-gradient Ascent Efficiency under Ellipsoidal Uncertainty)**.** *For a truncated ellipsoidal uncertainty set, the sub-gradient ascent component Algorithm 8 converges to within ε of the maximin optimal continuous assignment in $O\left(\frac{2Km}{n\varepsilon^2}\right)$ iterations.*

Proof. Because the sub-gradient is always contained in $\mathcal{V}$, we can easily upper-bound the norm of it. We have for all $\mathbf{A} \in \tilde{\mathcal{Z}}$ and $\mathbf{V} \in \mathcal{V}$ that

$$\|\nabla_{\mathbf{A}} \operatorname{USW}(\mathbf{A}, \mathbf{V})\|_2 \leq \max_{\mathbf{V} \in \mathcal{V}} \tfrac{1}{n} \|\mathbf{V}\|_2 .$$

If the ellipsoid has center $\boldsymbol{\mu}_{\mathcal{V}}$ and radius q, then by the triangle inequality,

$$\max_{\mathbf{V} \in \mathcal{V}} \|\mathbf{V}\|_2 \leq \|\boldsymbol{\mu}_{\mathcal{V}}\|_2 + q,$$

where q can be computed as the maximum eigenvalue of $\Sigma_{\mathcal{V}}$. Both $\|\boldsymbol{\mu}_{\mathcal{V}}\|_2$ and q are $O(\sqrt{nm})$, so we see that λ is $O(\sqrt{\frac{m}{n}})$. Applying Proposition 4.4.12 completes the proof. □

Finally, we can round using the extended Birkhoff von Neumann decomposition sampling algorithm [28, 64, 80]. This sampling algorithm generates an integral sample $\mathbf{A}_{\text{ROUND}}$ from the distribution defined by the continuous assignment matrix $\hat{\mathbf{A}}$. The sample $\mathbf{A}_{\text{ROUND}}$ still satisfies the constraints of $\tilde{\mathcal{Z}}$, and $\mathbb{E}[\mathbf{A}_{\text{ROUND}}] = \hat{\mathbf{A}}$.

The time complexity of this sampling algorithm is $O(mn(m+n))$, which is typically negligible compared to the complexity of sub-gradient ascent. Theorem 4.4.3 bounds the expected and probabilistic regret of adversarial projected sub-gradient ascent.

4.4.3 More Efficient Algorithms for Special Cases

The adversarial projected sub-gradient ascent method does not exploit the structure of the specific instantiations of Equation (4.1). It is often computationally expensive or intractable, as demonstrated empirically in Section 4.6. Despite the inherent complexities of these problems, we show that under specific assumptions, these problems reduce to forms that are easier to optimize. We then discuss a range of algorithms for efficiently optimizing the simplified problems.

We focus on the class of uncertainty sets defined by linear and ellipsoidal constraints:

$$\mathcal{V} = \left\{ \boldsymbol{v} \in \mathbb{R}^{nm} \mid \forall i \in [1, l],\, (\boldsymbol{v} - \hat{\boldsymbol{v}}_i)\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{v} - \hat{\boldsymbol{v}}_i) \leq \boldsymbol{r}_i^2, \boldsymbol{Q}\boldsymbol{v} \succeq \boldsymbol{e}, \boldsymbol{v} \succeq 0 \right\},$$

where the i^{th} ellipsoidal uncertainty set has center $\hat{\boldsymbol{v}}_i \in \mathbb{R}^{nm}_{0+}$, covariance matrix $\boldsymbol{\Sigma}_i \in \mathbb{R}^{nm \times nm}$, with radius $\boldsymbol{r}_i \in \mathbb{R}$, $\boldsymbol{Q} \in \mathbb{R}^{k \times nm}$, and $\boldsymbol{e} \in \mathbb{R}^k$ (k is the number of linear constraints). We will further assume that the covariance matrices corresponding to the ellipsoidal uncertainty sets are positive semi-definite. This limitation on the structure of uncertainty sets is not too restrictive; it is possible to construct such uncertainty sets for standard models using statistical bounds, as demonstrated for logistic regression in Section 4.4.1. When an integer allocation is either not feasible or computationally intractable, we relax the set of feasible integer assignments $\mathcal{Z}$ to the set of feasible continuous allocations $\tilde{\mathcal{Z}}$.

Utilitarian Welfare. We first consider allocations that optimize the utilitarian welfare under the worst valuation matrix in the uncertainty set. We formulate the problem as:

$$\max_{\boldsymbol{a} \in \mathcal{Z}} \min_{\boldsymbol{v} \in \mathcal{V}} \mathrm{USW}(\boldsymbol{a}, \boldsymbol{v}). \tag{4.5}$$

The objective and constraints of the inner-minimization problem described in (4.5) are convex, which confirms that the inner-minimization problem is also convex. The problem is strictly feasible, which satisfies Slater's condition [25] for strong duality. Therefore, by taking the dual of the inner-minimization problem, we can simplify the problem in (4.5) into a single equivalent maximization problem. We provide the dual formation in Proposition 4.4.15.

In the dual, let $\boldsymbol{\beta} \in \mathbb{R}^k_{0+}$ be the dual variable corresponding to the linear constraints $\boldsymbol{Q}\boldsymbol{v} \succeq \boldsymbol{e}$, $\boldsymbol{\lambda} \in \mathbb{R}^l_{0+}$ be the dual variable associated with the ellipsoidal constraints, and $\boldsymbol{\xi} \in \mathbb{R}^{nm}$ be the variable that combines the primal variable $\boldsymbol{a}$ with the dual variable of the non-negativity constraint on $\boldsymbol{v}$ for variable elimination. Furthermore, we define a set of feasible $\boldsymbol{\xi}$ as $\tilde{\mathcal{Z}}_\xi = \mathcal{Z} - \mathbb{R}^{nm}_{0+}$, which is Pareto-dominated by $\mathcal{Z}$.

Proposition 4.4.15. *The problem in* (4.5) *is equivalent to solving*

$$\max_{\substack{\boldsymbol{\xi} \in \tilde{\mathcal{Z}}_\xi, \boldsymbol{\lambda} \in \mathbb{R}^l_{0+}, \\ \boldsymbol{\beta} \in \mathbb{R}^k_{0+}}} \boldsymbol{p}^\intercal \boldsymbol{\Sigma}_{[L]}^{-1} \boldsymbol{q}^\intercal + \boldsymbol{\beta}^\intercal \boldsymbol{e} - \frac{1}{4} \|\boldsymbol{p}^\intercal \boldsymbol{\Sigma}_{[L]}^{-1/2}\|_2^2 + \sum_{i=1}^{l} \boldsymbol{\lambda}_i \|\hat{\boldsymbol{v}}_i^\intercal \boldsymbol{\Sigma}_i^{-1/2}\|_2^2 - \|\boldsymbol{q}^\intercal \boldsymbol{\Sigma}_{[L]}^{-1/2}\|_2^2 - \sum_{i=1}^{l} \boldsymbol{\lambda}_i \boldsymbol{r}_i^2, \tag{4.6}$$

where $\boldsymbol{p} = -\boldsymbol{\beta}^\intercal \boldsymbol{Q} + \boldsymbol{\xi}$ *and* $\boldsymbol{q} = \sum_{i=1}^{l} \boldsymbol{\lambda}_i \hat{\boldsymbol{v}}_i^\intercal \boldsymbol{\Sigma}_i^{-1}$, *and* $\boldsymbol{\Sigma}_{[L]} = \sum_{i=1}^{l} \boldsymbol{\lambda}_i \boldsymbol{\Sigma}_i^{-1}$. *Let* $\boldsymbol{\xi}^*$ *be the optimal* $\boldsymbol{\xi}$ *in* (4.6)*. Then, the optimal allocation* $\boldsymbol{a}^*$ *can be derived from* $\boldsymbol{\xi}^*$ *by solving:*

$$\frac{1}{n} \boldsymbol{a} \preceq \boldsymbol{\xi}^*, \, \boldsymbol{a} \in \mathcal{Z} \ ,$$

Proof. Let $\boldsymbol{a}' = \frac{\boldsymbol{a}}{n}$. Consider the inner-minimization problem:

$$\begin{aligned} &\min_{\boldsymbol{v}\in\mathbb{R}^{nm}} \text{USW}(\boldsymbol{a},\boldsymbol{v}) \\ &\forall i \in [1,l], (\boldsymbol{v}-\hat{\boldsymbol{v}}_i)\boldsymbol{\Sigma}_i^{-1/2}(\boldsymbol{v}-\hat{\boldsymbol{v}}_i) \leq \boldsymbol{r}_i^2 \\ &\boldsymbol{Qv} \succeq \boldsymbol{e} \\ &\boldsymbol{v} \succeq 0, \end{aligned} \tag{4.7}$$

We will use the Lagrangian method for computing the dual of the above problem. The Lagrangian for the above problem is given by

$$\begin{aligned} L(\boldsymbol{v},\boldsymbol{\lambda}\in\mathbb{R}^l_{0+},\boldsymbol{\beta}\in\mathbb{R}^k,\boldsymbol{\zeta}\in\mathbb{R}^{nm}) = \boldsymbol{a}'^{\intercal}\boldsymbol{v} + \sum_{i=1}^{l}\boldsymbol{\lambda}_i\left((\boldsymbol{v}-\hat{\boldsymbol{v}}_i)\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{v}-\hat{\boldsymbol{v}}_i)-\boldsymbol{r}_i^2\right) \\ -\boldsymbol{\beta}^{\intercal}(Q\boldsymbol{v}-\boldsymbol{e})-\boldsymbol{\zeta}^{\intercal}\boldsymbol{v}. \end{aligned} \tag{4.8}$$

From the first-order optimality conditions, we get

$$\begin{aligned} \frac{\partial L(\boldsymbol{v},\boldsymbol{\lambda}\in\mathbb{R}^l_{0+},\boldsymbol{\beta}\in\mathbb{R}^k,\boldsymbol{\zeta}\in\mathbb{R}^{nm})}{\partial\boldsymbol{v}} &= 0 \\ \boldsymbol{a}' + \sum_{i=1}^{l}2\boldsymbol{\lambda}_i(\boldsymbol{v}-\hat{\boldsymbol{v}}_i)\boldsymbol{\Sigma}_i^{-1} - \boldsymbol{\beta}^{\intercal}Q - \boldsymbol{\zeta} &= 0 \\ \implies \boldsymbol{v} = \frac{\sum_{i=1}^{l}2\boldsymbol{\lambda}_i\hat{\boldsymbol{v}}_i^{\intercal}\boldsymbol{\Sigma}_i^{-1} - (\boldsymbol{a}'-\boldsymbol{\beta}^{\intercal}Q-\boldsymbol{\zeta})}{\sum_{i=1}^{l}2\boldsymbol{\lambda}_i\boldsymbol{\Sigma}_i^{-1}}&. \end{aligned}$$

Substituting this value of $\boldsymbol{v}$ in (4.8), we get,

$$\begin{aligned}
\max_{\substack{\boldsymbol{\lambda}\in\mathbb{R}^{l},\\ \boldsymbol{\beta}\in\mathbb{R}^{nm},\\ \boldsymbol{\zeta}\in\mathbb{R}^{nm}}} \; & -\frac{1}{4}\left((\boldsymbol{a}'-\boldsymbol{\beta}^{\intercal}Q-\boldsymbol{\zeta})^{\intercal}\left(\sum_{i=1}^{l}\boldsymbol{\lambda}_i\boldsymbol{\Sigma}_i^{-1}\right)^{-1}(\boldsymbol{a}'-\boldsymbol{\beta}^{\intercal}Q-\boldsymbol{\zeta})\right)+\sum_{i=1}^{l}\boldsymbol{\lambda}_i\hat{\boldsymbol{v}}_i^{\intercal}\boldsymbol{\Sigma}_i^{-1}\hat{\boldsymbol{v}}_i \\
& -\left(\sum_{i=1}^{l}\boldsymbol{\lambda}_i\hat{\boldsymbol{v}}_i\boldsymbol{\Sigma}_i^{-1}\right)\left(\sum_{i=1}^{l}\boldsymbol{\lambda}_i\boldsymbol{\Sigma}_i^{-1}\right)^{-1}\left(\sum_{i=1}^{l}\boldsymbol{\lambda}_i\hat{\boldsymbol{v}}_i\boldsymbol{\Sigma}_i^{-1}\right)^{\intercal} \\
& +(\boldsymbol{a}'-\boldsymbol{\beta}^{\intercal}Q-\boldsymbol{\zeta})^{\intercal}\left(\sum_{i=1}^{l}\boldsymbol{\lambda}_i\boldsymbol{\Sigma}_i^{-1}\right)^{-1}\left(\sum_{i=1}^{l}\boldsymbol{\lambda}_i\hat{\boldsymbol{v}}_i^{\intercal}\boldsymbol{\Sigma}_i^{-1}\right)^{\intercal}-\sum_{i=1}^{l}\boldsymbol{\lambda}_i\boldsymbol{r}_i^2+\boldsymbol{\beta}^{\intercal}\boldsymbol{e} \qquad (4.9)\\
& \boldsymbol{\lambda}\succeq 0 \\
& \boldsymbol{\beta}\succeq 0 \\
& \boldsymbol{\zeta}\geq 0
\end{aligned}$$

Using change of variables $\boldsymbol{\zeta}=\boldsymbol{a}'-\boldsymbol{\xi}$, and combining the dual with the outer-maximization problem in (4.5), we get

$$\begin{aligned}
\max_{\substack{\boldsymbol{a}\in\mathcal{Z},\boldsymbol{\lambda}\in\mathbb{R}^{l}_{0+},\boldsymbol{\beta}\in\mathbb{R}^{k}_{0+},\\ \boldsymbol{\zeta}\in\mathbb{R}^{nm}_{0+},\boldsymbol{\xi}\in\mathbb{R}^{nm}}} \; & -\frac{1}{4}\|\boldsymbol{p}^{\intercal}\boldsymbol{\Sigma}_{[L]}^{-1/2}\|_2^2+\sum_{i=1}^{l}\boldsymbol{\lambda}_i\|\hat{\boldsymbol{v}}_i^{\intercal}\boldsymbol{\Sigma}_i^{-1/2}\|_2^2 \\
& -\|\boldsymbol{q}^{\intercal}\boldsymbol{\Sigma}_{[L]}^{-1/2}\|_2^2+\boldsymbol{p}^{\intercal}\boldsymbol{\Sigma}_{[L]}^{-1}\boldsymbol{q}^{\intercal}-\sum_{i=1}^{l}\boldsymbol{\lambda}_i\boldsymbol{r}_i^2+\boldsymbol{\beta}^{\intercal}\boldsymbol{e} \qquad (4.10)\\
& \text{s.t. } \boldsymbol{\zeta}=\boldsymbol{a}'-\boldsymbol{\xi},
\end{aligned}$$

where $\boldsymbol{p}=\boldsymbol{\beta}^{\intercal}Q+\boldsymbol{\xi}$, $\boldsymbol{\Sigma}_{[L]}=\sum_{i=1}^{l}\boldsymbol{\lambda}_i\boldsymbol{\Sigma}_i^{-1}$, and $\boldsymbol{q}=\sum_{i=1}^{l}\boldsymbol{\lambda}_i\hat{\boldsymbol{v}}_i^{\intercal}\boldsymbol{\Sigma}_i^{-1}$. Note that the above optimization problem is concave; from affine-composition rule in convex optimization, we retain the concavity of the objective after the change of variable and the allocation $\boldsymbol{a}$ only appears in a linear constraint which is convex.

We further simplify the above problem by eliminating the allocation variables $\boldsymbol{a}$ and the dual variable $\boldsymbol{\zeta}$ and subsequently deriving them from the solution of the resultant problem.

Note that in the above problem $\boldsymbol{a}'-\boldsymbol{\zeta}=\boldsymbol{\xi}$. Let $(\boldsymbol{a}^*,\boldsymbol{\zeta}^*)$ represent an optimal $(\boldsymbol{a},\boldsymbol{\zeta})$ pair for the problem in (4.10). Note that there can be multiple pairs of $(\boldsymbol{a},\boldsymbol{\zeta})$ that are optimal.

Let $\boldsymbol{a}'^{*} = \frac{1}{n}\boldsymbol{a}^{*}$. To eliminate $\boldsymbol{\zeta}$ and $\boldsymbol{a}$, we need to find a set of feasible $\boldsymbol{\xi}$, which we denote by $\tilde{\mathcal{Z}}_{\xi}$, such that there exists a $\boldsymbol{\xi}' \in \tilde{\mathcal{Z}}_{\xi}$ with $\boldsymbol{\xi}' = \boldsymbol{a}'^{*} - \boldsymbol{\zeta}^{*}$ for at least one optimal pair $(\boldsymbol{a}^{*}, \boldsymbol{\zeta}^{*})$. If there exists such a $\boldsymbol{\xi}' \in \tilde{\mathcal{Z}}_{\xi}$, then $\boldsymbol{\xi}'$ maximizes the objective in (4.10). Furthermore, $\tilde{\mathcal{Z}}_{\xi} = \mathcal{Z} - \mathbb{R}^{nm}_{0+} = \{\boldsymbol{\xi} \in \mathbb{R}^{nm} \mid \forall i \in N : \sum_{j=1}^{m} \boldsymbol{\xi}_{im+j} \leq \overline{\mathbf{k}}^{N}_{i}, \forall g_j \in M : \sum_{i=1}^{n} \boldsymbol{\xi}_{im+j} \leq \overline{\mathbf{k}}^{M}_{j}, \boldsymbol{\xi} \preceq \boldsymbol{c}\}$ satisfies the above criteria for optimality.

Thus, we can break down the problem in (4.10) into two sub-problems. In the first problem, we obtain the optimal value of $\boldsymbol{\lambda}$, $\boldsymbol{\xi}$ and $\boldsymbol{\beta}$ by solving:

$$\begin{aligned}\boldsymbol{\zeta}^{*}, \boldsymbol{\beta}^{*}, \boldsymbol{\xi}^{*} = \underset{\substack{\boldsymbol{\zeta} \in \mathbb{R}^{nm}_{0+}, \boldsymbol{\beta} \in \mathbb{R}^{k}_{0+}, \\ \boldsymbol{\xi} \in \tilde{\mathcal{Z}}_{\xi}}}{\arg\max} \quad & -\frac{1}{4} \|\boldsymbol{p}^{\intercal} \boldsymbol{\Sigma}_{[L]}^{-1/2}\|_2^2 + \sum_{i=1}^{l} \boldsymbol{\lambda}_i \|\hat{\boldsymbol{v}}_i^{\intercal} \boldsymbol{\Sigma}_i^{-1/2}\|_2^2 - \|\boldsymbol{q}^{\intercal} \boldsymbol{\Sigma}_{[L]}^{-1/2}\|_2^2 \\ & + \boldsymbol{p}^{\intercal} \boldsymbol{\Sigma}_{[L]}^{-1} \boldsymbol{q}^{\intercal} - \sum_{i=1}^{l} \boldsymbol{\lambda}_i \boldsymbol{r}_i^2 + \boldsymbol{\beta}^{\intercal} \boldsymbol{e},\end{aligned}$$

where $\boldsymbol{p} = -\boldsymbol{\beta}^{\intercal} \boldsymbol{Q} + \boldsymbol{\xi}$ and $\boldsymbol{q} = \sum_{i=1}^{l} \boldsymbol{\lambda}_i \hat{\boldsymbol{v}}_i^{\intercal} \boldsymbol{\Sigma}_i^{-1}$, and $\boldsymbol{\Sigma}_{[L]} = \sum_{i=1}^{l} \boldsymbol{\lambda}_i \boldsymbol{\Sigma}_i^{-1}$. The set of optimal $(\boldsymbol{a}, \boldsymbol{\zeta})$ pairs are computed by solving a system of equations:

$$\{(\boldsymbol{a}, \boldsymbol{\zeta}) \mid \boldsymbol{a} \in \mathcal{Z}, \frac{1}{n} \cdot \boldsymbol{a} - \boldsymbol{\zeta} = \boldsymbol{\xi}^{*}, \boldsymbol{\zeta} \in \mathbb{R}^{nm}_{0+}\}.$$

□

Proposition 4.4.15 shows that the optimal allocation for the problem in Equation (4.5) can be computed by first solving the concave cubic program in Equation (4.6) to obtain $\boldsymbol{\xi}^{*}$ and then deriving the optimal allocation $\boldsymbol{a}^{*}$ from $\boldsymbol{\xi}^{*}$ by solving a system of equations. The problem in Equation (4.6) is a single maximization problem with fewer variables and constraints as compared to the max-min problem in (4.5), making it simpler to solve. When the valuation uncertainty set is polyhedral, the problem in (4.6) simplifies further into a linear program (LP) which can be solved efficiently using standard LP solvers like Gurobi [73]. We present this result in Corollary 4.4.16. Moreover, when the valuation uncertainty set has a single ellipsoidal constraint with a non-negativity constraint, we can compute the exact optimal solution using iterated quadratic programming (IQP), as described in Corollary 4.4.17.

Corollary 4.4.16. *In the case where the uncertainty set $\mathcal{V}$ is defined purely by linear constraints, i.e., $\mathcal{V} = \{\boldsymbol{v} \in \mathbb{R}^{nm}_{0+} \mid \boldsymbol{Q}\boldsymbol{v} \succeq \boldsymbol{e}\}$, the optimal allocation $\boldsymbol{a}^*$ for the problem in* (4.5) *can be computed by solving the integer linear program:*

$$\begin{aligned} &\max_{\boldsymbol{a}\in\mathcal{Z},\boldsymbol{\beta}\in\mathbb{R}^k_{0+}} \boldsymbol{\beta}^\intercal \boldsymbol{e} \\ &\boldsymbol{\beta}^\intercal \boldsymbol{Q} \preceq \frac{1}{n}\boldsymbol{a}. \end{aligned} \tag{4.11}$$

This problem is totally unimodular, and therefore the optimal fractional solution for $\boldsymbol{a}$ corresponds to the optimal integer solution.

Proof. Consider the following inner-minimization problem. Let $\boldsymbol{a}' = \frac{\boldsymbol{a}}{n}$. Consider the inner-minimization problem:

$$\begin{aligned} &\min_{\boldsymbol{v}\in\mathbb{R}^{nm}} \mathrm{USW}(\boldsymbol{a},\boldsymbol{v}) \\ &\boldsymbol{Q}\boldsymbol{v} \succeq \boldsymbol{e} \\ &\boldsymbol{v} \succeq 0, \end{aligned}$$

We compute the dual of the above problem using the Lagrangian method.

$$\begin{aligned} L(\boldsymbol{v},\boldsymbol{\lambda}\in\mathbb{R}^l_{0+},\boldsymbol{\beta}\in\mathbb{R}^k_{0+},\boldsymbol{\zeta}\in\mathbb{R}^{nm}_{0+}) &= \boldsymbol{a}'^\intercal\boldsymbol{v} - \boldsymbol{\beta}^\intercal(\boldsymbol{Q}\boldsymbol{v}-\boldsymbol{e}) - \boldsymbol{\zeta}^\intercal\boldsymbol{v} \\ &= (\boldsymbol{a}' - \boldsymbol{\beta}^\intercal\boldsymbol{Q} - \boldsymbol{\zeta})^\intercal\boldsymbol{v} + \boldsymbol{\beta}^\intercal\boldsymbol{e} \\ L(\boldsymbol{\lambda}\in\mathbb{R}^l_{0+},\boldsymbol{\beta}\in\mathbb{R}^k_{0+},\boldsymbol{\zeta}\in\mathbb{R}^{nm}_{0+}) &= \begin{cases} \boldsymbol{\beta}^\intercal\boldsymbol{e} & (\boldsymbol{a}' - \boldsymbol{\beta}^\intercal\boldsymbol{Q} - \boldsymbol{\zeta}) \succeq 0 \\ -\infty & \text{otherwise} \end{cases} \end{aligned} \tag{4.12}$$

Therefore, the dual is given by

$$\begin{aligned} &\max_{\boldsymbol{\beta}\in\mathbb{R}^k_{0+},\boldsymbol{\zeta}\in\mathbb{R}^{nm}_{0+}} \boldsymbol{\beta}^\mathsf{T}\boldsymbol{e} \\ &\boldsymbol{\beta}^\mathsf{T}\boldsymbol{Q}-\boldsymbol{\zeta}\preceq\boldsymbol{a}'. \end{aligned} \tag{4.13}$$

Since $\boldsymbol{\zeta}$ is non-negative, we can eliminate it to get

$$\begin{aligned} &\max_{\boldsymbol{\beta}\in\mathbb{R}^k_{0+}} \boldsymbol{\beta}^\mathsf{T}\boldsymbol{e} \\ &\boldsymbol{\beta}^\mathsf{T}Q\preceq\boldsymbol{a}'. \end{aligned} \tag{4.14}$$

Combining the dual with the outer-maximization problem in (4.5) yields the final result.

□

Corollary 4.4.17. *Suppose the set $\mathcal{V}$ in* (4.5) *is defined by a single truncated ellipsoidal constraint $\mathcal{V}=\{\boldsymbol{v}\in\mathbb{R}^{nm}_{0+}\mid(\boldsymbol{v}-\hat{\boldsymbol{v}})\boldsymbol{\Sigma}_i^{-1}(\boldsymbol{v}-\hat{\boldsymbol{v}})\le r^2\}$. The problem in* (4.5) *is equivalent to*

$$\max_{\lambda\in\mathbb{R}_{0+},\boldsymbol{\xi}\in\tilde{\mathcal{Z}}_\xi}\left(\boldsymbol{\xi}^\mathsf{T}\hat{\boldsymbol{v}}-\frac{\|\boldsymbol{\xi}^\mathsf{T}\boldsymbol{\Sigma}^{\frac{1}{2}}\|_F^2}{4\lambda}-\lambda r^2\right). \tag{4.15}$$

The exact optimal solution ($\lambda^*,\boldsymbol{\xi}^*$) to Equation (4.15) *can be computed by alternately performing two steps until convergence: first, fixing $\boldsymbol{\xi}$ and optimizing λ, i.e., $\lambda=\|\boldsymbol{\xi}^\mathsf{T}\boldsymbol{\Sigma}^{\frac{1}{2}}\|_F^2/2r$, and second, fixing λ and solving a concave quadratic program to optimize $\boldsymbol{\xi}$. Furthermore, the optimal allocation $\boldsymbol{a}^*$ can be computed from $\boldsymbol{\xi}^*$ as in Proposition 4.4.15.*

Proof. Setting $k=0$ and $l=1$ in (4.6) yields the stated optimization problem. This dual is concave in λ and $\boldsymbol{\xi}$, since it is a combination of affine functions on λ and $\boldsymbol{\xi}$ and a concave function of λ and $\boldsymbol{\xi}$. Unfortunately, the objective in (4.15) is a cubic polynomial that is difficult to optimize exactly using standard solvers. However, since the objective is concave and differentiable, we can leverage block coordinate descent to achieve the global optimal solution, i.e., we can alternate between optimizing $(\boldsymbol{\zeta},\boldsymbol{\xi},\boldsymbol{a})$, which is a quadratic problem, and optimizing ($\boldsymbol{\lambda}$) which has a closed form solution $\lambda = \frac{\|\boldsymbol{\xi}^\intercal \Sigma^{\frac{1}{2}}\|_{\mathrm{F}}^2}{2r}$, until convergence. Since the objective is concave and differentiable, the above algorithm is guaranteed to converge to the global optimal solution [14, 177, 179].

□

Group Egalitarian Welfare. We now consider the problem where we aim to maximize the welfare corresponding to the worst group while using the robust approach for handling uncertainty. We can represent this problem as

$$\max_{\boldsymbol{a} \in \mathcal{Z}} \min_{\boldsymbol{v} \in \mathcal{V}} \min_{G \in \mathcal{G}} \mathrm{USW}(\boldsymbol{a}|_G\, , \boldsymbol{v}|_G\,). \tag{4.16}$$

This problem presents inherent challenges due to the non-smoothness of the inner-minimization problem and the joint constraint on the uncertainties of the valuation matrices of different groups. These factors make it difficult to compute the dual and reduce the problem or efficiently solve the problem using the quadratic program technique described in Corollary 4.4.17. To streamline this problem, we assume that the uncertainty sets for each group $G \in \mathcal{G}$ are independent of each other.

Assumption 4.4.18 (Independence of Groups)**.** *The uncertainty set $\mathcal{V}$ is a Cartesian product of individual groups' uncertainty sets, $\mathcal{V} \doteq \bigotimes_{G \in \mathcal{G}} \mathcal{V}_G$.*

This assumption is not unreasonable in practical scenarios. For example, conferences often group papers into disjoint tracks or require paper authors to select a single primary subject area. Although papers may have multiple secondary subject areas, the top-level grouping remains independent. Assumption 4.4.18 allows us to reorder the two inner-minimization problems without compromising generality:

$$\max_{\boldsymbol{a} \in \mathcal{Z}} \min_{G \in \mathcal{G}} \min_{\boldsymbol{v}|_G \in \mathcal{V}_G} \frac{1}{|G|} \boldsymbol{a}|_G\, \cdot \boldsymbol{v}|_G\, . \tag{4.17}$$

We note that the problem in (4.17) is a concave-convex optimization problem that can be solved exactly using the sub-gradient ascent method.

An alternative approach to optimizing the problem (4.17) involves taking the dual of the inner-most minimization and reordering the inner-minimization over groups and the inner-maximization problem over the dual variables to obtain a single max-min problem. This simplified problem can then be solved with iterated max-min quadratic programming. We illustrate this result in Proposition 4.4.19.

Proposition 4.4.19. *The problem in* (4.16) *is equivalent to solving*

$$\begin{aligned} \max_{\substack{\boldsymbol{\xi}\in\tilde{\mathcal{Z}}_{\xi},\\ \boldsymbol{\lambda}\in\mathbb{R}_{0+}^{\kappa\times l},\\ \boldsymbol{\beta}\in\mathbb{R}_{0+}^{\kappa\times k}}} \min_{G\in\mathcal{G}} \; & \boldsymbol{\beta}|_G^{\intercal}\, \boldsymbol{e}|_G + \boldsymbol{p}|_G^{\intercal}\, \boldsymbol{\Sigma}_*^{-1} \boldsymbol{q}|_G - \frac{1}{4} \|\boldsymbol{p}|_G^{\intercal}\, \boldsymbol{\Sigma}_*^{-1/2}\|_2^2 \\ & + \sum_{i=1}^{l} \left(\boldsymbol{\lambda}|_{G,i}\; \|\hat{\boldsymbol{v}}|_{G,i}^{\intercal}\, \boldsymbol{\Sigma}_*^{-1/2}\|_2^2 - \boldsymbol{\lambda}|_{G,i}\, \boldsymbol{r}_{G,i}^2 \right) - \|\boldsymbol{q}|_G^{\intercal}\, \boldsymbol{\Sigma}_*^{-1/2}\|_2^2, \end{aligned} \tag{4.18}$$

where $\forall G \in \mathcal{G} : \; \boldsymbol{p}|_G = (\boldsymbol{\xi}|_G - \beta|_G^{\intercal}\, \boldsymbol{Q}|_G)$, $\boldsymbol{q}|_G = \sum_{i=1}^{l} \boldsymbol{\lambda}|_{G,i}\; \hat{\boldsymbol{v}}|_{G,i}\, \boldsymbol{\Sigma}|_{G,i}^{-1}$, *and* $\boldsymbol{\Sigma}_* = \sum_{i=1}^{l}, \boldsymbol{\lambda}|_{G,i}$ $\boldsymbol{\Sigma}|_{G,i}^{-1}$. *Let* $\boldsymbol{\xi}^*$ *be the optimal* $\boldsymbol{\xi}$ *in* (4.18). *Then, the optimal allocation* $\boldsymbol{a}^*$ *satisfies the system of equations:*

$$\forall G \in \mathcal{G} : \frac{1}{|G|} \cdot \boldsymbol{a}|_G \preceq \boldsymbol{\xi}_G^*, \; \boldsymbol{a} \in \mathcal{Z}.$$

The dual variables $\boldsymbol{\lambda}\,|_G\,, \boldsymbol{\beta}\,|_G\,, \boldsymbol{\zeta}\,|_G$ and $\boldsymbol{\xi}\,|_G$ for each group G are interpreted as in Proposition 4.4.15. The optimization problem in (4.18) is concave with respect to the dual variables $\boldsymbol{\lambda}, \boldsymbol{\beta}$ and $\boldsymbol{\xi}$. Consequently, we can solve it using an approach similar to that in Corollary 4.4.17. Specifically, we employ the max-min iterated quadratic programming [123], alternately fixing $\boldsymbol{\lambda}$ and optimizing the rest of the dual variables $(\boldsymbol{\beta}, \boldsymbol{\xi})$ and vice versa until convergence.

Proof. Consider the following optimization problem.

$$\begin{gathered}
\max_{\boldsymbol{a}\in\mathcal{Z}} \min_{G\in\mathcal{G}} \min_{\boldsymbol{v}|_G\in\mathcal{V}_G} \frac{1}{|G|}\boldsymbol{a}|_G^{\intercal}\ \boldsymbol{v}|_G \\
\forall i \in [1,l], \forall G \in \mathcal{G}, \quad (\boldsymbol{v}|_G - \hat{\boldsymbol{v}}|_{G,i})\boldsymbol{\Sigma}|_{G,i}^{-1/2}\ (\boldsymbol{v}|_G - \hat{\boldsymbol{v}}|_{G,i}) \leq \boldsymbol{r}_{G,i}^2 \\
\forall G \in \mathcal{G}, \quad \boldsymbol{Q}|_G\ \boldsymbol{v}|_G \succeq \boldsymbol{e}|_G \\
\forall G \in \mathcal{G}, \quad \boldsymbol{v}|_G \succeq 0.
\end{gathered} \tag{4.19}$$

It is important to note that the inner-minimization is a convex optimization problem and the outer-maximization is a concave maximization problem. This is due to the fact that affine functions are either concave or convex and minimum of concave objectives is concave.

Furthermore, the inner-most minimization over the uncertainty set of valuation matrices is independent for each group. Thus, simply replacing each of these minimization problems with their respective duals yields the following problem.

$$\begin{aligned}
\max_{\boldsymbol{a}\in\mathcal{Z}} \min_{G\in\mathcal{G}} \max_{\substack{\boldsymbol{\lambda}|_G\in\mathbb{R}^l,\\ \boldsymbol{\beta}|_G\in\mathbb{R}^{|G|m},\\ \boldsymbol{\zeta}|_G\in\mathbb{R}^{|G|m}}} \quad & -\frac{1}{4}\left((\boldsymbol{a}'|_G - \boldsymbol{\beta}|_G^{\intercal}\ \boldsymbol{Q}|_G - \boldsymbol{\zeta}|_G)^{\intercal}\ (\sum_{i=1}^{l} \boldsymbol{\lambda}|_{G,i}\ \boldsymbol{\Sigma}|_{G,i}^{-1})^{-1}\ (\boldsymbol{a}'|_G - \boldsymbol{\beta}|_G^{\intercal}\ \boldsymbol{Q}|_G - \boldsymbol{\zeta}|_G) \right) \\
& + \sum_{i=1}^{l} \boldsymbol{\lambda}|_{G,i}\ \hat{\boldsymbol{v}}|_{G,i}^{\intercal}\ \boldsymbol{\Sigma}|_{G,i}^{-1}\ \hat{\boldsymbol{v}}|_{G,i} \\
& - \left(\sum_{i=1}^{l} \boldsymbol{\lambda}|_{G,i}\ \hat{\boldsymbol{v}}|_{G,i}\ \boldsymbol{\Sigma}|_{G,i}^{-1}\right) \left(\sum_{i=1}^{l} \boldsymbol{\lambda}|_{G,i}\ \boldsymbol{\Sigma}|_{G,i}^{-1}\right)^{-1} \left(\sum_{i=1}^{l} \boldsymbol{\lambda}|_{G,i}\ \hat{\boldsymbol{v}}|_{G,i}\ \boldsymbol{\Sigma}|_{G,i}^{-1}\right)^{\intercal} \\
& + (\boldsymbol{a}'|_G - \boldsymbol{\beta}|_G^{\intercal}\ \boldsymbol{Q}|_G - \boldsymbol{\zeta}|_G)^{\intercal} \left(\sum_{i=1}^{l} \boldsymbol{\lambda}|_{G,i}\ \boldsymbol{\Sigma}|_{G,i}^{-1}\right)^{-1} \left(\sum_{i=1}^{l} \boldsymbol{\lambda}|_{G,i}\ \hat{\boldsymbol{v}}|_{G,i}^{\intercal}\ \Sigma|_{G,i}^{-1}\right)^{\intercal} \\
& - \sum_{i=1}^{l} \boldsymbol{\lambda}|_{G,i}\ \boldsymbol{r}_{G,i}^2 + \boldsymbol{\beta}|_G^{\intercal}\ \boldsymbol{e}|_G \\
& \boldsymbol{\lambda}|_G \succeq 0 \\
& \boldsymbol{\beta}|_G \succeq 0 \\
& \boldsymbol{\zeta}|_G \geq 0.
\end{aligned} \tag{4.20}$$

Using the change of variables $\boldsymbol{\zeta}|_G = \boldsymbol{a}'|_G - \boldsymbol{\xi}|_G \ \forall G \in \mathcal{G}$, and combining the dual with the outer-maximization problem, we get

$$\begin{aligned} \max_{\boldsymbol{a}\in\mathcal{Z}} \ \min_{G\in\mathcal{G}} \ \max_{\substack{\boldsymbol{\lambda}|_G\in\mathbb{R}^l, \\ \boldsymbol{\beta}|_G\in\mathbb{R}^{|G|m}, \\ \boldsymbol{\zeta}|_G\in\mathbb{R}^{|G|m}}} \ & -\frac{1}{4}\|\boldsymbol{p}|_G^\intercal \ \boldsymbol{\Sigma}_*^{-1/2}\|_2^2 + \sum_{i=1}^{l} \boldsymbol{\lambda}|_{G,i} \ \|\hat{\boldsymbol{v}}|_{G,i}^\intercal \ \boldsymbol{\Sigma}|_{G,i}^{-1/2} \ \|_2^2 \\ & - \|\boldsymbol{q}|_G^\intercal \ \boldsymbol{\Sigma}_*^{-1/2}\|_2^2 - \boldsymbol{p}|_G^\intercal \ \boldsymbol{\Sigma}_*^{-1}\boldsymbol{q}|_G^\intercal - \sum_{i=1}^{l} \boldsymbol{\lambda}|_{G,i} \ \boldsymbol{r}_{G,i}^2 + \boldsymbol{\beta}|_G^\intercal \ \boldsymbol{e}|_G \\ & \text{s.t. } \ \boldsymbol{\zeta}|_G = \boldsymbol{a}'|_G - \boldsymbol{\xi}|_G, \end{aligned} \tag{4.21}$$

where $\forall G \in \mathcal{G}, \boldsymbol{p}|_G = (\boldsymbol{\beta}|_G^\intercal \ \boldsymbol{Q}|_G + \boldsymbol{\xi}|_G)$, $\boldsymbol{\Sigma}_* = \left(\sum_{i=1}^{l} \boldsymbol{\lambda}|_{G,i} \ \boldsymbol{\Sigma}|_{G,i}^{-1}\right)$, and $\boldsymbol{q}|_G = \sum_{i=1}^{l} \boldsymbol{\lambda}|_{G,i} \ \hat{\boldsymbol{v}}|_{G,i}^\intercal \ \boldsymbol{\Sigma}|_{G,i}^{-1}$.

Since the inner maximization for each group is independent of the other groups, we can re-order the inner minimization over groups and the inner-maximization problem. Thus, without loss of generality, we can write the above optimization problem as

$$\begin{aligned} \max_{\substack{\boldsymbol{a}\in\mathcal{Z},\boldsymbol{\zeta}\in\mathbb{R}^{nm},\boldsymbol{\lambda}\in\mathbb{R}_{0+}^{\kappa\times l}, \\ \boldsymbol{\beta}\in\mathbb{R}_{0+}^{\kappa\times k},\boldsymbol{\xi}\in\mathbb{R}^{nm}}} \ \min_{G\in\mathcal{G}} & -\frac{1}{4}\|\boldsymbol{p}|_G^\intercal \ \boldsymbol{\Sigma}_*^{-1/2}\|_2^2 + \sum_{i=1}^{l} \boldsymbol{\lambda}_{i,G}\|\hat{\boldsymbol{v}}_{i,G}^\intercal \boldsymbol{\Sigma}_*^{-1/2}\|_2^2 - \|\boldsymbol{q}|_G^\intercal \ \boldsymbol{\Sigma}_*^{-1/2}\|_2^2 \\ & - \boldsymbol{p}|_G^\intercal \ \boldsymbol{\Sigma}_*^{-1}\boldsymbol{q}|_G^\intercal - \sum_{i=1}^{l} \boldsymbol{\lambda}|_{G,i} \ \boldsymbol{r}_{G,i}^2 + \boldsymbol{\beta}|_G^\intercal \ \boldsymbol{e}|_G \\ & \text{s.t. } \ \boldsymbol{\xi}|_G = \frac{\boldsymbol{a}|_G}{|G|} - \boldsymbol{\zeta}|_G, \end{aligned} \tag{4.22}$$

Using the same technique as in the proof of Proposition 4.4.15, we eliminate the variables $\boldsymbol{a}$ and $\boldsymbol{\zeta}$ in the above problem and derive them from the optimal $\boldsymbol{\xi}$.

Eliminating $\boldsymbol{\zeta}$ and $\boldsymbol{a}$ in (4.22), we get the following optimization problem.

$$\boldsymbol{\lambda}^*, \boldsymbol{\beta}^*, \boldsymbol{\xi}^* = \underset{\substack{\boldsymbol{\lambda} \in \mathbb{R}_{0+}^{\kappa \times l}, \\ \boldsymbol{\beta} \in \mathbb{R}_{0+}^{\kappa \times k}, \\ \boldsymbol{\xi} \in \tilde{\mathcal{Z}}_\xi}}{\arg\max} \min_{G \in \mathcal{G}} - \frac{1}{4} \| \boldsymbol{p}|_G^\intercal \ \boldsymbol{\Sigma}_*^{-1/2} \|_2^2 + \sum_{i=1}^{l} \boldsymbol{\lambda}|_{G,i} \ \ \| \hat{\boldsymbol{v}}|_{G,i}^\intercal \ \boldsymbol{\Sigma}_*^{-1/2} \|_2^2 - \| \boldsymbol{q}|_G^\intercal \ \boldsymbol{\Sigma}_*^{-1/2} \|_2^2$$

$$- \boldsymbol{p}|_G^\intercal \ \boldsymbol{\Sigma}_*^{-1} \boldsymbol{q}|_G - \sum_{i=1}^{l} \boldsymbol{\lambda}|_{G,i} \ \ \boldsymbol{r}_{G,i}^2 + \boldsymbol{\beta}|_G^\intercal \ \boldsymbol{e}|_G, \tag{4.23}$$

where $\forall G \in \mathcal{G}$, $\boldsymbol{p}|_G = \ (\boldsymbol{\xi}|_G \ \ - \boldsymbol{\beta}|_G^\intercal \ \ \boldsymbol{Q}|_G)$, $\boldsymbol{q}|_G = \sum_{i=1}^{l} \boldsymbol{\lambda}|_{G,i} \ \ \hat{\boldsymbol{v}}|_{G,i} \ \ \boldsymbol{\Sigma}|_{G,i} \ \ ^{-1}$, and $\boldsymbol{\Sigma}_* = \left(\sum_{i=1}^{l}, \ \boldsymbol{\lambda}|_{G,i} \ \ \boldsymbol{\Sigma}|_{G,i} \ \ ^{-1} \right)$.

□

Interestingly, even when optimizing the egalitarian welfare objective with only polyhedral uncertainty sets, the robust egalitarian problem described in (4.18) simplifies to a straightforward linear program. This is akin to what we observe in the robust utilitarian case (Corollary 4.4.16). We formalize this finding in Corollary 4.4.20.

Corollary 4.4.20. *In the case where the uncertainty set $\mathcal{V}$ is defined only by linear constraints, i.e., $\mathcal{V} = \{\boldsymbol{v} \in \mathbb{R}^{nm} \mid \boldsymbol{Q}\boldsymbol{v} \succeq \boldsymbol{e}, \boldsymbol{v} \succeq 0\}$, the max-min-min problem in (4.16) is trivially transformable into a linear program.*

Proof. Substituting $l = 0$ in (4.18), we get

$$\begin{aligned} \max_{\boldsymbol{a} \in \mathcal{Z}, \boldsymbol{\beta} \in \mathbb{R}^{\kappa \times k}} \ \min_{G \in \mathcal{G}} \quad & \boldsymbol{\beta}|_G^\intercal \ \boldsymbol{e}|_G \\ & \boldsymbol{\beta}|_G^\intercal \ \boldsymbol{Q}|_G \preceq \boldsymbol{a}'|_G, \end{aligned} \tag{4.24}$$

where $\boldsymbol{a}'|_G = \frac{\boldsymbol{a}|_G}{|G|} \forall G \in \mathcal{G}$. Using simple algebraic manipulations, we can write the above optimization problem as

$$\begin{aligned} \max_{\boldsymbol{a} \in \mathcal{Z}, \boldsymbol{\beta} \in \mathbb{R}^{\kappa \times k}, t \in \mathbb{R}} \quad & t \\ & \forall G \in \mathcal{G} : t \leq \boldsymbol{\beta}|_G^\intercal \ \boldsymbol{e}|_G \\ & \forall G \in \mathcal{G} : \boldsymbol{\beta}|_G^\intercal \ \boldsymbol{Q}|_G \preceq \boldsymbol{a}'|_G, \end{aligned} \tag{4.25}$$

□

When the valuation uncertainty set is defined by a single ellipsoidal constraint per group, we can employ the iterated quadratic programming (Iterated QP) approach used in Corollary 4.4.17, alternately fixing $\boldsymbol{\lambda}$ and optimizing the rest of the dual variables $(\boldsymbol{\beta}, \boldsymbol{\xi})$ until convergence.

Corollary 4.4.21 (Group Egalitarian Welfare with Ellipsoidal Uncertainty)**.** *Suppose that the set $\mathcal{V}$ in* (4.16) *is defined by a single truncated ellipsoidal constraint per group i.e., $\mathcal{V} = \{\boldsymbol{v} \in \mathbb{R}^{nm} \mid \forall G \in \mathcal{G} : (\boldsymbol{v}|_G - \hat{\boldsymbol{v}}|_G)\boldsymbol{\Sigma}|_G^{-1}(\boldsymbol{v}|_G - \hat{\boldsymbol{v}}|_G) \leq \boldsymbol{r}_G^2, \boldsymbol{v} \succeq 0\}$. Then the problem in* (4.16) *is equivalent to solving*

$$\max_{\substack{\boldsymbol{\lambda} \in \mathbb{R}^{\kappa}_{0+} \\ \boldsymbol{\xi} \in \tilde{\mathcal{Z}}_{\xi}}} \min_{G \in \mathcal{G}} \boldsymbol{\xi}|_G^\intercal \, \hat{\boldsymbol{v}}|_G - \frac{\boldsymbol{\xi}|_G^\intercal \, \boldsymbol{\Sigma}|_G \, \boldsymbol{\xi}|_G}{4\boldsymbol{\lambda}_G} - \boldsymbol{\lambda}_G \boldsymbol{r}_G^2.$$

The exact optimal solution $(\boldsymbol{\lambda}^*, \boldsymbol{\xi}^*)$ to Equation (4.15) *can be computed by alternately performing two steps until convergence: first, fixing $\boldsymbol{\xi}$ and optimizing $\boldsymbol{\lambda}$, i.e., $\forall G \in \mathcal{G}$, $\boldsymbol{\lambda}_G = \boldsymbol{\xi}|_G^\intercal \boldsymbol{\Sigma}|_G \boldsymbol{\xi}|_G / 2\boldsymbol{r}_G$, and second, fixing $\boldsymbol{\lambda}$ and solving a concave quadratic program to optimize $\boldsymbol{\xi}$. The optimal allocation $\boldsymbol{a}^*$ can be computed from $\boldsymbol{\xi}^*$ as in Proposition 4.4.15.*

Proof. Consider the following optimization problem.

$$\max_{\boldsymbol{a} \in \mathcal{Z}} \min_{G \in \mathcal{G}} \min_{\boldsymbol{v}|_G \in \mathbb{R}^{|G|m}_{0+}} \boldsymbol{a}|_G^\intercal \, \boldsymbol{v}|_G$$

$$\forall G \in \mathcal{G} : (\boldsymbol{v}|_G - \hat{\boldsymbol{v}}|_G)^\intercal \boldsymbol{\Sigma}|_G^{-1} (\boldsymbol{v}|_G - \hat{\boldsymbol{v}}|_G) \leq \boldsymbol{r}_G^2$$

$$\boldsymbol{v}|_G \succeq 0.$$

Similar to the general version of the problem in (4.22), the inner-most minimization is a convex optimization problem and the outer-maximization is a concave maximization problem.

This is again due to the fact that affine functions are either concave or convex and minimum of concave objectives is concave.

The inner-most minimization over the uncertainty set of valuation matrices is independent for each group. Therefore, by simply replacing each of these minimization problems with their respective Lagrangian duals, as computed in Corollary 4.4.17, we obtain

$$\max_{\boldsymbol{a}\in\mathcal{Z}} \min_{G\in\mathcal{G}} \max_{\substack{\boldsymbol{\lambda}_G\in\mathbb{R}_{0+}\\ \boldsymbol{\zeta}|_G\in\mathbb{R}^{|G|m}_{0+}}} -\frac{1}{4}\left((\boldsymbol{a}|_G - \boldsymbol{\zeta}|_G)^\intercal \boldsymbol{\lambda}_G \boldsymbol{\Sigma}|_G \left(\boldsymbol{a}|_G - \boldsymbol{\zeta}|_G\right)\right) + (\boldsymbol{a}|_G - \boldsymbol{\zeta}|_G)^\intercal \hat{\boldsymbol{v}}|_G - \boldsymbol{\lambda}_G \boldsymbol{r}_G^2.$$

Note that the dual is computed following the approach outlined in the proof of Proposition 4.4.15.

Using the change of variables $\forall G \in \mathcal{G} : \boldsymbol{\xi}|_G = \boldsymbol{a}|_G - \boldsymbol{\zeta}|_G$, we get

$$\begin{aligned} \max_{\boldsymbol{a}\in\mathcal{Z}} \min_{G\in\mathcal{G}} \max_{\substack{\boldsymbol{\lambda}_G\in\mathbb{R}_{0+}\\ \boldsymbol{\xi}|_G\in\mathbb{R}^{|G|m}\\ \boldsymbol{\zeta}|_G\in\mathbb{R}^{|G|m}_{0+}}} \quad & \boldsymbol{\xi}|_G^\intercal \, \hat{\boldsymbol{v}}|_G - \frac{\boldsymbol{\xi}|_G^\intercal \, \boldsymbol{\Sigma}|_G \, \boldsymbol{\xi}|_G}{4\boldsymbol{\lambda}_G} - \boldsymbol{\lambda}_G \boldsymbol{r}_G^2 \\ \text{s.t.} \quad & \boldsymbol{\xi}|_G = \boldsymbol{a}|_G - \boldsymbol{\zeta}|_G \ . \end{aligned}$$

Since the inner maximization for each group is independent of the other groups, we can re-order the inner minimization over groups and the inner-maximization problem. Thus, without loss of generality, we can write the above optimization problem as

$$\begin{aligned} \max_{\substack{\boldsymbol{a}\in\mathcal{Z}\\ \boldsymbol{\lambda}\in\mathbb{R}^{\kappa}_{0+}\\ \boldsymbol{\xi}\in\mathbb{R}^{nm}\\ \boldsymbol{\zeta}\in\mathbb{R}^{nm}_{0+}}} \min_{G\in\mathcal{G}} \quad & \boldsymbol{\xi}|_G^\intercal \, \hat{\boldsymbol{v}}|_G - \frac{\boldsymbol{\xi}|_G^\intercal \, \boldsymbol{\Sigma}|_G \, \boldsymbol{\xi}|_G}{4\boldsymbol{\lambda}_G} - \boldsymbol{\lambda}_G \boldsymbol{r}_G^2 \\ \text{s.t.} \quad & \boldsymbol{\xi}|_G = \boldsymbol{a}|_G - \boldsymbol{\zeta}|_G \ . \end{aligned}$$

Using the same technique as in the proof of Proposition 4.4.15, we can simplify the problem by eliminating the variables $\boldsymbol{a}$ and $\boldsymbol{\zeta}$ in the above problem and then derive them from the optimal $\boldsymbol{\xi}$.

Eliminating $\boldsymbol{\zeta}$ and $\boldsymbol{a}$ in the above optimization problem, we get

$$\boldsymbol{\lambda}^*, \boldsymbol{\xi}^* = \underset{\substack{\boldsymbol{\lambda} \in \mathbb{R}^{\kappa}_{0+} \\ \boldsymbol{\xi} \in \tilde{\mathcal{Z}}_{\xi}}}{\arg\max} \min_{G \in \mathcal{G}} \boldsymbol{\xi}|_G^{\intercal} \, \hat{\boldsymbol{v}}|_G - \frac{\boldsymbol{\xi}|_G^{\intercal} \, \boldsymbol{\Sigma}|_G \, \boldsymbol{\xi}|_G}{4\boldsymbol{\lambda}_G} - \boldsymbol{\lambda}_G \boldsymbol{r}_G^2$$

$$\text{s.t. } \boldsymbol{\xi}|_G = \boldsymbol{a}|_G - \boldsymbol{\zeta}|_G,$$

where $\tilde{\mathcal{Z}}_{\xi} = \mathcal{Z} - \mathbb{R}^{nm}_{0+} = \{\boldsymbol{\xi} \in \mathbb{R}^{nm} \mid \forall a \in N : \sum_{i \in I} \boldsymbol{\xi}_{am+i} \leq \bar{\kappa}_a, \forall i \in M : \sum_{a \in N} \boldsymbol{\xi}_{am+i} \leq \bar{\psi}_i, \boldsymbol{\xi} \preceq \boldsymbol{c}\}$.

As in proposition 4.4.15, we can now determine the set of optimal $(\boldsymbol{a}, \boldsymbol{\zeta})$ pairs by solving the system of equations: $\{(\boldsymbol{a}^*, \boldsymbol{\zeta}^*) \mid \boldsymbol{a} \in \mathcal{Z}, G \in \mathcal{G} : \boldsymbol{a}|_G - \boldsymbol{\zeta}|_G = \boldsymbol{\xi}^*|_G, \boldsymbol{\zeta} \in \mathbb{R}^{nm}_{0+}\}$.

□

Monotonic Welfare Functions We extend our findings to a broader class of monotonic welfare functions. We show that when optimizing a monotonic welfare objective under the assumption that valuation uncertainty sets across groups are independent, we can decompose the problem into sub-problems such that we independently determine the worst valuation in the uncertainty set of each group, while jointly optimizing the allocations of different groups.

Proposition 4.4.22. *Consider an optimization problem of the form*

$$\max_{\boldsymbol{a} \in \mathcal{Z}} \min_{\boldsymbol{v} \in \mathcal{V}} W_M(\mathrm{USW}(\boldsymbol{a}|_{G_1}, \boldsymbol{v}|_{G_1}), \mathrm{USW}(\boldsymbol{a}|_{G_2}, \boldsymbol{v}|_{G_2}), \ldots, \mathrm{USW}(\boldsymbol{a}|_{G_\kappa}, \boldsymbol{v}|_{G_\kappa})), \tag{4.26}$$

where the welfare function $W_M(\cdot)$ is monotonic in the utility of groups. If Assumption 4.4.18 holds, then (4.26) *simplifies to*

$$\max_{\boldsymbol{a} \in \mathcal{Z}} W_M(\min_{\boldsymbol{v} \in \mathcal{V}} \mathrm{USW}(\boldsymbol{a}|_{G_1}, \boldsymbol{v}|_{G_1}), \min_{\boldsymbol{v} \in \mathcal{V}} \mathrm{USW}(\boldsymbol{a}|_{G_2}, \boldsymbol{v}|_{G_2}), \ldots, \min_{\boldsymbol{v} \in \mathcal{V}} \mathrm{USW}(\boldsymbol{a}|_{G_\kappa}, \boldsymbol{v}|_{G_\kappa})).$$

Proof. The result directly follows from the monotonic property of the welfare function and the independence of the uncertainty sets across groups. □

We note that the egalitarian problem in (4.16) is an instance of the class of optimization problem described in (4.26). Furthermore, when the allocation and valuation uncertainty sets are convex and compact, the problem in (4.26) can be solved using constrained convex-concave minimax optimization algorithms [50, 67, 176] or adversarial projected sub-gradient ascent (Algorithm 8). These approaches do not depend on Assumption 4.4.18, though the optimization may be simplified if independence does hold.

4.5 Stochastic Welfare Optimization

In this section, we optimize the CVaR of utilitarian and egalitarian welfare. This approach works when the distribution $\mathcal{D}_{\boldsymbol{v}}$ over the valuation matrix is known, or when we can sample from $\mathcal{D}_{\boldsymbol{v}}$. We demonstrate that when the distribution follows a Gaussian distribution, the CVaR of the utilitarian welfare has a simple representation that can be optimized without sample approximation using a projected gradient ascent method. In all other cases, we can approximately optimize CVaR using a sampling-based approach. In particular, when we have monotone, concave welfare functions, we can always approximate the CVaR objective using sampling. However, unlike in Propositions 4.5.1 and 4.5.7, where the approximated problem becomes linear, with arbitrary monotone, concave welfare functions the problem may require general concave optimization.

4.5.1 CVaR Allocation for Utilitarian Welfare

We wish to find an allocation that maximizes the CVaR_α of the utilitarian welfare. For any confidence level α, we formulate this problem as:

$$\max_{\boldsymbol{a}\in\mathcal{Z}} \mathrm{CVaR}_\alpha \left[\mathrm{USW}(\boldsymbol{a},\boldsymbol{v})\right] = \max_{\boldsymbol{a}\in\mathcal{Z}, b\in\mathbb{R}} \left\{ b - \frac{1}{\alpha} \mathop{\mathbb{E}}_{\boldsymbol{v}\sim\mathcal{D}_{\boldsymbol{v}}} \left[b - \mathrm{USW}\left(\boldsymbol{a},\boldsymbol{v}\right)\right]_+ \right\}, \tag{4.27}$$

where $(x)_+ = \max(x, 0)$ represents the positive part of x. Computing the exact expectation in this problem may not be feasible for every distribution $\mathcal{D}_{\boldsymbol{v}}$. Therefore, we adopt a sampling-based approach to approximately optimize the CVaR of utilitarian welfare. We begin by drawing h samples of the valuation matrix from $\mathcal{D}_{\boldsymbol{v}}$ represented as $\boldsymbol{v}^1, \boldsymbol{v}^2, \boldsymbol{v}^3, \ldots \boldsymbol{v}^h$. We then use these samples to solve the problem described in (4.27) by solving the linear program outlined in Proposition 4.5.1.

Proposition 4.5.1. *Given h samples $\boldsymbol{v}^1, \boldsymbol{v}^2, \boldsymbol{v}^3, \ldots \boldsymbol{v}^h$ from $\mathcal{D}_{\boldsymbol{v}}$, the optimal allocation for the problem in* (4.27) *can be approximately computed by solving*

$$\max_{\boldsymbol{a} \in \mathcal{Z}} \max_{\boldsymbol{y} \in \mathbb{R}^h_{0+}, b \in \mathbb{R}} \left(b - \frac{1}{\alpha} \sum_{j=1}^{h} \boldsymbol{y}_j \right) \qquad \forall j \in [1, h] : \ \boldsymbol{y}_j \geq \frac{1}{h} \left(b - \mathrm{USW}(\boldsymbol{a}, \boldsymbol{v}^j) \right). \tag{4.28}$$

Proof. For any random utility X, $\mathrm{CVaR}_\alpha[X]$ can be written as

$$\mathrm{CVaR}_\alpha[X] = \max_{b \in \mathbb{R}} b - \frac{1}{\alpha} \mathbb{E}\left[b - X\right]_+ , \tag{4.29}$$

where $(t)_+ = max(t, 0)$. Given a posterior distribution of valuations $\mathcal{D}_{\boldsymbol{v}}$, we generate h samples of the valuation matrix, i.e., $\boldsymbol{v}_1$, $\boldsymbol{v}_2 \ldots \boldsymbol{v}_h$, and use it to empirically estimate the expectation in (4.29).

$$\begin{aligned}
&\max_{\boldsymbol{a} \in \mathbb{R}^{n \times m}} \max_{y \in \mathbb{R}^h, b \in \mathbb{R}} \left(b - \frac{1}{\alpha} \sum_{j=1}^{h} y_j \right) \\
&\forall j \in [1, h] \, y_j \geq 0 \\
&\forall j \in [1, h] \, y_j \geq \frac{1}{h} b - \frac{1}{h} \mathrm{USW}(\boldsymbol{a}, \boldsymbol{v}^j) \\
&\boldsymbol{a} \in \mathcal{Z}.
\end{aligned} \tag{4.30}$$

□

The CVaR estimator used in (4.28) is a strongly consistent estimator [77], so the approximation error of the objective in (4.28) goes to 0 as $h \to \infty$. In Proposition 4.5.3, we bound the sample complexity of (4.28) when the valuation matrix is sub-Gaussian distributed.

Assumption 4.5.2. *Let $\tilde{w}$ represent the random welfare corresponding to a given allocation $\boldsymbol{a}$ and let f be its density function. Furthermore, let $\nu_\alpha(W; \boldsymbol{a}, \boldsymbol{v})$ denotes the α-percentile of the welfare corresponding to allocation $\boldsymbol{a}$. There exist universal constants $\eta, \delta' \geq 0$, s.t., $f(w) \geq \eta \, \forall w \in [\nu_\alpha(W; \boldsymbol{a}, \boldsymbol{v}) - \frac{\delta'}{2}, \nu_\alpha(W; \boldsymbol{a}, \boldsymbol{v}) + \frac{\delta'}{2}]$.*

For any allocation $\boldsymbol{a}$, let $\hat{c}_{h,\alpha}(\boldsymbol{a})$ represent the empirical estimate of CVaR of utilitarian welfare computed from h samples and $c_{h,\alpha}(\boldsymbol{a})$ represent the corresponding true value.

Proposition 4.5.3. *Suppose that $\boldsymbol{v}$ is a multivariate sub-Gaussian with mean $\hat{\boldsymbol{v}} \in \mathbb{R}^{nm}$ and covariance proxy $\boldsymbol{\Sigma} \in \mathbb{R}^{nm \times nm}$, i.e., $\exists K \geq 0$ s.t. $\mathbb{E}\left[\exp(\lambda(\boldsymbol{v} - \hat{\boldsymbol{v}})^\intercal \boldsymbol{z}))\right] \leq \exp({}^{\lambda^2 K^2 \boldsymbol{z}^\intercal \boldsymbol{\Sigma} \boldsymbol{z}}/_{2}), \forall \lambda \in \mathbb{R}, \forall \boldsymbol{z} \in \mathbb{R}^{nm}$ and that Assumption 4.5.2 holds. Let $|\mathcal{Z}|$ represent the number of feasible allocations, $\boldsymbol{a}' = \frac{1}{n}\boldsymbol{a}$, and $h > \frac{8\max(\max_{\boldsymbol{a}\in\mathcal{Z}} \boldsymbol{a}'^\intercal \Sigma \boldsymbol{a}', 8)\log\left(\frac{6|\mathcal{Z}|}{\delta}\right)}{\varepsilon^2(\alpha)^2 \min(\eta^2, 1)}$ where $\delta \in (0, 1)$. Then, $\Pr[\forall \boldsymbol{a} \in \mathcal{Z} : |\hat{c}_{h,\alpha}(\boldsymbol{a}) - c_\alpha(\boldsymbol{a})| \leq \varepsilon] \geq 1 - \delta$.*

Proof.

Assumption 4.5.4 (L.A. et al. [95])**.** *The random variable X is continuous with probability density function f that satisfies the following condition: There exists universal constants $\eta, \delta' \geq 0$ such that $f(x) \geq \eta \; \forall x \in [v_\alpha - \frac{\delta'}{2}, v_\alpha + \frac{\delta'}{2}]$, where $v_\alpha = F^{-1}(\alpha)$.*

Theorem 4.5.5 (L.A. et al. [95])**.** *Let $(X_i)_{i=1}^n$ be a sequence of i.i.d random variables. Let $\hat{c}_{n,\alpha}$ be the empirical CVaR estimates of X computed from the above samples. Suppose that $X_i, i = 1, \ldots n$ are $\sigma-$sub-Gaussian. Then for any $\varepsilon \geq 0$, we have*

$$\Pr\left[|c_{n,\alpha} - c_\alpha| > \varepsilon\right] \leq 6 \exp\left(\frac{-n(\alpha)^2 \min(\eta^2, 1)}{8\max(8, \sigma^2)}\right). \tag{4.31}$$

By assumption, the valuation vector $\boldsymbol{v}$ is a sub-Gaussian that satisfies the following condition: $\exists K \geq 0$ s.t. $\mathbb{E}\left[\exp(\lambda(\boldsymbol{v}-\hat{\boldsymbol{v}})^{\intercal}\boldsymbol{w}))\right] \leq \exp({}^{\lambda^2 K^2 \hat{\boldsymbol{v}}^{\intercal}\boldsymbol{\Sigma}\hat{\boldsymbol{v}}}/_{2}), \forall \lambda \in \mathbb{R}, \forall \boldsymbol{w} \in \mathbb{R}^{nm}$. Further, we know that if X is a σ-sub-Gaussian random variable then cX is also sub-Gaussian with variance proxy$=$ $c\sigma$. Applying these two properties, the utilitarian welfare for a given allocation $\boldsymbol{a}$ is also a sub-Gaussian with variance-proxy $=$ $K^2\boldsymbol{a}'^{\intercal}\Sigma\boldsymbol{a}'$.

For any allocation $\boldsymbol{a}$, let $\hat{c}_{h,\alpha}(\boldsymbol{a})$ represent the empirical estimate of CVaR of utilitarian welfare and $c_{h,\alpha}(\boldsymbol{a})$ represent the corresponding true value. Then, we can bound the error of approximating the CVaR of the utilitarian welfare for allocation $\boldsymbol{a}$ as

$$\Pr\left[|\hat{c}_{M,\alpha}(\boldsymbol{a}) - c_{\alpha}(\boldsymbol{a})| > \varepsilon\right] \leq 6\exp\left(\frac{-h(\alpha)^2 min(\eta^2, 1)}{8\max(8, K^2\boldsymbol{a}'^{\intercal}\Sigma\boldsymbol{a}')}\right). \tag{4.32}$$

Furthermore, the approximation error for all allocations can be upper-bounded as

$$\Pr\left[\forall \boldsymbol{a} \in \mathcal{Z}, |\hat{c}_{h,\alpha}(\boldsymbol{a}) - c_{\alpha}(\boldsymbol{a})| \leq \varepsilon\right] \leq 1 - \sum_{\boldsymbol{a}\in\mathcal{Z}} \Pr\left[|\hat{c}_{h,\alpha}(\boldsymbol{a}) - c_{\alpha}(\boldsymbol{a})| > \varepsilon\right]. \tag{4.33}$$

Combining (4.32) and (4.33) and setting $h > \frac{8\max(\max_{\boldsymbol{a}\in\mathcal{Z}} \boldsymbol{a}'^{\intercal}\Sigma\boldsymbol{a}', 8)\log\left(\frac{6|\mathcal{Z}|}{\delta}\right)}{\varepsilon^2(\alpha)^2\min(\eta^2,1)}$, yields

$$\Pr[\forall \boldsymbol{a} \in \mathcal{Z}, |\hat{c}_{h,\alpha}(\boldsymbol{a}) - c_{\alpha}(\boldsymbol{a})| \leq \varepsilon] \geq 1 - \delta. \tag{4.34}$$

□

When the valuation $\boldsymbol{v}$ is normally distributed, we can circumvent the sampling approach and instead solve the problem directly by optimizing a quadratic optimization problem (Proposition 4.5.6), which depends solely on the mean and covariance of the valuations $\boldsymbol{v}$.

Proposition 4.5.6. *If the valuation $\boldsymbol{v}$ is distributed as a multivariate Gaussian, i.e., $\boldsymbol{v} \sim \mathcal{N}(\hat{\boldsymbol{v}}, \boldsymbol{\Sigma})$, the optimization problem in* (4.27) *simplifies to*

$$\max_{\boldsymbol{a}\in\mathcal{Z}} \boldsymbol{a}^{\intercal}\hat{\boldsymbol{v}} - \frac{\phi(\Phi^{-1}(1-\alpha))}{\alpha}\sqrt{\boldsymbol{a}^{\intercal}\boldsymbol{\Sigma}\boldsymbol{a}}. \tag{4.35}$$

Proof. The proof simply follows from the fact that for any normally distributed random variable $X \sim \mathcal{N}(\mu, \sigma^2)$ with mean $\mu \in \mathbb{R}$ and $\sigma \in \mathbb{R}_{0+}$, $\text{CVaR}[X] = \mu - \frac{\phi(\Phi(\alpha)^{-1})}{1-\alpha}\sigma$. If the valuation for any group G is normally distributed as $\mathcal{N}(\hat{\boldsymbol{v}}_G, \boldsymbol{\Sigma}\mid_G)$, then the utility corresponding to that group has mean $= \frac{1}{|G|}\boldsymbol{a}\mid_G^{\intercal}\ \hat{\boldsymbol{v}}\mid_G$ and variance $= \left(\frac{1}{|G|}\right)^2 \boldsymbol{a}\mid_G^{\intercal}\ \boldsymbol{\Sigma}\mid_G \boldsymbol{a}|_G$. Therefore, for the utilitarian welfare objective mean$= \sum_G \frac{1}{|G|}\boldsymbol{a}|_G^{\intercal}\ \hat{\boldsymbol{v}}|_G$ and variance$= \sum_{G\in\mathcal{G}}\left(\left(\frac{1}{|G|}\right)^2 \boldsymbol{a}|_G^{\intercal}\ \boldsymbol{\Sigma}|_G\ \boldsymbol{a}|_G\right)$. Substituting by these values in the CVaR formulation for normal random variables, we get the stated results. □

(4.35) is concave and can be solved exactly using the projected gradient ascent method.

4.5.2 CVaR Allocation for Group Egalitarian Welfare

For our final objective, we wish to optimize egalitarian welfare under uncertainty using the CVaR approach. We formulate this optimization problem as

$$\begin{aligned} &\max_{\boldsymbol{a}\in\mathcal{Z}} \text{CVaR}_\alpha\left[\min_{G\in\mathcal{G}} \text{USW}(\boldsymbol{a}|_G, \boldsymbol{v}|_G)\right] \\ &\qquad = \max_{\boldsymbol{a}\in\mathcal{Z}, w\in\mathbb{R}}\left\{w - \frac{1}{\alpha}\mathbb{E}\left[\left(w - \min_{G\in\mathcal{G}}\frac{1}{|G|}\cdot\boldsymbol{a}|_G^{\intercal}\ \tilde{\boldsymbol{v}}|_G\right)_+\right]\right\}. \end{aligned} \tag{4.36}$$

To optimize the problem described in (4.36), we solve a linear program similar to the one used for optimizing the CVaR Utilitarian objective in (4.27).

Proposition 4.5.7. *Given h samples $\boldsymbol{v}^1, \boldsymbol{v}^2, \boldsymbol{v}^3, \dots \boldsymbol{v}^h$ from $\mathcal{D}_{\boldsymbol{v}}$, the optimal allocation for the problem in* (4.36) *can be approximately computed by solving*

$$\begin{aligned} &\max_{\boldsymbol{a}\in\mathcal{Z}}\ \max_{\boldsymbol{y}\in\mathbb{R}^m_{0+}, b\in\mathbb{R}}\left(b - \frac{1}{\alpha}\sum_{j=1}^{M}\boldsymbol{y}_j\right) \\ &\forall j\in[1,h], \forall G\in\mathcal{G}:\ \boldsymbol{y}_j \geq \frac{1}{h}\left(b - \frac{1}{|G|}\cdot\boldsymbol{a}|_G^{\intercal}\ \boldsymbol{v}|_G^{j}\right). \end{aligned} \tag{4.37}$$

Proof. Consider the CVaR of egalitarian welfare optimization problem, given by

$$\max_{\boldsymbol{a}\in\mathcal{Z}} \mathrm{CVaR}_\alpha \left[\min_{G\in\mathcal{G}} \frac{1}{|G|} \cdot \boldsymbol{a}|_G^\intercal \, \tilde{\boldsymbol{v}}|_G\right] = \max_{w\in\mathbb{R},\boldsymbol{a}\in\mathcal{Z}} \left\{w - \frac{1}{\alpha}\,\mathbb{E}\left[\left(w - \min_{G\in\mathcal{G}} \frac{1}{|G|} \cdot \boldsymbol{a}|_G^\intercal \, \tilde{\boldsymbol{v}}|_G\right)_+\right]\right\}. \tag{4.38}$$

Substituting the expectation in the above problem with the empirical expectation computed from the h samples of the valuation matrices, we get

$$\max_{w\in\mathbb{R},\boldsymbol{a}\in\mathcal{Z}} \left\{w - \frac{1}{\alpha}\frac{1}{h}\sum_{i=1}^{M}\left(w - \min_{G\in\mathcal{G}} \frac{1}{|G|} \cdot \boldsymbol{a}|_G^\intercal \, \boldsymbol{v}^i|_G\right)_+\right\}. \tag{4.39}$$

Introducing slack variables $\boldsymbol{y} \in \mathbb{R}^m$, we can write the above problem as

$$\begin{aligned}
&\max_{\boldsymbol{a}\in\mathcal{Z}} \max_{\boldsymbol{y}\in\mathbb{R}^h, b\in\mathbb{R}} \left(b - \frac{1}{\alpha}\sum_{j=1}^{m} \boldsymbol{y}_j\right)\\
&\forall j \in [1,h] : \ \boldsymbol{y}_j \geq 0\\
&\forall j \in [1,h] : \ \boldsymbol{y}_j \geq \frac{1}{h}\left(b - \min_{G\in\mathcal{G}} \frac{1}{|G|} \cdot \boldsymbol{a}|_G^\intercal \, \boldsymbol{v}|_G^j\right).
\end{aligned} \tag{4.40}$$

Without loss of generality, we can represent the above problem as

$$\begin{aligned}
&\max_{\boldsymbol{a}\in\mathcal{Z}} \max_{\boldsymbol{y}\in\mathbb{R}^m, b\in\mathbb{R}} \left(b - \frac{1}{\alpha}\sum_{j=1}^{m} \boldsymbol{y}_j\right)\\
&\forall j \in [1,h] : \ \boldsymbol{y}_j \geq 0\\
&\forall j \in [1,h], G \in \mathcal{G} : \ \boldsymbol{y}_j \geq \frac{1}{h}\left(b - \frac{1}{|G|} \cdot \boldsymbol{a}|_G^\intercal \, \boldsymbol{v}|_G^j\right).
\end{aligned} \tag{4.41}$$

□

When the valuation matrix $\boldsymbol{v}$ is normally distributed and the uncertainty sets of different groups are independent, the result is a quadratic program characterized by a linear objective and quadratic constraints, as detailed in Proposition 4.5.8.

Proposition 4.5.8. *If $\boldsymbol{v}|_{G_1}, \boldsymbol{v}|_{G_2}, \cdots, \boldsymbol{v}|_{G_\kappa}$ are i.i.d and normally distributed, i.e., $\forall G \in \mathcal{G}$, $\boldsymbol{v}|_G \sim \mathcal{N}(\hat{\boldsymbol{v}}|_G, \boldsymbol{\Sigma}|_G)$, then, the optimization problem in* (4.36) *simplifies to*

$$\begin{aligned}
&\max_{\boldsymbol{a}\in\mathcal{Z}, t\in\mathbb{R}} \; t \\
&s.t.\ \forall G \in \mathcal{G} : \left(\frac{1}{|G|} \cdot \boldsymbol{a}|_G^\intercal \, \hat{\boldsymbol{v}}|_G - t\right)^2 \geq \left(\frac{1}{|G|} \cdot \frac{\phi(\Phi^{-1}(1-\alpha))}{(\alpha)}\right)^2 \boldsymbol{a}|_G^\intercal \, \boldsymbol{\Sigma}|_G \, \boldsymbol{a}|_G \\
&\forall G \in \mathcal{G} : \left(\frac{1}{|G|} \cdot \boldsymbol{a}|_G^\intercal \, \hat{\boldsymbol{v}}|_G - t\right) \geq 0.
\end{aligned} \tag{4.42}$$

Proof. The proof simply follows from the fact that for any normally distributed random variable $X \sim \mathcal{N}(\mu, \sigma^2)$ with mean $\mu \in \mathbb{R}$ and $\sigma \in \mathbb{R}_{0+}$, $\text{CVaR}[X] = \mu - \frac{\phi(\Phi(1-\alpha)^{-1})}{\alpha}\sigma$. If the valuation for any group G is normally distributed as $\mathcal{N}(\hat{\boldsymbol{v}}|_G, \boldsymbol{\Sigma}|_G)$, then the utility corresponding to that group has mean $= \frac{1}{|G|}\boldsymbol{a}|_G^\intercal \, \hat{\boldsymbol{v}}|_G$ and variance $= \left(\frac{1}{|G|}\right)^2 \boldsymbol{a}|_G^\intercal \, \boldsymbol{\Sigma}|_G \, \boldsymbol{a}|_G$. Substituting these values in (4.37), we get

$$\max_{\boldsymbol{a}\in\mathcal{Z}} \min_{G\in\mathcal{G}} \left(\frac{1}{|G|} \cdot \boldsymbol{a}|_G^\intercal \, \hat{\boldsymbol{v}}_G\right) - \frac{1}{|G|} \cdot \frac{\phi(\Phi^{-1}(\alpha))}{(1-\alpha)} \sqrt{\sum_{G\in\mathcal{G}} \boldsymbol{a}|_G^\intercal \, \boldsymbol{\Sigma}|_G \, \boldsymbol{a}|_G}. \tag{4.43}$$

Introducing a slack variable t to represent a lower bound on the group utilities and rearranging the terms we get, we get

$$\begin{aligned}
&\max_{\boldsymbol{a}\in\mathcal{Z}, t\in\mathbb{R}} \; t \\
&\forall G \in \mathcal{G} : \left(\frac{1}{|G|} \cdot \boldsymbol{a}|_G^\intercal \, \hat{\boldsymbol{v}}|_G - t\right) \geq \left(\frac{1}{|G|} \cdot \frac{\phi(\Phi^{-1}(\alpha))}{(1-\alpha)}\right) \sqrt{\boldsymbol{a}|_G^\intercal \, \boldsymbol{\Sigma}|_G \, \boldsymbol{a}|_G} \\
&\forall G \in \mathcal{G} : \left(\frac{1}{|G|} \cdot \boldsymbol{a}|_G^\intercal \, \hat{\boldsymbol{v}}|_G - t\right) \geq 0.
\end{aligned} \tag{4.44}$$

Squaring the quadratic constraint on both sides and adding a constraint to ensure that the non-negativity of the L.H.S of each each group constraint in (4.44) is retained after squaring, gives us the final result. □

The problem in (4.42) is a second order conic program (SOCP) and can be solved using popular SOCP solvers in CVXPY library [53]. In general, the sampling approach to CVaR requires solving linear programs with a large number of samples to be effective, which makes them computationally expensive. One potential solution is to leverage importance sampling methods to reduce the variance of the estimator [51, 182].

Having laid out strategies for optimizing USW and GESW in the maximin robust and CVaR stochastic optimization regimes, we now explore these objectives on real data.

4.6 Experiments

We compare the uncertainty-unaware solutions (optimizing for the USW or GESW of a central estimate $\hat{\boldsymbol{v}}$) to the robust and CVaR approaches. Although we do not have access to ground truth performance measures, we demonstrate through simulations that the robust and CVaR approaches can retain higher true performance than the uncertainty-unaware solution, when noise levels are non-negligible.

We use two main experimental setups here.

ICLR. We create uncertainty sets using five years of ICLR data, by constructing a multivariate Gaussian and taking a confidence interval (as outlined by the example in (4.4)). We also truncate all values to be within $[0, 1]$. We then apply the sub-gradient ascent algorithm.

We use the OpenReview API to collect all papers submitted (both accepted and rejected) to five recent iterations of ICLR (2018–2022). Following recent work, we use the pool of authors for each year as the reviewer pool, since we do not have access to the true reviewer identities for these conferences. The number of reviewers and papers for each conference year is shown in Table 4.2.

For each author in each year, we collect the multiset of keywords from papers the author submitted to ICLR in the current or previous years. We then follow a procedure similar to that of AAAI 2021 [100] to convert keywords into a mean vector $\boldsymbol{\mu} \in \mathbb{R}^{nm}$, and we also construct a covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}_{\geq 0}^{nm \times nm}$ for paper-reviewer affinity scores. Vector $\boldsymbol{p}$ is set so $\boldsymbol{p}_i$ is an indicator for keyword i on paper p. Vector $\boldsymbol{r}$ is initially set so $\boldsymbol{r}_i$ is the number of times the keyword i appears on a paper written by that reviewer in this or previous years' conferences. We then modify the values (but not the ordering) of $\boldsymbol{r}$ such that the minimum non-zero value is 0.2, the maximum value is 1, and the remaining non-zero values are evenly-spaced between 0.2 and 1. Let $\boldsymbol{\lambda} \in \mathbb{R}^V$ be such that $\boldsymbol{\lambda}_i = \left(\frac{1}{2}\right)^{i-1}$. *Sorted* represents the function that sorts values of a vector in decreasing order, $M_{\boldsymbol{p}}$ and $M_{\boldsymbol{r}}$ denote the number of non-zero entries in $\boldsymbol{p}$ and $\boldsymbol{r}$ respectively, and $X = \sum_{i=1}^{M_{\boldsymbol{p}}} \left(\frac{1}{2}\right)^{i-1}$. We set $\boldsymbol{\mu}_{pr} = \frac{\boldsymbol{\lambda} \cdot Sorted(\boldsymbol{p} \circ \boldsymbol{r})}{X}$, $\boldsymbol{\Sigma}_{im+j,im+j} = (M_{\boldsymbol{p}} M_{\boldsymbol{r}})^{-2}$, and all off-diagonal entries of $\boldsymbol{\Sigma}$ are 0. This procedure was chosen to roughly mirror the procedure used by AAAI 2021. For each year of ICLR, we set the uncertainty set $\mathcal{V}$ for robust optimization to be the 95% confidence interval for the distribution $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ (as in (4.4)), intersected with the unit hypercube (Lemma 4.4.9).

For each year of ICLR, we sample without replacement 60% of the reviewers and 60% of the papers 100 times, to produce more data for statistical robustness of our experiments. We assume that all papers require 3 reviews and all reviewers can review up to 6 papers. There are no conflicts of interest.

AAMAS. We also run experiments on three reviewer assignment datasets containing only bids. The datasets contain bids from the International Conference on Autonomous Agents and Multiagent Systems (AAMAS) 2015, 2016, and 2021 [107, 108, 109].

Reviewers issue bids of `yes`, `maybe`, `no`, or `no response`. We run two experiments with this data. In one, we binarize the bids such that `yes` and `maybe` are considered affirmative and `no` is considered negative, while in the other we convert the bids to numerical scores such that `yes` is 1, `maybe` is .5, and `no` is 0.01. Under the binarized model, we fit a logistic matrix factorization model to predict whether the bid is affirmative or negative, and in the continuous model, we fit a Gaussian process matrix factorization model [97]. We derive probability distributions and uncertainty sets from these models.

Both models define probability distributions over outcomes, which we use to compute and evaluate the CVaR of utilitarian and egalitarian welfare. For the logistic model, we build a polyhedral uncertainty set by estimating the cross-entropy loss on a held-out test set, and for the Gaussian process model we use the uncertainty set described by (4.4).

For the binarized bids, we first set aside some of the observed bids as a test set. We estimate the missing bids and the bids for the held-out test pairs using logistic matrix factorization. Setting a hidden dimension size d, we construct two matrices $\mathbf{X} \in \mathbb{R}^{n \times d}$ and $\mathbf{Y} \in \mathbb{R}^{m \times d}$. We set $d = 20$. Let $\mathbf{V}^*$ denote the true binarized bid matrix, where we observe entries for the training set pairs $(a, i) \in \mathcal{T}$. We predict the probability of an affirmative bid as $\sigma((\mathbf{X}\mathbf{Y}^\intercal)_{a,i})$ where σ is the logistic sigmoid function. We select $\mathbf{X}$ and $\mathbf{Y}$ to minimize

$$\sum_{(a,i) \in \mathcal{T}} -\mathbf{V}^*_{a,i} \ln\left(\sigma((\mathbf{X}\mathbf{Y}^\intercal)_{a,i})\right) - \mathbf{V}^*_{a,i} \ln\left((\mathbf{X}\mathbf{Y}^\intercal)_{a,i}\right).$$

For CVaR, we take samples from the distribution defined by $\sigma(\mathbf{X}\mathbf{Y}^\intercal)$, assuming all pairs are independently-distributed. We also construct an uncertainty set as described at the end of Section 4.4.1 using the cross-entropy loss on the test pairs.

Under the Gaussian process matrix factorization model [97], we simply predict a mean and variance of a normal distribution for each reviewer-paper pair. We can then sample values independently for each pair, or give a confidence interval for the joint Gaussian with $mn - 1$ degrees of freedom.

These datasets do not contain groups of papers and reviewers, so we create 4 roughly balanced clusters of reviewers and papers for each dataset. Given the real-valued bids in the set $\{0.01, .5, 1\}$ we set unknown bids to be 0. We then construct a graph with all reviewers and papers as nodes, and the bid score between reviewers and papers is the edge weight. All inter-reviewer and inter-paper edges are set to 0 edge weight. We apply spectral embedding with 5 dimensions to transform the nodes into vectors, and cluster the resulting vectors into 4 clusters to obtain 4 groups containing both papers and reviewers. To ensure a balance of reviewers and papers across clusters, we employ Lloyd's algorithm for KMeans clustering with the modification that during each assignment step we enforce a lower bound on the number of papers and number of reviewers assigned to each cluster.

We define our valid set of assignments $\mathcal{Z}$ as follows. For each paper $i \in N$, we set $\underline{\mathbf{k}}_i^N = \overline{\mathbf{k}}_i^N = 3$ for all a in AAMAS 2015, and $\underline{\mathbf{k}}_i^N = \overline{\mathbf{k}}_i^N = 2$ for all i in AAMAS 2016 and 2021. For each reviewer $g \in M$, we set $\underline{\mathbf{k}}_g^M = 0$ and $\overline{\mathbf{k}}_g^M = 15$ for 2015 and 2016 and 4 for 2021. We optimize and evaluate $\text{CVaR}_{0.01}$; we take $4,000$ samples from the distribution to optimize for CVaR using the sampling-based approach, and we take $10,000$ samples to estimate the CVaR for evaluation. We optimize and evaluate the adversarial welfares at the $\delta = 0.3$ level (there is a 70% chance the true values lie in the uncertainty set). All results are averaged over 5 runs of subsampling 20% of each dataset.

4.6.1 Results

Table 4.2: Adversarial and average welfare (mean ± standard deviation) for naïve LP, `FairFlow`, `PR4A`, `FairSequence`, and Algorithm 8 on five ICLR conferences.

Year	m	n	Adversarial USW · 100 (↑)					Average USW · 100 (↑)				
			LP	FF	PR4A	FS	Alg. 8	LP	FF	PR4A	FS	Alg. 8
2018	1657	546	**17**±3	7±3	**17**±3	16±3	16±3	**179**±2	134±12	177±2	177±2	160±4
2019	2620	851	22±2	12±2	22±2	22±2	**27**±3	**184**±1	139±9	**184**±1	183±1	161±3
2020	4123	1327	17±2	11±2	18±2	17±2	**23**±2	**187**±1	158±8	**187**±1	186±1	166±5
2021	4662	1557	23±2	18±2	23±2	23±2	**33**±3	**192**±1	177±2	**192**±1	191±1	174±6
2022	5023	1576	28±2	23±2	28±2	27±2	**38**±2	**191**±1	177±1	190±1	190±1	172±3

Overall Performance. The results of the ICLR experiments are shown in Table 4.2. Welfare is scaled by 100 for ease of comparison. Adversarial welfare is consistently highest (**bold**) using Algorithm 8, except for 2018, which is within one standard deviation.

Table 4.3: Performance of different allocations across each metric on the AAMAS 2015 dataset.

Allocation	Evaluation Objective (↑)					
	USW	GESW	CVaR USW	CVaR GESW	Rob. USW	Rob. GESW
USW	$\mathbf{1.00 \pm 0}$	$\mathbf{1.00 \pm 0}$	$\mathbf{1.00 \pm 0}$	$\mathbf{1.00 \pm 0}$	0 ± 0	0 ± 0
GESW	0.97 ± 0.01	$\mathbf{1.00 \pm 0}$	0.97 ± 0.01	0.97 ± 0.02	0 ± 0	0 ± 0
CVaR USW	$\mathbf{1.00 \pm 0}$	0.99 ± 0	$\mathbf{1.00 \pm 0}$	0.99 ± 0	0 ± 0	0 ± 0
CVaR GESW	0.98 ± 0	0.99 ± 0	0.97 ± 0.01	$\mathbf{1.00 \pm 0}$	0 ± 0	0 ± 0
Rob. USW	0.92 ± 0.01	0.90 ± 0.02	0.92 ± 0.01	0.90 ± 0.02	$\mathbf{1.00 \pm 0}$	$\mathbf{1.00 \pm 0}$
Rob. GESW	0.89 ± 0.04	0.85 ± 0.06	0.89 ± 0.04	0.86 ± 0.06	0.88 ± 0.02	$\mathbf{1.00 \pm 0}$

Table 4.4: Performance of different allocations across each metric on the AAMAS 2016 dataset.

Allocation	Evaluation Objective (↑)					
	USW	GESW	CVaR USW	CVaR GESW	Rob. USW	Rob. GESW
USW	$\mathbf{1.00 \pm 0}$	$\mathbf{1.00 \pm 0}$	$\mathbf{1.00 \pm 0}$	$\mathbf{1.00 \pm 0}$	0 ± 0	0 ± 0
GESW	0.99 ± 0	$\mathbf{1.00 \pm 0}$	0.99 ± 0	0.99 ± 0.01	0 ± 0	0 ± 0
CVaR USW	0.99 ± 0	0.98 ± 0.01	0.99 ± 0	0.98 ± 0.01	0 ± 0	0 ± 0
CVaR GESW	0.99 ± 0.01	0.99 ± 0.01	0.98 ± 0.01	$\mathbf{1.00 \pm 0}$	0 ± 0	0 ± 0
Rob. USW	0.91 ± 0.02	0.87 ± 0.03	0.91 ± 0.02	0.90 ± 0.03	$\mathbf{1.00 \pm 0}$	$\mathbf{1.00 \pm 0}$
Rob. GESW	0.76 ± 0.05	0.66 ± 0.04	0.76 ± 0.05	0.65 ± 0.05	0.74 ± 0.10	$\mathbf{1.00 \pm 0}$

Table 4.3 shows the results for the binarized version of AAMAS 2015 bids. For the binarized AAMAS 2016 and 2021 datasets, Tables 4.4 and 4.5 show the performance of the baseline USW and GESW maximizing allocations, the CVaR$_{0.01}$ USW and GESW maximizing allocations, and the adversarially-robust USW and GESW maximizing allocations at the $\delta = 0.3$ level. Because so many of the bids in AAMAS 2021 are recorded as `no`, since `no` is the default bid, we randomly select 90% of the `no` bids to be converted to `no response`.

Tables 4.6 to 4.8 show the same results for the Gaussian matrix factorization version of the 3 datasets, with the CVaR$_{0.01}$ estimated by sampling from the estimated Gaussian distribution, and the adversarial welfare computed over the truncated ellipsoidal uncertainty set corresponding to the $1 - \delta$ confidence interval of the Gaussian. Results are not reported

Table 4.5: Performance of different allocations across each metric on the AAMAS 2021 dataset.

Allocation	Evaluation Objective (↑)					
	USW	GESW	CVaR USW	CVaR GESW	Rob. USW	Rob. GESW
USW	**1.00 ± 0**	**1.00 ± 0**	**1.00 ± 0**	**1.00 ± 0**	0 ± 0	0.40 ± 0.49
GESW	**1.00 ± 0**	**1.00 ± 0**	**1.00 ± 0**	**1.00 ± 0**	0 ± 0	0.40 ± 0.49
CVaR USW	**1.00 ± 0**	**1.00 ± 0**	**1.00 ± 0**	**1.00 ± 0**	0 ± 0	0.40 ± 0.49
CVaR GESW	**1.00 ± 0**	**1.00 ± 0**	0.99 ± 0	**1.00 ± 0**	0 ± 0	0.40 ± 0.49
Rob. USW	0.85 ± 0.04	0.69 ± 0.14	0.84 ± 0.05	0.64 ± 0.19	**1.00 ± 0**	**1.00 ± 0**
Rob. GESW	0.48 ± 0.09	0.32 ± 0.12	0.43 ± 0.09	0.20 ± 0.12	0.07 ± 0.08	**1.00 ± 0**

for the adversarial GESW approach, since the basic sub-gradient ascent approach fails to converge even after $1,000$ iterations.

Table 4.6: Performance of different allocations across each metric on the Gaussian AAMAS 2015 dataset.

Allocation	Evaluation Objective (↑)					
	USW	GESW	CVaR USW	CVaR GESW	Rob. USW	Rob. GESW
USW	**1.00 ± 0**	0.95 ± 0.03	**1.00 ± 0**	0.94 ± 0.04	0.61 ± 0.19	0.34 ± 0.34
GESW	0.87 ± 0.08	**1.00 ± 0**	0.86 ± 0.09	0.98 ± 0.02	0.42 ± 0.30	0.32 ± 0.35
CVaR USW	**1.00 ± 0**	0.94 ± 0.03	**1.00 ± 0**	0.96 ± 0.04	0.63 ± 0.19	0.35 ± 0.34
CVaR GESW	0.90 ± 0.06	0.99 ± 0.01	0.90 ± 0.07	**1.00 ± 0**	0.51 ± 0.26	0.36 ± 0.33
Rob. USW	0.86 ± 0.07	0.76 ± 0.12	0.88 ± 0.06	0.80 ± 0.10	**1.00 ± 0**	0.99 ± 0.01
Rob. GESW	0.75 ± 0.13	0.77 ± 0.12	0.76 ± 0.13	0.82 ± 0.09	0.87 ± 0.09	**1.00 ± 0**

Each row shows the metrics for the allocation produced by the method which optimizes for the objective shown in the left-most column. Objective values are normalized by dividing by the maximum value of that objective per seed. All methods have 0 adversarial welfare, even at the $\delta = 0.3$ level, indicating that if robustness to adversarial noise is desired, it is very important to consider this objective explicitly. We approximate the optimal $\text{CVaR}_{0.01}$ using 1000 samples, which leaves some room for sampling error as evidenced by the strong performance of the baseline USW and GESW allocations on the $\text{CVaR}_{0.01}$ measure. However, relatively little noise is actually present in this dataset, as the $\text{CVaR}_{0.01}$ is relatively high for both USW and GESW in all cases.

Table 4.7: Performance of different allocations across each metric on the Gaussian AAMAS 2016 dataset.

Allocation	Evaluation Objective (↑)					
	USW	GESW	CVaR USW	CVaR GESW	Rob. USW	Rob. GESW
USW	$\mathbf{1.00 \pm 0}$	0.99 ± 0.01	$\mathbf{1.00 \pm 0}$	0.99 ± 0.02	0.47 ± 0.27	0.25 ± 0.38
GESW	0.91 ± 0.06	$\mathbf{1.00 \pm 0}$	0.91 ± 0.07	0.98 ± 0.01	0.37 ± 0.32	0.24 ± 0.38
CVaR USW	$\mathbf{1.00 \pm 0}$	0.98 ± 0.02	$\mathbf{1.00 \pm 0}$	0.99 ± 0.01	0.52 ± 0.25	0.27 ± 0.37
CVaR GESW	0.92 ± 0.05	0.98 ± 0.02	0.92 ± 0.06	$\mathbf{1.00 \pm 0}$	0.41 ± 0.31	0.28 ± 0.37
Rob. USW	0.84 ± 0.08	0.77 ± 0.12	0.86 ± 0.07	0.84 ± 0.09	$\mathbf{1.00 \pm 0}$	$\mathbf{1.00 \pm 0}$
Rob. GESW	0.73 ± 0.14	0.76 ± 0.13	0.74 ± 0.13	0.84 ± 0.09	0.85 ± 0.09	$\mathbf{1.00 \pm 0}$

Table 4.8: Performance of different allocations across each metric on the Gaussian AAMAS 2021 dataset.

Allocation	Evaluation Objective (↑)					
	USW	GESW	CVaR USW	CVaR GESW	Rob. USW	Rob. GESW
USW	$\mathbf{1.00 \pm 0}$	$\mathbf{1.00 \pm 0.01}$	$\mathbf{1.00 \pm 0}$	$\mathbf{1.00 \pm 0.01}$	0.53 ± 0.26	0.21 ± 0.40
GESW	0.80 ± 0.12	$\mathbf{1.00 \pm 0}$	0.79 ± 0.12	0.99 ± 0.01	0.24 ± 0.39	0.20 ± 0.40
CVaR USW	$\mathbf{1.00 \pm 0}$	$\mathbf{1.00 \pm 0.01}$	$\mathbf{1.00 \pm 0}$	$\mathbf{1.00 \pm 0.01}$	0.53 ± 0.26	0.21 ± 0.40
CVaR GESW	0.85 ± 0.08	$\mathbf{1.00 \pm 0}$	0.84 ± 0.08	$\mathbf{1.00 \pm 0}$	0.36 ± 0.34	0.20 ± 0.40
Rob. USW	0.81 ± 0.11	0.69 ± 0.16	0.81 ± 0.11	0.71 ± 0.16	$\mathbf{1.00 \pm 0}$	$\mathbf{1.00 \pm 0.01}$
Rob. GESW	0.70 ± 0.17	0.68 ± 0.17	0.71 ± 0.17	0.70 ± 0.16	0.88 ± 0.10	$\mathbf{1.00 \pm 0}$

Importance of the GESW Objective We run several experiments to further highlight the distinction between the USW and GESW objectives. Figure 4.1 shows the results of this experiment. We synthetically add extra papers to the AAMAS 2015 dataset, by first copying uniformly at random some papers from the dataset and then modifying them. We modify the papers by dividing their valuations and setting all but the top few valuations to 0. We then consider GESW when treating the real papers as one group and the artificial papers as another group. Results are reported as we vary the divisor applied to artificially scale the artificial group's valuations, the size of the artificial group (reported in the figure as the

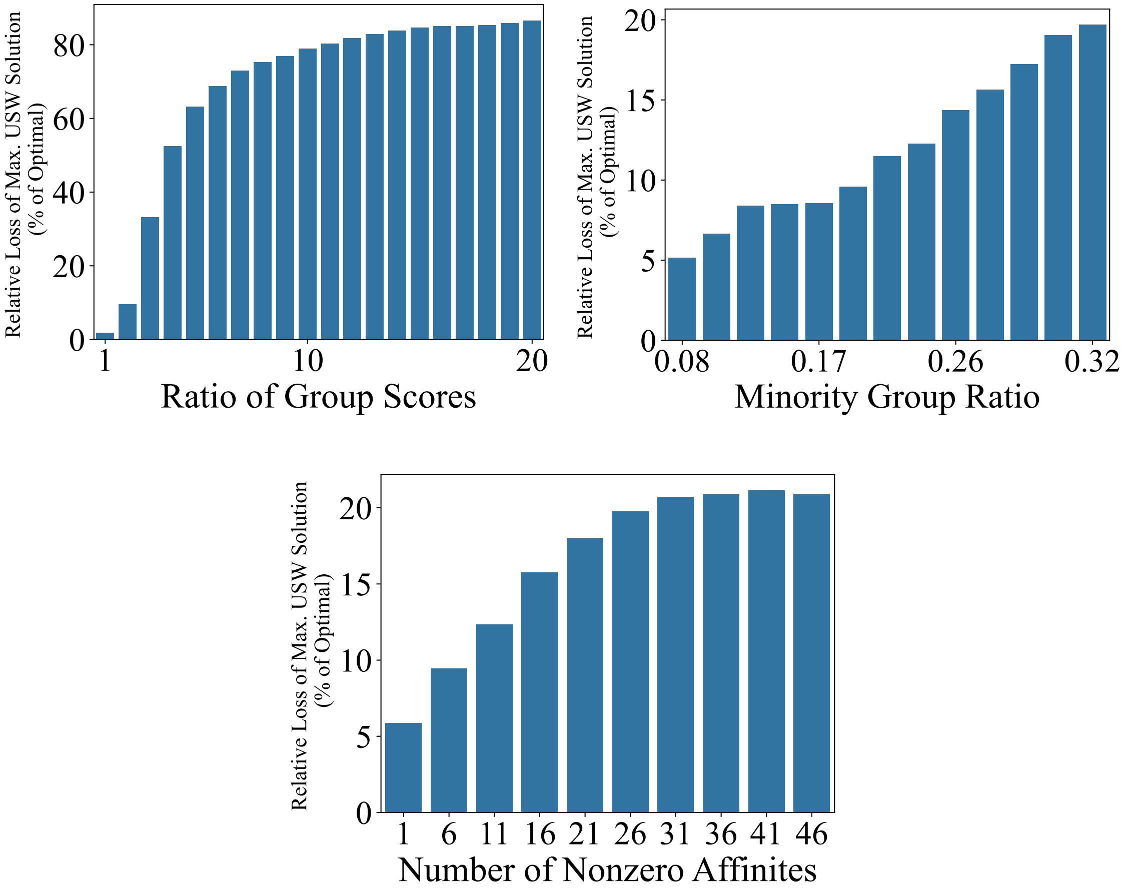


Figure 4.1: Relative loss (in GESW) of the maximum USW solution, compared to the optimal GESW solution.

ratio of the minority group size to the total number of papers), and the number of valuations per paper that are artificially set to 0. As all three parameters increase, the GESW of the USW-optimal solution decreases relative to the GESW-optimal solution.

CVaR vs. Robustness under Left-Skewed Distributions CVaR USW behaves similarly to the uncertainty-unaware USW maximal approach in Figure 4.3.

The utility of CVaR is higher when welfare distributions exhibit a left fat-tail, meaning a greater probability mass is concentrated in the left tail. Unlike the robust (minimax) approach, the CVaR method, particularly at higher α values, balances between extreme pessimism and optimizing for the average performance. This is advantageous in allocation problems with high uncertainty, where worst-case optimization can lead to overly pessimistic and inefficient outcomes.

However, it's crucial to differentiate between when to apply CVaR and when the robust approach is more appropriate. The robust approach is better suited to scenarios with extremely high stakes, where any failure is unacceptable—such as life-or-death situations (e.g., allocating medical experts). It is also effective in low-uncertainty contexts where optimizing for the worst case is reasonable and does not significantly reduce efficiency.

There are several reasons why CVaR USW behaves very similarly to the uncertainty-unaware USW maximal. When the valuation matrix is sampled from a Gaussian distribution, the USW is just the mean of normal random variables. The standard deviation of the utilitarian welfare is $\sqrt{\sum_a \sum_i \sigma_{a,i}}/nm$. Due to this the variance of the utilitarian measure is fairly low, and USW is a more stable measure compared to GESW. Second, we were sampling valuations from symmetrical Gaussian distributions and so the noise in the valuations was (mostly) getting averaged out. Finally, we have a large number of items with very small variance. In AAMAS 2015 and 2016, around $8 - 9\%$ of the entries have variance less than 0.005.

We perform an experiment where we model valuations using a negatively-skewed normal distribution with the same means and variances as those estimated for AAMAS 2015. We see increasing robustness of CVaR relative to uncertainty-unaware USW (Table 4.9). The difference is sharper as the skew parameter gets more negative. We optimize and evaluate for $\text{CVaR}_{0.3}$, and robust USW is optimized at the $\delta = 0.3$ level, for a normal distribution/ellipsoid. This is actually somewhat optimistic, since the true worst-case will be a bit more extreme with the additional skewness.

Table 4.9: $\text{CVaR}_{0.3}$ of USW under left-skewed Gaussians for the uncertainty-unaware, CVaR, and robust approaches.

Skew	CVaR USW	USW	Robust USW
-0.5	**1.64**	1.56	1.01
-1	**1.45**	1.21	0.86
-2	**1.33**	0.96	0.75
-5	**1.29**	0.84	0.70
-10	**1.28**	0.82	0.52

Robustness under Increasing Uncertainty. To demonstrate the importance of optimizing for the adversarial case, we perform an experiment where we simulate the effect of adding many low-quality, high-variance reviewers to the MIDL dataset (used in Chapter 2), and a similar experiment where we systematically overestimate some of the affinities for a subset of papers. These reviewers can often appear in modern conference reviewing. PhD students with few papers will tend to have higher variance in true expertise relative to document-based similarity scores derived from their prior work, while reviewers with fewer bids will tend to have higher variance in true interest relative to their bids. Intuitively, the systematic overestimation of papers' affinities can occur when paper authors submit a keyword that does not have the exact meaning they expected or is listed by reviewers in the wrong subcommunity. We believe that this occurs quite often, and leave a rigorous examination of this important question to future work.

We assume the true affinity of the 177 original reviewers for each paper is equal to the affinity score present in the public dataset, but is *noisily estimated.* Thus, we assume that the conference organizers have access to estimated affinity scores which are equal to the original affinity scores plus normally distributed noise, $\text{Normal}(0, .02)$. We also assume the conference organizers know that the estimation error is distributed according to $\text{Normal}(0, .02)$ for each request-expert pair. We also add a number of “dummy” reviewers to the dataset. For each

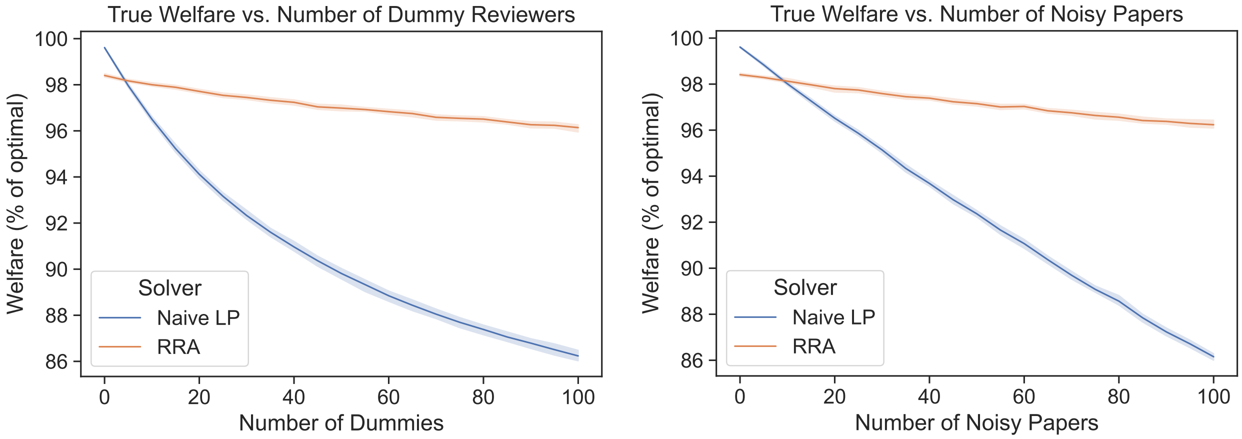


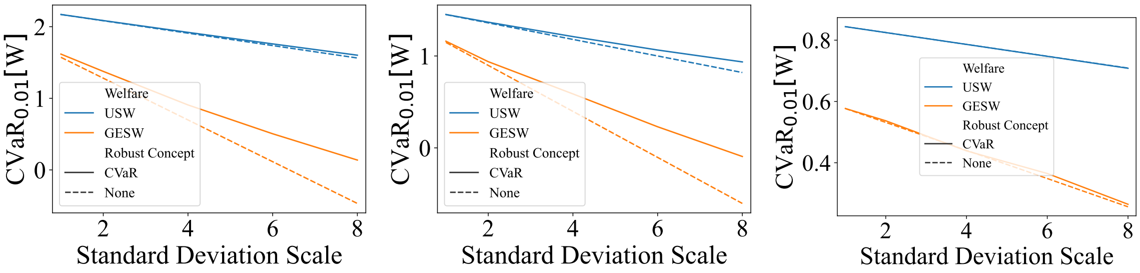
Figure 4.2: Left: True welfare (as percentage of optimal, when $\mathbf{V}^*$ is known) of naïve LP approach vs. Algorithm 8 on MIDL 2018 dataset with increasing number of "dummy" reviewers. Right: True welfare on MIDL 2018 dataset with increasing number of noisy papers.

dummy reviewer, we set the true affinity of that reviewer to be .1 for all papers, and sample the estimated affinity from Normal$(.1, .15)$. As with the original reviewers, we assume the conference organizers know the estimation error is distributed according to Normal$(0, .15)$ for each request-expert pair. This simple setup implies a multivariate Gaussian distribution over the true affinity scores, which are unknown to the conference organizers. We take a 95% confidence interval of this distribution and intersect it with the unit hypercube to define a truncated-ellipsoidal uncertainty set. We then assign reviewers using the naïve LP and Algorithm 8, and compare the *true*, but unknown, welfare for each approach.

For the setting with noisy papers, we take a subset of papers and identify the 20^{th} to 30^{th} (non-inclusive) ranked reviewers for each paper in decreasing order of affinity. For each paper, we add .3 to the estimated affinity for those 10 reviewers. We then assume that conference organizers estimate the standard deviation of these request-expert pairs to be .15, while the remaining standard deviations are estimated at .02.

The results are shown in Figure 4.2. We report the welfare of naïve LP and Algorithm 8 as a percentage of optimal, averaged over 100 runs per number of dummy reviewers/papers.

We also report the minimum and maximum over all 100 runs for each setting. As we add more dummy reviewers or noisy papers, the true welfare of the naïve LP approach drops by up to roughly 15%, while Algorithm 8 maintains high true welfare.

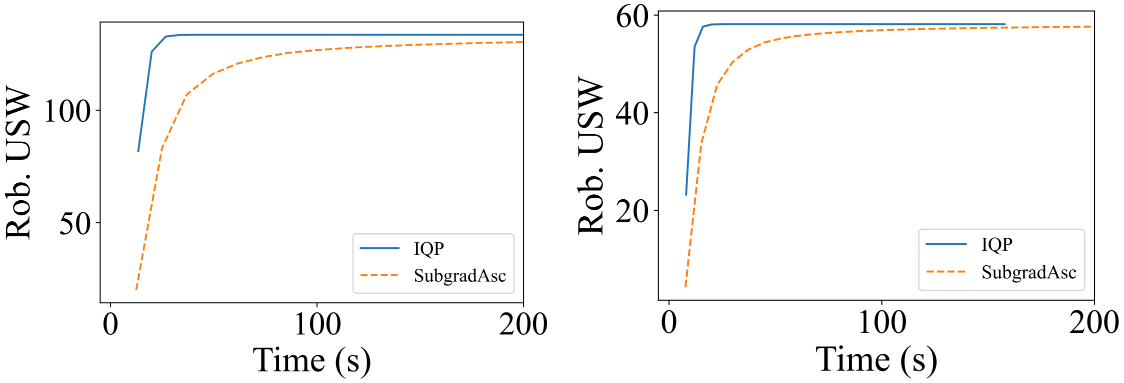


(a) AAMAS 2015 (b) AAMAS 2016 (c) AAMAS 2021

Figure 4.3: CVaR$_{0.01}$ as noise increases for all three datasets, using the estimated Gaussian distribution.

We perform a similar experiment for the AAMAS datasets, to demonstrate the impact of increasing noise on the CVaR metric. Figure 4.3 shows the CVaR$_{0.01}$ on the Gaussian version of all three datasets as we artificially increase the amount of noise. We multiply the standard deviations of the Gaussian distributions by a scalar and optimize for the CVaR or the central estimate of the USW and GESW. We then plot CVaR$_{0.01}$ as noise increases. Although the CVaR approach is less important at low noise levels, the CVaR of welfare decreases for both welfare measures as noise increases. GESW has a sharper decline than USW. We see that as the noise increases, the CVaR$_{0.01}$ of the baseline USW and GESW maximizing allocations drops off relative to the same value for the CVaR-optimized allocation.

Runtime. Finally, for the soft robust optimization setting with ellipsoidal uncertainty sets (derived from confidence intervals over the Gaussian process matrix factorization), we compare the IQP approach Corollary 4.4.17 to projected sub-gradient ascent on the original max-min problem. We find that IQP converges much faster than the subgradient ascent



(a) AAMAS 2015

(b) AAMAS 2016

Figure 4.4: Convergence behavior of the Iterated Quadratic Program (IQP) vs. Adversarial Projected Sub-gradient Ascent approach on AAMAS 2015 and 2016, solving for utilitarian welfare under the estimated Gaussian distribution.

algorithm; see Figure 4.4. Sub-gradient ascent fails to converge in $1,000$ iterations for the adversarial GESW objective on all datasets and the USW objective on AAMAS 2021.

4.7 Costs of Accounting for Uncertainty

In this and the previous chapter, we outlined an end-to-end pipeline for predicting response quality and then assigning experts to requests accounting for uncertainty in predictions. However, accounting for uncertainty requires compromising on other deciderata. In Chapter 2, we focused primarily on the deciderata of fairness to individual requests, welfare efficiency, computation speed, and algorithmic simplicity. Our robustness concepts target welfare efficiency measures, so there is no trade-off for that objective. We show that group egalitarian welfare can be robustly optimized, but it is not generally possible to have robustness on individually fair objectives like egalitarian welfare [91, 157]. It may be possible to have robustness over aggregates of individual measures; for example, the sum of the total envy between papers. Though we did not analyze that objective here, similar techniques to those outlined in this chapter can be used. In addition, the robust objectives we optimize for in this chapter can be augmented with additional bounds on the central estimates of valuation matrices, for example optimizing for CVaR or robust welfare subject to the constraint that our allocation must be EF1 on the centrally-estimated valuations. This additional constraint may result in extra space and runtime requirements, but is not a fundamental limitation.

No matter the approach, robustness to uncertainty necessarily introduces additional complexity and runtime. The degree to which the additional runtime and implementation complexity overshadows the benefits of robustness depends on the degree of noise in the valuations. However, as discussed in Section 3.2, it is very challenging to obtain data linking inputs and outputs for reviewer assignment. Thus, we must leave to future work the question of exactly how much noise exists in predictions of downstream review quality.

4.8 Conclusion

In conclusion, we explore the stochastic and robust optimization regimes for utilitarian and group-wise welfare objectives. The robust optimization algorithms depend on the form of the uncertainty set. Some special cases of uncertainty sets (singleton, hyperrectangular, or spherical) reduce to problems without uncertainty, which can all be solved via linear programming. The general problem is NP-hard, and our adversarial projected sub-gradient ascent algorithm provides approximate solutions with error guarantees dependent on the uncertainty set. We show that when the uncertainty set has linear constraints only, the resulting problem is an LP and can be solved efficiently. Under ellipsoidal constraints, we demonstrate an iterative quadratic programming approach converges much faster than adversarial projected sub-gradient ascent for utilitarian welfare. In the stochastic regime, we lay out the sample complexity of CVaR for the utilitarian welfare objective.

We show two natural settings where the robust optimization approach recovers a higher welfare than the naive approach using central estimates of valuations. Using a simple model, we show that the worst-case welfare on 5 years of conferences simulated from real ICLR data is much better using adversarial projected sub-gradient ascent than any existing approach. We also demonstrate the feasibility of estimating probability distributions and uncertainty sets on three years of bid data from AAMAS, and show that the robust and CVaR approaches combat the uncertainty present in these three datasets. Together with the valuation estimation discussion in Chapter 3, we have proposed an end-to-end pipeline for expert assignment.

CHAPTER 5
CONCLUSION

This thesis proposes an end-to-end pipeline for fast, efficient, fair, and robust expert assignment. In Chapter 2, we modify classic techniques from the fair division literature to obtain a fast, fair, and efficient expert assignment algorithm. In Chapter 3, we demonstrate through an end-to-end example how using past performance of experts can help improve expert assignments. Chapter 4 shows how, given these predictive distributions or high-probability uncertainty sets, we can optimize for high quality expert assignments. We hope that this work encourages further exploration of data driven expert assignment as a general problem. The most important next step for the work of Chapters 3 and 4 is to demonstrate its utility in real conference settings.

Publications Related to This Thesis

This thesis is based on the following publications:

1. Justin Payan and Yair Zick. I will have order! Optimizing orders for fair reviewer assignment. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI)*, pages 440–446, 2022.

2. Cyrus Cousins, Neha Nayak Kennard, Sheshera Mysore, Justin Payan, and Yair Zick. Who you gonna call? Optimizing expert assignment with predictive models. *Submitted*, 2024.

3. Cyrus Cousins, Justin Payan, and Yair Zick. Into the unknown: Assigning reviewers to papers with uncertain affinities. In *Proceedings of the 16th International Symposium on Algorithmic Game Theory (SAGT)*, pages 179–197, 2023.

4. Cyrus Cousins, Elita Lobo, Justin Payan, and Yair Zick. Fair and welfare-efficient constrained multi-matchings under uncertainty. In *Proceedings of the 38th Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

Other Publications

In addition to expert assignment, I studied other topics in fair division, combinatorial optimization, and natural language processing through my Ph.D., resulting in the following publications:

1. Ananya Gupta, Eric Johnson, Justin Payan, Aditya Roy, Ari Kobren, Swetasudha Panda, Michael Wick, and Jean-Baptiste Tristan. Online post-processing in rankings for fair utility maximization. In *Proceedings of the 14th International Conference on Web Search and Data Mining (WSDM)*, pages 454–462, 2021.

2. Justin Payan, Yuval Merhav, He Xie, Satyapriya Krishna, Anil Ramakrishna, Mukund Sridhar, and Rahul Gupta. Towards realistic single-task continuous learning research for NER. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3773–3783, 2021.

3. Justin Payan, Swaroop Mishra, Singh Mukul, Carina Negreanu, Christian Poelitz, Chitta Baral, Subhro Roy, Rasika Chakravarthy, Benjamin Van Durme, and Elnaz Nouri. InstructExcel: A benchmark for natural language instruction in Excel. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4026–4043, 2023.

4. Justin Payan, Rik Sengupta, and Vignesh Viswanathan. Relaxations of envy-freeness over graphs. In *Proceedings of the 22nd International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 2652–2654, 2023.

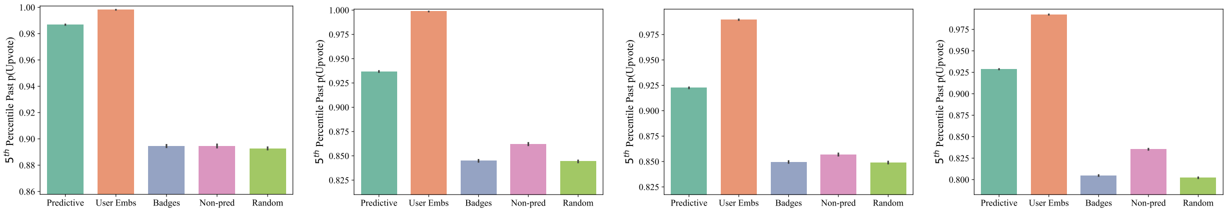
5. Hadi Hosseini, Justin Payan, Rik Sengupta, Rohit Vaish, and Vignesh Viswanathan. Graphical house allocation with identical valuations. *Autonomous Agents and Multi-Agent Systems*, 38(42), 2024.

Although this work does not fit into my primary project on data-driven expert assignment, my collaborations have been incredibly rewarding personally and academically, and some of this work indirectly informed my thesis. Peer review and expert assignment can benefit greatly from the intersection of natural language processing and combinatorial optimization, as Chapter 3 shows. I hope to continue this thread of research in the future.

APPENDIX

ADDITIONAL METRICS FROM STACKEXCHANGE EXPERIMENTS

Figure A.1 shows the performance of all methods on the 5^{th} percentile of user's historical probability of upvote given vote for all StackExchange websites. This metric is very similar to the median of user's historical performance (Figure 3.5).



(a) CS

(b) Biology

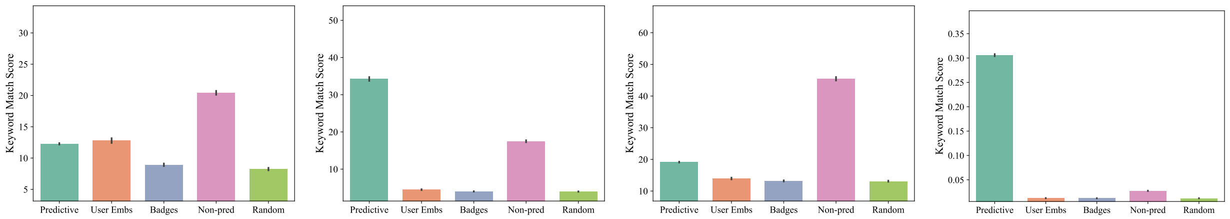
(c) Chemistry

(d) Academia

Figure A.1: Comparison of all models on 5^{th} percentile of historical probability of upvote given vote, on all 4 StackExchange websites.

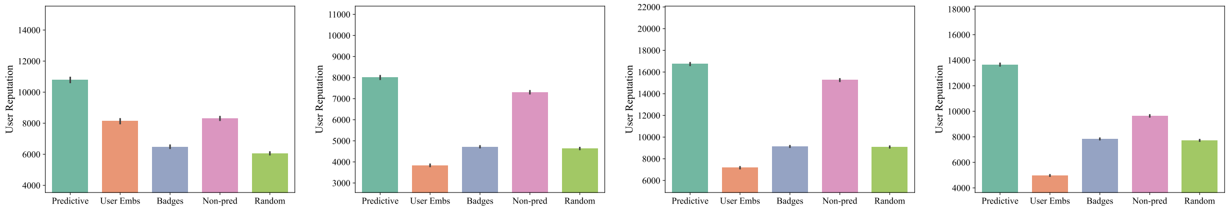
Figure A.2 shows performance measured by the keyword matching score. This measure shows different behavior to the cosine similarity measure (Figure 3.8), which is also a content-based similarity measure. Here, we see that the predictive model outperforms the non-predictive baseline on keyword similarity on two StackExchanges. It is possible that the predictive model is choosing to rely on either keyword-based similarity or cosine similarity, depending on the dataset.

Figure A.3 shows performance as measured by average of the assigned users' reputation scores. In comparing the predictive model and the non-predictive baseline, these plots reflect



(a) CS (b) Biology (c) Chemistry (d) Academia

Figure A.2: Comparison of all models on keyword matching score, on all 4 StackExchange websites.

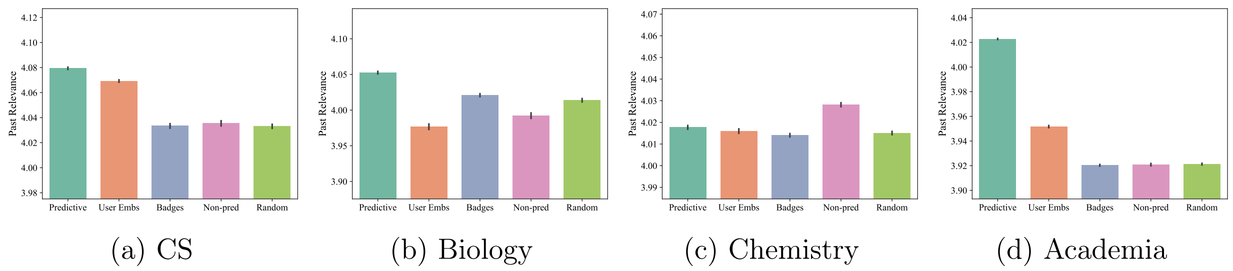


(a) CS (b) Biology (c) Chemistry (d) Academia

Figure A.3: Comparison of all models on user reputation, on all 4 StackExchange websites.

what we already observed from Figures 3.5, 3.6 and A.1, that the predictive model outperforms all models on measures of positive user historical features. However, we do see a marked decrease in the performance of the trained user embedding model, which is curious. Perhaps this implies that past a high enough reputation threshold, reputation is useful for predicting performance but not essential.

Finally, we present Figures A.4 and A.5 showing the assigned users' average LLM-annotated "Informativeness" and "Relevance" over previous answers. These results do not add much to the discussion over the results on "Usefulness" in Figure 3.6.



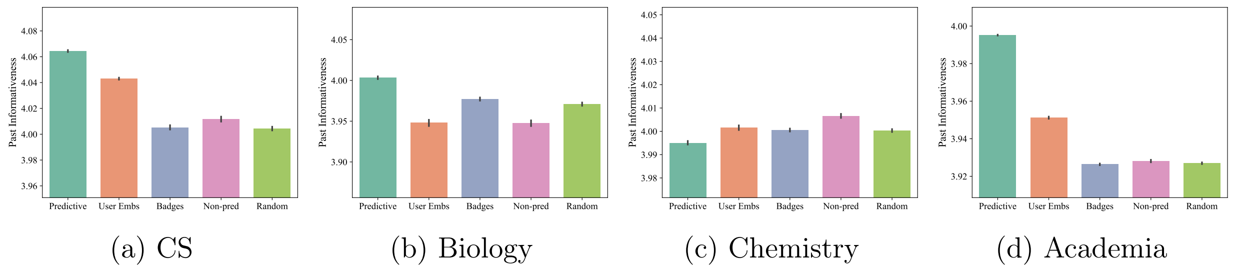
(a) CS

(b) Biology

(c) Chemistry

(d) Academia

Figure A.4: Comparison of all models on LLM-annotated “Relevance” score, on all 4 StackExchange websites.



(a) CS

(b) Biology

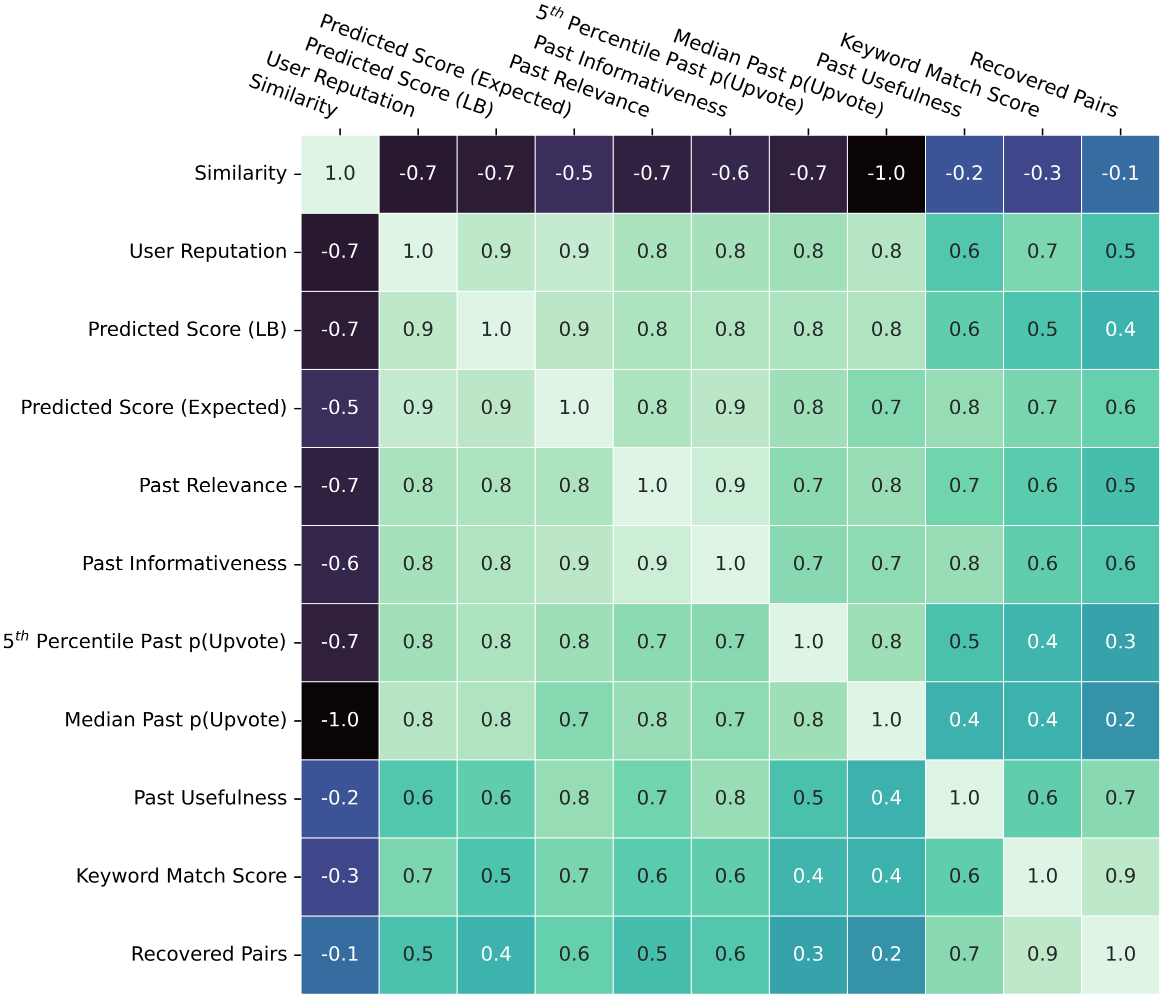
(c) Chemistry

(d) Academia

Figure A.5: Comparison of all models on LLM-annotated “Informativeness” score, on all 4 StackExchange websites.

A.1 Correlation of Rankings

Figures A.6, A.8, A.10 and A.12 display the average correlations (over 1,000 samples) of the rankings over assignments produced by each metric. High positive values indicate the metrics tend to rank assignments in the same order, low negative values indicate the metrics tend to rank assignments in opposite order. Values near zero indicate low correlation in rankings in either direction. The p-values of these statistics are displayed in Figures A.7, A.9, A.11 and A.13.



	Similarity	User Reputation	Predicted Score (LB)	Predicted Score (Expected)	Past Relevance	Past Informativeness	5^{th} Percentile Past p(Upvote)	Median Past p(Upvote)	Past Usefulness	Keyword Match Score	Recovered Pairs
Similarity	1.0	-0.7	-0.7	-0.5	-0.7	-0.6	-0.7	-1.0	-0.2	-0.3	-0.1
User Reputation	-0.7	1.0	0.9	0.9	0.8	0.8	0.8	0.8	0.6	0.7	0.5
Predicted Score (LB)	-0.7	0.9	1.0	0.9	0.8	0.8	0.8	0.8	0.6	0.5	0.4
Predicted Score (Expected)	-0.5	0.9	0.9	1.0	0.8	0.9	0.8	0.7	0.8	0.7	0.6
Past Relevance	-0.7	0.8	0.8	0.8	1.0	0.9	0.7	0.8	0.7	0.6	0.5
Past Informativeness	-0.6	0.8	0.8	0.9	0.9	1.0	0.7	0.7	0.8	0.6	0.6
5^{th} Percentile Past p(Upvote)	-0.7	0.8	0.8	0.8	0.7	0.7	1.0	0.8	0.5	0.4	0.3
Median Past p(Upvote)	-1.0	0.8	0.8	0.7	0.8	0.7	0.8	1.0	0.4	0.4	0.2
Past Usefulness	-0.2	0.6	0.6	0.8	0.7	0.8	0.5	0.4	1.0	0.6	0.7
Keyword Match Score	-0.3	0.7	0.5	0.7	0.6	0.6	0.4	0.4	0.6	1.0	0.9
Recovered Pairs	-0.1	0.5	0.4	0.6	0.5	0.6	0.3	0.2	0.7	0.9	1.0

Figure A.6: Correlation of ranking of assignments by each metric for `cs`.

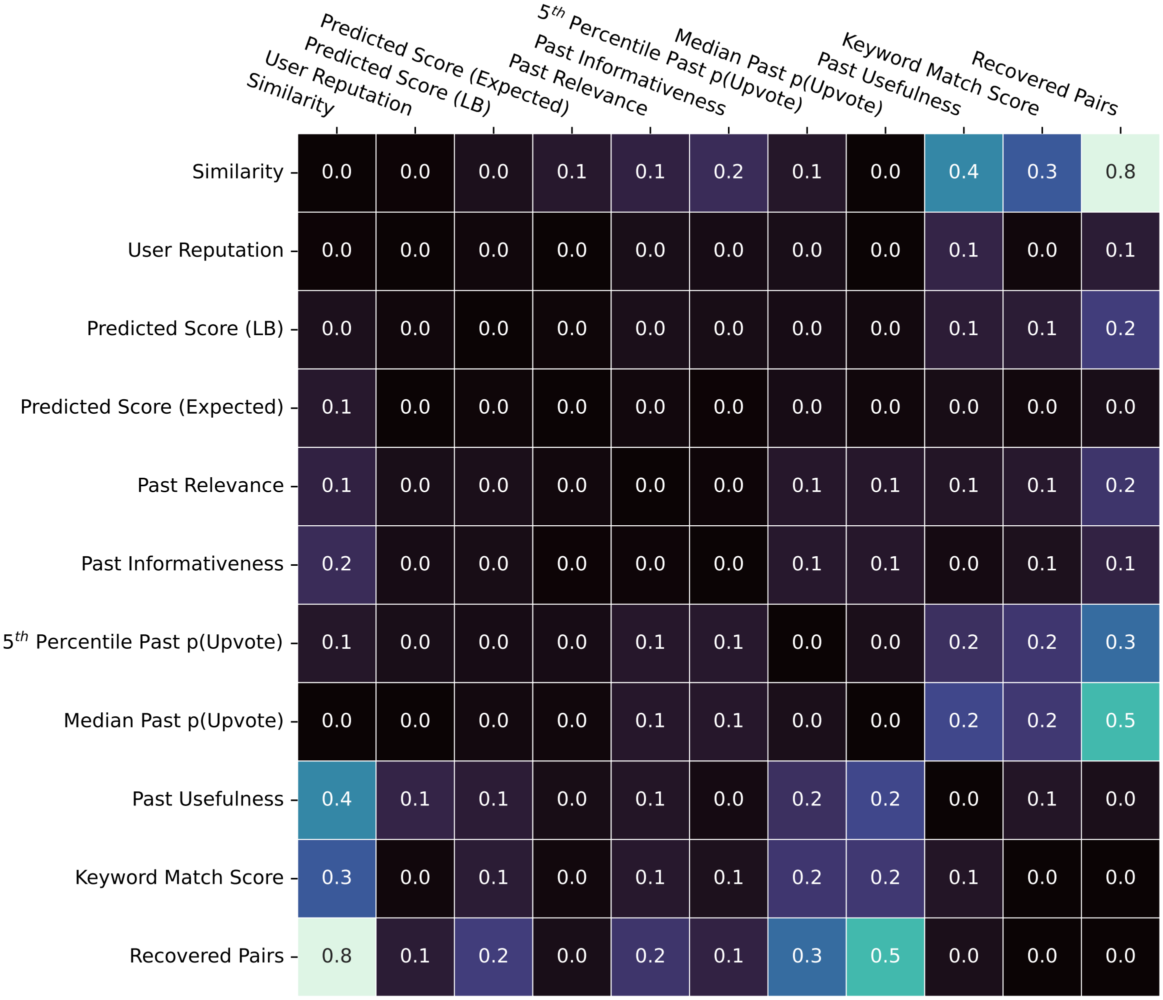


Figure A.7: P-values for correlation of ranking of assignments by each metric for `cs`.

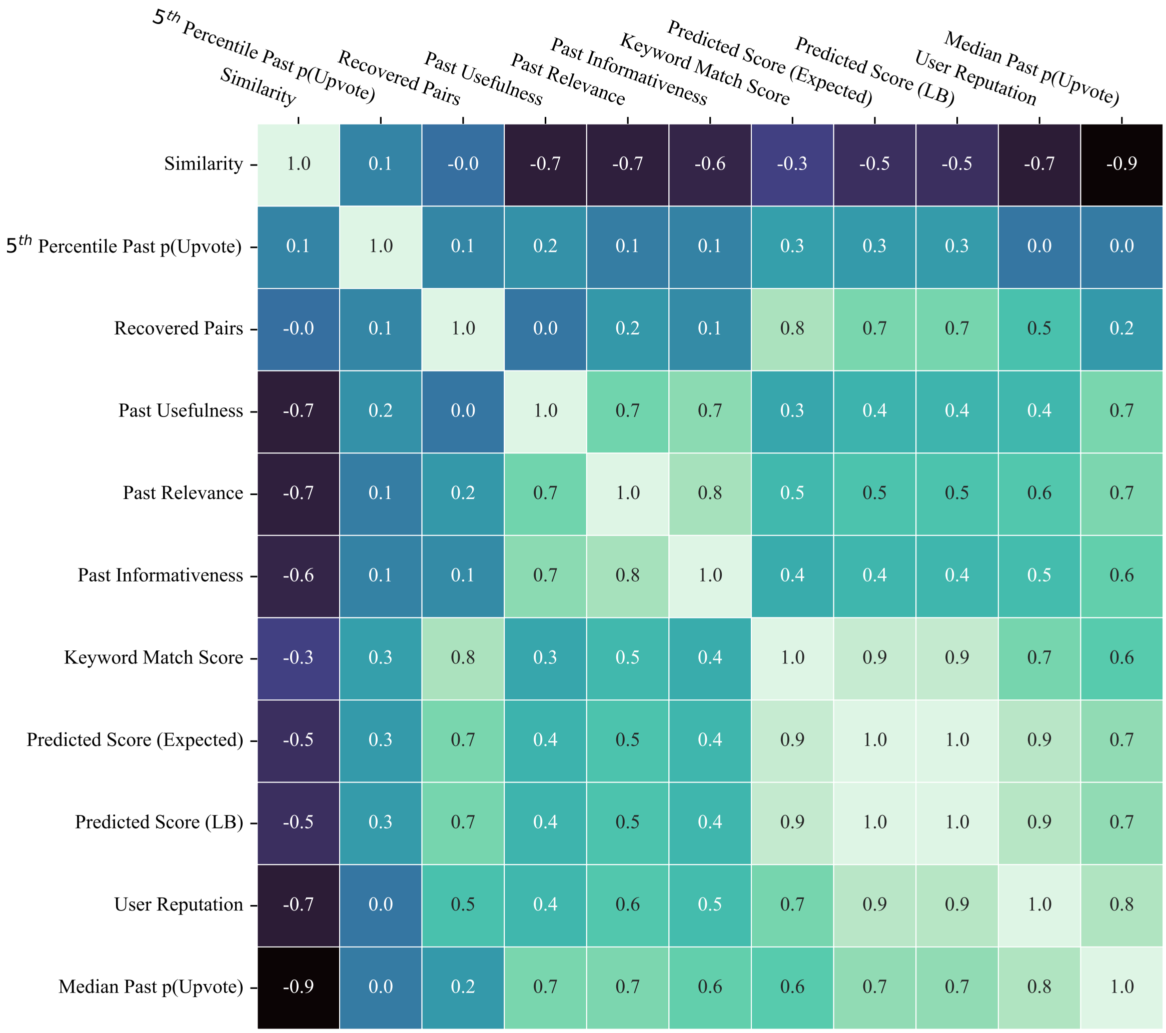
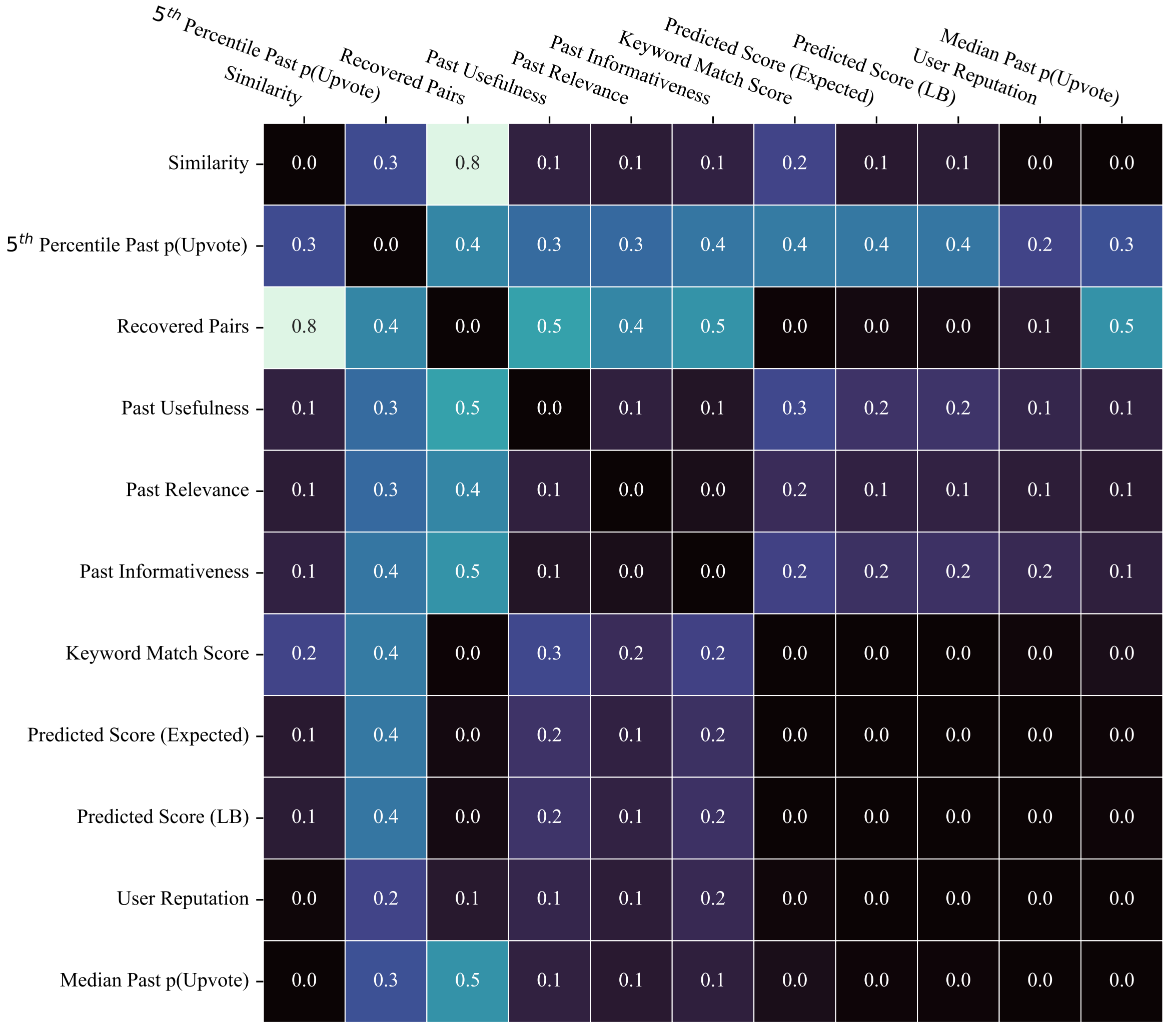


Figure A.8: Correlation of ranking of assignments by each metric for `biology`.



	Similarity	5^{th} Percentile Past p(Upvote)	Recovered Pairs	Past Usefulness	Past Relevance	Past Informativeness	Keyword Match Score	Predicted Score (Expected)	Predicted Score (LB)	User Reputation	Median Past p(Upvote)
Similarity	0.0	0.3	0.8	0.1	0.1	0.1	0.2	0.1	0.1	0.0	0.0
5^{th} Percentile Past p(Upvote)	0.3	0.0	0.4	0.3	0.3	0.4	0.4	0.4	0.4	0.2	0.3
Recovered Pairs	0.8	0.4	0.0	0.5	0.4	0.5	0.0	0.0	0.0	0.1	0.5
Past Usefulness	0.1	0.3	0.5	0.0	0.1	0.1	0.3	0.2	0.2	0.1	0.1
Past Relevance	0.1	0.3	0.4	0.1	0.0	0.0	0.2	0.1	0.1	0.1	0.1
Past Informativeness	0.1	0.4	0.5	0.1	0.0	0.0	0.2	0.2	0.2	0.2	0.1
Keyword Match Score	0.2	0.4	0.0	0.3	0.2	0.2	0.0	0.0	0.0	0.0	0.0
Predicted Score (Expected)	0.1	0.4	0.0	0.2	0.1	0.2	0.0	0.0	0.0	0.0	0.0
Predicted Score (LB)	0.1	0.4	0.0	0.2	0.1	0.2	0.0	0.0	0.0	0.0	0.0
User Reputation	0.0	0.2	0.1	0.1	0.1	0.2	0.0	0.0	0.0	0.0	0.0
Median Past p(Upvote)	0.0	0.3	0.5	0.1	0.1	0.1	0.0	0.0	0.0	0.0	0.0

Figure A.9: P-values for correlation of ranking of assignments by each metric for `biology`.

	Similarity	Past Relevance	Past Informativeness	Keyword Match Score	Recovered Pairs	Median Past p(Upvote)	User Reputation	Predicted Score (Expected)	Predicted Score (LB)	Past Usefulness	5^{th} Percentile Past p(Upvote)
Similarity	1.0	0.3	0.5	-0.2	-0.1	-0.9	-0.7	-0.7	-0.8	-0.4	-0.4
Past Relevance	0.3	1.0	0.8	-0.1	-0.0	-0.3	-0.2	-0.2	-0.2	0.2	0.1
Past Informativeness	0.5	0.8	1.0	-0.3	-0.3	-0.6	-0.6	-0.5	-0.5	-0.1	-0.1
Keyword Match Score	-0.2	-0.1	-0.3	1.0	0.9	0.3	0.6	0.5	0.4	0.4	0.1
Recovered Pairs	-0.1	-0.0	-0.3	0.9	1.0	0.3	0.6	0.5	0.5	0.5	0.1
Median Past p(Upvote)	-0.9	-0.3	-0.6	0.3	0.3	1.0	0.9	0.9	0.9	0.6	0.5
User Reputation	-0.7	-0.2	-0.6	0.6	0.6	0.9	1.0	0.9	0.9	0.6	0.4
Predicted Score (Expected)	-0.7	-0.2	-0.5	0.5	0.5	0.9	0.9	1.0	1.0	0.7	0.6
Predicted Score (LB)	-0.8	-0.2	-0.5	0.4	0.5	0.9	0.9	1.0	1.0	0.7	0.6
Past Usefulness	-0.4	0.2	-0.1	0.4	0.5	0.6	0.6	0.7	0.7	1.0	0.5
5^{th} Percentile Past p(Upvote)	-0.4	0.1	-0.1	0.1	0.1	0.5	0.4	0.6	0.6	0.5	1.0

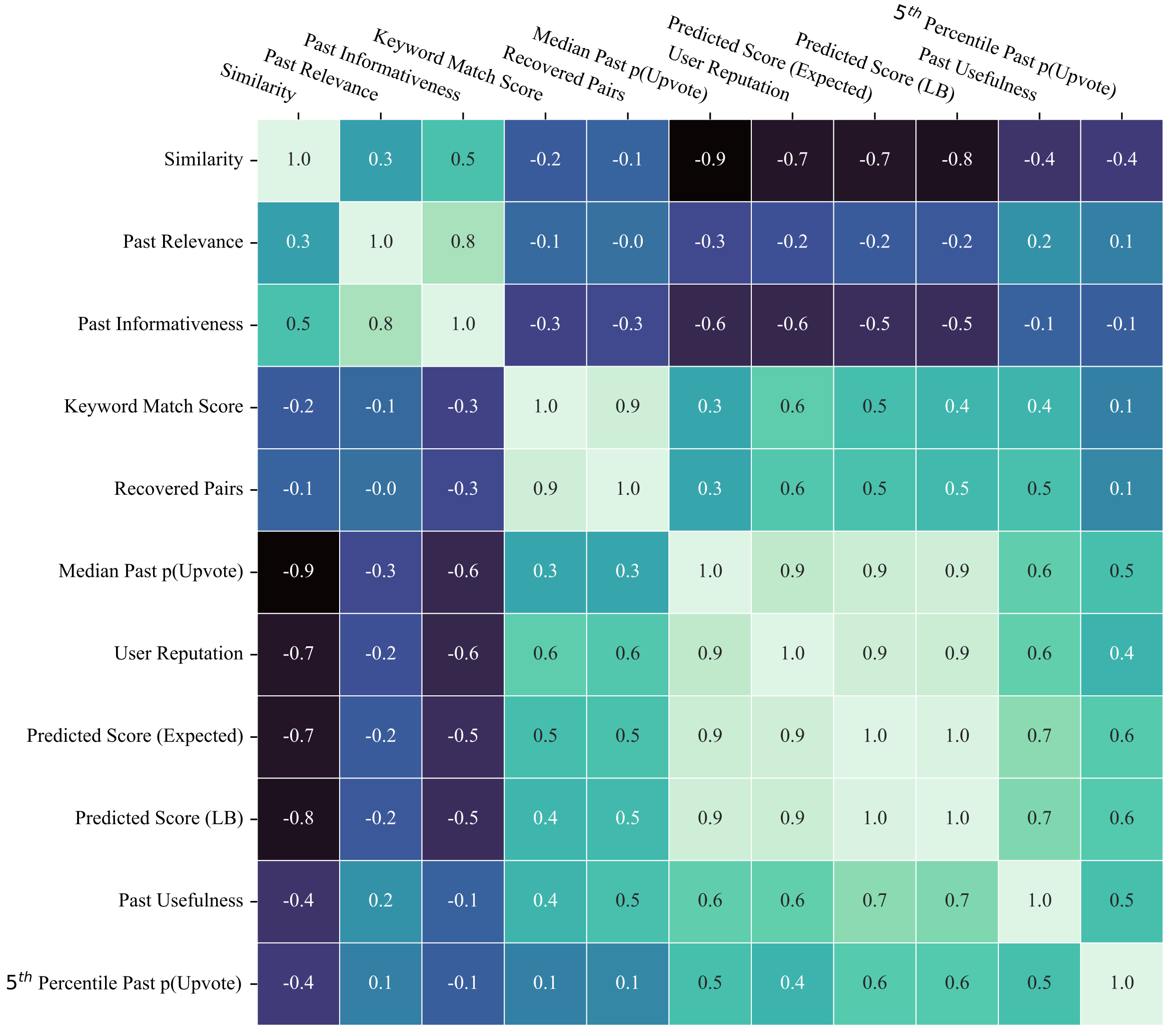


Figure A.10: Correlation of ranking of assignments by each metric for `chemistry`.

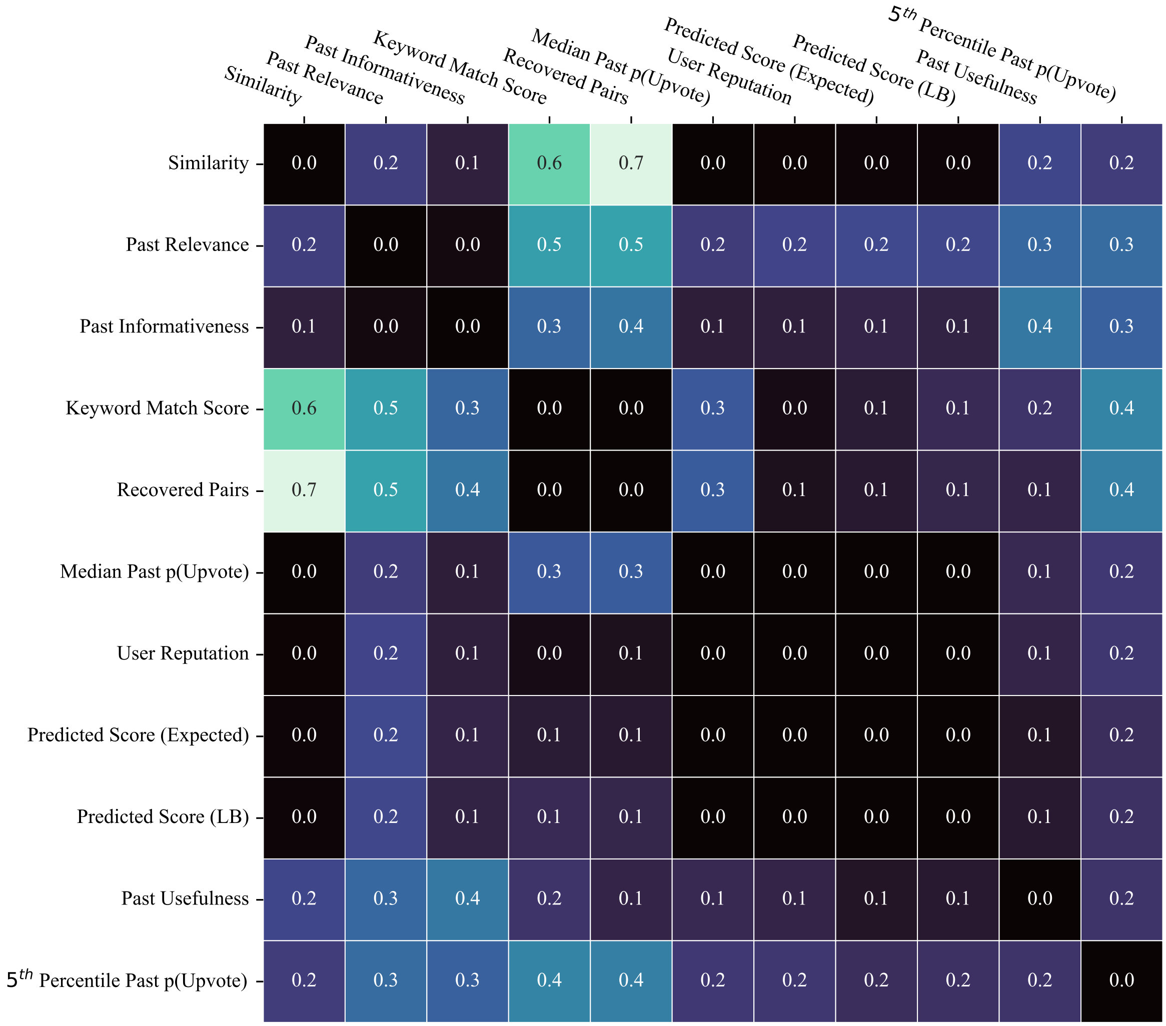


Figure A.11: P-values for correlation of ranking of assignments by each metric for `chemistry`.

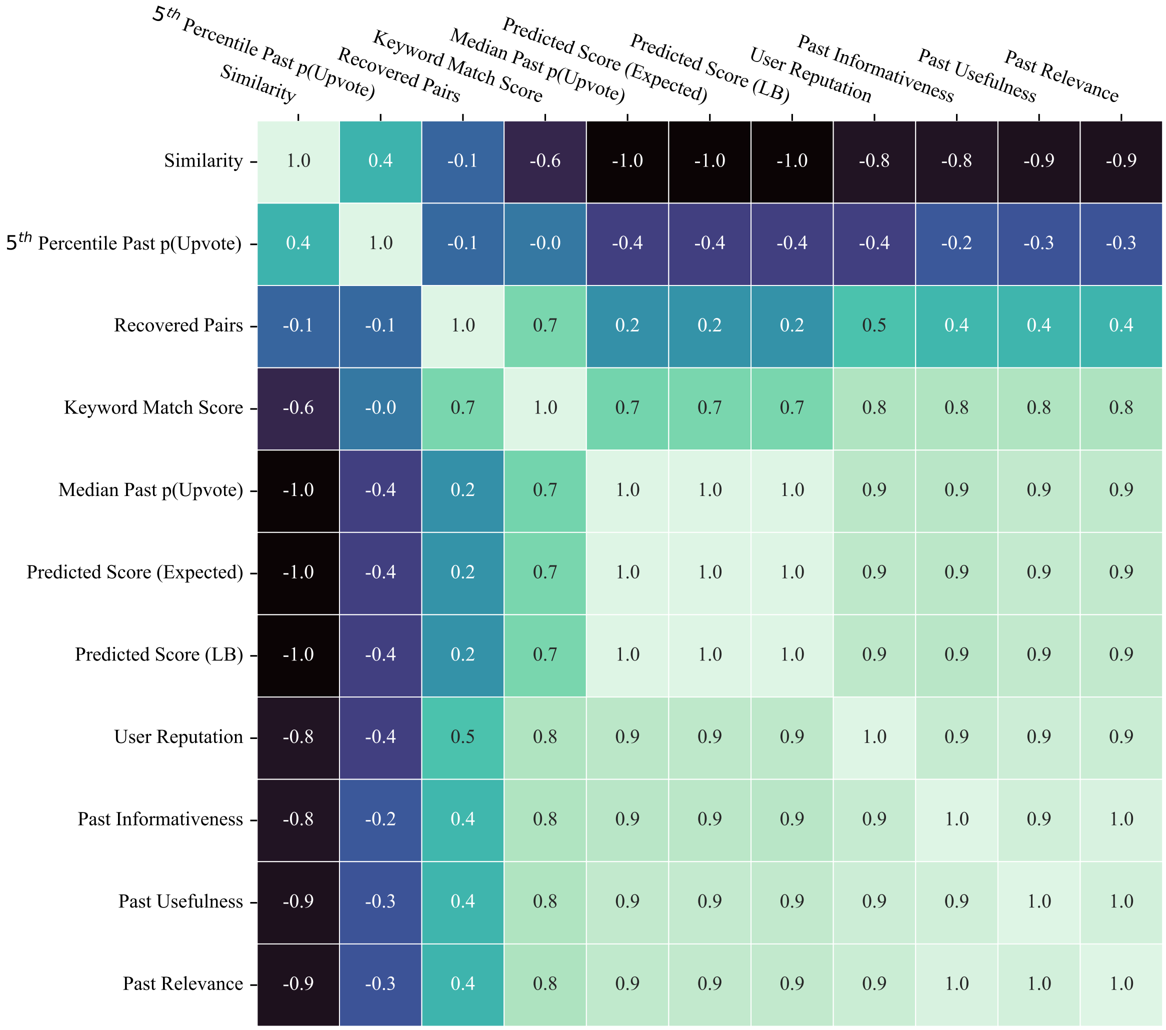


Figure A.12: Correlation of ranking of assignments by each metric for `academia`.

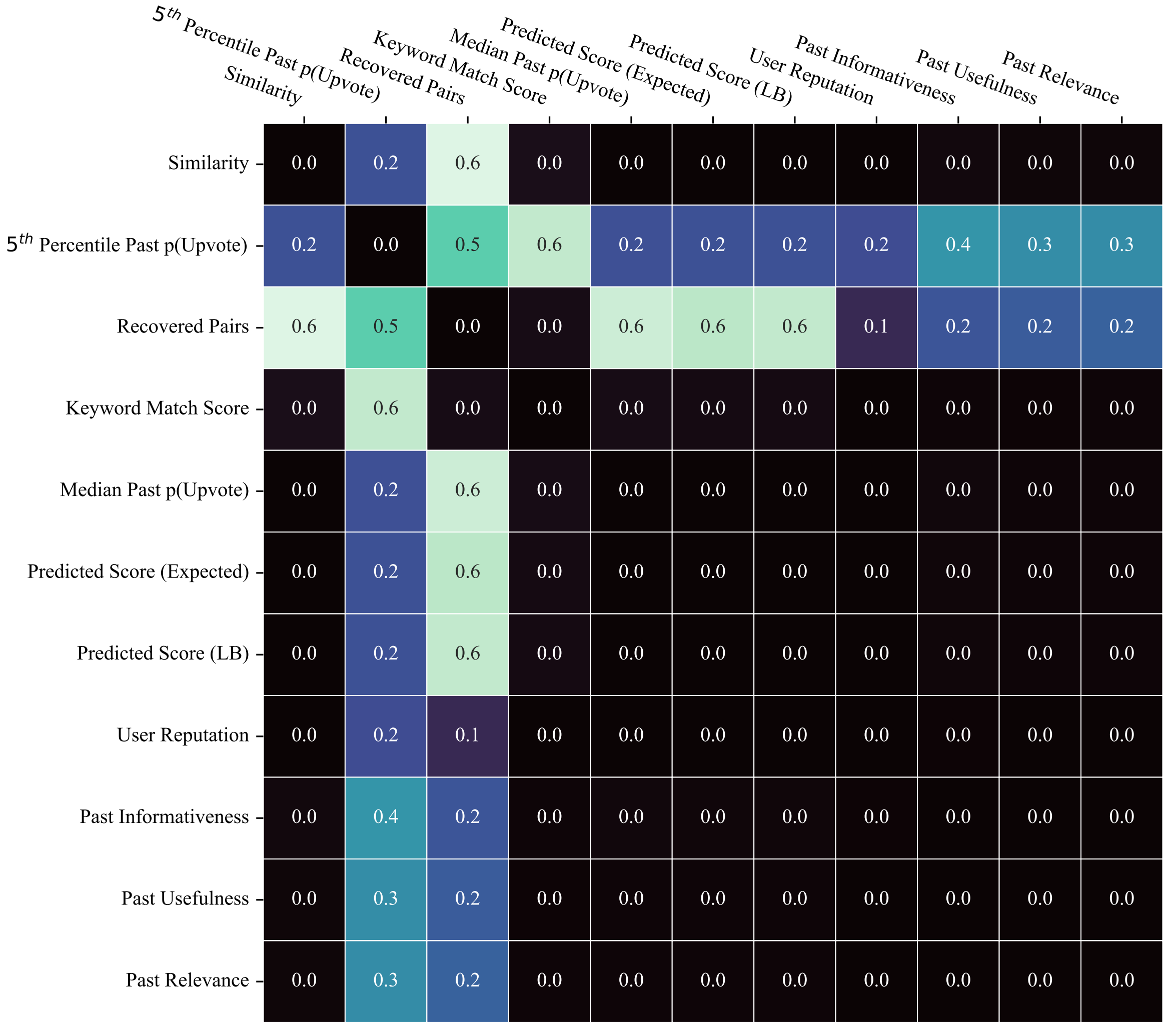


Figure A.13: P-values for correlation of ranking of assignments by each metric for `academia`.

BIBLIOGRAPHY

[1] Faez Ahmed, John P Dickerson, and Mark Fuge. Diverse weighted bipartite b-matching. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 35–41, 2017.

[2] Meltem Aksoy, Seda Yanik, and Mehmet Fatih Amasyali. Reviewer assignment problem: A systematic review of the literature. *Journal of Artificial Intelligence Research*, 76: 761–827, 2023.

[3] Amine Allouah, Christian Kroer, Xuan Zhang, Vashist Avadhanula, Anil Dania, Caner Gocmen, Sergey Pupyrev, Parikshit Shah, and Nicolas Stier. Robust and fair work allocation. *arXiv preprint arXiv:2202.05194*, 2022.

[4] Aris Anagnostopoulos, Luca Becchetti, Carlos Castillo, Aristides Gionis, and Stefano Leonardi. Power in unity: Forming teams in large-scale community systems. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 599–608, 2010.

[5] Igor Averbakh. Minmax regret solutions for minimax optimization problems with uncertainty. *Operations Research Letters*, 27(2):57–65, 2000.

[6] Haris Aziz, Toby Walsh, and Lirong Xia. Possible and necessary allocations via sequential mechanisms. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 468–474, 2015.

[7] Haris Aziz, Thomas Kalinowski, Toby Walsh, and Lirong Xia. Welfare of sequential allocation mechanisms for indivisible goods. In *Proceedings of the 22nd European Conference on Artificial Intelligence (ECAI)*, pages 787–794, 2016.

[8] Haris Aziz, Xin Huang, Nicholas Mattei, and Erel Segal-Halevi. The constrained round robin algorithm for fair and efficient allocation. *arXiv preprint arXiv:1908.00161*, 2019.

[9] Haris Aziz, Evi Micha, and Nisarg Shah. Group fairness in peer review. *Proceedings of the 37th Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 64885–64895, 2023.

[10] Moshe Babaioff, Tomer Ezra, and Uriel Feige. Fair and truthful mechanisms for dichotomous valuations. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, pages 5119–5126, 2021.

[11] Moshe Babaioff, Tomer Ezra, and Uriel Feige. Fair-share allocations for agents with arbitrary entitlements. *Mathematics of Operations Research*, 2023.

[12] Siddharth Barman and Paritosh Verma. Existence and computation of maximin fair allocations under matroid-rank valuations. In *Proceedings of the 20th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 169–177, 2021.

[13] Siddharth Barman, Ganesh Ghalme, Shweta Jain, Pooja Kulkarni, and Shivika Narang. Fair division of indivisible goods among strategic agents. In *Proceedings of the 18th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 1811–1813, 2019.

[14] Amir Beck and Luba Tetruashvili. On the convergence of block coordinate descent type methods. *SIAM Journal on Optimization*, 23(4):2037–2060, 2013.

[15] Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust optimization*. Princeton University Press, 2009.

[16] Nawal Benabbou, Mithun Chakraborty, Ayumi Igarashi, and Yair Zick. Finding fair and efficient allocations when valuations don't add up. In *Proceedings of the 13th International Symposium on Algorithmic Game Theory (SAGT)*, pages 32–46, 2020.

[17] George Bennett. Probability inequalities for the sum of independent random variables. *Journal of the American Statistical Association*, 57(297):33–45, 1962.

[18] Dimitris Bertsimas, David B Brown, and Constantine Caramanis. Theory and applications of robust optimization. *SIAM Review*, 53(3):464–501, 2011.

[19] Alina Beygelzimer, Yann Dauphin, Percy Liang, and Jennifer Wortman Vaughan. The NeurIPS 2021 consistency experiment, 2021. URL https://blog.neurips.cc/2021/12/08/the-neurips-2021-consistency-experiment/. Accessed August 20, 2024.

[20] John R Birge and Francois Louveaux. *Introduction to stochastic programming*. Springer Science & Business Media, 2011.

[21] Arpita Biswas and Siddharth Barman. Fair division under cardinality constraints. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 91–97, 2018.

[22] Craig Boutilier. Computational decision support: Regret-based models for optimization and preference elicitation. In Thomas Zentall and Philip Crowley, editors, *Comparative Decision Making*, chapter 14, pages 423–453. Oxford University Press, 2013.

[23] Craig Boutilier, Relu Patrascu, Pascal Poupart, and Dale Schuurmans. Constraint-based optimization and utility elicitation using the minimax decision criterion. *Artificial Intelligence*, 170(8-9):686–713, 2006.

[24] Sylvain Bouveret and Jérôme Lang. A general elicitation-free protocol for allocating indivisible goods. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 73–78, 2011.

[25] Stephen Boyd, Stephen P Boyd, and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, 2004.

[26] Niv Buchbinder, Moran Feldman, Joseph Naor, and Roy Schwartz. A tight linear time (1/2)-approximation for unconstrained submodular maximization. In *Proceedings of the 53rd Symposium on Foundations of Computer Science (FOCS)*, pages 649–658, 2012.

[27] Eric Budish. The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *Journal of Political Economy*, 119(6):1061–1103, 2011.

[28] Eric Budish, Yeon-Koo Che, Fuhito Kojima, and Paul Milgrom. Implementing random assignments: A generalization of the Birkhoff-von Neumann theorem. In *Cowles Summer Conference*, 2009.

[29] Jan Buermann, Enrico H Gerding, and Baharak Rastegari. Fair allocation of resources with uncertain availability. In *Proceedings of the 19th International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 204–212, 2020.

[30] Rainer E. Burkard, Mauro dell'Amico, and Silvano Martello. *Assignment Problems*. SIAM, 2012.

[31] T Tony Cai and Wen-Xin Zhou. Matrix completion via max-norm constrained optimization. *Electronic Journal of Statistics*, 10(1):1493–1525, 2016.

[32] Juan Miguel Campanario and Erika Acedo. Rejecting highly cited papers: The views of scientists who encounter resistance to their discoveries from other scientists. *Journal of the American Society for Information Science and Technology*, 58(5):734–743, 2007.

[33] Ioannis Caragiannis and Nidhi Rathi. Optimizing over serial dictatorships. In *Proceedings of the 16th International Symposium on Algorithmic Game Theory (SAGT)*, pages 329–346, 2023.

[34] Ioannis Caragiannis, David Kurokawa, Hervé Moulin, Ariel D Procaccia, Nisarg Shah, and Junxing Wang. The unreasonable fairness of maximum Nash welfare. *ACM Transactions on Economics and Computation (TEAC)*, 7(3):1–32, 2019.

[35] Mithun Chakraborty, Ayumi Igarashi, Warut Suksompong, and Yair Zick. Weighted envy-freeness in indivisible item allocation. *ACM Transactions on Economics and Computation (TEAC)*, 9(3):1–39, 2021.

[36] Mithun Chakraborty, Ulrike Schmidt-Kraepelin, and Warut Suksompong. Picking sequences and monotonicity in weighted fair division. *Artificial Intelligence*, 301:103578, 2021.

[37] Mithun Chakraborty, Erel Segal-Halevi, and Warut Suksompong. Weighted fairness notions for indivisible items revisited. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI)*, pages 4949–4956, 2022.

[38] Laurent Charlin and Richard Zemel. The Toronto paper matching system: An automated paper-reviewer assignment system. In *Proceedings of the 2013 ICML Workshop on Peer Reviewing and Publishing Models*, 2013.

[39] Laurent Charlin, Richard Zemel, and Craig Boutilier. A framework for optimizing paper matching. In *Proceedings of the 27th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 86–95, 2011.

[40] Laurent Charlin, Richard Zemel, and Craig Boutilier. Active learning for matching problems. In *Proceedings of the 29th International Conference on Machine Learning (ICML)*, pages 139–146, 2012.

[41] NeurIPS 2024 Organizing Committee. NeurIPS reviewer guidelines, 2024. URL https://neurips.cc/Conferences/2024/ReviewerGuidelines. Accessed July 30, 2024.

[42] Don Conry, Yehuda Koren, and Naren Ramakrishnan. Recommender systems for the conference paper assignment problem. In *Proceedings of the 3rd ACM Conference on Recommendation Systems (RecSys)*, pages 357–360, 2009.

[43] Cyrus Cousins. An axiomatic theory of provably-fair welfare-centric machine learning. In *Proceedings of the 35th Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 16610–16621, 2021.

[44] Cyrus Cousins, Justin Payan, and Yair Zick. Into the unknown: Assigning reviewers to papers with uncertain affinities. In *Proceedings of the 16th International Symposium on Algorithmic Game Theory (SAGT)*, pages 179–197, 2023.

[45] Cyrus Cousins, Vignesh Viswanathan, and Yair Zick. Dividing good and great items among agents with bivalued submodular valuations. In *Proceedings of the 19th Conference on Web and Internet Economics (WINE)*, pages 225–241, 2023.

[46] Cyrus Cousins, Vignesh Viswanathan, and Yair Zick. The good, the bad and the submodular: Fairly allocating mixed manna under order-neutral submodular preferences. In *Proceedings of the 19th Conference on Web and Internet Economics (WINE)*, pages 207–224, 2023.

[47] Cyrus Cousins, Elita Lobo, Justin Payan, and Yair Zick. Fair and welfare-efficient constrained multi-matchings under uncertainty. In *Proceedings of the 38th Annual Conference on Neural Information Processing Systems (NeurIPS)*, 2024.

[48] Cyrus Cousins, Neha Nayak Kennard, Sheshera Mysore, Justin Payan, and Yair Zick. Who you gonna call? Optimizing expert assignment with predictive models. *Submitted*, 2024.

[49] Abhimanyu Das and David Kempe. Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection. In *Proceedings of the 28th International Conference on Machine Learning (ICML)*, pages 1057–1064, 2011.

[50] Constantinos Daskalakis, Stratis Skoulakis, and Manolis Zampetakis. The complexity of constrained min-max optimization. In *Proceedings of the 53rd Annual ACM Symposium on Theory of Computing (STOC)*, page 1466–1478, 2021.

[51] Anand Deo and Karthyek Murthy. Efficient black-box importance sampling for VaR and CVaR estimation. In *Proceedings of the 2021 Winter Simulation Conference (WSC)*, pages 1–12, 2021.

[52] Siddartha Devic, David Kempe, Vatsal Sharan, and Aleksandra Korolova. Fairness in matching under uncertainty. *Proceedings of the 40th International Conference on Machine Learning (ICML)*, pages 7775–7794, 2023.

[53] Steven Diamond and Stephen Boyd. CVXPY: A python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

[54] Kate Donahue and Jon Kleinberg. Fairness and utilization in allocating resources with uncertain demand. In *Proceedings of the 3rd ACM Conference on Fairness, Accountability and Transparency (FAT*)*, pages 658–668, 2020.

[55] Evan Dong and Cyrus Cousins. Decentering imputation: Fair learning at the margins of demographics. In *Proceedings of the 2022 ICML Queer in AI Workshop*, 2022.

[56] Amitay Dror, Michal Feldman, and Erel Segal-Halevi. On fair division under heterogeneous matroid constraints. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, pages 5312–5320, 2021.

[57] John C Duchi and Hongseok Namkoong. Learning models with uniform performance via distributionally robust optimization. *The Annals of Statistics*, 49(3):1378–1406, 2021.

[58] Susan T Dumais and Jakob Nielsen. Automating the assignment of submitted manuscripts to reviewers. In *Proceedings of the 15th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 233–244, 1992.

[59] Ethan X Fang, Han Liu, Kim-Chuan Toh, and Wen-Xin Zhou. Max-norm optimization for robust matrix recovery. *Mathematical Programming*, 167(1):5–35, 2018.

[60] Tanner Fiez, Nihar Shah, and Lillian Ratliff. A SUPER* algorithm to optimize paper bidding in peer review. In *Proceedings of the 36th Annual Conference on Uncertainty in Artificial Intelligence (UAI)*, pages 580–589, 2020.

[61] M. L. Fisher, G. L. Nemhauser, and L. A. Wolsey. An analysis of approximations for maximizing submodular set functions—II. In M. L. Balinski and A. J. Hoffman, editors, *Polyhedral Combinatorics*, pages 73–87. Springer Berlin Heidelberg, 1978.

[62] Hengyi Fu and Sanghee Oh. Quality assessment of answers with user-identified criteria and data-driven features in social Q&A. *Information Processing & Management*, 56(1): 14–28, 2019.

[63] Dilrukshi Gamage, Thomas Staubitz, and Mark Whiting. Peer assessment in moocs: Systematic literature review. *Distance Education*, 42(2):268–289, 2021.

[64] Rajiv Gandhi, Samir Khuller, Srinivasan Parthasarathy, and Aravind Srinivasan. Dependent rounding and its applications to approximation algorithms. *Journal of the ACM (JACM)*, 53(3):324–360, 2006.

[65] Naveen Garg, Telikepalli Kavitha, Amit Kumar, Kurt Mehlhorn, and Julián Mestre. Assigning papers to referees. *Algorithmica*, 58(1):119–136, 2010.

[66] Corrado Gini. On the measure of concentration with special reference to income and statistics. *Colorado College Publication, General Series*, 208(1):73–79, 1936.

[67] Denizalp Goktas and Amy Greenwald. Convex-concave min-max Stackelberg games. *Proceedings of the 35th Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 2991–3003, 2021.

[68] Alexander Goldberg, Ivan Stelmakh, Kyunghyun Cho, Alice Oh, Alekh Agarwal, Danielle Belgrave, and Nihar B Shah. Peer reviews of peer reviews: A randomized controlled trial and other experiments. *arXiv preprint arXiv:2311.09497*, 2023.

[69] Paul Gölz and Ariel D Procaccia. Migration as submodular optimization. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI)*, pages 549–556, 2019.

[70] Bram L Gorissen, İhsan Yanıkoğlu, and Dick Den Hertog. A practical guide to robust optimization. *Omega*, 53:124–137, 2015.

[71] Negin Gorlezaei, Patrick Jaillet, and Zijie Zhou. Online resource allocation with samples. *arXiv preprint arXiv:2210.04774*, 2022.

[72] Ananya Gupta, Eric Johnson, Justin Payan, Aditya Roy, Ari Kobren, Swetasudha Panda, Michael Wick, and Jean-Baptiste Tristan. Online post-processing in rankings for fair utility maximization. In *Proceedings of the 14th International Conference on Web Search and Data Mining (WSDM)*, pages 454–462, 2021.

[73] Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2023. URL https://www.gurobi.com. Accessed August 21, 2024.

[74] David Hartvigsen, Jerry C Wei, and Richard Czuchlewski. The conference paper-reviewer assignment problem. *Decision Sciences*, 30(3):865–876, 1999.

[75] Chien-Ju Ho and Jennifer Wortman Vaughan. Online task assignment in crowdsourcing markets. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence (AAAI)*, pages 45–51, 2012.

[76] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

[77] L. Jeff Hong and Guangwu Liu. Monte Carlo estimation of value-at-risk, conditional value-at-risk and their sensitivities. In *Proceedings of the 2011 Winter Simulation Conference (WSC)*, pages 95–107, 2011.

[78] Hadi Hosseini, Justin Payan, Rik Sengupta, Rohit Vaish, and Vignesh Viswanathan. Graphical house allocation with identical valuations. *Autonomous Agents and Multi-Agent Systems*, 38(42), 2024.

[79] HuggingFace. Huggingface sentencetransformers multi-qa-mpnet-base-cos-v1, 2024. URL https://huggingface.co/sentence-transformers/multi-qa-mpnet-base-cos-v1. Accessed July 31, 2024.

[80] Steven Jecmen, Hanrui Zhang, Ryan Liu, Nihar Shah, Vincent Conitzer, and Fei Fang. Mitigating manipulation in peer review via randomized reviewer assignments. In *Proceedings of the 34th Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 12533–12545, 2020.

[81] Steven Jecmen, Nihar B Shah, Fei Fang, and Vincent Conitzer. Tradeoffs in preventing manipulation in paper bidding for reviewer assignment. In *Proceedings of the 2022 ICLR Workshop on ML Evaluation Standards*, 2022.

[82] Steven Jecmen, Minji Yoon, Vincent Conitzer, Nihar B Shah, and Fei Fang. A dataset on malicious paper bidding in peer review. In *Proceedings of the 2023 International World Wide Web Conference (WWW)*, pages 3816–3826, 2023.

[83] Yiqiao Jin, Qinlin Zhao, Yiyang Wang, Hao Chen, Kaijie Zhu, Yijia Xiao, and Jindong Wang. Agentreview: Exploring peer review dynamics with LLM agents. *arXiv preprint arXiv:2406.12708*, 2024.

[84] Thomas Kalinowski, Nina Narodytska, and Toby Walsh. A social welfare optimal sequential allocation procedure. In *Proceedings of the 23rd International Joint Conference on Artificial Intelligence (IJCAI)*, pages 227–233, 2013.

[85] Maryam Karimzadehgan and ChengXiang Zhai. Constrained multi-aspect expertise matching for committee review assignment. In *Proceedings of the 18th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 1697–1700, 2009.

[86] Maryam Karimzadehgan, ChengXiang Zhai, and Geneva Belford. Multi-aspect expertise matching for review assignment. In *Proceedings of the 17th ACM International Conference on Information and Knowledge Management (CIKM)*, pages 1113–1122, 2008.

[87] Samir Khan, Martin Saveski, and Johan Ugander. Off-policy evaluation beyond overlap: Sharp partial identification under smoothness. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, pages 23734–23757, 2024.

[88] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[89] Aniket Kittur and Robert E Kraut. Harnessing the wisdom of crowds in Wikipedia: quality through coordination. In *Proceedings of the 2008 ACM Conference on Computer Supported Cooperative Work (CSCW)*, pages 37–46, 2008.

[90] Teun Kloek and Herman K Van Dijk. Bayesian estimates of equation system parameters: An application of integration by Monte Carlo. *Econometrica*, 46(1):1–19, 1978.

[91] Ari Kobren, Barna Saha, and Andrew McCallum. Paper matching with local fairness constraints. In *Proceedings of the 25th International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 1247–1257, 2019.

[92] Ngai Meng Kou, Leong Hou U, Nikos Mamoulis, and Zhiguo Gong. Weighted coverage based reviewer assignment. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data (SIGMOD)*, pages 2031–2046, 2015.

[93] Panos Kouvelis and Gang Yu. *Robust discrete optimization and its applications*. Springer Science & Business Media, 2013.

[94] NK Kwak and Changwon Lee. A linear goal programming model for human resource allocation in a health-care organization. *Journal of Medical Systems*, 21:129–140, 1997.

[95] Prashanth L.A., Krishna Jagannathan, and Ravi Kolla. Concentration bounds for CVaR estimation: The cases of light-tailed and heavy-tailed distributions. In *Proceedings of the 37th International Conference on Machine Learning (ICML)*, pages 5577–5586, 2020.

[96] Neil D. Lawrence. The NIPS experiment, 2014. URL https://inverseprobability.com/2014/12/16/the-nips-experiment. Accessed August 20, 2024.

[97] Neil D Lawrence and Raquel Urtasun. Non-linear matrix factorization with Gaussian processes. In *Proceedings of the 26th International Conference on Machine Learning (ICML)*, pages 601–608, 2009.

[98] Jason Lee, Ben Recht, Nathan Srebro, Joel Tropp, and Ruslan Salakhutdinov. Practical large-scale optimization for max-norm regularization. *Proceedings of the 23rd Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 1297–1305, 2010.

[99] Daniel Levy, Yair Carmon, John C Duchi, and Aaron Sidford. Large-scale methods for distributionally robust optimization. In *Proceedings of the 34th Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 8847–8860, 2020.

[100] Kevin Leyton-Brown, Mausam, Yatin Nandwani, Hedayat Zarkoob, Chris Cameron, Neil Newman, and Dinesh Raghu. Matching papers and reviewers at large conferences. *Artificial Intelligence*, 331:104119, 2024.

[101] Baochun Li and Y Thomas Hou. The new automated IEEE INFOCOM review assignment system. *IEEE Network*, 30(5):18–24, 2016.

[102] Jing Wu Lian, Nicholas Mattei, Renee Noble, and Toby Walsh. The conference paper assignment problem: Using order weighted averages to assign indivisible goods. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI)*, pages 1138–1145, 2018.

[103] Richard J Lipton, Evangelos Markakis, Elchanan Mossel, and Amin Saberi. On approximately fair allocations of indivisible goods. In *Proceedings of the 5th ACM Conference on Electronic Commerce (EC)*, pages 125–131, 2004.

[104] Mingrong Liu, Yicen Liu, and Qing Yang. Predicting best answerers for new questions in community question answering. In *Proceedings of the 11th International Conference on Web-Age Information Management (WAIM)*, pages 127–138, 2010.

[105] Elita A Lobo, Mohammad Ghavamzadeh, and Marek Petrik. Soft-robust algorithms for batch reinforcement learning. *arXiv preprint arXiv:2011.14495*, 2020.

[106] Cheng Long, Raymond Chi-Wing Wong, Yu Peng, and Liangliang Ye. On good and fair paper-reviewer assignment. In *Proceedings of the 13th IEEE International Conference on Data Mining (ICDM)*, pages 1145–1150, 2013.

[107] Nicholas Mattei and Toby Walsh. Preflib: A library of preference data. In *Proceedings of the 3rd International Conference on Algorithmic Decision Theory (ADT)*, pages 259–270, 2013.

[108] Nicholas Mattei and Toby Walsh. A preflib.org retrospective: Lessons learned and new directions. *Trends in Computational Social Choice. AI Access Foundation*, pages 289–309, 2017.

[109] Nicholas Mattei, Simon Rey, and Toby Walsh. AAMAS bidding data, 2024. URL https://preflib.simonrey.fr/dataset/00037. Accessed July 31, 2024.

[110] Alessio Mazzetto, Cyrus Cousins, Dylan Sam, Stephen H Bach, and Eli Upfal. Adversarial multi class learning under weak supervision with performance guarantees. In *Proceedings of the 38th International Conference on Machine Learning (ICML)*, pages 7534–7543, 2021.

[111] Colin McDiarmid. On the method of bounded differences. *Surveys in Combinatorics*, 141(1):148–188, 1989.

[112] Reshef Meir, Jérôme Lang, Julien Lesca, Nicholas Mattei, and Natan Kaminsky. A market-inspired bidding scheme for peer review paper assignment. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, pages 4776–4784, 2021.

[113] David Mimno and Andrew McCallum. Expertise modeling for matching papers with reviewers. In *Proceedings of the 13th International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 500–509, 2007.

[114] Luisa Montanari, Ulrike Schmidt-Kraepelin, Warut Suksompong, and Nicholas Teh. Weighted envy-freeness for submodular valuations. In *Proceedings of the 38th AAAI Conference on Artificial Intelligence (AAAI)*, pages 9865–9873, 2024.

[115] Sheshera Mysore, Mahmood Jasim, Andrew McCallum, and Hamed Zamani. Editable user profiles for controllable text recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 993–1003, 2023.

[116] Hongseok Namkoong and John C Duchi. Stochastic gradient methods for distributionally robust optimization with f-divergences. *Proceedings of the 30th Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 2208–2216, 2016.

[117] Graham Neubig, John Wieting, Arya McCarthy, Amanda Stent, Natalie Schluter, and Trevor Cohn. ACL reviewer matching code, 2020. URL https://github.com/acl-org/reviewer-paper-matching. Accessed August 22, 2024.

[118] Regina O'Dell, Mirjam Wattenhofer, and Roger Wattenhofer. The paper assignment problem. Technical report, ETH Zurich, Department of Computer Science, 2005.

[119] OpenReview. Openreview matching code, 2024. URL https://github.com/openreview/openreview-matcher. Accessed July 31, 2024.

[120] OpenReview. Paper-reviewer affinity modeling for OpenReview, 2024. URL https://github.com/openreview/openreview-expertise. Accessed September 12, 2024.

[121] OpenReview. Openreview: How to run a paper matching, 2024. URL https://docs.openreview.net/how-to-guides/paper-matching-and-assignment/how-to-do-automatic-assignments/how-to-run-a-paper-matching. Accessed July 31, 2024.

[122] James Oxley. *Matroid Theory.* Oxford Graduate Texts in Mathematics. Oxford Univerity Press, 2nd edition, 2011.

[123] A.R. Pankov, E.N. Platonov, and Konstantin Siemenikhin. Minimax quadratic optimization and its application to investment planning. *Automation and Remote Control*, 62:1978–1995, 2001.

[124] Justin Payan and Yair Zick. I will have order! Optimizing orders for fair reviewer assignment. In *Proceedings of the 31st International Joint Conference on Artificial Intelligence (IJCAI)*, pages 440–446, 2022.

[125] Justin Payan and Yair Zick. Reviewer assignment code, 2024. URL https://github.com/justinpayan/ReviewerAssignmentCode. Accessed July 31, 2024.

[126] Justin Payan, Yuval Merhav, He Xie, Satyapriya Krishna, Anil Ramakrishna, Mukund Sridhar, and Rahul Gupta. Towards realistic single-task continuous learning research for NER. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 3773–3783, 2021.

[127] Justin Payan, Swaroop Mishra, Singh Mukul, Carina Negreanu, Christian Poelitz, Chitta Baral, Subhro Roy, Rasika Chakravarthy, Benjamin Van Durme, and Elnaz Nouri. InstructExcel: A benchmark for natural language instruction in Excel. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4026–4043, 2023.

[128] Justin Payan, Rik Sengupta, and Vignesh Viswanathan. Relaxations of envy-freeness over graphs. In *Proceedings of the 22nd International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 2652–2654, 2023.

[129] Dominik Peters, Ariel D Procaccia, and David Zhu. Robust rent division. In *Proceedings of the 36th Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 13864–13876, 2022.

[130] Malte L Peters and Stephan Zelewski. Assignment of employees to workplaces under consideration of employee competences and preferences. *Management Research News*, 30(2):84–99, 2007.

[131] András Prékopa. *Stochastic programming*, volume 324. Springer Science & Business Media, 2013.

[132] Eric Price. The NIPS experiment, 2014. URL https://blog.mrtz.org/2014/12/15/the-nips-experiment.html. Accessed August 20, 2024.

[133] David Pujol, Ryan McKenna, Satya Kuppam, Michael Hay, Ashwin Machanavajjhala, and Gerome Miklau. Fair decision making using privacy-protected data. In *Proceedings of the 2020 ACM Conference on Fairness, Accountability and Transparency (FAT*)*, pages 189–199, 2020.

[134] Yujie Qian, Jie Tang, and Kan Wu. Weakly learning to match experts in online community. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3841–3847, 2018.

[135] Nils Reimers and Iryna Gurevych. Sentence-BERT: Sentence embeddings using siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3982–3992, 2019.

[136] Yuqing Ren, Haifeng Zhang, and Robert E Kraut. How did they build the free encyclopedia? A literature review of collaboration and coordination among Wikipedia editors. *ACM Transactions on Computer-Human Interaction*, 31(1):1–48, 2023.

[137] Microsoft Research. Microsoft conference management toolkit, 2024. URL https://cmt3.research.microsoft.com/. Accessed July 31, 2024.

[138] Microsoft Research. Microsoft conference management toolkit: CHAIR HOW-TO: Automatic assignment of reviewer, 2024. URL https://cmt3.research.microsoft.com/docs/help/chair/auto-assign-reviewers.html. Accessed July 31, 2024.

[139] Philippe Rigaux. An iterative rating method: Application to web-based conference management. In *Proceedings of the 19th ACM Symposium on Applied Computing (SAC)*, pages 1682–1687, 2004.

[140] R Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *Journal of Risk*, 2:21–42, 2000.

[141] Inbal Rozenzweig, Reshef Meir, Nicholas Mattei, and Ofra Amir. Mitigating skewed bidding for conference paper assignment. In *Proceedings of the 22nd International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 573–581, 2023.

[142] Andrzej Ruszczyński and Alexander Shapiro. Stochastic programming models. *Handbooks in Operations Research and Management Science*, 10:1–64, 2003.

[143] Martin Saveski, Steven Jecmen, Nihar Shah, and Johan Ugander. Counterfactual evaluation of peer-review assignment policies. *Proceedings of the 37th Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 58765–58786, 2023.

[144] Sara Schroter, Leanne Tite, Andrew Hutchings, and Nick Black. Differences in review quality and recommendations for publication between peer reviewers suggested by authors or by editors. *JAMA*, 295(3):314–317, 2006.

[145] Scikit-learn. Scikit-learn agglomerative clustering, 2024. URL https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html#sklearn.cluster.AgglomerativeClustering. Accessed July 31, 2024.

[146] Scikit-learn. Scikit-learn logistic regression, 2024. URL https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html. Accessed July 31, 2024.

[147] Chirag Shah and Jefferey Pomerantz. Evaluating and predicting answer quality in community QA. In *Proceedings of the 33rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 411–418, 2010.

[148] Nihar B Shah. Challenges, experiments, and computational solutions in peer review. *Communications of the ACM*, 65(6):76–87, 2022.

[149] Nihar B Shah, Joseph K Bradley, Abhay Parekh, Martin Wainwright, and Kannan Ramchandran. A case for ordinal peer-evaluation in MOOCs. In *Proceedings of the 2013 NIPS Workshop on Data Driven Education*, 2013.

[150] Lloyd S Shapley. A value for n-person games. In *Contributions to the Theory of Games*, volume 2, pages 307–317. Princeton University Press, 1953.

[151] NZ Shor, Krzysztof C Kiwiel, and Andrzej Ruszcayǹski. *Minimization methods for non-differentiable functions.* Springer-Verlag, 1985.

[152] Nyssa J. Silbiger and Amber D. Stubler. Unprofessional peer reviews disproportionately harm underrepresented groups in STEM. *PeerJ*, 7:e8247, 2019.

[153] Rita J Simon, Von Bakanic, and Clark McPhail. Who complains to journal editors and what happens. *Sociological Inquiry*, 56(2):259–271, 1986.

[154] StackExchange. StackExchange data dump, 2023. URL https://archive.org/details/stackexchange. Accessed December 18, 2023.

[155] StackExchange. StackExchange tour, 2024. URL https://stackexchange.com/tour. Accessed July 31, 2024.

[156] Ivan Stelmakh. Towards fair, equitable, and efficient peer review. In *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI)*, pages 15736–15737, 2021.

[157] Ivan Stelmakh, Nihar B Shah, and Aarti Singh. PeerReview4All: Fair and accurate reviewer assignment in peer review. In *Proceedings of the 30th International Conference on Algorithmic Learning Theory (ALT)*, pages 828–856, 2019.

[158] Ivan Stelmakh, Charvi Rastogi, Nihar B Shah, Aarti Singh, and Hal Daumé III. A large scale randomized controlled trial on herding in peer-review discussions. *PLoS ONE*, 18(7):e0287443, 2023.

[159] Ivan Stelmakh, John Wieting, Graham Neubig, and Nihar B Shah. A gold standard dataset for the reviewer assignment problem. *arXiv preprint arXiv:2303.16750*, 2023.

[160] Ivan Stelmakh, Nihar Shah, and Aarti Singh. PeerReview4All code, 2024. URL https://github.com/niharshah/peerreview4all. Accessed July 31, 2024.

[161] Warut Suksompong and Nicholas Teh. On maximum weighted nash welfare for binary valuations. *Mathematical Social Sciences*, 117:101–108, 2022.

[162] Warut Suksompong and Nicholas Teh. Weighted fair division with matroid-rank valuations: Monotonicity and strategyproofness. *Mathematical Social Sciences*, 126: 48–59, 2023.

[163] Jiankai Sun, Abhinav Vishnu, Aniket Chakrabarti, Charles Siegel, and Srinivasan Parthasarathy. Coldroute: Effective routing of cold questions in stack exchange sites. *Data Mining and Knowledge Discovery*, 32:1339–1367, 2018.

[164] Jiankai Sun, Jie Zhao, Huan Sun, and Srinivasan Parthasarathy. Endcold: An end-to-end framework for cold question routing in community question answering services. In *Proceedings of the 29th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3244–3250, 2021.

[165] Mao Tan, Zhuocen Dai, Yuling Ren, Toby Walsh, and Martin Aleksandrov. Minimal-envy conference paper assignment: Formulation and a fast iterative algorithm. In *Proceedings of the 5th Asian Conference on Artificial Intelligence Technology (ACAIT)*, pages 667–674, 2021.

[166] Wenbin Tang, Jie Tang, Tao Lei, Chenhao Tan, Bo Gao, and Tian Li. On optimization of expertise matching with various constraints. *Neurocomputing*, 76(1):71–83, 2012.

[167] Camillo J Taylor. On the optimal assignment of conference papers to reviewers. Technical report, University of Pennsylvania, 2008.

[168] Andrew Tomkins, Min Zhang, and William D Heavlin. Reviewer bias in single-versus double-blind peer review. *Proceedings of the National Academy of Sciences*, 114(48): 12708–12713, 2017.

[169] Rohan Tondulkar, Manisha Dubey, and Maunendra Sankar Desarkar. Get me the best: Predicting best answerers in community question answering sites. In *Proceedings of the 12th ACM Conference on Recommendation Systems (RecSys)*, pages 251–259, 2018.

[170] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.

[171] Susan Van Rooyen, Nick Black, and Fiona Godlee. Development of the review quality instrument (RQI) for assessing peer reviews of manuscripts. *Journal of Clinical Epidemiology*, 52(7):625–629, 1999.

[172] Fernanda B Viégas, Martin Wattenberg, Jesse Kriss, and Frank Van Ham. Talk before you type: Coordination in Wikipedia. In *Proceedings of the 40th Hawaii International Conference on System Sciences (HICSS)*, page 78, 2007.

[173] Vignesh Viswanathan and Yair Zick. A general framework for fair allocation under matroid rank valuations. In *Proceedings of the 24th ACM Conference on Economics and Computation (EC)*, pages 1129–1152, 2023.

[174] Vignesh Viswanathan and Yair Zick. Yankee swap: A fast and simple fair allocation mechanism for matroid rank valuations. In *Proceedings of the 22nd International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, pages 179–187, 2023.

[175] Xianzhi Wang, Chaoran Huang, Lina Yao, Boualem Benatallah, and Manqing Dong. A survey on expert recommendation in community question answering. *Journal of Computer Science and Technology*, 33:625 – 653, 2018.

[176] Yuanhao Wang and Jian Li. Improved algorithms for convex-concave minimax optimization. In *Proceedings of the 34th Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 4800–4810, 2020.

[177] Stephen J Wright. Coordinate descent algorithms. *Mathematical programming*, 151(1): 3–34, 2015.

[178] Xin Xia, David Lo, Ying Ding, Jafar M Al-Kofahi, Tien N Nguyen, and Xinyu Wang. Improving automated bug triaging with specialized topic model. *IEEE Transactions on Software Engineering*, 43(3):272–297, 2016.

[179] Yangyang Xu and Wotao Yin. A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion. *SIAM Journal on Imaging Sciences*, 6(3):1758–1789, 2013.

[180] Liu Yang, Minghui Qiu, Swapna Gottipati, Feida Zhu, Jing Jiang, Huiping Sun, and Zhong Chen. CQARank: Jointly model topics and expertise in community question answering. In *Proceedings of the 22nd ACM International Conference on Information and Knowledge Management (CIKM)*, pages 99–108, 2013.

[181] Sha Yuan, Yu Zhang, Jie Tang, Wendy Hall, and Juan Bautista Cabota. Expert finding in community question answering: a review. *Artificial Intelligence Review*, 53:843–874, 2018.

[182] Tony Sit Yue Xing and Hoi Ying Wong. Variance reduction for risk measures with importance sampling in nested simulation. *Quantitative Finance*, 22(4):657–673, 2022.

[183] Wenbin Zhang and Jeremy C Weiss. Fair decision-making under uncertainty. In *Proceedings of the 21st IEEE International Conference on Data Mining (ICDM)*, pages 886–895, 2021.

[184] Xiquan Zhao and Yangsen Zhang. Reviewer assignment algorithms for peer review automation: A survey. *Information Processing and Management*, 59(5):103028, 2022.

[185] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, and Eric Xing. Judging LLM-as-a-judge with MT-bench and Chatbot Arena. *Proceedings of the 37th Annual Conference on Neural Information Processing Systems (NeurIPS)*, pages 46595–46623, 2023.

[186] Zhemin Zhu, Delphine Bernhard, and Iryna Gurevych. A multi-dimensional model for assessing the quality of answers in social Q&A. In *Proceedings of the 14th International Conference on Information Quality (ICIQ)*, pages 264–265, 2009.