Robust Rank Aggregation and its Applications

Yuangang Pan

Faculty of Engineering and Information Technology
University of Technology Sydney

This dissertation is submitted for the degree of
Doctor of Philosophy

September 2019

Certificate of Original Authorship

I hereby declare that this thesis, is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the Faculty of Engineering and Information Technology at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise reference or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis. The contents of this dissertation are original and have not been submitted in whole or in part for qualifications at any other academic institution. This research is supported by the Australian Government Research Training Program.

Production Note:

Signature: Signature removed prior to publication.

Date: 24 Dec 2019

I would like to dedicate this thesis to my family who, through thick and thin, has been there for me. Their support and drive are what have made me who I am.

Acknowledgements

First and foremost, my utmost gratitude goes to my supervisor Professor Ivor W. Tsang, for his cordiality and generousness in offering me this research opportunity in his group. Life has not been easy for me, especially at the beginning of this adventure. He has always been there for me, serving as a friend and a mentor. His illuminating instructions and insightful guidance helped me to avoid being unnecessarily stuck and kept me focused on the nature of science itself. His consistent support and encouragement helped me to survive the many struggles I experienced on my Ph.D. journey. His depth and breadth of knowledge, insightful vision, and wisdom have not only guided me through my Ph.D. research but also shaped what I have achieved in this dissertation. His dedication and professionalism also set a great example from which I can learn and benefit from in my future career.

During my time in Sydney, I have had the privilege of meeting many excellent people. Being more senior to me, I have appreciated how they have generously shared many insightful thoughts on life and research. Having them as companions in my daily life makes me feel life brighter. Their enthusiasm in research, diligent attitude, depth and breadth of knowledge guided me through my Ph.D. research and shaped what I have achieved. My warmest gratitude, therefore, goes to Dr. Qiong Wang, Dr. Yali Du, Dr. Jiangchao Yao, Yiming Jiang, Licheng Feng, Prof. Weijie Chen, Prof. Weiwei Liu, Dr. Muming Zhao, Dr. Donna Xu, Yueming Lyu, Xiaoqi Jiang and Xu Chen for many illuminating conversations on life and research.

Thankyou also to the University of Technology Sydney for offering me this rare opportunity. My appreciation also goes to those fine minds in our "Eating Group". Your participation has made this journey for me so memorable. I would like to thank especially Dr. Yan Zhang, Yaxin Shi, Yinghua Yao, Jing Li, Xiaowei Zhou, Xingrui Yu. Thank you all!

Last but not least, I owe the deepest debt of gratitude to my family for their unbounded love and unselfish support. Those miseries my parents suffered for me will be buried in my heart forever.

Abstract

Rank aggregation (RA) refers to the task of recovering the total order over a set of items, given a collection of preferences over the items. The flexible collection of preferences enables successful application of RA in various fields, e.g., image rating and bioinformatics. A basic assumption underlying the vanilla RA is that all preferences are provided by homogeneous users. However, this assumption is rarely satisfied in real applications, due to the complex real situation. Therefore, RA usually suffers from model misspecification, namely the inconsistency between the collected preferences and the homogeneity assumption. Another challenge associated with RA is the scalability issue. In particular, RA usually involves ranking over tens of thousands of items, leading to an exponential volume of preferences for aggregation. Therefore, an inappropriate inference method would limit the application of the proposed model.

This thesis considered RA under model misspecification in the following three scenarios:

- In a crowdsourcing scenario, sufficient annotations from each user are available, which enables exploration of user heterogeneity to account for model misspecification. Therefore, I proposed a reliable CrowdsOUrced Plackett-LucE (COUPLE) model, which introduces an uncertainty vector to make a fine-grained categorization of users. Meanwhile, a general Bayesian Moment Matching (OnlineGBMM) was proposed, to ensure an analytic Bayesian update with an almost twice differentiable likelihood function.

- In a general setting, typical model augmentation methods would cause overfitting, because insufficient annotations from each user are available. Inspired by the distributional robust literature, I proposed CoarsenRank, which performs regular RA over a neighborhood of preferences. The resultant inference would enjoy robustness against model misspecification. To this end, I first defined a neighborhood of the rank dataset using relative entropy. Then, I instantiated CoarsenRank with three popular probability ranking models and discussed the optimization strategies.

- RA for mental fatigue monitoring. Common practices for mental fatigue monitoring refer to predicting the reaction time (RT) by aggregating the EEG signal from multiple heterogeneous EEG channels. Let us consider the RT as the item score and view each EEG channel as a user. The mental fatigue monitoring task could be formulated as RA under model misspecification, particularly in a crowdsourcing scenario. To address this problem, a Self-Weight Ordinal REgression (SWORE) model with Brain Dynamics table (BDtable) is proposed. The SWORE model could give a reliable evaluation of brain dynamics preferences from multiple channels, while the BDtable is employed to calibrate the SWORE model by utilizing the proposed online generalized Bayesian moment matching (OGMM) algorithm.

Table of contents

List of figures

List of tables

Chapter 1

Introduction

Originally formulated in social choice theory [Lijphart, 1994, Saari, 1999], rank aggregation (RA) has been the subject of a renewed interest in the machine learning community. The array of fields in which RA has been applied include ordinal peer grading [Raman and Joachims, 2014], online image-rating [Knight and Keith, 2005], meta-search [Desarkar et al., 2016] and online product recommendation [Liu, 2009]. This chapter briefly introduces the background of RA, the challenges facing the application of it, and the contributions of my thesis.

1.1 Background

RA refers to the task of recovering the total order over a set of items, given a collection of pairwise, partial or full preferences over the items [Lin, 2010]. Typically, there are two kinds of judgments widely adopted to express human preferences, ratings and rankings [Niu et al., 2015]. Compared to rating items, the preference is a more natural expression of user opinions that can provide more consistent results [Raman and Joachims, 2014]. Therefore, RA is a practical and useful approach to summarize user preferences [de Borda, 1781].

1.1.1 Ordinal Data Collection

A straightforward method for ordinal evaluation is to resort to a small number of domain experts to yield a consensus full preference according to a certain criterion. However, such a naive approach can be expensive, or even infeasible, for large scale data [Shah et al., 2013, Luaces et al., 2015]. For example, massive open online courses (MOOCs) are being touted as a revolution in education delivery, but traditional teacher graders are no longer available for an online class with massive student numbers. Peer grading is widely adopted to address this problem, which first divides students' assignments into small groups and then asks each

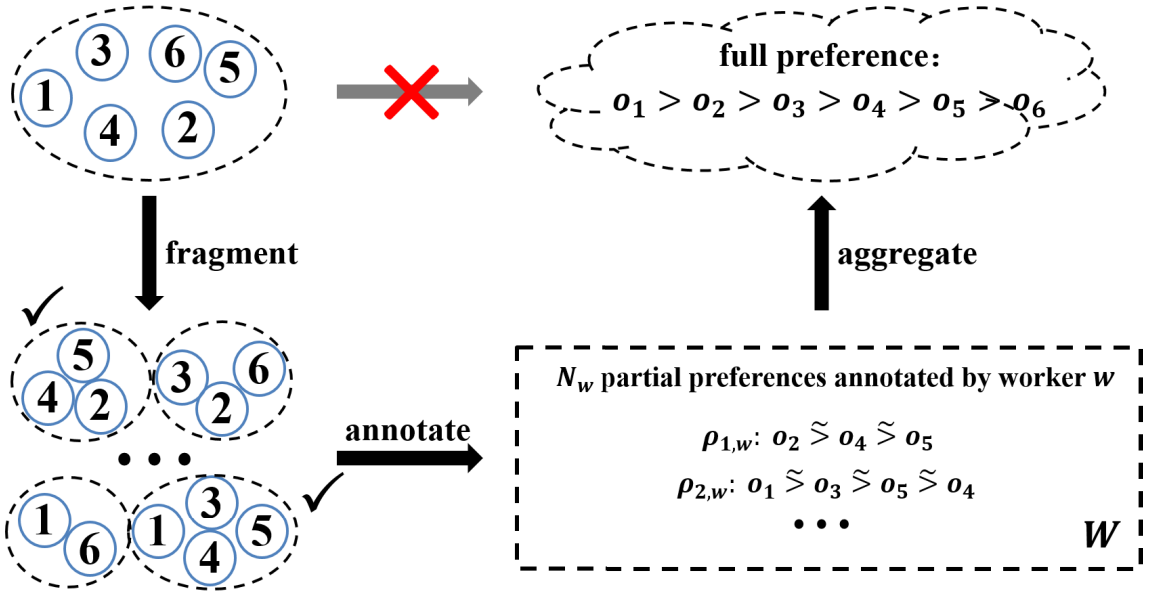


Figure 1.1 Crowdsourced Partial Preferences Collection. The tasks (subsets) with "✓" are assigned to the worker w. The notation W in the corner denotes that W crowd workers complete the annotation independently.

student grader to order the assignments within each group independently.The collected partial preferences are aggregated into one full preference using some aggregation rules at the end. For the same reason, a more efficient solution for ordinal evaluation is to post unlabeled data to a crowdsourcing marketplace (for examples, MTurk[1], Innocentive[2], CrowdFlower[3], and Allourideas[4]), where a big crowd of low-paid workers can be hired instantaneously to perform labeling tasks [Sheng et al., 2008, Snow et al., 2008, Hsueh et al., 2009, Nowak and Rüger, 2010].

In particular, I summarize a general data collection setting: Crowdsourced Partial Preferences Collection, illustrated in Figure 1.1. *Fragment*: a large set of items is randomly broken into several subsets; *Annotate*: crowd workers annotate multiple (overlapped) subsets independently to yield partial preferences; *Aggregation*: aggregation rules are used to aggregate the noisy partial preferences from crowd workers into one unanimous global preference. I assume that each crowd worker has their own beliefs as to the correct preferences for all

[1]https://www.mturk.com/
[2]https://www.innocentive.com/
[3]https://www.figure-eight.com/
[4]https://www.allourideas.org/

Table 1.1 The applications of rank aggregation

Reference	Applications
Raman and Joachims [2014]	ordinal peer grading
Kolde et al. [2012]	gene list integration and meta-analysis
Chen et al. [2013]	reading level assessment
Xu et al. [2018]	image quality assessment & world college ranking
Baltrunas et al. [2010]	recommendations
Knight and Keith [2005]	image-rating
Domshlak et al. [2007]	schema matching
Ailon et al. [2008]	clustering
Sarkar et al. [2014]	feature selection
Fagin et al. [2003]	classification
Rayana and Akoglu [2014]	event detection
Berrada and Cheney [2019]	anomaly detection

items and will annotate each partial preference according to those beliefs. To control the difficulty of the task, the number of items in each subset is restricted to be much smaller than the total number of items (for examples, ≤ 7) [Raman and Joachims, 2014].

Furthermore, the preferences could arise not only by explicitly querying crowd workers, but also through passive data collection, i.e., by observing user purchasing behavior [Baltrunas et al., 2010], clicks on search engine results [Dwork et al., 2001], etc. The flexible collection of preferences enables successful application of RA in various fields, from image rating [Liang and Grauman, 2014] to peer grading [Raman and Joachims, 2014], and bioinformatics [Kim et al., 2014]. I summarize the applications in Table 1.1. A comprehensive review of RA can be found in Fürnkranz and Hüllermeier [2010], Lin [2010] and references therein.

Overall, the final goal of RA is to reliably and efficiently aggregate the partial preferences[5] collected from multiple heterogeneous users into one consensus global ranking over all items. In this thesis, I assume that the global ranking has a single ground-truth ranking, which is a fundamental assumption in many aggregation models.

1.1.2 Rank Aggregation Methods

In literature, many RAs have been proposed to find a consensus ranking r_* over these partial preferences $\mathscr{R}_N = \{\rho_n\}_{n=1}^N$. They all fall into the framework of minimizing a distance

[5]The number of items compared in each preference could be varied.

between predicted ranking and input preferences.

$$\min_{r} \sum_{n=1}^{N} D(r, \rho_n). \tag{1.1}$$

According to different distances used for optimization, they can be mainly divided into three categories: pointwise, pairwise and listwise approaches.

Pointwise Rank Aggregation Methods

Pointwise rank aggregation methods utilize the order information per item from all the input preferences to define the ranking function or objective function. The distance measure between the consensus ranking and the input ranking is decomposed into position difference per item, for example, Borda Count [de Borda, 1781] and Median Rank [Fagin et al., 2003].

Borda Count minimizes the average Spearman Rank Coefficient and obtains the optimal ranking by the mean position of each item. Median Rank optimizes the average Spearman Footrule Distance between the consensus ranking and each ranking input, and obtains the optimal solution by sorting items according to their median rank in ascending order. Kolde et al. [2012] resorted to order statistics, which computes a significant score for each item by comparing the input rankings with the expected behavior of randomly shuffled preferences.

Note that pointwise rank aggregation methods require inputs to be all full rankings. This could be extended to top-k rankings by assuming the unseen items are ranked at the bottom. However, this approach is intractable to scale to the crowdsourced partial preferences collection scenario (Figure 1.1) where only partial preferences are available.

Pairwise Rank Aggregation Methods

Pairwise rank aggregation methods organize their inputs in a pairwise way, whether for ranking functions or optimization objective functions.

In terms of graph-based methods, for example, Condorcet Fuse [Montague and Aslam, 2002] constructs the Condorcet graph with all items and its arc representing the pairwise comparison results between two items by majority voting, and a Hamiltonian path is obtained from this graph by QuickSort. With all the aggregated pairwise comparisons summarized in a tournament, a consensus ranking can be achieved via minimizing the pairwise disagreement cost in this tournament. In terms of matrix-based methods. With all the inputs summarized in a pairwise comparison matrix, singular vector projection [Gleich and Lim, 2011] minimizes the nuclear norm of this pairwise comparison matrix by rank-2 factorization. In terms of probabilistic approaches, a generative probability of pairwise comparisons is defined

and a gradient-based approach is adopted to optimize the likelihood function [Bradley and Terry, 1952, Maydeu-Olivares, 1999], that is, minimizing the Kullback-Leibler divergence in Equation 1.1.

Pairwise rank aggregation method is the most popular in real applications due to its simplicity and the popularity of pairwise comparisons. Furthermore, it could be generalized to partial preferences with a simple rank-breaking method, namely first breaking each partial preference into a set of pairwise comparisons and then modeling each pairwise comparison independently with the adopted pairwise ranking model [Soufiani et al., 2014, Khetan and Oh, 2016].

Listwise Rank Aggregation Methods

Listwise rank aggregation methods model the partial or full ranking lists directly. They measure the distance between the consensus probability distributions and the likelihood of the collected ranking lists. Popular distances are Kullback-Leibler divergence [Kullback and Leibler, 1951] and Kendall tau rank distance [Kendall, 1948].

In particular, the Plackett-Luce model [Luce, 1959, Plackett, 1975] is usually used along with Kullback-Leibler divergence to model partial preferences. Then, a maximum likelihood estimation is considered to optimize an equivalent objective function in Equation 1.1. Mallow's model [Mallows, 1957] is another popular generative probability function, to model partial preferences using the Kendall tau distance [Kendall, 1948]. Qin et al. [2010] addresses the limitations of Luce's and Mallow's rank aggregation models in terms of expressiveness or computational complexity. They proposed a new model, which was defined by a coset-permutation distance, and modeled the generation of a permutation as a stagewise process. This model has rich expressiveness and low complexity.

1.2 The Challenges of Rank Aggregation

The first challenge is the issue of model misspecification, which refers to the inconsistency between the collected ranking lists and the ranking model assumption. To be specific, a basic assumption underlying most RAs is that all ranking lists are provided by homogeneous users, sharing the same annotation accuracy and agreeing with the single ground truth ranking [Dwork et al., 2001, Li et al., 2017]. However, this assumption is rarely satisfied due to the flexible data construction and complex real situation [Gormley and Murphy, 2005, Kolde et al., 2012, Mollica and Tardella, 2017]. This is because typical crowdsourced tasks are tedious and annotators usually come from a diverse pool, including genuine experts,

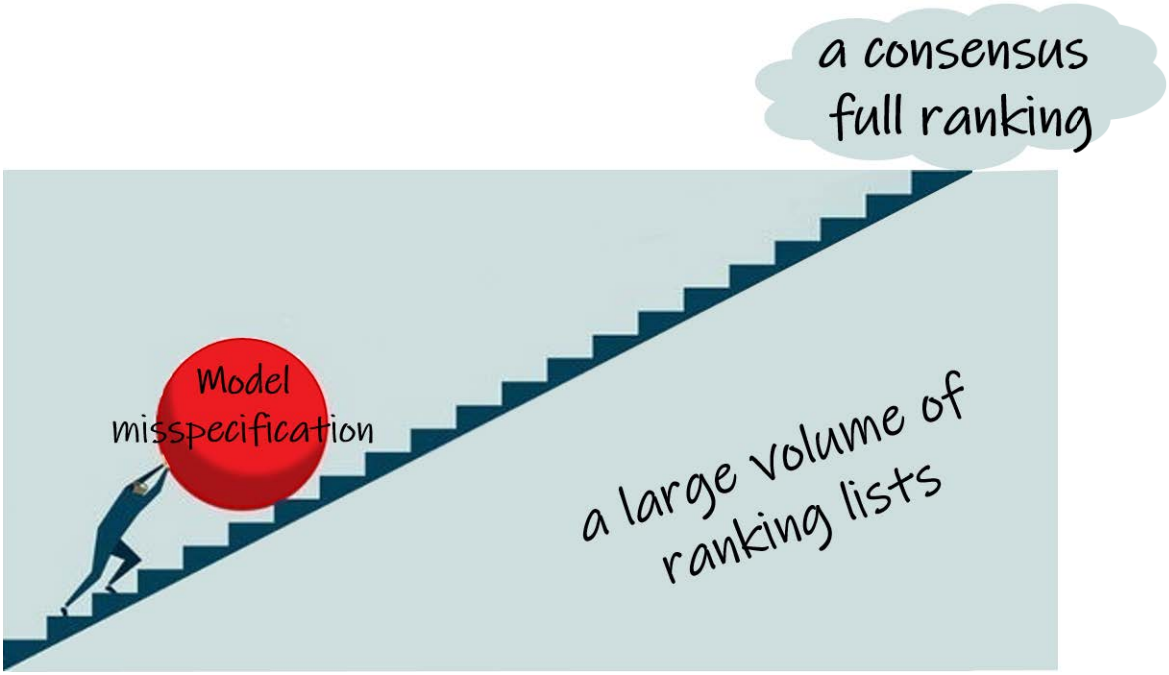


Figure 1.2 Challenges of Rank Aggregation

amateurs, biased workers, and malicious annotators. Therefore, annotations generated by the crowd suffer from low quality.

Various strategies are introduced to ensure the reliability of the collected ranking lists. Some works [Fu et al., 2014, 2015, Xu et al., 2018] formulate the above problem as an outlier detection task. They treat the inconsistent ranking lists as outliers and remove them from the learning data before or during the RA task. Naive majority voting strategy requires each pair is assigned to multiple annotators so as to identify and discard noisy data provided by unreliable annotators. They, thus, require a large amount of pairwise labels to be collected. Xu et al. [2013] formulated the outlier detection task as a LASSO problem. Regularization paths of the LASSO problem could provide order on samples tending to be outliers. Another trend of existing methods models the annotator quality and resorts to an augmentation of the ranking model to account for additional perturbation caused by each annotator. Particularly, the reliability of users is considered in Chen et al. [2013], which studied RA in a crowdsourcing environment for pairwise/trinary preferences. Raman and Joachims [2014] introduced a general framework to aggregate ordinal peer grading from users while considering the user reliability. Model augmentation methods indeed alleviate model misspecification issues to some extent when aggregating crowdsourced partial preferences. It is because sufficient annotations from each crowd worker are available in a crowdsourcing scenario, which enables

an exploration of the heterogeneity of users. However, model misspecification still remains open in a general setting, where only one preference is usually available for each user. In such a case, model augmentation methods would cause overfitting since at least one parameter would be estimated for each preference.

Another challenge to address is the scalability. As mentioned, rank aggregation tasks usually involve massive items in real applications. For ordinal peer grading, tens of thousands of students usually participate in an same online class [Wulf et al., 2014]. For the online game competition, for examples, chess and Go, an unlimited number of competitors would be involved [Elo, 1978, Herbrich et al., 2007]. This would lead to a large volume of ranking lists for aggregation.

Traditional inferences for aggregation models usually lead to a massive computational cost. For example, with a large number of preferences (for examples, 5×10^5 preferences), a gradient descent-based Plackett-Luce model would suffer from high computation costs. Mallow's model resorts to sampling-based algorithms, for examples, Gibbs sampling, which are too slow to infer the model's parameters. To enhance the reliability of RA for low-quality preferences, namely RA against model misspecification, even simple extensions on vanilla RA would lead to exponential high computation costs for the large volume of ranking lists in real-world challenges.

The above challenges raise a question: Can a robust RA be built, along with an efficient inference approach, for a large volume of noisy preferences encountered in real applications?

1.3 Thesis Contributions

The main contributions of this thesis can be summarized in the following three sections.

1.3.1 Stagewise Learning for Crowdsourced Partial Preferences

In this section, I consider model misspecification in a crowdsourcing scenario, where sufficient annotations from each crowd worker are available for exploring the worker heterogeneity. In particular, a reliable CrowdsOUrced Plackett-LucE (COUPLE) model is proposed for aggregating crowdsourced partial preferences, combined with an efficient Bayesian learning technique. COUPLE models a k-ary preference as a series of sequential comparison stages. In each stage, one item from a number of alternatives is selected preferentially as a "local winner" without replacement. Because many crowd workers have limited expertise, a simple stagewise strategy can be easily confounded by the unreliable decisions crowd workers have made. Hence, to ensure the reliability of the estimated ranking, I proposed a robust

learning paradigm, called stagewise learning. This paradigm considers the indecision in crowd workers' choices when selecting a local winner. In each stage, I identified several potential local winners according to an estimate of each crowd worker's expertise. Once all stages are complete, this robust learning strategy recovers the ground truth from the noisy preferences with a certain level of probability. To ensure COUPLE's scalability, I proposed an efficient online Bayesian moment matching method, which ensures COUPLE is scalable to large-scale datasets. Specifically, I designed analytic rules to efficiently update the posterior of COUPLE after each observation, which naturally leads to an online update facility. *Contributions:*

- I presented a CrowdsOUrced Plackett-LucE (COUPLE) model to directly aggregate crowdsourced partial preferences, which avoids the statistical inconsistencies caused by rank-breaking.
- To ensure reliability, I introduced an uncertainty vector to model the quality of each worker, which recovers the ground truth from their noisy preferences with a certain level of probability.
- To ensure the scalability of COUPLE, I proposed an Online Generalized Bayesian Moment Matching (OnlineGBMM) algorithm to update the posterior of COUPLE analytically, which ensures COUPLE is scalable to large-scale datasets.

Outcome:

- This contribution is accepted by Machine Learning-2018 [Pan et al., 2018].

1.3.2 Robust Rank Aggregation against Model Misspecification

In this section, I consider model misspecification in a general setting, where only one preference is usually available for each user. Typical model augmentation methods would cause overfitting in such a scenario, because at least one parameter would be estimated for each preference. In particular, a novel robust RA approach, called CoarsenRank is introduced. The main idea of CoarsenRank is to perform RA over the neighborhood of the noisy ranking data, which enables CoarsenRank against most potential perturbations [Volpi et al., 2018, Chen and Paschalidis, 2018b]. However, it is intractable to directly make inferences over the neighborhood of the noisy ranking data because of the unlimited samples involved. It also prohibits gradient-based solutions adopted for distributional robustness in the optimization community, due to the particularity of the ranking data [Blanchet et al., 2016, Gao et al., 2017]. For the sake of tractability, the relative entropy is adopted as the

divergences metric to define the neighborhood [Ben-Tal et al., 2013, Namkoong and Duchi, 2017]. I further introduced a prior distribution for the unknown size of the neighborhood to avoid parameter tuning and derive a much-simplified formula for CoarsenRank. Furthermore, CoarsenRank is instantiated using three popular probability ranking models, followed by the corresponding optimization strategies. In particular, a surrogate objective is proposed for the Coarsened Thurstone model, which enables various gradient-based optimizations. Regarding the Coarsened Bradley-Terry/ Plackett-Luce model, a data augmentation trick is adopted to eliminate the annoying normalization term and a tractable closed-form solution is derived.

Contributions:

- I introduced a novel robust rank aggregation method called CoarsenRank. CoarsenRank performs RA over the neighborhood of the ranking data instead of original data directly. CoarsenRank is the first rank aggregation method against model misspecification and enjoys distributional robustness.
- I obtained a computationally efficient formula for CoarsenRank, which introduces only one extra hyperparameter to vanilla ranking models. Furthermore, I derived a tractable closed-form solution and introduce a data-driven criterion for choosing the hyperparameter.

Outcome:

- This contribution is currently under review by Machine Learning.

1.3.3 Real-time Mental Fatigue Monitoring

In this section, I apply RA for real-time mental fatigue monitoring. Common practices for mental fatigue monitoring refer to predicting the reaction time (RT) to some emergency by aggregating the EEG signal from multiple heterogeneous EEG channels. Let us consider the RT as the item score to construct brain dynamics-related preferences, and view each EEG channel as a crowd worker. The mental fatigue monitoring task could then be formulated as RA under model misspecification, particularly in a crowdsourcing scenario, while involving the EEG signals as the features. This framework consists of two components, a Self-Weight Ordinal REgression (SWORE) model and a Brain Dynamics table (BDtable). The SWORE model learns from brain dynamics preferences from multiple noisy channels by learning the reliability of each channel explicitly within the aggregation process, while the BDtable maintains the landmark EEG signals and the corresponding RTs as the reference in real-time (online). Whenever a new EEG signal x_t comes at time t, the SWORE model could give a coarse estimation of its reaction time RT_t by interpolating it among the bunch of

maintained RTs using the brain dynamics preferences. An online generalized Bayesian moment matching (OGMM) algorithm is further proposed for Bayesian posterior updating. Once the real reaction time RT_t is available, the BDtable would help online calibrating of the SWORE model by utilizing the simple analytic update rules introduced in the OGMM algorithm.

Contributions:

- I proposed a real-time mental fatigue monitoring system using EEG in real-time while maintaining a high prediction performance.
- I proposed a Self-Weight Ordinal REgression (SWORE) model and a Brain Dynamics table (BDtable). The SWORE model reliably aggregates brain dynamics preferences from multiple noisy channels, while the BDtable is used to calibrate the SWORE model in real-time (online).
- I proposed an online generalized Bayesian moment matching (OGMM) algorithm for Bayesian posterior updating. OGMM can handle complex models where the likelihood are twice differentiable.
- I explored the reliability of SWORE in a real-time mental fatigue evaluation task and analyze the parameter sensitivity and model uncertainty of the SWORE with regard to the OGMM algorithm.

Outcome:

- This contribution is currently under review by IEEE Transactions on Pattern Analysis and Machine Intelligence.

1.4 Thesis Outline

This thesis focuses on robust RA against model misspecification. In addition, I designed efficient solutions to ensure that the robust model is scalable to large-scale datasets in real challenges. Furthermore, I applied the robust RA to the mental fatigue evaluation task. Specifically, I proposed a real-time mental fatigue evaluation framework in the context of mental fatigue of drivers, by utilizing the brain dynamics related preferences. This thesis is organized as follows:

- Chapter 2 introduces the related work;

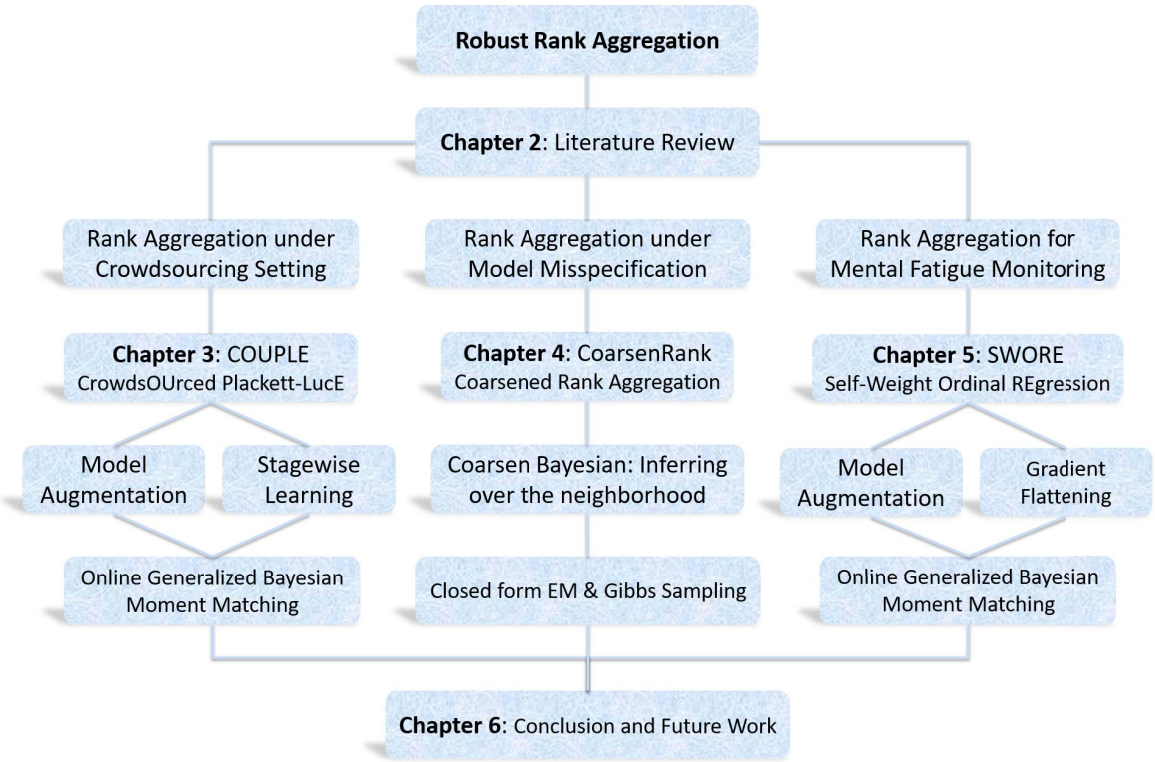


Figure 1.3 The organization of this thesis.

- Chapter 3 introduces a stagewise learning RA paradigm for crowdsourced preferences;
- Chapter 4 studies robust RA against model misspecification;
- Chapter 5 introduces online Bayesian RA for real-time mental fatigue monitoring;
- Chapter 6 concludes this thesis and presents possible future works.

The organization of this thesis is shown in Figure 1.3.

1.5 Publications

1. **Yuangang Pan**, Weijie Chen, Gang Niu, Ivor W. Tsang, Masashi Sugiyama. Fast and Robust Rank Aggregation against Model Misspecification. under review by Machine Learning

2. **Yuangang Pan**, Weijie Chen, Ivor W. Tsang, Masashi Sugiyama. Exploring Disentangled Features among Multiple Responses. under review by Machine Learning

3. **Yuangang Pan**, Avinash K Singh, Yueming Lyu, Ivor W. Tsang, Chin-teng Lin: Online Brain Dynamics Ranking with Real-time Monitoring. under review by IEEE Transactions on Pattern Analysis and Machine Intelligence

4. **Yuangang Pan**, Avinash K Singh, Ivor W. Tsang, Chin-teng Lin, Masashi Sugiyama: Mental Fatigue Monitoring using Brain Dynamics Preferences. under review by Neural Computation

5. Yaxin Shi, Donna Xu, **Yuangang Pan**, Ivor W. Tsang. "Multi-Context Label Embedding." AAAI (2019).

6. Bo Han, Quanming Yao, **Yuangang Pan**, Ivor W. Tsang, Xiaokui Xiao, Qiang Yang, Masashi Sugiyama. "Millionaire: A Hint-guided Approach for Crowdsourcing." Machine Learning 108.5 (2019): 831-858.

7. **Yuangang Pan**, Bo Han, and Ivor W. Tsang. "Stagewise learning for noisy k-ary preferences." Machine Learning 107.8-10 (2018): 1333-1361.

8. Bo Han*, **Yuangang Pan***, and Ivor W. Tsang. "Robust Plackett–Luce model for k-ary crowdsourced preferences." Machine Learning 107.4 (2018): 675-702. (Equal)

Chapter 2

Literature Review

This thesis focuses on listwise rank aggregation methods. Particularly, I assume a generative ranking model for the collected preferences. Meanwhile, I explore efficient inference algorithms to scale this rank aggregation model to a large volume of preferences emerged in real applications. The overall structure of this Chapter are summarized in Figure 2.1.

2.1 Problem Statement

Considering an item set $\mathscr{O} = \{o_1, o_2, \ldots, o_M\}$, the observed dataset $\mathscr{R}_N$ denotes a collection of partial preferences over subsets of $\mathscr{O}$, that is, $\mathscr{R}_N = \{\rho_1, \rho_2, \ldots, \rho_N\}$, $\rho_n : \rho_n^1 > \rho_n^2 > \ldots > \rho_n^k$ and $\{\rho_n^1, \rho_n^2, \cdots, \rho_n^k\} \subseteq \mathscr{O}$. Let $|\rho_n| = k$ denote the number of items compared in ρ_n, where $2 \leq k \ll M$. The notation $\rho_n^i > \rho_n^j$ denotes item ρ_n^i is preferred over item ρ_n^j. Furthermore, I assume a parameterized generative ranking model P_θ for the collected preferences $\mathscr{R}_N$. Let P_o denote the unknown rank generation model in practice.

Therefore, the rank aggregation task, that is, aggregating the collected preferences $\mathscr{R}_N$ into a total order over all items in $\mathscr{O}$, is equivalent to find an optima θ to minimize a distance between the proposed ranking model P_θ and the real rank generation model P_o, given the collected ranking lists $\mathscr{R}_N$. Namely,

$$\min_{\theta \in \Theta} D(P_\theta, P_o). \tag{2.1}$$

According to different choices of generative ranking models P_θ and distance measures $D(.,.)$, a great variety of rank aggregation methods could be derived.

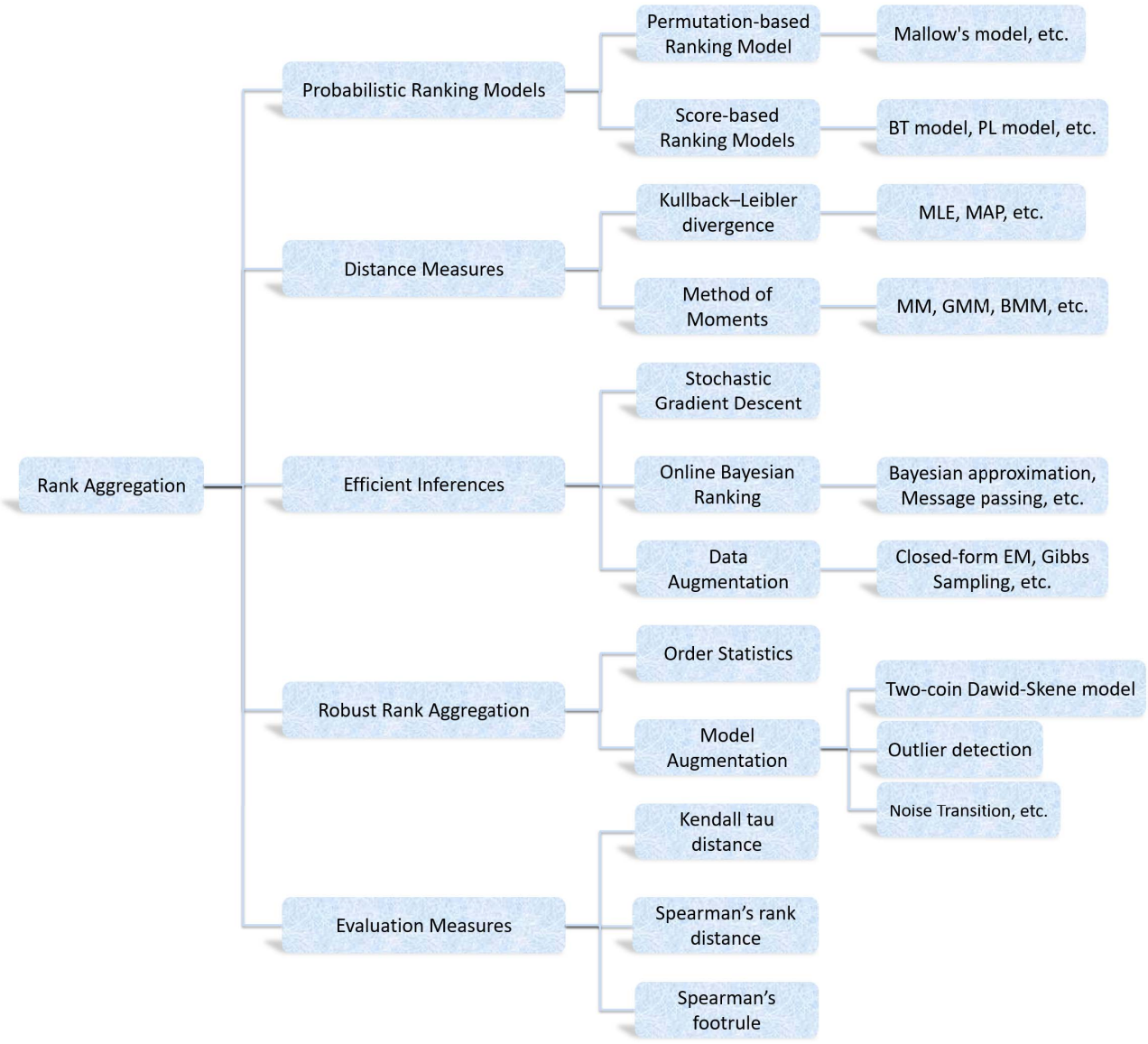


Figure 2.1 The overall structure of literature review.

2.2 Probabilistic Ranking Models

Probabilistic ranking models deal with learning probability distributions over permutations (that is, rankings or preferences over items). They solely concern preferences, paying little attention to features. There are two main paradigms: permutation-based and score-based ranking models.

2.2.1 Permutation-based Ranking Model

Permutation-based models are based on the notion of distances [Mallows, 1957, Fligner and Verducci, 1986], which express the probability of a permutation in terms of its distance to a reference permutation. The most prominent example of these models is Mallow's model, an exponential model that depends on the definition of a distance for permutations. Let ρ be a ranking list, then Mallow's model specifies:

$$P(\rho|r) = \frac{1}{\psi(\sigma)} \exp(-\sigma d(\rho, r)), \quad \text{where } \psi(\sigma) = \sum_{\rho} \exp(-\sigma d(\rho, r)). \tag{2.2}$$

Here $\sigma \in \mathbb{R}_+$ is a spread parameter, r is the reference permutation and $d(\rho, r)$ represents a distance between ρ and r. Note that r is the mode, and the closer a ranking list ρ is to r, the larger $p(\rho)$ is . There are a total of six different distances listed in Diaconis [1988], giving rise to the family of the distance-based probability models. The alternative distance measures considered are Kendall tau distance, Spearman's rank distance, Hamming, Ulam, Cayley and Spearman's footrule. Among in, Kendall's-τ has become the most recurrent in both theoretical and applied studies, due to its nice theoretical properties.

However, permutation-based models are often impractical for the large-scale problem, because, (1) the normalization term usually requires high computational cost due to discrete distance computation; and (2) a maximum likelihood estimation involves an impossible discrete search for ranking over a large volume of items.

2.2.2 Score-based Ranking Models

Score-based models assign a true score for each item and express the probability of a ranking list in terms of item-specific scores. In particular, they assume that users provide rankings of subsets of items by comparing the approximate version of these scores corrupted by additive noise. When restricted to comparing pairs of items, Thurstone's model gives rise to the Bradley-Terry model [Bradley and Terry, 1952] if the noise follows the Gumbel distribution, and to the Thurstone Case V model [Thurstone, 1927b] if the noise is normal. Namely, for a pairwise comparison $\rho : o_i > o_j$, I have

$$P(\rho) = \begin{cases} \frac{e^{\theta_i}}{e^{\theta_i} + e^{\theta_j}} & \text{Bradley-Terry model,} \\ \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\frac{1}{\sqrt{2}}(\theta_i - \theta_j)} e^{-x^2/2} dx & \text{Thurstone Case V model.} \end{cases} \tag{2.3}$$

where θ denotes the score associated with each item.

Furthermore, the Bradley-Terry model can be generalized to describe comparisons of multiple items. When data consists of partial rankings (as opposed to pairwise comparisons), Plackett-Luce model [Luce, 1959, Plackett, 1975] can be represented as,

$$P(\rho) = \prod_{i=1}^{k} \frac{e^{\theta_{\rho^i}}}{\sum_{j=i}^{k} e^{\theta_{\rho^j}}}. \tag{2.4}$$

where $\rho : \rho^1 > \rho^2 > \ldots > \rho^k$ denotes the listwise preference. θ is the element-wise score vector.

2.3 Distance Measures

Various distribution divergences could be adopted to implement the distance measures $D(.,.)$. These would trigger different properties of the learning model and further influence the choice of inference strategy.

2.3.1 Kullback–Leibler divergence

The Kullback-Leibler (KL) divergence is a natural distance function from a "true" probability distribution p to a "target" probability distribution q. Namely,

$$D_{KL}(p\|q) = \int p(x) \cdot \log \frac{p(x)}{q(x)} dx. \tag{2.5}$$

The KL divergence brings superior properties into the rank aggregation problem Equation 2.1. It could be proved to be equivalent to the maximum likelihood estimation (MLE), by eliminating the unknown "true" probability. In particular,

$$\begin{aligned} \theta_{\min D_{\mathrm{KL}}} &= \arg\min_{\theta} D_{KL}(P_o\|P_\theta) \overset{i}{\approx} \arg\min_{\theta} D_{KL}(F_N(\mathscr{R}_N)\|P_\theta) \\ &= \arg\min_{\theta} E_{\rho\sim\mathscr{R}_N}[\log P_o(\rho) - \log P_\theta(\rho)] \\ &\overset{ii}{=} \arg\max_{\theta} E_{\rho\sim\mathscr{R}_N}[\log P_\theta(\rho)] = \arg\max_{\theta} \sum_{n=1}^{N} \log P_\theta(\rho_n) = \theta_{\max \mathrm{MLE}}. \end{aligned} \tag{2.6}$$

where i holds because of the definition of the empirical data distribution $F_N(\mathscr{R}_N) = \frac{1}{N}\sum_{n=1}^{N} \delta_{\rho_n}(x)$. ii holds due to the elimination of the constant entropy term.

Many rank aggregation approaches fall into this category due to its nice mathematical properties (the equivalent MLE formulation). It is convenient for considering an augmentation

of the ranking model to account for the additional function of the aggregation model. See Chen et al. [2013], Xu et al. [2013], Raman and Joachims [2014] for more details.

2.3.2 Method of Moments

The method of moments is a fairly simple approach for parameter estimation and yields consistent estimators by matching a set of generalized moment conditions between the "true" probability distribution p and the "target" probability distribution q. Namely,

$$E_{x\sim p(x)}\left[x^k\right] = E_{x\sim q(x)}\left[x^k\right], \qquad k = 1,2,\ldots,K. \tag{2.7}$$

Moment-based estimates can still be quickly and easily calculated for an unknown probability distribution. For example, this is preferred to MLE when estimating parameters of a utility function, instead of parameters of a known probability distribution. Moment methods show great potential for rank aggregation due to the special structure of the ranking lists [Soufiani et al., 2013, Zhao et al., 2016, 2018].

Furthermore, moment methods would be a good alternative to derive closed-form updating rules for parameter estimation with non-conjugate likelihood functions. In particular, online learning algorithms are introduced to scale the rank aggregation approaches to a large-scale/sequential scenario following the Bayesian theorem [Weng and Lin, 2011, Chen et al., 2013, Han et al., 2018].

2.4 Efficient Inferences for Ranking Models

The scalability is an indispensable aspect of the learning algorithm for rank aggregation tasks, due to the widespread large-scale scenario [Wulf et al., 2014] or streaming scenario [Elo, 1978, Herbrich et al., 2007] in real applications.

2.4.1 Online Stochastic Ranking

Stochastic Gradient Descent (SGD) is a natural choice to extend the learning algorithm to the large-scale scenario, which updates the gradient of one or mini-batch samples each time. SGD is widely adopted for rank aggregation model since it requires nothing but the gradient, which is usually accessible for most methods. For example, Chen et al. [2013], Raman and Joachims [2014] resorted to SGD to derive an online updating paradigm.

2.4.2 Online Bayesian Ranking

Online Bayesian ranking is another practical technique to handle large-scale datasets, following the Bayesian theorem. In these methods, the global rankings are updated with streaming preferences. Elo [Elo, 1978] and Glicko [Glickman, 1999] are famous online ranking systems. Herbrich et al. [2007] developed TrueSkill, which constructs a graphical model and inferences with approximate message passing. Weng and Lin [2011] introduced a Bayesian approximation method to derive simple analytic rules for inference in k-ary preferences aggregation. Their methods, OnlineBT and OnlinePL, achieve competitive accuracy with the TrueSkill system, but are much faster, as they both rely on analytical update rules rather than the iterative procedures in TrueSkill.

2.4.3 Data Augmentation

Although SGD could greatly free me from the burden caused by the large-scale scenario, it is still far from deriving an efficient learning algorithm. The main inferential issue lies in the presence of the annoying normalization term in the probability ranking model. Direct applications of SGD result in a less informative gradient updating rule, requiring more iterations over the whole dataset.

Note that the Gumbel distribution is employed as a distribution of the support parameters and the conjugacy of the Gamma density with the Gumbel distribution. Inspired by such an observation, Caron and Doucet [2012] introduced an auxiliary Gamma random variable for each normalization term, which leads to a joint distribution without suffering from the annoying normalization terms. Therefore, a simple and effective solution Expectation-Maximum (EM) solution is derived accordingly.

2.5 Robust Rank Aggregation

Rank aggregation aggregates preferences collected from different data sources into one unanimous global preference. However, most existing methods fail to consider the heterogeneity of data sources [Tsiporkova and Boeva, 2006] and reliability suffer as a result of problems with the crowdsource settings [Vuurens et al., 2011].

2.5.1 Robust Rank Aggregation using Order Statistics

Order statistic refers to positions of each item in the ranking lists. Order statistic itself is a robust but informative indicator for ranking the item. In particular, the significance of

the order statistic w.r.t. one item, being superior over expectation by chance, reveals the importance of this item over other alternatives and usually used for ranking the item. Order statistics was first utilized for robust rank aggregation by Stuart et al. [2003], Aerts et al. [2006]. While being robust to noise, these methods require simulations to define significance thresholds and do not support incomplete lists. These limitations were then theoretically addressed in Kolde et al. [2012] and this methodology is generalized to the top-k ranking.

Although the rank aggregation methods using order statistic are computationally efficient and statistically stable, they are not suitable for the large-scale setting considered in this thesis. In particular, the large-scale setting means that only particular preferences are available during data collection, instead of top-k preferences and let alone full preferences.

2.5.2 Robust Rank Aggregation via Model Augmentation

The collected preferences are viewed as a noisy perturbation of some idealized preferences, and some robust rank aggregation methods resort to an augmentation of the ranking model to account for additional noises. Vitelli et al. [2014] used a Bayesian framework to deal with the preference uncertainty, modelling the worker quality implicitly. Chen et al. [2013], Han et al. [2018] studied RA in a crowdsourcing environment for pairwise/trinary preferences, modelling the user reliability following the two-coin Dawid-Skene model [Raykar et al., 2010]. Raman and Joachims [2014] introduced a general framework to aggregate ordinal peer gradings from peer graders while exploring each grader's reliability by introducing a scale factor.

To better capture the perturbation pattern of each user, model augmentation based methods usually assume sufficient samples from each user are collected. However, each user usually provides one preference in real applications, which would cause overfitting since at least one parameter, denoting user reliability variable, would be estimated with regard to each preference [Sajjadi et al., 2016]. The same problem also arises for Xu et al. [2017], which formulates the robust RA as outlier detection and introduces a deviation factor for each preference to account for the unknown noise-perturbation. Actually, these previous attempts simply amount to convolving the original ranking model with some pre-assumed perturbation mechanism. It leads to a new model with a few more parameters but is just as bound to be misspecified w.r.t. other overlooked perturbations. Furthermore, model augmentation inevitably introduces extra computation cost for the learning model, which raises a higher requirement for optimization when aggregating a large volume of preferences.

2.6 Evaluation Measures

Let $\bar{\theta}$ be the consensus full ranking list over $\mathscr{O}$ estimated by some rank aggregation method, while θ_* be the ground truth. There are many well-defined metrics to measure the distance between two permutations, for example, Kendall tau distance [Kendall, 1938] and Spearman's rank distance [Daniel, 1990].

Kendall tau distance The Kendall tau distance is one of the most common measures, which counts the pairwise disagreements between items from two rankings. Namely,

$$\tau^1(\bar{\theta}, \theta_*) = \frac{1}{C_M^2} \sum_{i<j} (\mathbb{1}[(\bar{\theta}^i > \bar{\theta}^j) \wedge (\theta_*^i < \theta_*^j)] + \mathbb{1}[(\bar{\theta}^i < \bar{\theta}^j) \wedge (\theta_*^i > \theta_*^j)]), \tag{2.8}$$

where $C_M^2 = \frac{1}{2}M(M-1)$ denotes total number of pairs. $\mathbb{1}$ is an indicator function, which equals to 1 if the condition is true and 0 otherwise. $d(\bar{\theta}, \theta_*)$ ranges from 0 (identical) to 1 (reverse).

Accordingly, the Kendall tau correlation is usually defined as:

$$S(\bar{\theta}, \theta_*) = 1 - \tau^1(\bar{\theta}, \theta_*), \tag{2.9}$$

$S(\bar{\theta}, \theta_*)$ will be equal to 1 if the two lists are identical and 0 if one list is the reverse of the other.

Spearman's rank distance The Spearman's rank distance is a nonparametric measure of the strength and direction of the association that exists between two ranking lists. It is usually formulated as:

$$\tau^2(\bar{\theta}, \theta_*) = \frac{6 \sum_m^M d_i}{M(M^2-1)}, \tag{2.10}$$

where M denotes the number of items. $d_i = \theta^i - \theta_*^i$ is the difference between the order of the item i in two ranking lists. $\tau^2(\bar{\theta}, \theta_*)$ will always be between 1 (reverse) and -1 (identical).

Then, then Spearman's rank correlation is can be represented as folows,

$$S(\bar{\theta}, \theta_*) = 1 - \tau^2(\bar{\theta}, \theta_*), \tag{2.11}$$

$S(\bar{\theta}, \theta_*)$ will be equal to 1 if the two ranking lists are identical and -1 if one list is the reverse of the other.

Spearman's footrule Spearman's footrule refers to another two popular rank measures. It calculates the sum of absolute values or square values of the difference of each item between two ranking lists, respectively.

$$\tau^3(\bar{\theta}, \theta_*) = \sum_{i=1}^{n} |\bar{\theta}^i - \theta_*^i|, \tag{2.12a}$$

$$\tau^4(\bar{\theta}, \theta_*) = \sum_{i=1}^{M} (\bar{\theta}^i - \theta_*^i)^2, \tag{2.12b}$$

The relationships of $\tau^3(\bar{\theta}, \theta_*)$ and $\tau^4(\bar{\theta}, \theta_*)$ with other commonly used non-parametric measures (i.e., Kendall tau distance and Spearman's rank distance) and its asymptotic normality have been studied by Diaconis and Graham [1977].

Chapter 3

COUPLE: Stagewise Learning for Noisy Preferences

In this section, I consider model misspecification in a crowdsourcing scenario, where sufficient annotations from each crowd worker are available for exploring the worker heterogeneity. In particular, a reliable CrowdsOUrced Plackett-LucE (COUPLE) model is introduced for aggregating crowdsourced noisy preferences. To ensure reliability, I introduce an uncertainty vector for each crowd worker in COUPLE, which helps to recover (in expectation) the ground truth of the noisy preferences. Furthermore, I propose an Online Generalized Bayesian Moment Matching (OnlineGBMM) algorithm, which ensures that COUPLE is scalable to large-scale datasets. Comprehensive experiments on four large-scale synthetic datasets and three real-world datasets show that COUPLE with OnlineGBMM achieves substantial improvements in reliability and noisy worker detection over the well-known approaches.

3.1 Towards the robust aggregation of noisy preferences

In this section, I first discuss the deficiency of classical methods for aggregating noisy preferences. Then, I propose the COUPLE model and analyze its reliability and difficulty of optimization.

3.1.1 Intractability of classical models

For a k-ary preference (ranking list of k items), a popular ranking model is the Plackett-Luce (PL) model. PL model relies on Luce's axiom of choice [Luce, 1959], i.e., the odds of choosing an item over another do not depend on the set of items from which the choice is made. Suppose I have a set of k items $\xi = \{o_1, o_2, \ldots, o_k\}$. Under Luce's axiom, the

probability of selecting an item i from ξ is given by $e^{\theta_i}/\sum_{t=1}^{k} e^{\theta_t}$, where θ_i represents the **score** (real-value constant) associated with item i. The larger the score θ_i, the higher-position the item i ranked in the preference. Considering a k-ary preference $\rho : o_1 > o_2 > \ldots > o_k$ as a sequence of choices: the top-ranked item is chosen first, followed by the second-ranked item from the remaining items, and so on. It follows that the probability of the preference ρ is

$$f_{PL}(\rho|\theta) = \prod_{i=1}^{k} \frac{e^{\theta_{\rho^i}}}{\sum_{t=i}^{k} e^{\theta_{\rho^t}}}, \tag{3.1}$$

where ρ^i is the i-th ranked item in ρ. The above model is also derived in Plackett [1975]; hence the name the Plackett-Luce model. Given the assumption of single ground truth, Equation 3.1 actually defines the likelihood of each preference over the subset, based on the score (θs) for each item. The more a preference is consistent with the ground truth (over the subset), the larger the probability value of f_{PL} (Equation 3.1). However, since the parameter θs are unknown, f_{PL} itself cannot be viewed as an indicator to discriminate the high-quality preferences from the low-quality ones. Further, the parameters θs are usually estimated with a maximum likelihood estimation (MLE), which aims to find the parameter θs (of f_{PL} model) that best fit the data, without distinction of the preference quality. As previously mentioned, preferences from crowdsourcing platforms are often noisy, and low-quality preferences could easily skew estimations of parameter θs. Therefore, the performance of a vanilla Plackett-Luce model suffers when aggregating crowdsourced k-ary preferences.

Chen et al. [2013] proposed CrowdBT to reliably aggregate crowdsourced pairwise preferences while modelling the worker quality η_w of crowd worker w, $\forall w \in \{1,2,\ldots,W\}$. Let η_w represent the conditional probability that the pairwise preferences annotated by crowd worker w accords with the ground truth. Namely, $\eta_w = P(o_i \tilde{>} o_j | o_i > o_j)$, where $o_i \tilde{>} o_j$ denotes the pairwise preference annotated by crowd worker w, and $o_i > o_j$ is the ground truth order between item o_i and o_j. According to the law of total probability, I have $P(o_i \tilde{>} o_j) = \eta_w P(o_i > o_j) + (1-\eta_w)P(o_i < o_j)$. However, CrowdBT was originally designed for pairwise preferences, so it cannot directly model crowdsourced k-ary preferences. A simple practice used to generalize CrowdBT into crowdsourced k-ary preference is rank-breaking[1][Soufiani et al., 2013, Shah et al., 2015, Negahban et al., 2016]. Specifically, for a k-ary preference $\rho : o_1 \tilde{>} o_2 \tilde{>} \ldots \tilde{>} o_k$, full rank-breaking refers to the pair indexes $\mathscr{A} = \{(i,j)|i > j, i,j \in \{1,2,\ldots,k\}\}$; adjacent rank-breaking refers to the pair indexes $\mathscr{B} = \{(i,j)|j = i+1, i \in \{1,2,\ldots,k-1\}\}$; and position-$i$ ($< k$) rank-breaking refers to the pair indexes $\mathscr{C} = \{(i,j)|j > i, j \in \{1,2,\ldots,k\}\}$. Therefore, I can break each k-ary preference

[1]Rank-breaking refers to the idea of splitting the observed preference into a set of pairwise comparisons and applying estimators tailored for pairwise preferences treating each piece of comparisons as independent.

into a set of pairwise preferences. Then, CrowdBT would model each pairwise comparison independently. Below, I take $k = 3$ as an example. The likelihood of a ternary preference with full rank-breaking can be expressed as:

$$P(o_a \tilde{>} o_b \tilde{>} o_c) = \prod_{(i,j)\in\mathscr{A}} \left\{ \eta_w P(o_i > o_j) + (1-\eta_w) P(o_i < o_j) \right\},$$

where $\mathscr{A} = \{(a,b),(a,c),(b,c)\}$.

However, due to the ignored dependencies among the pairwise preferences, inappropriate rank-breaking approach would result in inconsistent estimates according to Lemma 1. Further, the computational burden caused by the full rank-breaking would limit its application with large k preferences. See Section 3.3.4 for more details.

Lemma 1 *(Corollary 1 in [Soufiani et al., 2014]) Given a k-ary preference, the only consistent rank-breaking for the Bradley-Terry model is the full rank-breaking.* ■

The above analysis motivates me to directly integrate worker quality into the Plackett-Luce model. This not only avoids the inconsistency and computational burden caused by rank-breaking, but also reliably aggregates noisy k-ary preferences. Again, my aim is to reliably aggregate the noisy k-ary preferences annotated by crowd workers. Unlike the series of practices proposed by Raman and Joachims [2014] that try to reduce the adverse impact of noisy preferences, I intend to to recover the ground truth by capturing each worker's annotation pattern from the noisy preferences. To compare k items, there are $k!$ distinct permutations, which constitutes a finite partition of the entire permutation space. Therefore, I aim to traverse the entire permutation space of each k-ary preference and identify the ground truth.

Inspired by CrowdBT, it is intuitive to traverse the entire permutation space explicitly, where each permutation can be unique indexed based on its distance to the noisy preference (a similar indexing strategy to the permutation-based ranking model). Formally, let ρ denote the noisy preferences, and $\rho[i]$ be the permutation indexed as i-th. Then I have $\eta_w^i = P(\rho|\rho[i])$, which denotes the conditional probability that I observe ρ selected given that the i-th indexed permutation $\rho[i]$ is the ground truth.

However, this approach has the following drawbacks: (1) It requires that all preferences are the same length, and crowdsourced k-ary preferences may not satisfy this inflexible constraint. (2) The permutation-based indexing method is not scalable as k increases. For example, for moderate $k = 6$, the length of η_w reaches $6!(720)$, which is intractable for inference. Therefore, I need to design a clever and practical indexing method to traverse the permutation space.

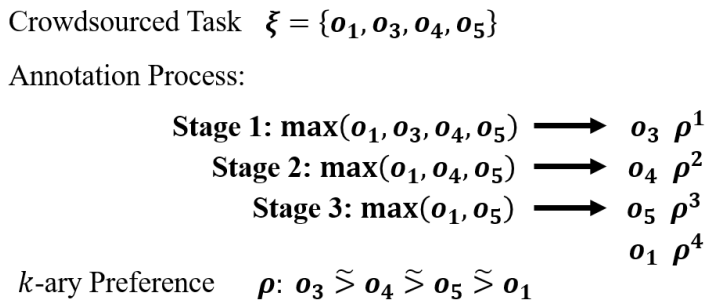


Figure 3.1 Stagewise annotation process

3.1.2 The CrowdsOUrced Plackett-LucE (COUPLE) model

To assist in explaining COUPLE, it is helpful to revisit the Plackett-Luce model from the beginning. The Plackett-Luce model can be regarded as a stagewise model [Volkovs and Zemel, 2012] that constructs a preference through a series of sequential stages. In each stage, compared to all the remaining alternatives, the item selected preferentially (without replacement) is regarded as the "local winner". In particular, in stage 1, item o_3 is selected as the "local winner" and ranked first. Then, o_3 is removed from the candidate set and another "local winners" o_4 is selected in stage 2; o_5 is selected in stage 3, and so on (See Figure 3.1).

Inspired by the stagewise annotation process, I introduce the following stagewise learning strategy, namely the learning process is broken down into a number of sub-tasks that are completed in stages. The idea is to inject ranking information into the learning model gradually so as to focus on modelling the "local winner" in each stage, rather than modelling the complex ranking as a whole. Following the stagewise learning strategy, Plackett-Luce model decomposes each k-ary preference into a series of sequential stages and models each stage independently. Therefore, the likelihood function for the k-ary preference ρ can also be expressed as follows:

$$P(\rho|\theta) = \prod_{i=1}^{k} P(X_i = \rho^i|\theta) = \prod_{i=1}^{k} \delta(\theta_{\rho^i}), \tag{3.2}$$

where $X_i \stackrel{\Delta}{=} \max(\rho^i, \rho^{i+1}, \ldots, \rho^k)$ denotes the "local winner" in stage i. Further, the softmax function $\delta(\theta_{\rho^i}) = e^{\theta_{\rho^i}} / \sum_{t=i}^{k} e^{\theta_{\rho^t}}$ is used to model the probability that item ρ^i is selected as the "local winner" in stage i.

Remark 1 *Given the assumption of a single ground truth, f_{PL} actually defines the likelihood of each preference over the subset. Hence, the preferences with a higher degree of consistency*

to the ground truth (over the subset) have a higher f_{PL} value. There are some rules that govern the behaviour characteristics of crowd workers and the likelihood (f_{PL}) of their corresponding annotations: (1) Expert workers have a clear understanding about the contrast among items. Hence, their annotated preferences are usually fully consistent with the ground truth (the order of θs). Therefore, the likelihood f_{PL} of their preferences are usually the largest, almost 1. *(2) Amateur workers may mistakenly annotate the preferences due to their limited expertise with item contrast. In these cases, their annotated preferences are often slightly inconsistent with the ground truth (the order of θs), and the likelihood f_{PL} of the preference is usually smaller than the annotations by experts. Therefore, if I directly model the noisy k-ary preferences provided by amateur workers without making any distinctions about the quality of the preferences, the Plackett-Luce model's reliability would inevitably degrade.* ■

The stagewise learning strategy is a scalable approach for k-ary preferences in two aspects: (1) each stage can be further assumed independent, and can be updated in a distributed fashion. (2) Since the "local winner" in each stage is much easier to model than the entire preference, it is more flexible to introduce the latent variable with new functions (i.e. indicator for worker quality). Due to crowd workers' limited expertise, a vanilla Plackett-Luce model (Equation 3.2) yields some deviations in modelling the noisy preferences. Hence, based on the introduced stagewise learning strategy, I first split each preference into a series of sequential stages and models each stage independently. Then I focus on recovering the ground truth in each stage, rather than directly identifying the ground truth of the whole preference.

For a K-ary preference [2], at the first stage with K items to be compared (See Figure 3.2), I only need to identify the actual item that is ranked first among the candidate set of size K. Note that each candidate can be uniquely indexed based on the order in the original K-ary noisy preference. Formally, let ρ denote the noisy preferences, where ρ^t is the t-th ranked item (indexed as t-th in the candidate set). I omit the subscripts n and w for brevity. Next, I introduce the uncertainty vector η_w for each crowd worker w to model the worker quality. The length of η_w for any crowd worker w is set to the maximal preference length K. Further, I assume $\eta_w = [\eta_w^1, \eta_w^2, , \ldots, \eta_w^K]$ with $\sum_{t=1}^{K} \eta_w^t = 1$, while each entry $\eta_w^t = P(\tilde{X} = \rho^1 | X = \rho^t)$ denotes the conditional probability that I observe ρ^1 selected given t-th indexed item ρ^t being the ground truth. $\tilde{X}$ denotes the "local winner" selected by crowd worker w. X represents the item that should have been selected according to the ground truth. The strategy of stagewise learning overcomes the deficiency with permutation-based approach (Section 3.1.1), which

[2] K is the maximal preference length, where $K = \max_{n,w} l_{\rho_{n,w}}$, $n = 1, 2, \ldots, N_w$ and $w = 1, 2, \ldots, W$. N_w is the number of k-ary preferences annotated by crowd worker w and $l_{\rho_{n,w}}$ represents the length of preference $\rho_{n,w}$.

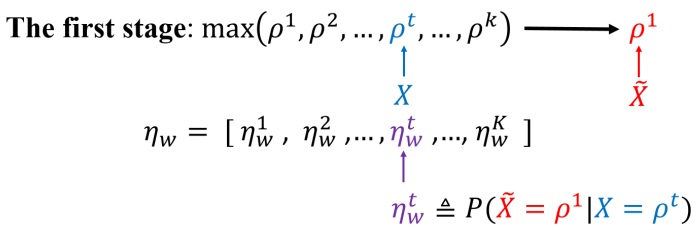


Figure 3.2 An intuitive example for the first stage of the robust stagewise learning strategy.

needs to enumerate all possible permutations, and significantly reduce the parameter space from $K!$ to K.

However, given a K-ary preference ρ, each stage will have a different number of items to compare, which means different entries of the uncertainty vector will be active in each stage. Therefore, a single uncertainty vector is not suitable for processing all stages simultaneously. To avoid this issue, I propose the **renormalization trick**, namely normalizing the active entries in each stage, to populate the definition of the uncertainty vector to subsequent stages. Formally, in the i-th stage, let $\bar{\eta}_w^{t-i+1} = \eta_w^{t-i+1} / \sum_{v=1}^{K-i+1} \eta_w^v$, where $t = i, i+1, \ldots, K$. Following the above rules, I can model the follow-up stages sequentially until there is only one candidate left (Figure 3.3).

Specifically, only two items are compared in stage $K-1$. However, the corresponding active entries $[\eta_w^1, \eta_w^2]$ do not constitute a valid distribution because $\eta_w^1 + \eta_w^2 \neq 1$. Therefore, I normalize the active entries to ensure at least one of the two items being selected. Similarly, in the general stage i, I have $k-i+1$ candidates, which is less than the maximal preference length K. Only the top $k-i+1$ entries of η_w are active. Then, I apply the renormalization trick on the active entries $[\eta_w^1, \eta_w^2, \ldots, \eta_w^{k-i+1}]$, and generalize the definition of uncertainty vector accordingly.

Remark 2 *Assume* $\eta_w = (\eta_w^1, \eta_w^2, \ldots, \eta_w^K)^T$ *and* $\eta_w \sim \mathrm{Dir}(\eta_w | \alpha_w) = \frac{\Gamma(\sum_{i=1}^K \alpha_w^i)}{\prod_{i=1}^K \Gamma(\alpha_w^i)} \prod_{i=1}^K (\eta_w^i)^{(\alpha_w^i - 1)}$, *if* $\bar{\eta}_w^i$ *is renormalized based on the first* k *elements of* η_w, *i.e.,* $\bar{\eta}_w^i = \eta_w^i / \sum_{v=1}^k \eta_w^v$, $i = 1, 2, \ldots, k$, *then I have*

$$\bar{\eta}_w^i \sim \mathrm{Dir}(\bar{\eta}_w | \bar{\alpha}_w) = \frac{\Gamma(\sum_{i=1}^k \alpha_w^i)}{\prod_{i=1}^k \Gamma(\alpha_w^i)} \prod_{i=1}^k (\bar{\eta}_w^i)^{(\alpha_w^i - 1)},$$

where $\bar{\eta}_w = (\bar{\eta}_w^1, \bar{\eta}_w^2, \ldots, \bar{\eta}_w^k)^T$ *and* $\bar{\alpha}_w = (\alpha_w^1, \alpha_w^2, \ldots, \alpha_w^k)^T$.

According to my definition, I can make the following observations: (1) For an expert worker w, η_w^t *decreases exponentially with* t, *as experts have a clearer understanding about the contrast between the items. (2) An amateur worker* w *may hesitate over comparable items*

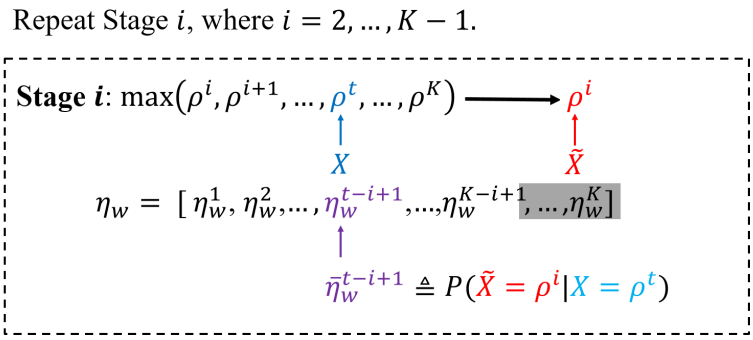


Figure 3.3 Robust stagewise learning strategy. For brevity, I omit the subscripts i which indicates the stage in which the item has been selected.

due to limited expertise. That is to say, η_w^1, denoting the conditional probability that the selected "local winner" accords with the ground truth, does not gain an absolute advantage over other entries $\eta_w^t (t \geq 2)$, especially η_w^2. ■

Therefore, after integrating the Plackett-Luce model with the introduced uncertainty vector, the likelihood of the k-ary preference ρ in stage i can be represented as:

$$P(\tilde{X} = \rho^i | \theta, \eta_w) = \sum_{t=i}^{k} P(\tilde{X} = \rho^i | X = \rho^t) P(X = \rho^t | \theta) = \sum_{t=i}^{k} \bar{\eta}_w^{(t-i+1)} \delta(\theta_{\rho^t}). \tag{3.3}$$

Combining Equation 3.2 and Equation 3.3, I propose a reliable COUPLE model for a collection of crowdsourced preferences $\mathscr{R}$, which can be expressed as follows:

$$\begin{aligned} P(\mathscr{R} | \theta, \{\eta_w\}_{w=1}^W) &= \prod_{w=1}^{W} P(\mathscr{R}_w | \theta, \eta_w) = \prod_{w=1}^{W} \prod_{n=1}^{N_w} P(\rho_{n,w} | \theta, \eta_w) \\ &= \prod_{w=1}^{W} \prod_{n=1}^{N_w} \prod_{i=1}^{l_{\rho_{n,w}}} P(\tilde{X} = \rho_{n,w}^i | \theta, \eta_w) = \prod_{w=1}^{W} \prod_{n=1}^{N_w} \prod_{i=1}^{l_{\rho_{n,w}}} \sum_{t=i}^{l_{\rho_{n,w}}} \bar{\eta}_w^{(t-i+1)} \delta(\theta_{\rho_{n,w}^t}), \end{aligned} \tag{3.4}$$

where η_w is the uncertainty vector for each crowd worker w. This uncertainty vector reveals crowd worker w's indecision to select the "local winner" in each stage. The optimization difficulty of COUPLE model (Equation 3.4) is discussed in Section 3.1.4. In particular, I resort to the Bayesian framework to infer the uncertainty vector η_w, and choose a tailor-designed prior distribution to circumvent the need to directly optimize the normalization.

3.1.3 Reliability of COUPLE model

In this subsection, I characterize the reliability of COUPLE through the score parameter θ and the quality parameter $\{\eta_w\}_{w=1}^{W}$.

Analysis of parameters θ: Assume in stage i of a k-ary preference ρ, worker w selects item ρ^i as the "local winner": (1) In terms of easy tasks, $\theta_{\rho^i} \gg \{\theta_{\rho^{i+1}}, \theta_{\rho^{i+2}}, \ldots, \theta_{\rho^k}\}$ denotes item ρ^i exhibits significant advantages over other items. According to Equation 3.3, I have $P(\tilde{X} = \rho^i|\theta, \eta_w) \approx \eta_w^1$, namely, the likelihood function for stage i is mainly dependent on worker's expertise. (2) In terms of a difficult task, $\theta_{\rho^i} \approx \ldots \approx \theta_{\rho^{i+m}} \gg \{\theta_{\rho^{i+m+1}}, \ldots, \theta_{\rho^k}\}$, I have $P(\tilde{X} = \rho^i|\theta, \eta_w) \approx \frac{1}{m+1}$, which means COUPLE cannot distinguish these $(m+1)$ items $\{\rho^i, \rho^{i+1}, \ldots, \rho^{i+m}\}$ regardless of the worker's expertise.

Analysis of parameters η_w : (1) If worker w is an expert, I have $\eta_w^1 \approx 1$ and $\eta_w^r \approx 0$ for $r \geq 2$, which means the ground truth item wound be selected in each stage with no hesitation. (2) Amateur workers tend to make more mistakes about similar items, which means a choice needs to be made between m potential "local winners" at some stages. Fortunately, these m items appear in abutting positions in a preference. Therefore, I have $\sum_{r=1}^{m} \eta_w^r \approx 1$ and usually $m = 2$. (3) If worker w is a spammer, I have $\eta_w^1 \approx \eta_w^2 \approx \ldots \approx \eta_w^K$. Thus, the likelihood for each stage equals to some constant, which means COUPLE cannot distinguish the items and discard all the preferences $\mathscr{R}_w$ annotated by worker w. (4) Malicious workers intentionally select inferior items in each stage. COUPLE places more weight on $\eta_w^r (r > 1)$ instead of η_w^1 to correct the order of the items.

3.1.4 Optimization Difficulty of COUPLE model

In this subsection, I discuss the optimization difficulty of COUPLE model.

The proposed COUPLE model is formulated as a Maximum Likelihood Estimation (MLE) problem. The aim is simply to estimate the model parameters (θs and $\{\eta_w\}_{w=1}^{W}$) by maximizing Equation 3.4. In principle, any solution strategies for MLE can be used as a candidate to solve this problem. What actually makes this problem difficult or even intractable for traditional MLE solutions lies in the introduced latent variable $\{\eta_w\}_{w=1}^{W}$ (a.k.a. uncertainty vector) and the renormalization trick required in each stage. In the following, I leverage three examples [Bishop, 2006]: Coordinate Gradient Descent (CGD) algorithm (Common practice for MLE), Expectation Maximization (EM) algorithm (Common practice for MLE with latent variable) and Markov Chain Monte Carlo (MCMC) method (Common practice for MLE with complex formula) to intuitively illustrate the difficulty of the proposed model.

In terms of CGD, I need to calculate the first order partial derivatives of the log likelihood (Equation 3.4) w.r.t. the parameters θs and η_w, respectively. However, because the sum (integration over the latent variable η_w) is inside of the product of Equation 3.4, the partial derivatives w.r.t. the parameters θs and η_w become extremely complex. Further, since η_w is restricted to $[0,1]$, the box-constrained optimization would lead to an inaccurate and inefficient solution. Taking these two points into account, I shelved CGD and moved on to other possible candidates.

In terms of EM, it avoids calculating the derivative to the sum of latent variable directly, and instead resorts to a surrogate lower bound for optimization. Therefore, EM, a silver bullet for MLE with latent variable, seems a promising approach for Equation 3.4. However, due to the introduced renormalization trick for η_w in each stage, I still need to calculate derivative w.r.t. $\bar{\eta}_w$ instead of η_w directly. Therefore, the renormalization trick makes the derivatives w.r.t. η_w remain complex. Moreover, I still need to conduct box-constrained optimization to η_w over the feasible region $[0,1]$. In other words, EM does significantly simplify the optimization over parameter θs, but is still not able to complete the complex optimization over parameter η_w.

In terms of MCMC, it is a competitive candidate for parameter estimation, especially for complex model. By constructing a Markov chain that has the desired distribution as its equilibrium distribution (i.e. posterior distribution w.r.t. the model parameter θs and $\{\eta_w\}_{w=1}^{W}$), samples of the desired distribution can be obtained by observing the chain after a number of steps. Then I estimate the parameters of the posterior distribution by calculating a set of sufficient moments of the collected samples. According to the law of large numbers, the more samples collected, the more closely the moments of the sample should match the actual moments of desired distribution. However, due to the intrinsic properties of large-scale samples (large W and large N_w in Equation 3.4) and the high dimensionality of the parameters (large number of items, large number of crowd workers) in my problem, MCMC's sampling process would become extremely inefficient. Therefore, MCMC is not a good option for Equation 3.4.

The above difficulties prompted me to reject common practices and seek a tailor-made, but powerful, solution for the specific problem.

Bayesian moment matching (BMM) [Jaini et al., 2017] is a Bayesian approach used to estimate the model parameters. Specifically, it estimates the parameters of the approximate posterior by matching a set of sufficient moments of the exact complex posterior. Therefore, BMM can be viewed as an equivalent substitution of MCMC from the perspective of moment matching: BMM resorts to approximation to match the moments, while MCMC leverages the collected samples to match the moments. Under the independence assumption for

samples, BMM can be further extended to the sequential update strategy, OnlineBMM (see Figure 3.4). That is, the approximate posterior is updated after each sample instead of the whole dataset. Therefore, BMM has some inherent advantages over MCMC when dealing with large-scale datasets. In terms of the inefficiency of sampling-based methods (i.e. MCMC) for parameters with high dimensionality, the optimization-based methods (i.e. BMM) can naturally circumvent the curse of dimensionality. Further, based on the Bayesian theorem, BMM only needs to process the whole dataset once (see Figure 3.5) and can be updated for new samples online. These advantages prompted me to consider BMM as a basic framework for Equation 3.4. See Section 3.3 for more details.

3.2 Connection to related models

Connection to Plackett-Luce model: If worker w is an expert, then I have $\eta_w^1 \approx 1$, which means worker w selects the "local winner" in each stage with no hesitation. That is, in a general stage i of a k-ary preference ρ, I have $P(\tilde{X} = \rho^i|\theta,\eta_w) = \sum_{t=i}^{k} \bar{\eta}_w^{t-i+1} P(X = \rho^t|\theta) \approx P(X = \rho^i|\theta)$ for worker w. Therefore, COUPLE 3.4 degenerates into the vanilla Plackett-Luce model 3.1 when dealing with preferences from domain experts.

Connection to CrowdBT: CrowdBT extends the Bradley-Terry model to aggregate pairwise preferences by considering the quality of the worker [Chen et al., 2013]. The worker quality actually denotes the probability that worker w agrees with the ground truth; while COUPLE directly integrates worker quality into the Plackett-Luce model with an uncertainty vector η_w for each worker w. The uncertainty vector represents worker w's indecision about selecting the "local winner" in each stage. When COUPLE deals with pairwise preferences, I have $|\eta_w| = K = \max_{w,n} l_{\rho_{n,w}} \equiv 2$, $\forall w \in \{1,2,\ldots,W\}$ and $\forall n \in \{1,2,\ldots,N_w\}$. According to the definition of η_w in Section 3.1.2, η_w^1 represents the conditional probability that the item ranked first according to the worker's belief also accords with the ground truth. Therefore, η_w^1 also reveals the accuracy of worker w. Overall, COUPLE (Equation 3.4) degenerates into CrowdBT when dealing with pairwise preferences.

Connection to methods in [Raman and Joachims, 2014]: COUPLE focuses on modelling the human annotation process and aims to recover the ground truth from the noisy preferences. Whereas, the methods in Raman and Joachims [2014] try to reduce the negative impact of noisy preferences. In other words, COUPLE: trusts high-quality preferences from expert workers; recovers the ground truth for low-quality preferences from amateur or malicious workers; and reduces the negative impact of random preferences from spammer workers. The methods proposed by Raman and Joachims [2014] trust the high-quality preferences from expert workers just as COUPLE does, but indiscriminately reduce the impact of

low-quality preferences from non-expert workers. Therefore, benefiting from the fine-grained categorization of noisy workers, COUPLE can distil more useful information from the noisy preferences.

Connection to classical mixture models: COUPLE operates on the assumption of a single ground truth, where the ground truth preference is a unanimous global preference shared by the vast majority of workers. Therefore, workers whose preferences are consistent with the ground truth preference are classified as experts; otherwise they are classified as noisy workers. Although the heterogeneity is a very common phenomenon in human annotation data, the proportions of the mixture components are distributed quite unevenly among the annotations [Turner and Miller, 2012, Vitelli et al., 2014, Khare et al., 2015]. Usually, most workers agree with the major component, while the remaining few workers agree with one of the other minor components. If I adopt the classical mixture formulation, the model is easy to be underfitting for each minor component due to its insufficient number of supporting samples. In addition, most of the time, only the major component supported by the majority workers needs to be estimated. Therefore, I assume a single ground truth rather than the multiple ground truths assumed in the classical mixture formulation. Further, I introduce a worker-specific uncertainty vector to weaken the influence of the minor components. This vector identifies the minority workers as the noisy workers and eliminate their preferences during the aggregation process. Specifically, for a crowd worker w who agrees with the major component (the ground-truth), the first entry η_w^1 of his/her uncertainty vector is close to 1, which denotes he/she is an expert; while for a crowd worker w who agrees with one of the minor components, he/she would be classified as noisy worker, since η_w^1 does not dominate his/her uncertainty vector η_w. Section 3.1.3 contains some details on an even finer-grained categorization.

3.3 Online Bayesian moment matching for COUPLE

Bayesian moment matching [Jaini et al., 2017] is a scalable technique for estimating a model's parameters. It estimates the approximate posterior by matching a set of sufficient moments of the exact complex posterior after each observation. However, due to the non-conjugate likelihood function (Plackett-Luce model), the moments for score θ have no closed-form integrations. To address this issue, I combine COUPLE with a generalized Bayesian moment matching (GBMM) technique that helps to circumvent the need to compute some of the intractable moments. Based on the efficient posterior updating procedures, I propose the OnlineGBMM algorithm, which makes COUPLE scalable to large-scale datasets.

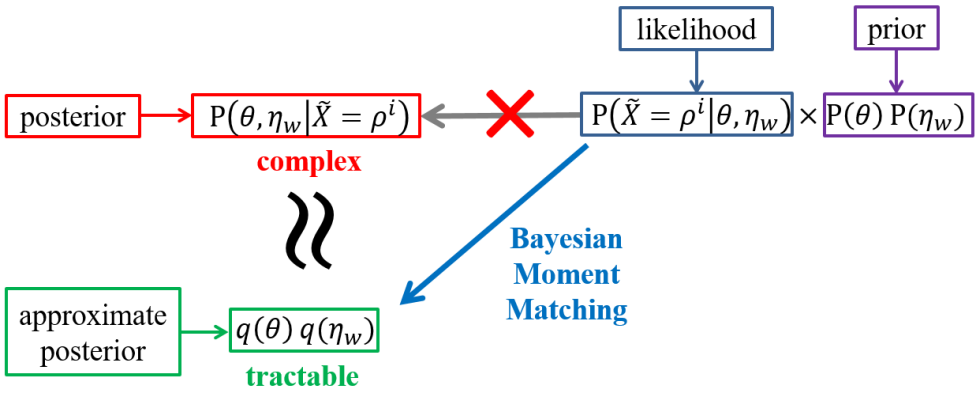


Figure 3.4 Bayesian Moment Matching: (1) define $q(\theta)q(\eta_w)$ in the same form with the prior; (2) match the moments between $q(\eta_w)$ and $P(\eta_w|\tilde{X} = \rho^i)$; (3) match the moments between $q(\theta)$ and $P(\theta|\tilde{X} = \rho^i)$.

3.3.1 Main routine of Bayesian Moment Matching(BMM)

As shown in Figure 3.4, I first extend COUPLE to its Bayesian version [3]. Specifically, I introduce a Normal prior $N(\theta_r|\mu_r, \sigma_r^2)$ for each score $\theta_r, r = 1, 2, \ldots, M$ and a Dirichlet prior $\text{Dir}(\eta_w|\alpha_w)$ for each uncertainty vector $\eta_w, w = 1, 2, \ldots, W$.

Benefiting from the stagewise learning strategy, I could decompose a crowdsourced preference ρ into a series of sequential stages, and update one stage instead of the entire preference each time. Generally, the likelihood function for a general stage i of preference ρ is $P(\tilde{X} = \rho^i|\theta, \eta_w)$ (See Equation 3.3). Accordingly, the posterior can be represented as follows,

$$P(\theta, \eta_w|\tilde{X} = \rho^i) = \frac{P(\tilde{X} = \rho^i|\theta, \eta_w)\prod_{r=1}^{M} N(\theta_r|\mu_r, \sigma_r^2)\text{Dir}(\eta_w|\alpha_w)}{P(\tilde{X} = \rho^i)}. \tag{3.5}$$

The main issue with Equation 3.5 is that the posterior $P(\theta, \eta_w|\tilde{X} = \rho^i)$ is hard to compute. To keep the computation tractable, I project the posterior into the same form with the prior (product of a Dirichlet with normals), by matching a set of sufficient moments of the approximate posterior with the exact posterior (See Figure 3.4):

1. Match the moments between $q(\eta_w)$ and $P(\eta_w|\tilde{X} = \rho^i)$. As η_w subjects to Dirichlet distribution, the approximate posterior $q(\eta_w)$ needs to satisfy the moment constraints: $\int \eta_w^t q(\eta_w)\,d\eta_w = \int \eta_w^t P(\eta_w|\tilde{X} = \rho^i)\,d\eta_w$ and $\int (\eta_w^t)^2 q(\eta_w)\,d\eta_w = \int (\eta_w^t)^2 P(\eta_w|\tilde{X} =$

[3] Here, I clarify that the Bayesian version of COUPLE is different from Thurstonian model [Maydeu-Olivares, 1999]. Although COUPLE and Thurstonian model all adopt the single ground-truth assumption, the hyperparameters σ_i^2 estimated by COUPLE are completely independent of workers, while Thurstonian model will learn a worker-specific variance $\sigma_{i,w}^2$ for each worker w.

$\rho^i)\, d\eta_w$, $t = 1,2,\ldots,K$. Fortunately, I can solve the constraints with closed-form integration [Rashwan et al., 2016], obtaining the posterior parameters $(\alpha_w)^{new}$ accordingly.

2. Match the moments between $q(\theta)$ and $P(\theta|\tilde{X} = \rho^i)$. As θ subjects to a normal distribution, a set of sufficient moment constraints are: $\mu = \int \theta q(\theta) d\theta = \int \theta P(\theta|\tilde{X} = \rho^i) d\theta$ and $\Sigma = \int (\theta - \mu)(\theta - \mu)^T q(\theta) d\theta = \int (\theta - \mu)(\theta - \mu)^T P(\theta|\tilde{X} = \rho^i) d\theta$.

However, due to the non-conjugacy between the likelihood[4] $P(\tilde{X} = \rho^i|\theta)$ (Equation 3.3) and the normal prior $\prod_{r=1}^{M} N(\theta_r|\mu_r, \sigma_r^2)$, the posterior $P(\theta|\tilde{X} = \rho^i)$ is too complex. Therefore, the posterior parameters $\{\mu_r^{new}, (\sigma_r^2)^{new}\}_{r=1}^{M}$ cannot be computed analytically, because the integrations in the moment constraints are intractable.

3.3.2 Generalized Bayesian Moment Matching (GBMM)

In cases with a non-conjugate likelihood with normal prior, I follow the strategy introduced by [Weng and Lin, 2011]. Weng and Lin [2011] proposed an efficient Bayesian approximation method based on Stein's Lemma [Woodroofe, 1989] to estimate the posterior parameters analytically.

Proposition 1 *Let $Z = (Z_1, Z_2, \ldots, Z_M)^T$, where $Z_r = \frac{\theta_r - \mu_r}{\sigma_r} \sim N(0,1)$, $r = 1,2,\ldots,M$. Assume $l(Z)$ is the likelihood $P(\tilde{X} = \rho^i|\theta)$ and almost twice differentiable. Upon the completion of stage i, the posterior parameters $(\mu_r^{new}, (\sigma_r^2)^{new})$ of score θ_r can be estimated as:*

$$\mu_r^{new} = \mu_r + \sigma_r E[\frac{\partial l(Z)/\partial Z_r}{l(Z)}], \tag{3.6a}$$

$$(\sigma_r^2)^{new} = \sigma_r^2 (1 + E[\frac{\partial^2 l(Z)/\partial^2 Z_r}{l(Z)}]_{rr} - E[\frac{\partial l(Z)/\partial Z_r}{l(Z)}]^2), \tag{3.6b}$$

where $r = 1,2,\ldots,M$.

According to the Bayesian approximation method introduced in Proposition 1, the posterior parameters $\{\mu_r^{new}, (\sigma_r^2)^{new}\}_{r=1}^{M}$ of the approximate posterior $q(\theta)$ can be estimated by a differential operation instead of an integral operation. Therefore, my GBMM can handle complex situations where the likelihood function is only required to be almost twice differentiable.

[4] $P(\tilde{X} = \rho^i|\theta) = E_{\mathrm{Dir}(\eta_w|\alpha_w)}\left[P(\tilde{X} = \rho^i|\theta, \eta_w)\right]$.

3.3.3 Posterior update

Given a crowdsourced k-ary preference ρ, I first decouple the complex likelihood into independent stages according to the stagewise learning strategy, and then updated hyperparameters in stages. In a general stage i, I first update the hyperparameters α_w, then updated the hyperparameters (μ, σ^2).

Quality update hyperparameters α_w

To update the hyperparameters α_w, I first integrate out θ to obtain the intermediate likelihood[5]

$$P(\tilde{X} = \rho^i|\eta_w) = E_{N(\theta|\mu,\sigma^2)}\left[P(\tilde{X} = \rho^i|\theta,\eta_w)\right] = \sum_{t=1}^{k-i+1}(\eta_w^t \times R_t),$$

where $R_t = E_{N(\theta|\mu,\sigma^2)}[e^{\theta_{\rho^{i+t-1}}} / \sum_{m=i}^{k} e^{\theta_{\rho^m}}]$. Note that in Bayesian framework, I do not directly conduct the renormalization on η_w, but rather choosed a tailor-designed prior distribution, which yields the same effect. Further, R_t can be calculated by its 2^{nd}-order Taylor approximation at μ.

Let $R = P(\tilde{X} = \rho^i) = E_{\mathrm{Dir}(\eta_w|\alpha_w)}\left[P(\tilde{X} = \rho^i|\eta_w)\right] = \sum_{t=1}^{k-i+1}\left(\frac{\alpha_w^t}{\sum_{m=1}^{k-i+1}\alpha_w^m} \times R_t\right)$ be the normalization constant, then the posterior distribution of η_w can be represented as:

$$P(\eta_w|\tilde{X} = \rho^i) = \frac{P(\tilde{X} = \rho^i|\eta_w)\mathrm{Dir}(\eta_w|\alpha_w)}{P(\tilde{X} = \rho^i)} = \frac{P(\tilde{X} = \rho^i|\eta_w)\mathrm{Dir}(\eta_w|\alpha_w)}{R}. \tag{3.7}$$

Note that I only need to calculate the moments of the first $k-i+1$ entries of α_w, because the intermediate likelihood $P(\tilde{X} = \rho^i|\eta_w)$ only depends on the first $k-i+1$ entries of η_w. According to Section 3.3.1, the sufficient moments $(E[\eta_w^t], E[(\eta_w^t)^2])$ for hyperparameter α_w^t can be calculated analytically [Bishop, 2006] as follows:

$$\begin{aligned} E[\eta_w^t] =& \frac{\alpha_w^t(\sum_{v=1}^{\tilde{k}}(R_v \times \alpha_w^v) + R_t)}{R(\sum_{v=1}^{\tilde{k}}\alpha_w^v + 1)(\sum_{v=1}^{\tilde{k}}\alpha_w^v)}, \\ E[(\eta_w^t)^2] =& \frac{\alpha_w^t(\alpha_w^t + 1)(\sum_{v=1}^{\tilde{k}}(R_v \times \alpha_w^v) + 2R_t)}{R(\sum_{v=1}^{\tilde{k}}\alpha_w^v + 2)(\sum_{v=1}^{\tilde{k}}\alpha_w^v + 1)(\sum_{v=1}^{\tilde{k}}\alpha_w^v)}, \end{aligned} \tag{3.8}$$

where $\tilde{k} = k-i+1$. The derivation can be found in the Appendix for the sake of completeness.

[5] "intermediate likelihood" denotes the corresponding likelihood with respect to a single stage instead of the whole preference or the whole dataset.

Then, I update the hyperparameter α_w^t of Dirichlet distribution $\mathrm{Dir}(\eta_w|\alpha_w)$ as follow:

$$(\alpha_w^t)^{new} = \frac{(E[\eta_w^t] - E[(\eta_w^t)^2])E[\eta_w^t]}{E[(\eta_w^t)^2] - (E[\eta_w^t])^2}, \tag{3.9}$$

where $t \in \{1,2,\ldots,k-i+1\}$.

Score update for hyperparameters (μ,σ^2)

To update the hyperparameters (μ,σ^2), I first integrat out η_w to obtain the intermediate likelihood

$$l(\theta) = E_{\mathrm{Dir}(\eta_w|\alpha_w)}\left[P(\tilde{X}=\rho^i|\theta,\eta_w)\right] = \frac{\sum_{r=i}^{k}(\alpha_w^{(r-i+1)} \times e^{\theta_{\rho^r}})}{(\sum_{v=1}^{k-i+1}\alpha_w^v) \times (\sum_{m=i}^{k} e^{\theta_{\rho^m}})}.$$

Note that only the moments of the scores, which are involved in the intermediate likelihood $l(\theta)$, will change during each stage update. Let $Z = z_{1:M}$, where $Z = \frac{\theta-\mu}{\sigma} \sim N(\mathbf{0},\mathbf{1})$. According to Equation 3.6a, I can directly calculate the posterior parameter $(\mu_{\rho^r})^{new}$ as follows:

$$(\mu_{\rho^r})^{new} \approx \mu_{\rho^r} + \sigma_{\rho^r}\frac{\partial l(Z)/\partial z_{\rho^r}}{l(Z)}\Big|_{Z=\mathbf{0}} = \mu_{\rho^r} + \sigma_{\rho^r}^2(\frac{\alpha_w^{(r-i+1)} \times e^{\mu_{\rho^r}}}{\Psi} - \frac{e^{\mu_{\rho^r}}}{\psi}), \tag{3.10}$$

where $r \in \{i,i+1,\ldots,k\}$, $\psi = \sum_{m=i}^{k} e^{\mu_{\rho^m}}$ and $\Psi = \sum_{m=i}^{k}(\alpha_w^{m-i+1} \times e^{\mu_{\rho^m}})$. I set $Z=\mathbf{0}$, so that θ is replaced by μ. Such an approximation is reasonable as I expect that the posterior density of θ to be concentrated on μ [Weng and Lin, 2011]. According to Equation 3.6b, I can directly estimate the posterior parameter $(\sigma_{\rho^r}^2)^{new}$ as follows:

$$\begin{aligned}(\sigma_{\rho^r}^2)^{new} &\approx \sigma_{\rho^r}^2(1 + \frac{\partial^2 l(Z)/\partial^2 z_{\rho^r}}{l(Z)}\Big|_{Z=\mathbf{0}} - (\frac{\partial l(Z)/\partial z_{\rho^r}}{l(Z)}\Big|_{Z=\mathbf{0}})^2) \\ &\approx \sigma_{\rho^r}^2 \max(1 + \sigma_{\rho^r}^2(\frac{\alpha_w^{(r-i+1)} \times e^{\mu_{\rho^r}}(\Psi - \alpha_w^{(r-i+1)} \times e^{\mu_{\rho^r}})}{\Psi^2} - \frac{e^{\mu_{\rho^r}}(\psi - e^{\mu_{\rho^r}})}{\psi^2}), \kappa),\end{aligned} \tag{3.11}$$

where $r \in \{i,i+1,\ldots,k\}$ and κ is a positive value to ensure a positive variance.

3.3.4 The OnlineGBMM algorithm

The OnlineGBMM for COUPLE model (Figure 3.5) is summarized in Algorithm 1 according to the above analysis. It is notable that the quality update and score update can both be completed with analytic solutions (Equation 3.9, 3.10, 3.11). As a result of the efficient

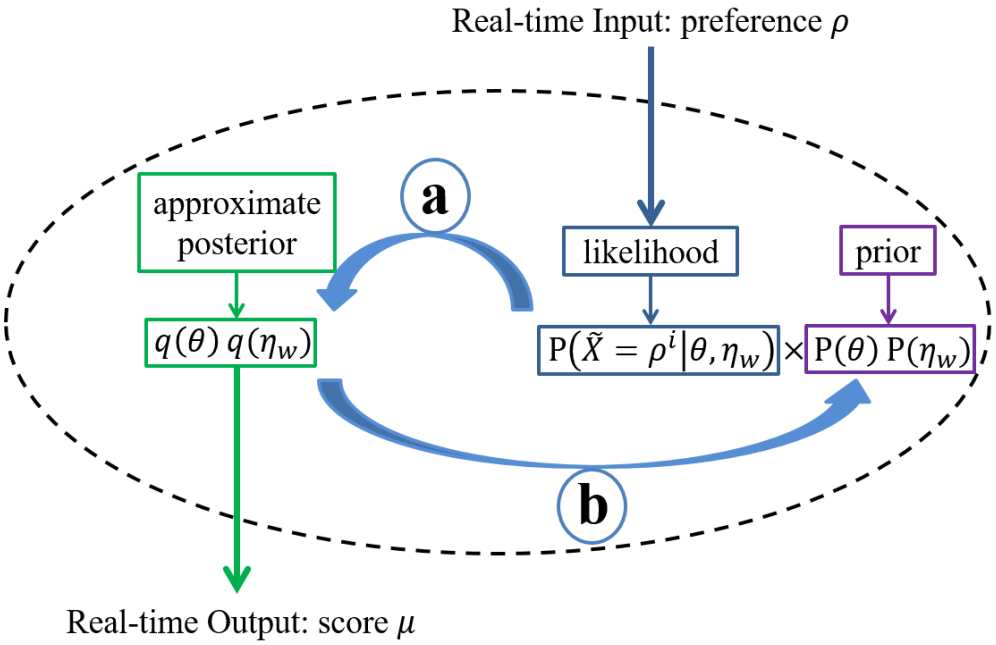


Figure 3.5 Online Generalized Bayesian Moment Matching (OnlineGBMM) for COUPLE: Step (a) estimate $q(\theta)q(\eta_w)$ with generalized Bayesian moment matching; Step (b) replace prior $P(\theta)P(\eta_w)$ with approximate posterior $q(\theta)q(\eta_w)$.

posterior updating procedure, OnlineGBMM allows COUPLE to inherently handle streaming preferences in real-time. Note that in Algorithm 1 I introduce a reverse update strategy during the preference updates.

Remark 3 Reverse Update Strategy: *Looking at Figure 3.3, in stage* 1*, I have k potential candidates for "local winner" for each k-ary preference, so the top k entries of the hyperparameter* α_w *need to be updated; with the updates of stages, in stage i, I have only* $k-i+1$ *potential candidates, therefore, only the top* $k-i+1$ *entries of* α_w *need to be updated. This means fewer entries of* α_w *will be updated in later stages as there are fewer candidates. To better propagate the update information, I propose the reverse update strategy, which update from the highest stage* $k-1$ *to the lowest stage* 1*. That is, the top two entries of* α_w *are updated in stage* $k-1$*. In stage i,* $k-i+1$ *entries need to be updated, so that the updating information from higher stages propagates to the fresh updated entries through the renormalization trick. In stage* 1*, all k items compete to become the "local winner", and the top k entries of* α_w *will be updated. Although the fresh active entry* α_w^k *is only updated once during the preference updates, it assembles the update information from all former stages through renormalizing.*

Algorithm 1 Online Generalized Bayesian Moment Matching (OnlineGBMM) for COUPLE

Initialization: Prior distribution parameters $\{\mu, \sigma^2, \{\alpha_w\}_{w=1}^{W}\}$.
Real-time Input: a k-ary preference ρ along with worker index w.
  for $stage = k-1, 2, \ldots, 1$ **do**
    Quality Update: α_w by Equation 3.9.
    Score Update: μ, σ^2 by Equation 3.10, 3.11.
  end
Real-time Output: Ranking items by sorting the obtained *mu*.

In Table 3.1, I compared the computation cost of COUPLE with three state-of-art online ranking models: (1) online Bradley-Terry (OnlineBT) [Weng and Lin, 2011]. (2) online Plackett-Luce (OnlinePL) [Weng and Lin, 2011]. (3) CrowdBT [Chen et al., 2013]. First, I broke each k-ary preference for the BT-based models into C_k^2 all possible pairwise preferences, then aggregate each pairwise preference independently. As all the methods are implemented with online learning, I only considered the computation cost of updating one k-ary preference. Note that I did not report the computation cost for PeerGrader [6], because it relies on SGD to estimate the model parameters and needs to process the whole dataset several times to converge. Therefore, as Bayesian methods only need to process the dataset once, SGD is inferior to the Bayesian online updating methods in terms of efficiency. Figure 3.7 provides empirical verification of this comparison for reference.

Table 3.1 computation cost of COUPLE and other models. Assume t_1 is the cost of extracting a pairwise preference from a k-ary preference and t_2 is the cost of completing an update in Algorithm 1.

	Split Number	Score Updated	Quality Updated	computation cost
OnlinePL	0	$2k$	0	$2kt_2$
COUPLE	0	$C_k^2 + k - 1$	$4C_k^2 + 5(k-1)$	$\big(5C_k^2 + 6(k-1)\big)t_2$
OnlineBT	C_k^2	2×2	0	$C_k^2 t_1 + 4C_k^2 t_2$
CrowdBT	C_k^2	2×2	7	$C_k^2 t_1 + 11C_k^2 t_2$

It is obvious in Table 3.1 that, of all the methods, CrowdBT has the largest computation cost to update one k-ary preference, while OnlinePL has the lowest computation cost. I further compared the time cost of COUPLE and other models on one real dataset (Section 3.4.5), and the empirical results are consistent with my analysis in Table 3.1.

[6] http://peergrading.org/

3.4 Experiments

In this section, I evaluated the reliability of COUPLE on four large-scale synthetic datasets, followed by experiments in two real-world applications - ordinal peer grading and online image-rating - to further verify the reliability of COUPLE in real-world situations.

3.4.1 Experiment setup

Datasets: I generated *synthetic datasets* similar to the method described in CrowdBT [Chen et al., 2013]. Assume that I have an item set $\mathcal{O} = \{o_1, o_2, \ldots, o_M\}$ with the ground-truth preference. Each task, composed of a subset selected randomly from $\mathcal{O}$, was corrupted by W crowd workers with different uncertainly vectors $\{\eta_w\}_{w=1}^{W}$ following a Dirichlet distribution $\mathrm{Dir}(\eta_w|\alpha_0)$. I controlled the quality of dataset by choosing the proper hyperparameters α_0 while retaining diversity among crowd workers.

To verify the reliability of COUPLE in *ordinal peer grading*, I used two *PeerGrading* datasets (PO = Poster, FR = Final Report) introduced by Raman and Joachims [2014]. They were collected as part of a senior-undergraduate and masters-level class. There are 42 assignments (items), 148 students (crowd workers) and 7 TAs participated in the PO dataset. The FR dataset contains 44 assignments (items), 153 students (crowd workers) and 9 TAs. More information can be found in [Raman and Joachims, 2014]. This size of class is appropriate, since it is large enough for collecting a substantial number of peer grades, meanwhile, it allows TA gradings to serve as the ground truth.

To further demonstrate the superiority of COUPLE in *online image-rating*, I built a facial image dataset (the *BabyFace* dataset) based on images of children's facial microexpressions with 18 levels from happy to angry. According to the crowdsourced k-ary preference setting, I divided 18 microexpressions into 816 distinct subsets, with each subset including three different microexpressions [7]. I submitted them to Amazon Mechanical Turk and collected the preferences from 105 crowd workers. I only considered workers who have at least 60 annotations, which yielded a collection of 3074 crowdsourced preferences annotated by 41 crowd workers. Further, I asked a further seven people to provide a credible unanimous (global) preference of the 18 microexpressions.

Baselines and Metrics: I compared COUPLE with three online rank aggregation models and two ordinal peer grading methods: (1) online Bradley-Terry (OnlineBT) [Weng and Lin, 2011]; (2) online Plackett-Luce (OnlinePL) [Weng and Lin, 2011]; (3) CrowdBT [Chen et al., 2013]; and (4)PeerGrader [Raman and Joachims, 2014]. I adapted the Kendall tau correlation (Equation 2.9) to evaluate the accuracy .

[7] I fixed the size of subsets for the convenience of comparing computation cost.

Parameter Initialization: I assigned a normal prior $N(0,1)$ for $\theta_i \ \forall i \in \{1,2,\ldots,M\}$ in all experiments. Inspired by [Chen et al., 2013], I initialized each hyperparameter α_w of uncertainty vector $\eta_w \ \forall w \in \{1,2,\ldots,W\}$ with 10 gold tasks with known ground-truth preferences in *synthetic simulations*. This method was also applied to the hyperparameter (α_w, β_w) of worker quality $\eta_w \ \forall w \in \{1,2,\ldots,W\}$ in CrowdBT. The parameter initialization for the *BabyFace* dataset was consistent with the method used for the synthetic simulations.

There is no access to the gold preferences for the *PeerGrading* datasets, as the average number of preferences annotated by each worker is too small (six for PO and two for FR). Fortunately, Raman and Joachims [2014] demonstrated that most students are high-quality (expert) workers with PO and FR datasets. According to my analysis in Section 3.1.2, η_w^i decreases exponentially with i for an expert worker w. Therefore, I assumed $\alpha_w = a_0 \times [a^{-1}a^{-2}\ldots a^{-K}]$, where $a_0 > 0$ and $a > 1$, resulting in $E[\eta_w^1] = \frac{1}{1+a^{-1}+\ldots+a^{-(K-1)}}$, where a large a denotes that worker w has a higher degree of confident when making decision.

In terms of the hyperparameter (α_w, β_w) for CrowdBT, I have $E[\eta_w] = \frac{\alpha_w}{\alpha_w+\beta_w}$ for worker w. That is to say, a large α_w represents highly accurate preferences annotated by worker w. For a fair comparison, I set $a_0 = 10, a = 6$ in COUPLE and $\alpha_w = 5, \beta_w = 1 \ \forall w \in \{1,2,\ldots,W\}$ in CrowdBT, namely $E[\eta_w^1] \approx 0.83$ and $E[\eta_w] \approx 0.83$ for all workers in COUPLE and CrowdBT, respectively. This parameter initialization is consistent with the assumption that most students are expert workers in *PeerGrading* datasets (PO and FR).

3.4.2 Empirical results on large-scale synthetic datasets

First, I investigated the reliability of COUPLE on large-scale synthetic datasets. According to the analysis in Section 3.1.2, the hyperparameter α_0 was set to $(5,1,0.1,0.01)$, $(5,4,1,0.1)$, $(5,4,3,3)$ and $(2,5,4,1)$ to simulate datasets from expert, amateur, spammer and malicious workers, respectively. COUPLE can be applied to large k. I set $k \leq 4$ for better controlling the characteristics of synthetic datasets. I set $L = 1,000$, $W = 500$, and assigned each worker $T = 900$ tasks. The amount of generated preferences reaches $W \times T = 450,000$. In addition, I ran the Algorithm 1 with a random sample sequence; the results are presented in Figure 3.6. Note that PeerGrader takes too much time cost to produce a result (Figure 3.7), so its accuracy on large-scale synthetic datasets could not be recorded.

Figure 3.6 shows that: (1) On all settings, COUPLE delivered a performance superior to other baselines; (2) On amateur, spammer and malicious settings, the advantage of COUPLE over CrowdBT became noticeable gradually, since COUPLE is able to correct mis-ordered items in noisy preferences in expectation while CrowdBT discard noisy preferences directly; (3) It is clear that all PL-based models showed minor improvements over corresponding BT-based models, since BT-based models must break each k-ary preferences into pairwise

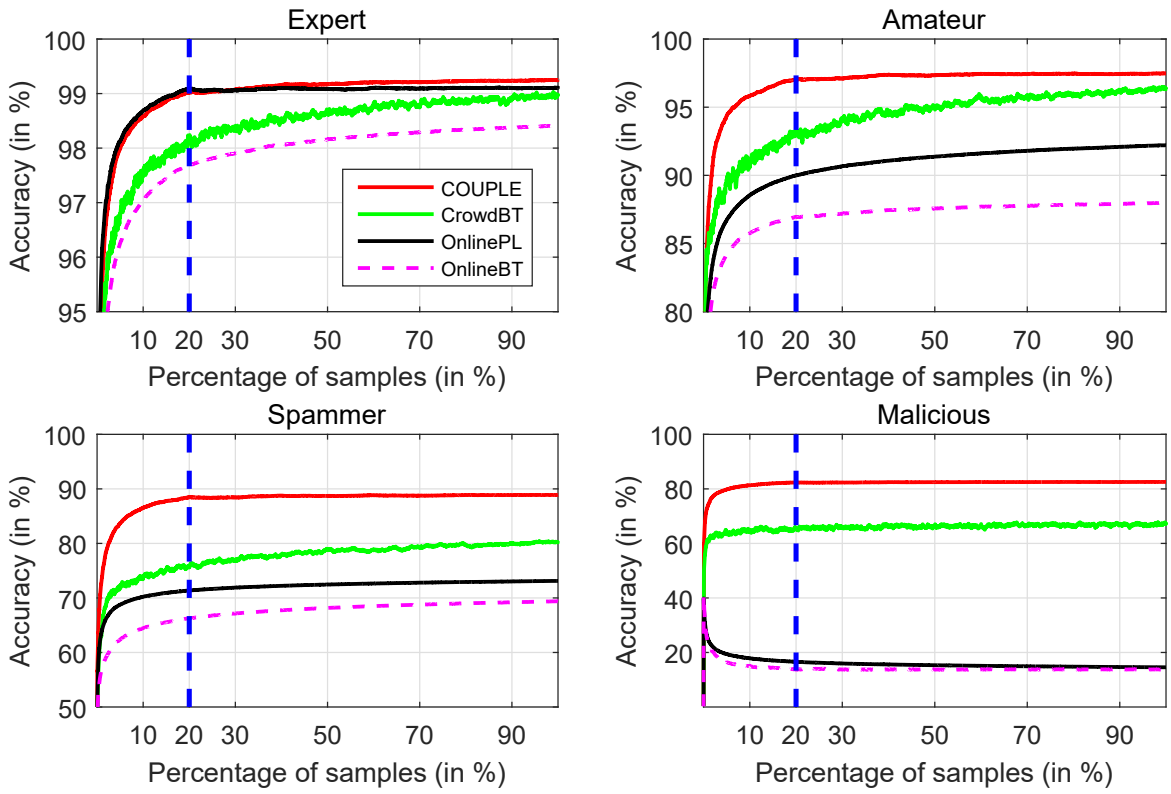


Figure 3.6 To verify the *reliability* of COUPLE preliminarily, the accuracy (%) with varying percentage of samples on large-scale synthetic datasets is provided.

preferences before aggregation, which may introduces some biases; and (4) The accuracy of COUPLE reached stability at 20% of the samples on all settings. Therefore, COUPLE is able to produce reliable results by aggregating incomplete dataset.

3.4.3 Empirical results in ordinal peer grading

In this section, I explored the reliability of COUPLE on *PeerGrading* datasets (PO and FR). First, I duplicated the real datasets ten times to reduce the adverse effects of other random factors and to ensure all models converge. Then, I ran the experiment 10^3 times to collect the results shown in Table 3.2.

Table 3.2 demonstrates that: (1) COUPLE showed obvious advantages over other baselines in terms of reliability; (2) COUPLE and CrowdBT consistently outperformed OnlinePL and OnlineBT, since they both consider worker quality; (3) The PL-based methods were more reliable than the BT-based methods because crowdsourced (noisy) preferences might magnify the effect of statistical inconsistencies, even though a full rank-breaking method was

Table 3.2 To verify the *reliability* of COUPLE in ordinal peer grading, the accuracy (%) on two ordinal peer grading datasets, i.e., PO and FR datasets, is provided. Accuracy is represented by a mean with standard deviation. As PeerGrader is SGD-based algorithms, I iterated PeerGrader until convergence and only measured the accuracy only once.

Dataset	**COUPLE**	CrowdBT	OnlinePL	OnlineBT	PeerGrader
PO	81.05 ± 0.99	78.03 ± 0.88	77.66 ± 1.89	73.64 ± 2.99	78.73
FR	78.73 ± 0.58	77.75 ± 0.32	76.68 ± 0.87	71.94 ± 0.91	70.35

used; and (4) PeerGrader was more accurate than OnlinePL and OnlineBT on the PO dataset, and even higher than CrowdBT. However, on the FR dataset, the PeerGrader's accuracy was inferior to OnlinePL and OnlineBT. Because PeerGrader focuses on the random noise, it cannot accurately model the annotation noise introduced by humans. Hence, it is reasonable that PeerGrader may fail in some real-world applications.

Noisy Worker Detection: According to the definition, COUPLE introduces an uncertainty vector η_w for each worker w, where $E\left[\eta_w^1\right] = \alpha_w^1 / \sum_{i=1}^K \alpha_w^i$ represents the probability that worker w selected the ground truth in each stage. Whereas CrowdBT introduces work quality η_w, which denotes the accuracy of the preferences annotated by worker w, where $E[\eta_w] = \frac{\alpha_w}{\alpha_w + \beta_w}$. Furthermore, PeerGrader also introduces a variable η_w, denoting the reliability of each crowd worker w (higher is better). Hereafter, I leveraged these three values as indicators to detect noisy workers. Table 3.3 lists the six lowest-quality workers detected by COUPLE, CrowdBT, and PeerGrader.

Table 3.3 Six workers with lowest work quality identified by COUPLE, CrowdBT, or PeerGrader on PO and FR datasets, respectively.

Dataset	**COUPLE**	**CrowdBT**	**PeerGrader**
Poster (PO)	$4, 75, 86, 103, 111, 124$	$30, 103, 118, 124, 157, 169$	$12, 52, 79, 103, 112, 124$
Final Report(FR)	$30, 57, 82, 87, 125, 131$	$1, 57, 125, 131, 134, 141$	$2, 20, 30, 57, 87, 125$

It is worth noting that it is impossible to assess the reliability of the noisy workers identified by the three models because no ground truths for these workers are available.

Next, I conducted a series of experiments to verify the efficacy of COUPLE, CrowdBT and PeerGrader in terms of noisy worker detection. For the PO dataset, I first removed the preferences annotated by six noisy workers with the lowest indicator value detected by COUPLE, CrowdBT and PeerGrader. Then, I ran the five models on the cleaned PO dataset again. This process was repeated for FR dataset. I repeated the experiment 10^3 times and

collected the results shown in Table 3.4. All parameters were consistent with the previous experiments on the PO and FR datasets.

Table 3.4 To verify the efficacy of COUPLE and CrowdBT on *noisy worker detection*, the accuracy (%) on cleaned PO and FR datasets is provided. The accuracy is represented by a mean with standard deviation. "PO(COUPLE)" denotes the cleaned PO dataset processed by COUPLE. This definition applies to other similar notations.

Cleaned Dataset	**COUPLE**	CrowdBT	OnlinePL	OnlineBT	PeerGrader
PO(COUPLE)	81.29 ± 0.53	79.77 ± 0.40	78.59 ± 0.97	77.18 ± 0.86	79.96
PO(CrowdBT)	81.10 ± 0.79	78.72 ± 0.73	78.10 ± 1.29	76.38 ± 0.98	80.49
PO(PeerGrader)	81.69 ± 0.65	81.82 ± 0.9	81.58 ± 0.75	81.29 ± 0.64	79.44
FR(COUPLE)	78.83 ± 0.39	78.60 ± 0.21	78.23 ± 0.46	74.38 ± 0.42	70.50
FR(CrowdBT)	78.76 ± 0.41	78.06 ± 0.30	77.34 ± 0.54	73.00 ± 0.44	70.94
FR(PeerGrader)	76.51 ± 0.27	76.03 ± 0.31	76.18 ± 0.72	72.38 ± 0.35	68.73

In comparing to results for PO dataset in Table 3.2 to the cleaned PO datasets in Table 3.4, I make the following observations: (1) The accuracy of COUPLE stabilized, but the accuracy of CrowdBT increased. This means that COUPLE is more reliable than CrowdBT for noisy preferences; (2) The standard deviations of the online models were lower because of the improved quality of the cleaned datasets; (3) The accuracy for CrowdBT, OnlinePL, and OnlineBT with the PO (COUPLE) dataset was higher than that with the PO (CrowdBT) dataset, respectively. This demonstrates that COUPLE is more reliable than CrowdBT for noisy worker detection; (4) Similar to the observations for the original PO dataset, PeerGrader achieved higher or comparable accuracy to CrowdBT on all cleaned PO datasets. However, PeerGrader's accuracy was inferior to OnlinePL and OnlineBT on all cleaned FR datasets. This observation further verifies my analysis in Section 3.2 that the series of methods introduced by Raman and Joachims [2014] cannot model the nature of human annotation noise well, and, therefore, those methods are likely to fail in real-world applications.

3.4.4 Empirical results in online image-rating

I further explored the reliability of COUPLE on the *BabyFace* dataset. Following the experiment on the *PeerGrading* datasets, I first duplicated the *BabyFace* dataset five times to reduce the adverse effects of random unknown factors. Then, I repeated the experiment 10^3 times and measured the accuracy of all models in terms of mean and standard deviation, as shown in Table 3.5.

Table 3.5 To verify the *reliability* of COUPLE in the real-world challenges, the accuracy (%) on the *BabyFace* dataset is provided. To verify the efficacy of COUPLE and other methods on *noisy worker detection*, the accuracy (%) on the Cleaned *BabyFace* dataset is provided. The accuracy is represented by the mean with the standard deviation. As PeerGrader is SGD-based algorithms, I iterated PeerGrader until convergence and collect the accuracy only once.

Dataset	**COUPLE**	CrowdBT	OnlinePL	OnlineBT	PeerGrader
BabyFace	92.49 ± 0.35	90.33 ± 0.33	88.24 ± 0.03	$88.24 \pm 3.48 \times 10^{-3}$	92.16
Cleaned *BabyFace*	92.30 ± 0.08	92.06 ± 0.61	92.80 ± 0.05	$92.10 \pm 1.20 \times 10^{-3}$	92.14

From the accuracy of the five models on the *BabyFace* dataset (the second line in Table 3.5), I observe that: (1) Although COUPLE delivered comparable accuracy to PeerGrader on the *BabyFace* dataset, COUPLE is more efficient because it relies on Bayesian analytical updating rules, while PeerGrader relies on gradient information for updates. (2) COUPLE, CrowdBT, and PeerGrader outperformed OnlinePL and OnlineBT, which demonstrates the superiority of the three models in term of modelling worker quality. Since OnlinePL and OnlineBT do not model the quality of crowd workers, their accuracies was easily affected by noisy preferences; and (3) it is interesting to note that the standard derivation of OnlineBT was smaller. Since BT-based methods resort to full rank-breaking to split each k-ary preference into C_k^2 pairwise preferences, OnlineBT generates more preferences and is more stable on small datasets. However, CrowdBT does not enjoy the benefit of rank-breaking; therefore, I conjecture that the noisy preferences would magnify the effect of statistical inconsistency, even with a consistent rank-breaking method.

Noisy Worker Detection: Following the procedure in Section 3.4.3, I leveraged the uncertainty vector η_w (COUPLE), worker quality η_w (CrowdBT) and worker reliability η_w (PeerGrader as indicators to identify the noisy workers in *BabyFace* dataset. The six lowest-quality workers, detected by COUPLE, CrowdBT, or PeerGrader, are presented in Table 3.6.

Table 3.6 Six workers with lowest work quality identified by COUPLE, CrowdBT, or PeerGrader on *BabyFace* datasets.

Dataset	**COUPLE**	**CrowdBT**	**PeerGrader**
BabyFace	$1, 2, 5, 12, 13, 17$	$1, 2, 5, 12, 13, 17$	$1, 2, 5, 12, 13, 17$

It is quite interesting that COUPLE, CrowdBT and PeerGrader detected the same six noisy workers for the *BabyFace* dataset. Similar to the setup on the *PeerGrading* datasets, I first removed the preferences annotated by these six noisy workers. Then, I ran the five

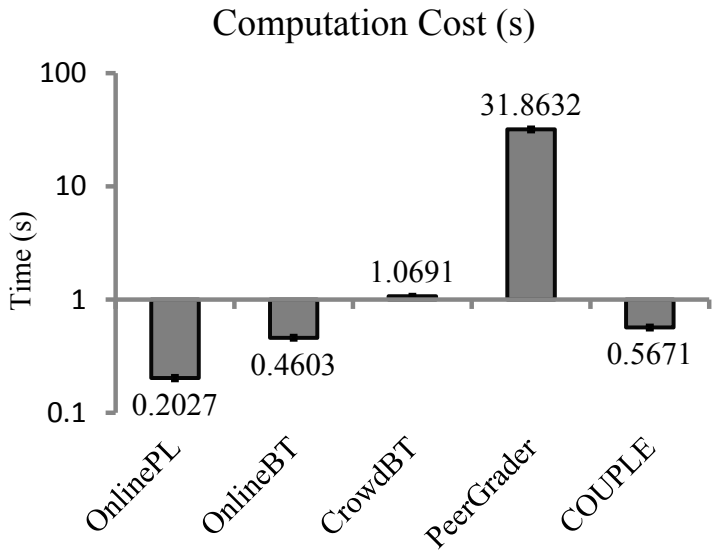


Figure 3.7 To verify the *complexity analysis* in Table 3.1, I collected the time cost of the four models when I conducted the experiment on the *BabyFace* dataset.

models on the cleaned *BabyFace* dataset again. I repeated the experiment 10^3 times and collected the results (in the bottom line of Table 3.5).

In comparing the results with the *BabyFace* dataset in Table 3.5, I observe that: (1) The accuracy of COUPLE stabilized at around 92.30%, which means that COUPLE is reliable for preferences provided by noisy workers; (2) The accuracy of OnlinePL, OnlineBT improved significantly on the cleaned *BabyFace* dataset. It means that the three models indeed detected the actual noisy (low-quality) workers in the dataset. They can be used to clean the noisy dataset by removing the preferences from the detected noisy workers; (3) The standard deviations of all online methods reduced as the quality of the dataset improved; and (4) it is notable that the accuracy of CrowdBT also improved on the cleaned *BabyFace* dataset. It is because CrowdBT tries to reduce the impact of low-quality preferences from crowd workers, while COUPLE aims to recover the ground the truth from noisy preferences. Therefore, COUPLE distils more useful information from the noisy preferences, resulting in higher accuracy with noisy rank aggregation.

3.4.5 Computation Cost

The *BabyFace* dataset is a satisfactory candidate for verifying the complexity analysis, as shown in Table 3.1. Since the length of the preferences in the *BabyFace* dataset is fixed to three, according to Table 3.1, the computation cost of processing each preference remains constant for COUPLE and other baselines. Hence, the time cost of all online models increase

linearly with the number of samples, which means that the proportional relations between the time cost of four models should be consistent with that in Table 3.1. I duplicated the *BabyFace* dataset five times and repeated the experiment 100 times to measure the time cost in terms of mean and standard deviation for all models, as shown in Figure 3.7. As PeerGrader is SGD-based algorithms, I set the iteration number to one and collect the time cost for fair comparison. Empirical results were implemented in Matlab (2015b) with an Intel i5 processor (2.30 GHz) and 8 GB random-access memory (RAM).

The theoretical analysis in Table 3.1 is consistent with my observations in Figure 3.7: (1) The computation cost of CrowdBT was much higher than that of other models, because CrowdBT needs to break each k-ary preferences into C_k^2 pairwise comparisons before aggregation; (2) The proportion of the time cost between COUPLE and OnlinePL was about 2.8, which is close to the theoretical result $17:6$ in Table 3.1 when $k=3$. Similar observations can be made of the other pairs; (3) The computation cost of COUPLE was higher than those of OnlinePL and OnlineBT because I introduced an uncertainly vector to model the worker quality. However, COUPLE was more reliable than all other baselines at the acceptable expense of greater computation cost; and (4) The time cost for PeerGrader was much higher than the online updating methods, because online methods update with simple Bayesian analytical updating rules, while PeerGrader uses the complex gradient to update the coupled parameters.

3.5 Summary of This Chapter

In this chapter, I outline a method to reliably aggregate large-scale noisy preferences annotated by crowd workers into one global preference using a reliable crowdsourced Plackett-Luce model, called (COUPLE) combined with an efficient Bayesian learning technique. To ensure reliability, an uncertainty vector in COUPLE recovers (in expectation) the ground truth from each worker's noisy preferences. An online Bayesian moment matching technique ensures that COUPLE scales naturally to large-scale preferences. Empirical results show that COUPLE combined with the OnlineGBMM algorithm delivers substantially more reliable results than current approaches. In future, I intend to extend this research in several ways. With active learning, different policies for COUPLE could be designed to select samples more wisely, so as to maximize gain against some criteria. Additionally, a theoretical analysis of OnlineGBMM's convergence rate and the approximation accuracy of GBMM would allow COUPLE to be applied to more complex situations.

Chapter 4

CoarsenRank: Rank Aggregation against Model Misspecification

In this chapter, I present a novel rank aggregation approach, called CoarsenRank. The main idea of CoarsenRank is to perform regular rank aggregation over a neighborhood of the collected inconsistent preferences, which enables CoarsenRank against noise agnostic perturbation within a neighborhood. To this end, I first define a neighborhood of the preference dataset using relative entropy. Then, I instantiate CoarsenRank with three popular probability ranking models and discuss the optimization strategies. In particular, a tractable closed-form solution is derived for Coarsened Bradley-Terry / Plackett-Luce model. Experiments on real-world datasets show that CoarsenRank is fast and robust, achieving consistent improvement over baseline methods.

4.1 Rank aggregation under model misspecification

In this section, I discuss RA under model misspecification. The term "model misspecification" here refers to the mismatch between the ranking model and the ranking dataset, namely the collected user preferences do not strictly satisfy the user homogeneity assumption of the ranking model.

4.1.1 Problem statement

Let $\mathscr{R}_N$ denote a collection of partial preferences $\{\rho_1, \rho_2, \ldots, \rho_N\}$ over the item set $\mathscr{O} = \{o_1, o_2, \ldots, o_M\}$. $\rho_n : \rho_n^1 > \rho_n^2 > \ldots > \rho_n^k$, where $\{\rho_n^1, \rho_n^2, \cdots, \rho_n^k\} \subseteq \mathscr{O}$. The goal of rank aggregation is then to aggregate the collected preferences $\mathscr{R}_N$ into a consensus order over

all items in $\mathscr{O}$. The consensus order should achieve the maximum agreement among all preferences.

In this section, I focus my work on the probability ranking model. Particularly, it assumes there exists a generative model P_o from which the preferences $\mathscr{R}_N$ are sampled, i.e., $\mathscr{R}_N = \{\rho_n | \rho_n \sim P_o, n = 1,2,\ldots,N\}$. However, the real data generation model P_o is hardly accessible due to the complexity of the real situation. For the sake of easier modeling, a parameterized rank model P_θ is usually adopted under the assumption of homogeneity of users. Let $\mathscr{P}$ be the set of all probability rank models under the homogeneity assumption. Then a maximum likelihood estimation (MLE) for RA can be formulated as follows,

$$\max_{P_\theta \in \mathscr{P}} P_\theta(\mathscr{R}_N), \qquad \text{where} \quad \mathscr{R}_N = \{\rho_n | \rho_n \sim P_o, n = 1,2,\ldots,N\}. \tag{4.1}$$

$P_\theta(\mathscr{R}_N) = \prod_{n=1}^{N} P_\theta(\rho_n)$ is the likelihood of the preferences $\mathscr{R}_N$. P_θ is usually instantiated with Thurstone model [Thurstone, 1927a,b], Bradley-Terry model [Bradley and Terry, 1952], Plackett-Luce model [Plackett, 1975, Luce, 1959], etc.

The model misspecification would arise when preferences were not strictly collected from a homogeneous user community due to the flexible data construction and the complex real situation (See Figure 4.1). For example, the reliability of each user would not be the same and the single total order assumption would be no longer satisfied. Therefore, a MLE for RA under model misspecification can be formulated as follows,

$$\max_{P_\theta \in \mathscr{P}, P_o \notin \mathscr{P}} P_\theta(\mathscr{R}_N), \qquad \text{where} \quad \mathscr{R}_N = \{\rho_n | \rho_n \sim P_o, n = 1,2,\ldots,N\}. \tag{4.2}$$

Here comes my robust rank aggregation against model misspecification, namely how to achieve a reliable total order from the collected preferences $\mathscr{R}_N$ using a misspecified ranking model P_θ. For the sake of explanation, let $\mathfrak{R}_N$ represent a virtual dataset $\{\varrho_n \mid \varrho_n \sim P_\theta, n = 1,2,\ldots,N\}$, which consists of idealized preferences and satisfies the homogeneity assumption. Then, RA under model misspecification can be formulated as noisy RA, where the collected preferences are viewed as a noisy perturbation of some idealized preferences.

4.1.2 Previous attempts: convolving the ranking model with specific perturbation mechanisms

When encountering model misspecification, a remedy solution for accessing a correct rank model could be

$$P_\theta(\rho_n) = \int P_\theta(\varrho_n) P(\varrho_n | \rho_n) d\ \varrho_n = \int P(\rho_n, \varrho_n) d\ \varrho_n, \quad \text{where } \rho_n \sim P_o,\ \varrho_n \sim P_\theta. \tag{4.3}$$

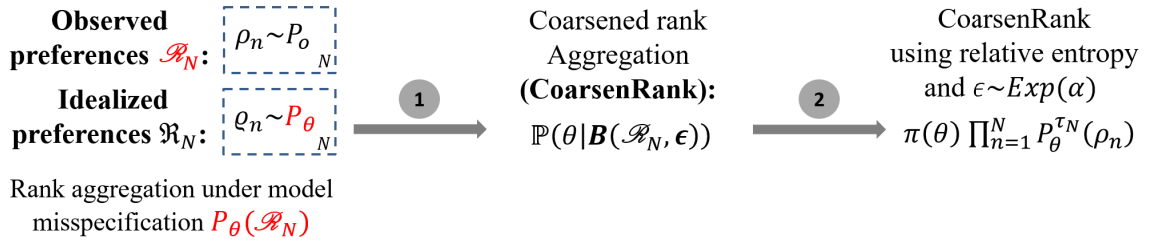


Figure 4.1 The logic stream of CoarsenRank. Condition 1: perform rank aggregation over a neighborhood of the collected preferences (See Equation 4.7 and 4.8). Condition 2: adopt relative entropy as the divergence measure and assign an exponential prior for the size of the neighborhood (See Corollaries 1 and 2).

Therefore, previous approaches usually resort to an augmentation of the ranking model to account for additional error/noise/uncertainty caused misspecification. For the sake of tractability, the perturbation mechanism is usually defined at the sample level. According to Equation 4.3, there are essentially two ways of implementing this:

- One intuitive approach is to correct each preference by pre-assuming some perturbation distribution, i.e., $P(\varrho_n|\rho_n)$. However, this simply amounts to convolving the original model distribution P_θ with the chosen perturbation distribution, leading to a new model that has a few more parameters but is just as bound to be misspecified w.r.t. other overlooked perturbations.

- The second approach would be to model the joint distribution $P(\rho_n, \varrho_n)$ directly, which needs to take into consideration all potential perturbations. Essentially, it needs to be a nonparametric model for $P(\rho_n, \varrho_n)$, but would easily be computationally intractable.

However, the perturbation patterns leading to model misspecification vary from setting to setting, it is impossible to design a universal practice that can be generalized to most settings. Therefore, in this chapter, I perform rank aggregation under model misspecification from another perspective.

4.1.3 CoarsenRank: rank aggregation over the neighborhood of the ranking data

Note that in many situations, it is impractical to correct the model, and these are the situations my method is intended to address. I am concerned with robust rank aggregation against model misspecification (Equation 4.2) in general, not just robust rank aggregation against one particular kind of perturbation considered in previous work. Motivated by the recent advances

of robust Bayesian inference [Miller and Dunson, 2018, Volpi et al., 2018], *inferring over the neighborhood of original dataset would equip the learning model with distributional robustness*.

In the following, I first give the definition of the neighborhood in the sense of ranking data,

Definition 1 (sample-level neighborhood) *Let $B(\mathscr{R}_N, \varepsilon)$ denote the neighborhood of the ranking dataset $\mathscr{R}_N$ with size ε.*

$$B(\mathscr{R}_N, \varepsilon) = \{\rho' | D(\rho', \rho) < \epsilon,\ \exists \rho \in \mathscr{R}_N\}. \tag{4.4}$$

where $D(\cdot,\cdot)$ denotes some distance measure between two ranking lists, e.g., Kendall tau distance [Kendall, 1938], Spearman's rank distance [Daniel, 1990].

Definition 2 (distribution-level neighborhood) *Let $B(\mathscr{R}_N, \varepsilon)$ denote the neighborhood of the ranking dataset $\mathscr{R}_N$ with size ε.*

$$B(\mathscr{R}_N, \varepsilon) = \{\mathscr{R}'_N | D(\mathscr{R}'_N, \mathscr{R}_N) < \epsilon\}. \tag{4.5}$$

where $D(\cdot,\cdot)$ denotes some distance measure between two ranking datasets. The distance measure between two datasets is usually defined as the divergence of their corresponding empirical distributions. Popular divergence measure between distributions are Kullback-Leibler (KL) divergence [Kullback and Leibler, 1951], f-divergence [Ali and Silvey, 1966] and Wasserstein metric [Villani, 2008].

Proposition 2 *For any ranking dataset $\mathscr{R}_N$, it must be a subset (element) of its sample-level (distribution-level) neighborhood. Namely*

$$\begin{aligned} &\mathscr{R}_N \subseteq B(\mathscr{R}_N, \varepsilon), && \textbf{\textit{sample-level neighborhood,}} \\ &\mathscr{R}_N \in B(\mathscr{R}_N, \varepsilon), && \textbf{\textit{distribution-level neighborhood.}} \end{aligned}$$

Proof: In terms of sample-level neighborhood, I have

$$\forall \rho_n \in \mathscr{R}_N, \quad \exists \rho* = \rho_n,\ \text{s.t.},\ D(\rho*, \rho) = 0 < \epsilon\ .$$

Then $\rho_n \in B(\mathscr{R}_N, \varepsilon)$ holds. Accordingly, $\mathscr{R}_N \subseteq B(\mathscr{R}_N, \varepsilon)$ holds according to Equation 4.4.

In terms of distribution-level neighborhood, I have

$$D(\mathscr{R}_N, \mathscr{R}_N) = 0 < \epsilon\ .$$

Therefore, $\mathscr{R}_N \in B(\mathscr{R}_N, \varepsilon)$ holds according to Equation 4.5.

Note that the proof is valid for any particular choice of the distance metric, either sample level or distribution level. ■

Definition 3 (empirical data distribution) *Let $F_N(\mathscr{R}_N)$ and $F_N(\mathfrak{R}_N)$ denote the empirical distributions of the ranking dataset $\mathscr{R}_N$ and $\mathfrak{R}_N$, respectively.*

$$\begin{aligned} F_N(\mathscr{R}_N) &= \frac{1}{N}\sum_{n=1}^{N} \delta_{\rho_n}(x), \\ F_N(\mathfrak{R}_N) &= \frac{1}{N}\sum_{n=1}^{N} \delta_{\varrho_n}(x), \end{aligned} \tag{4.6}$$

where $\delta_x(y)$ is the Dirac delta function, which is zero everywhere except at $x = y$, where it is infinite [Weisstein, 2004]. In this section, I assume that the empirical distribution converges to the corresponding data generating distribution, namely $F_N(\mathscr{R}_N) \to P_o$ and $F_N(\mathfrak{R}_N) \to P_\theta$ when $N \to +\infty$.

For the sake of brevity, I introduce my work following the definition of distribution-level neighborhood. In particular, I assume that *the idealized ranking dataset $\mathfrak{R}_N$ locates in nearby of the collected preferences $\mathscr{R}_N$, i.e., $\mathfrak{R}_N \in B(\mathscr{R}_N, \varepsilon)$*, then my Coarsened rank aggregation (CoarsenRank) can be formulated as follows,

$$\max_{\theta\in\Theta} P_\theta(\mathfrak{R}_N), \qquad \text{where} \quad \mathfrak{R}_N \in B(\mathscr{R}_N, \varepsilon). \tag{4.7}$$

Θ denotes the parameter space. Note that I use the word "Coarsen" to emphasize the learning paradigm that pursues the distributional robustness by inferring over the neighborhood of the dataset [Miller and Dunson, 2018].

Two equivalent (but compact) formulations of CoarsenRank (Equation 4.7) could be represented as follows,

$$\max_{\theta\in\Theta} \mathbb{P}(\theta | B(\mathscr{R}_N, \varepsilon)) \quad \Longleftrightarrow \quad \max_{\theta\in\Theta} \mathbb{P}(\theta | D(\mathscr{R}_N, \mathfrak{R}_N) < \epsilon). \tag{4.8}$$

where the equivalence between the two objectives in Equation 4.8 is according to the definition of distribution-level neighborhood.

As I can see from Equation 4.8, (1) CoarsenRank degenerates to vanilla rank aggregation method (Equation 4.1) when the collected preferences satisfy the homogeneity assumption; (2) CoarsenRank would be robust to noise agnostic perturbations as long as the idealized ranking dataset locates in nearby of the collected preferences when model misspecification

arises; and (3) CoarsenRank would fail to output a reliable total rank when the collected preferences significantly violate the homogeneity assumption. The same is true for other vanilla rank aggregation methods and most of robust RAs which fail to capture this perturbation. Therefore, compare with the previous methods, CoarsenRank is robust to most potential perturbations within a neighborhood, not only to some pre-assumed perturbations.

Remark 4 (The Coarsen mechanism VS. the distributional robustness) *The Coarsen mechanism* $max_{\theta\in\Theta}\mathbb{P}(\theta|B(X,\varepsilon))$ *shares a similar formula with the minimax distributional robustness* $min_{\theta\in\Theta}\max_{\mathscr{X}\in B(X,\varepsilon)}\sum_{\mathscr{X}}[\ell(\theta)]$ *[Sinha et al., 2017]. The Coarsen mechanism aims to maximize the likelihood over the neighborhood of original dataset X. The minimax distributional robustness aims to minimize the loss using the worst data samples in the neighborhood of original dataset X. The neighborhood in the Coarsen mechanism is usually defined at the distribution-level where a Bayesian criterion is adopted to estimate the proper size of the neighborhood for each dataset; while the neighborhood in the minimax distributional robustness is usually defined at the sample-level and a fixed size neighborhood is adopted once for all. The choice of distance measures* $D(\cdot,\cdot)$ *influences robustness guarantees and computability in both two methods.* ■

4.2 Coarsened rank aggregation

In this section, I first illustrate how CoarsenRank enables me to perform rank aggregation in a way that is robust to model misspecification. Meanwhile, a simplified formula is derived for CoarsenRank, which introduces only one extra hyperparameter to vanilla ranking models. Then, I instantiate CoarsenRank framework with three popular probability ranking models and discussed their optimization strategies, respectively.

4.2.1 Inferring over the neighborhood brings distributional robustness

Assuming that the empirical distribution defined in Equation 4.6 converges to the corresponding data generating distribution, namely $F_N(\mathscr{R}_N)\to P_o$ and $F_N(\mathfrak{R}_N)\to P_\theta$ when $N\to+\infty$, I come to Theorem 1. Note that this result is essentially Theorem 5.3 in [Miller and Dunson, 2018]. The derivation can be found in the Appendix for the sake of completeness.

Theorem 1 *Suppose* $D(\mathscr{R}_N,\mathfrak{R}_N)$ *is an almost surely (a.s.)-consistent estimator*[1] *of* $D(P_o,P_\theta)$*, namely* $D(\mathscr{R}_N,\mathfrak{R}_N)\xrightarrow[N\to+\infty]{a.s.}D(P_o,P_\theta)$*, where* $F_N(\mathscr{R}_N)\to P_o$ *and* $F_N(\mathfrak{R}_N)\to P_\theta$ *when* $N\to$

[1] In probability theory, an event happens almost surely (a.s.) if it happens with probability one.

$+\infty$. *Assume* $\mathbb{P}(D(P_o,P_\theta)=\epsilon)=0$ *and* $\mathbb{P}(D(P_o,P_\theta)<\epsilon)>0$*, then I have*

$$\mathbb{P}(\theta|D(\mathscr{R}_N,\mathfrak{R}_N)<\epsilon) \xrightarrow[N\to+\infty]{a.s.} \mathbb{P}(\theta|D(P_o,P_\theta)<\epsilon), \tag{4.9}$$

for any $\theta\in\Theta$ *such that* $\int|\theta|\mathbb{P}(d\theta)<\infty$.

Theorem 1 is a general conclusion in robust Bayesian inference [Miller and Dunson, 2018]. It justifies my motivation to pursue robustness in a distributional sense. In what follows, I extend Theorem 1 to some variants which possess nice properties for robust rank aggregation.

Level of distributional robustness

The value of the parameter ϵ is usually difficult to set without sufficient prior knowledge. I treat it as a random variable and introduce a prior on it. Then, I come to the following conclusion.

Corollary 1 *Assume* $\theta\sim\Pi(\theta)$*, the approximate posterior can be further simplified.*

$$\mathbb{P}(\theta|D(P_o,P_\theta)<\epsilon)=\frac{\Pi(\theta)\mathbb{P}(D(P_o,P_\theta)<\epsilon\,|\theta)}{\mathbb{P}(D(P_o,P_\theta)<\epsilon)}\propto\exp(-\alpha D(P_o,P_\theta))\Pi(\theta), \tag{4.10}$$

when random variable ϵ *subjects to an exponential prior, i.e.,* $\epsilon\sim\text{Exp}(\alpha)$.

Proof: Note that since $\epsilon\sim\text{Exp}(\alpha)$, I have

$$\mathbb{P}(D(P_o,P_\theta)<\epsilon\,|\theta)=1-\mathbb{P}(D(P_o,P_\theta)>\epsilon\,|\theta)=\exp(-\alpha D(P_o,P_\theta)).$$

where the second equation holds because the exponential prior is independent from θ. Substitute $\mathbb{P}(D(P_o,P_\theta)<\epsilon\,|\theta)$ in Equation 4.10 with $\exp(-\alpha D(P_o,P_\theta))$ and complete the proof. ∎

Indeed, a very large class of distributions can be adopted as prior $\pi(\alpha)$. A case of particular interest arises when $\epsilon\sim\text{Exp}(\alpha)$, since it leads to a computationally simple formula via maintaining an exponential formulation. The efficacy of the exponential prior is verified in the experiment (See Section 4.5).

Inspired by the exponential formulation of the posterior derived in Equation 4.10, I give the following derivations (Equation 4.11) to explain why the standard posterior is lack of

robustness.

$$\begin{aligned}\mathbb{P}(\theta|\mathscr{R}_N=\mathfrak{R}_N)=\frac{\Pi(\theta)P_\theta(\mathscr{R}_N)}{\int\Pi(\theta)P_\theta(\mathscr{R}_N)d\theta}&\propto\Pi(\theta)P_\theta(\mathscr{R}_N)=\Pi(\theta)\exp(\sum_{n=1}^{N}\log P_\theta(\rho_n))\\&\overset{i}{=}\Pi(\theta)\exp(N\int F_N(\mathscr{R}_N)\log P_\theta)\overset{ii}{\approx}\Pi(\theta)\exp(N\int P_o\log P_\theta)\\&\overset{iii}{\propto}\Pi(\theta)\exp(N\mathscr{D}_{\mathrm{KL}}(P_o\|P_\theta)).\end{aligned}\tag{4.11}$$

where *i* holds because the defination of the empirical data distribution $F_N(\mathscr{R}_N)=\frac{1}{N}\sum_{n=1}^{N}\delta_{\rho_n}(x)$. *ii* indicates Monte Carlo approximation. *iii* holds due to the omitting of the entropy term $\int P_o\log P_o$, which is a constant w.r.t. the model parameter θ. The standard posterior (Equation 4.11) diverges to infinity under model misspecification as $N\to+\infty$ but $P_o\neq P_\theta$, while the approximate posterior (Equation 4.10) remains stable.

Types of distributional robustness and tractability

The choice of $D(\cdot,\cdot)$ in $\mathbb{P}(\theta|D(P_o,P_\theta)<\epsilon)$ (Equation 4.9) affects both the richness of the robustness types I wish to cover as well as the tractability of the resulting optimization problem. The Wasserstein metric is a popular option in previous approaches on distributional robustness [Blanchet et al., 2016, Gao et al., 2017, Volpi et al., 2018], which exhibits super tolerance to adversarially corrupted outliers [Chen and Paschalidis, 2018a,b] and also allows robustness to unseen data [Abadeh et al., 2015, Sinha et al., 2017]. Meanwhile, [Ben-Tal et al., 2013, Namkoong and Duchi, 2017] adopted f-divergences in pursuit of tractable optimization approaches. Considering the particularity of rank aggregation task: (1) no generalization test is required for the RA task since I only need to aggregate the whole ranking dataset into one consensus full rank; (2) high complexity of the ranking model itself, I adopt relative entropy for $D(\cdot,\cdot)$, since it allows standard inference with no additional computational burden and helps to exhibit robustness to most types of perturbations.

Corollary 2 *If $D(\mathscr{R}_N,\mathfrak{R}_N)$ is an almost surely (a.s.)-consistent estimator of $\mathscr{D}_{\mathrm{KL}}(P_o\|P_\theta)=\int P_o\log\frac{P_o}{P_\theta}$, and ϵ is subject to an exponential prior, i.e., $\epsilon\sim\mathrm{Exp}(\alpha)$, I obtain the following approximation to the posterior:*

$$\mathbb{P}(\theta|D(\mathscr{R}_N,\mathfrak{R}_N)<\epsilon)\ \approx\ \Pi(\theta)\prod_{n=1}^{N}P_\theta^{\tau_N}(\rho_n),\tag{4.12}$$

where $\approx$ denotes that the term on the left is approximately equal to a term, which is proportional to the expression on the right, and $\tau_N=\frac{1/N}{1/N+1/\alpha}$.

Remark 5 (Connection between CoarsenRank and the standard posterior) *Because* $\epsilon \sim \text{Exp}(\alpha)$*, I have* $\mathbb{E}(\epsilon) = \frac{1}{\alpha}$ *denoting the expected discrepancy of the collected preferences* $\mathscr{R}_N$ *w.r.t.* $\mathfrak{R}_N$*. Further,* $\mathbb{E}(\epsilon)$ *approximates to zero as* $\alpha \to +\infty$*, which means the misspecification does not exist in the limit. Meanwhile, the robust posterior Equation 4.20 degenerates to the standard posterior as* $\tau_N = \frac{1/N}{1/N+1/\alpha}$ *approximates to* 1 *when* $\alpha \to +\infty$*.* ■

4.2.2 Coarsened probability ranking model

I derive a general robust rank aggregation framework in Equation 4.12, wherein P_θ could be instantiated with many probability ranking models.

Coarsened Thurstone model

I first review the basic Thurstone model [Thurstone, 1927b], which is a popular ranking model to model pairwise comparisons. Particularly, it assumes that the score $\theta_m \in R$ for each item o_m has a Gaussian distribution $N(\mu_m, \sigma_m^2)\ \forall m = 1, 2, \ldots, M$. For simplicity, I only consider the Case V Thurstone model with $\sigma_m = 1$ for all items. In particular, the comparison between any two items o_i and o_j also follows a Gaussian distribution $N(\mu_i - \mu_j, 2)$.

For a pair comparison $\rho_n : \rho_n^1 > \rho_n^2$, Thurstone model assumes

$$P_\theta(\rho_n) = \Phi\left(\frac{\triangle\theta_{\rho_n}}{\sqrt{2}}\right) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\frac{\triangle\theta_{\rho_n}}{\sqrt{2}}} \exp\left(-\frac{t^2}{2}\right) dt, \tag{4.13}$$

where $\triangle\theta_{\rho_n} = \theta_{\rho_n^1} - \theta_{\rho_n^2}$ denotes difference between the score of two items in ρ_n. Φ is the cumulative distribution function (CDF) of the standard normal distribution.

According to Corollary 2, an example of CoarsenRank (Equation 4.8) using Thurstone model can be represented as follow:

$$\text{Equation 4.8} \propto\!\!\!\!\!\!\sim\ \ \Pi(\theta) \prod_{n=1}^{N} P_\theta^{\tau_N}(\rho_n) = \frac{\Pi(\theta)}{(\sqrt{2\pi})^{N\tau_N}} \left[\prod_{n=1}^{N} \int_{-\infty}^{\frac{\triangle\theta_{\rho_n}}{\sqrt{2}}} \exp\left(-\frac{t^2}{2}\right) dt\right]^{\tau_N}, \tag{4.14}$$

where $\tau_N = \frac{1/N}{1/N+1/\alpha}$. Equation 4.14 can be only applied to pairwise comparisons. When encountering listwise preferences, I need to split each listwise preferences into pairwise comparisons and then perform CoarsenRank on the new dataset, alternatively [Khetan and Oh, 2016].

Remark 6 (Optimization intractability and my strategy) *The cumulative distribution function* Φ *is a special function, which cannot be expressed in terms of elementary functions.*

Therefore, it is inefficient or intractable to optimize Equation 4.8 directly. Inspired by the work which explores the connection between sigmoid function and Gaussian distribution [Weng and Lin, 2011], I consider approximating the cumulative distribution function Φ with the sigmoid function. In particular,

$$\Phi(\frac{\triangle\theta_{\rho_n}}{\sqrt{2}}) \approx \frac{1}{1+exp(-\lambda\triangle\theta_{\rho_n})}, \tag{4.15}$$

where λ is set as $2/\sqrt{\pi}$ so that the two probability curves have the same slope at $\triangle\theta_{\rho_n}=0$. A more accurate approximation with the second order moments guarantee can be found in Daunizeau [2017]. Then, Equation 4.14 can be further approximated as

$$\textit{Equation 4.14} \;\approx\; \frac{\Pi(\theta)}{(\sqrt{2\pi})^{N\tau_N}} \prod_{n=1}^{N} \frac{1}{\left[1+exp(-\lambda\triangle\theta_{\rho_n})\right]^{\tau_N}}, \tag{4.16}$$

where regular gradient-based optimization approaches could be carried out. ■

Coarsened Bradley-Terry model

A closely related model to the Thurstone model is the the Bradley-Terry (BT) model [Bradley and Terry, 1952]. For any pairwise comparison $\rho_n : \rho_n^1 > \rho_n^2$, the BT model assumes

$$P_\theta(\rho_n) = \frac{\theta_{\rho_n^1}}{\theta_{\rho_n^1}+\theta_{\rho_n^2}}, \tag{4.17}$$

where $\theta_m \in R_+^M$ is a positive support parameter for item o_m, $\forall m = 1,2,\ldots,M$.

According to Corollary 2, an example of my CoarsenRank (Equation 4.8) using BT model can be represented as follow:

$$\text{Equation 4.8} \;\propto\; \Pi(\theta)\prod_{n=1}^{N} P_\theta^{\tau_N}(\rho_n) = \Pi(\theta)\prod_{n=1}^{N}\left[\frac{\theta_{\rho_n^1}}{\theta_{\rho_n^1}+\theta_{\rho_n^2}}\right]^{\tau_N}, \tag{4.18}$$

where $\tau_N = \frac{1/N}{1/N+1/\alpha}$. Similar to Thurstone model, Equation 4.18 can only model pairwise comparisons. I still adopt the rank breaking strategy to split each listwise preferences into pairwise comparisons and perform my CoarsenRank on the new dataset [Khetan and Oh, 2016].

Remark 7 (Optimization intractability and my strategy) *The main inferential issue related to Equation 4.18 concerns the presence of the annoying normalization terms $\theta_{\rho_n^1}+\theta_{\rho_n^2}$, $\forall n = 1,2,\ldots,N$, that do not permit the direct maximization of the posterior. Further, the*

nonnegative constraint over the model parameters θ rules out the direct applications of gradient-based optimization approaches.

Motivated by Caron and Doucet [2012], I introduce the data augmentation trick to address the above-mentioned difficulty. Considering the fact that the Gumbel distribution is employed as a distribution of the support parameters and the conjugacy of the Gamma density with the Gumbel distribution, I follow Caron and Doucet [2012] and introduce an auxiliary Gamma random variable for each normalization term, which leads to a joint distribution without suffering from the annoying normalization terms. ■

Coarsened Plackett-Luce model

Here I instantiate P_θ with the popular Plackett-Luce (PL) model [Plackett, 1975, Luce, 1959]. Different from previous Thurstone mode and BT model, PL model is a more general probability ranking model, which could model listwise rankings of a finite set of items directly. Note that PL model incorporates BT model as a special case.

For a ranking list $\rho_n : \rho_n^1 > \rho_n^2 > \cdots > \rho_n^k$, the PL model assumes

$$P_\theta(\rho_n) = \prod_{i=1}^{k-1} \frac{\theta_{\rho_n^i}}{\theta_{\rho_n^i} + \theta_{\rho_n^{i+1}} + \cdots + \theta_{\rho_n^k}}, \tag{4.19}$$

where $\theta_m \in R_+^M$ is a positive support parameter associated with item o_m, $\forall m = 1,2,\ldots,M$. Comparing Equation 4.19 to Equation 4.17, PL model degenerates to BT model when modeling pairwise comparisons, i.e., $k \equiv 2$.

According to Corollary 2, an example of my CoarsenRank (Equation 4.8) instantiated using PL model can be represented as follow:

$$\text{Equation 4.8} \propto \Pi(\theta) \prod_{n=1}^{N} P_\theta^{\tau_N}(\rho_n) = \Pi(\theta) \prod_{n=1}^{N} \left[\prod_{i=1}^{k-1} \frac{\theta_{\rho_n^i}}{\theta_{\rho_n^i} + \theta_{\rho_n^{i+1}} + \cdots + \theta_{\rho_n^k}} \right]^{\tau_N}, \tag{4.20}$$

where $\tau_N = \frac{1/N}{1/N+1/\alpha}$. k denotes the length of each preference, which could be variant for different preferences.

Remark 8 (Optimization intractability and my strategy) *The same inferential issue arises for Equation 4.20. Following my analysis in Remark 7, I avoid this issue using the data augmentation trick. In particular, I introduce an auxiliary Gamma random variable for each normalization term $\theta_{\rho_n^i} + \theta_{\rho_n^{i+1}} + \cdots + \theta_{\rho_n^k}$, $\forall n = 1,2,\ldots,N, i = 1,2,\ldots,k-1$. Then, the resultant joint distribution would no longer suffers from the annoying normalization terms.* ■

4.2.3 CoarsenRank VS. the Mallows model

In this Chapter, I consider RA against model misspecification focusing on the score-based ranking model for the sake of tractability. The connection between my CoarsenRank and the permutation-based ranking model is also an interesting topic to explore. In particular, according to Equation 2.2, the likelihood function of Mallow's could be represented as

$$\begin{aligned} P(\mathscr{R}_N|r) &= \frac{1}{\psi^N(\sigma)} \prod_{n=1}^{N} \exp(-\sigma D(\rho_n, r)) = \frac{1}{\psi^N(\sigma)} \exp(-\sigma \sum_{n=1}^{N} D(\rho_n, r)) \\ &\overset{i}{=} \frac{1}{\psi^N(\sigma)} \exp(-\sigma D(\mathscr{R}_N, r)) \overset{ii}{\propto} \exp(-\sigma D(\mathscr{R}_N, r)), \end{aligned} \tag{4.21}$$

where $\sigma \in \mathbb{R}_+$ is a spread parameter, r is the reference permutation and $d(\rho, r)$ represents a distance between ρ and r. $\mathscr{R}_N = \{\rho_1, \rho_2, \ldots, \rho_N\}$ denotes the collected preferences. i is valid since we define $D(\mathscr{R}_N, r) = \sum_{n=1}^{N} D(\rho_n, r)$. ii holds due to the omission of the data non-relevant normalization term.

Permutation-based models are often impractical for the large-scale problem, because: (1) the normalization term $\psi(\alpha)$ usually requires high computational cost due to discrete distance computation; and (2) a maximum likelihood estimation involves an impossible discrete search for ranking over a large volume of items.

Remark 9 (Connection between CoarsenRank and MM) *Comparing Eq. 4.21 to our CoarsenRank formulation (Eq. 4.10), we can find that the difference between CoarsenRank and MM mainly lies in the definition of distance $D(\cdot,\cdot)$. Namely, distribution-level distance, i.e., $D_{\mathrm{KL}}(\cdot,\cdot)$ is adopted for CoarsenRank, while sample-level distance, e.g., Kendall tau distance, is adopted for MM. The superiority of CoarsenRank over MM lies in three aspects: (1) In terms of inference, an efficient inference strategy is discussed in the Remark after each variant of CoarsenRank, respectively; (2) In terms of explanation, our CoarsenRank formulation is derived from the Coarsen mechanism while assigning an exponential prior for the size of the neighborhood; (3) In terms of optimization w.r.t. the hyperparameter α, we avoid parameter turning by adopting data-driving strategy for choosing α, while Mallow's mode does not enjoy this convenience.*

4.3 Efficient Bayesian inference

In Section 4.2.2, I instantiate CoarsenRank framework (Equation 4.8) with three vanilla probability ranking models, i.e., Coarsened Thurstone model (Equation 4.14), Coarsened BT model (Equation 4.18), and Coarsened PL model (Equation 4.20). Meanwhile, I discuss the

optimization intractability and my optimization strategies w.r.t. to each model respectively. In the rest of this section, I focus on Coarsened PL model only and refer it as CoarsenRank for the following reasons: (1) Thurstone model and BT model are constrained to pairwise preferences, while the rank breaking strategy would lead to computational inefficient; (2) PL model can be applied to preferences with various length and incorporates BT model as a special case.

4.3.1 Data augmentation

According to the discussion in Remark 7, I resort to the data augmentation trick to eliminate the annoying normalization terms peculiar to the ranking model Equation 4.19, which helps to deduce an efficient inference method for Equation 4.20. First, I reformulate Equation 4.20 as follows:

$$\begin{aligned}\text{Equation 4.20} &= \prod_{n=1}^{N}\left[\prod_{i=1}^{k-1}\frac{\theta_{\rho_n^i}}{\theta_{\rho_n^i}+\theta_{\rho_n^{i+1}}+\cdots+\theta_{\rho_n^k}}\right]^{\tau_N} \\ &= \prod_{n=1}^{N}\prod_{i=1}^{k-1}\left[\frac{\theta_{\rho_n^i}}{\theta_{\rho_n^i}+\theta_{\rho_n^{i+1}}+\cdots+\theta_{\rho_n^k}}\right]^{\tau_N} = \prod_{n=1}^{N}\prod_{i=1}^{k-1}\left(\frac{\theta_{\rho_n^i}}{\eta_n^i}\right)^{\tau_N},\end{aligned} \tag{4.22}$$

where $\eta_n^i = \theta_{\rho_n^i}+\theta_{\rho_n^{i+1}}+\cdots+\theta_{\rho_n^k}$.

In terms of each normalization term η_n^i in Equation 4.22, I introduce an auxiliary variable ξ_n^i, $\forall n = 1,2,\ldots,N$ and $\forall i = 1,2,\ldots,k-1$. Let each η_n^i be subject to a Gamma distribution, i.e., $\mathrm{Gam}(\xi|p,q) = \frac{q}{\Gamma(p)}(q\xi)^{p-1}e^{-q\xi}$. Here $\Gamma(p)$ is the gamma function evaluated at p. More specifically, I define the posterior distribution of ξ_n^i as follows,

$$P(\xi_n^i|\rho_n,\theta) = \mathrm{Gam}(\xi_n^i|1,\eta_n^i) = \eta_n^i e^{-\xi_n^i\eta_n^i}. \tag{4.23}$$

Now, I can deal with the joint distribution directly, which leads to significant simplifications for optimization. Further, I utilize a Gamma prior to instantiate the prior distribution $\Pi(\theta)$, which naturally satisfies the nonnegative constraint of θ, i.e., $\theta \sim \mathrm{Gam}(\theta|a,b) = \prod_{m=1}^{M}\mathrm{Gam}(\theta_m|a_m,b_m)$). Therefore, the full likelihood of the CoarsenRank model (Equation 4.22) can be formulated as follows,

$$\begin{aligned}P(\mathscr{R}_N,\Xi,\theta,\epsilon\,|\{a_m,b_m\}_{m=1}^M,\alpha) &= \Pi(\theta)\prod_{n=1}^{N}\left(P_\theta(\rho_n)\prod_{i=1}^{k-1}P(\xi_n^i|\rho_n,\theta)\right)^{\tau_N} \\ &= \prod_{m=1}^{M}\mathrm{Gam}(\theta_m|a_m,b_m)\prod_{n=1}^{N}\prod_{i=1}^{k-1}\left(\theta_{\rho_n^i}\cdot e^{-\xi_n^i\eta_n^i}\right)^{\tau_N},\end{aligned} \tag{4.24}$$

Algorithm 2 Closed form EM for Coarsened rank aggregation (CoarsenRank)

1: **Input:** the collection of preferences $\mathscr{R}_N$.
2: **Initialization:** hyperparameters$\{a_m, b_m\}_{m=1}^M$ for θ.
3: **for** $t = 1, 2, \ldots, T$ **do**
4: *E-step:* calculate the posterior expectation of auxiliary variable ξ_n^i according to Equation 4.25.
5: *M-step:* update θ_m according to Equation 4.27 $\forall m = 1, 2, \ldots M$.
6: *C-step:* normalize θ, i.e., $\theta_m = C \cdot \theta_m / \sum_t \theta_t, \forall m = 1, 2, \ldots M$.
7: **end for**
8: **Output:** item score θ.

where $\tau_N = \frac{1/N}{1/N+1/\alpha}$. $\mathscr{R}_N = \{\rho_1, \rho_2, \ldots, \rho_n\}$ denotes the observed preferences. $\Xi = \{\xi_n^i\}$ denotes the introduced auxiliary variables. $\epsilon \sim \text{Exp}(\alpha)$ is the discrepancy between the collected preferences $\mathscr{R}_N$ and its idealized counterpart $\mathfrak{R}_N$, measured in relative entropy. (a_m, b_m) is initialized to (1, 2) $\forall m = 1, 2, ..., M$. I fix $\{a_m, b_m\}_{m=1}^M$ in this chapter to eliminate their coupling effects with other factors in CoarsenRank (Equation 4.24).

4.3.2 EM algorithm with closed-formed updating rules

Concerning the presence of the introduced auxiliary variables Ξ, I resort to the Expectation-Maximization (EM) framework, which is a silver bullet to compute the maximum-likelihood solution or maximum a posterior estimation in the presence of latent variables.

Expectation step (E-step) In the expectation step, I calculate the expectation of each auxiliary variable ξ_n^i w.r.t. its posterior distribution $P(\xi_n^i | \rho_n, \theta)$:

$$\mathbb{E}_{P(\xi_n^i|\rho_n,\theta)}[\xi_n^i] = \frac{1}{\theta_{\rho_n^i} + \theta_{\rho_n^{i+1}} + \cdots + \theta_{\rho_n^k}} = \frac{1}{\eta_n^i}. \tag{4.25}$$

where $n = 1, 2, \ldots, N$ and $i = 1, 2, \ldots, k-1$. Then, the expectation of the complete-data log-likelihood function w.r.t. the posterior of the introduced auxiliary variables Ξ can be represented as follows:

$$\begin{aligned} &\mathbb{E}_{P(\Xi|\mathscr{R}_N,\theta)} \left[\log P(\mathscr{R}_N, \Xi, \theta, \epsilon \,|\{a_m, b_m\}_{m=1}^M, \alpha)\right] \\ &= \sum_{m=1}^{M} \left[(a_m - 1)\log\theta_m - b_m\theta_m\right] + \sum_{n=1}^{N}\sum_{i=1}^{k-1} \left[\tau_N \log\theta_{\rho_n^i} - \tau_N \mathbb{E}[\xi_n^i]\eta_n^i\right] + \text{constant}. \end{aligned} \tag{4.26}$$

where $P(\Xi|\mathscr{R}_N, \theta) = \prod_{n=1}^N \prod_{i=1}^{k-1} P(\xi_n^i | \rho_n, \theta)$ following Equation 4.25.

Maximization step (M-step) In the maximization step, I maximize the objective function Equation 4.26 by setting its gradient w.r.t. θ_m to zero and obtain the following estimates for θ_m $\forall m = 1,2,\ldots,M$:

$$\theta_m = \frac{\tau_N \sum_{n=1}^{N} \sum_{i=1}^{k-1} (\varphi_{n,i}^m) + a_m - 1}{\tau_N \sum_{n=1}^{N} \sum_{i=1}^{k-1} (\psi_{n,i}^m \cdot \mathbb{E}[\xi_n^i]) + b_m}, \tag{4.27}$$

where $\varphi_{n,i}^m = \begin{cases} 1 & \rho_n^i = m \\ 0 & \text{otherwise} \end{cases}$ and $\psi_{n,i}^m = \begin{cases} 1 & m \in \{\rho_n^i, \ldots, \rho_n^k\}, \\ 0 & \text{otherwise.} \end{cases}$

Calibration for real application (C-step) In real applications, the number of items involved in partial comparisons usually varies significantly. Some items may appear frequently in the ranking list due to their popularity, while others only appear a few times due to their professionality. In such cases, the final ranking will not be unique or even not converge. To ensure a unique solution and to avoid overfitting, regularization may be used. To ensure the nonnegativity of the parameter θ, I perform normalization over θ. Namely, $\theta = C \cdot \theta / \sum_m \theta_m$. I fix $C = M/2$ in the experiment for simplicity.

Overall, the EM algorithm for Coarsened rank aggregation (CoarsenRank) is summarized in Algorithm 2.

4.3.3 Gibbs sampling for Coarsened rank aggregation

In CoarsenRank (Equation 4.24), there are two types of latent variables, i.e., Ξ and θ. According to the definition in Equation 4.23, the posterior distribution of ξ_n^i can be represented as

$$P(\xi_n^i | \rho_n, \theta) = \text{Gam}(\xi_n^i | 1, \eta_n^i) = \eta_n^i e^{-\xi_n^i \eta_n^i}, \tag{4.28}$$

where $n = 1,2,\ldots,N$ and $i = 1,2,\ldots,k$. Similarly, the full conditional distributions of θ_m $\forall m = 1,2,\ldots,M$ are still members of the Gamma family. According to Equation 4.27, the posterior distribution $P(\theta_m | \mathscr{R}_N, \Xi)$ can be represented as

$$P(\theta_m | \mathscr{R}_N, \Xi) = \text{Gam}\left(\theta_m \mid \tau_N \sum_{n=1}^{N} \sum_{i=1}^{k-1} (\varphi_{n,i}^m) + a_m - 1, \tau_N \sum_{n=1}^{N} \sum_{i=1}^{k-1} (\psi_{n,i}^m \cdot \mathbb{E}[\xi_n^i]) + b_m\right). \tag{4.29}$$

where $\varphi_{n,i}^m = \begin{cases} 1 & \rho_n^i = m \\ 0 & \text{otherwise} \end{cases}$ and $\psi_{n,i}^m = \begin{cases} 1 & m \in \{\rho_n^i, \ldots, \rho_n^k\}, \\ 0 & \text{otherwise.} \end{cases}$

Algorithm 3 Gibbs Sampling for Coarsened rank aggregation (CoarsenRank)

1: **Input:** the collection of preferences $\mathscr{R}_N$.
2: **Initialization:** hyperparameters $\{a_m, b_m\}_{m=1}^M$ for θ.
3: **for** $t = 1, 2, \ldots, T$ **do**
4: sample ξ_n^i from $P(\xi_n^i | \rho_n, \theta)$ according to Equation 4.28, $\forall n = 1, 2, \ldots, N$, $\forall i = 1, 2, \ldots, k$.
5: sample θ_m from $P(\theta_m | \mathscr{R}_N, \Xi)$ according to Equation 4.29, $\forall m = 1, 2, \ldots, M$.
6: **end for**
7: **Output:** item score $\theta_{M \times T}$.

Therefore, the Gibbs sampling procedure for Coarsened rank aggregation (CoarsenRank) can be summarized in Algorithm 3.

4.3.4 A data-driven strategy for choosing α

Regarding the hyperparameter optimization, I focus on exploring the effects of the hyperparameter α in this section. Other hyperparameters were not further explored in the experiment since these are not my focus in this section.

Regarding the hyperparameter α, I have no prior basis for choosing parameter α in Equation 4.10. Therefore, the following diagnostic curve can help to make a data-driven choice. Let $f(\alpha)$ be a measure of fit to the data and $g(\alpha)$ be a measure of model complexity. Following [Spiegelhalter et al., 1998], I use the posterior expected log-likelihood for $f(\alpha)$, and the difference between the log-likelihood evaluated at the posterior mean of the parameters and the posterior expected log-likelihood for $g(\alpha)$. Specifically, I define

$$f(\alpha) = \mathbb{E}_{q_\alpha(\theta|\mathscr{R}_N)}[\log P_\theta(\mathscr{R}_N)] \quad \text{and} \quad g(\alpha) = \log P_{\mathbb{E}(\theta)}(\mathscr{R}_N) - f(\alpha), \tag{4.30}$$

where $q_\alpha(\theta|\mathscr{R}_N)$ is an approximate posterior distribution for θ and $\mathbb{E}(\theta) = \mathbb{E}_{q_\alpha(\theta|\mathscr{R}_N)}[\theta]$ is the posterior expectation of θ.

Therefore, the adopted Deviance Information Criterion (DIC) [Spiegelhalter et al., 1998] can be represented as

$$\text{DIC} = g(\alpha) - f(\alpha). \tag{4.31}$$

As α ranges from 0 to $+\infty$, DIC traces out a curve in $\mathbb{R}$, and the technique is to choose α with the lowest DIC or where DIC levels off.

Table 4.1 Comparison between various ranking models in terms of their noisy assumption and ranking model assumption. "—" denotes the assumption does not apply to a particular ranking model.

Baselines		**Distance measure**	**Noisy data assumption**		**Ranking model assumption**		
		distribution or sample level?	perturbation $P(\rho_n \mid \varrho_n)$	neighborhood size ϵ	model P_θ	prior $\Pi(\theta)$	posterior $P(\xi_n^i \mid \rho_n, \theta)$
Vanilla	TH	distribution	—	—	Thurstone	Gaussian	—
	BT	distribution	—	—	Bradley-Terry	Gamma	Gamma
	PL	distribution	—	—	Plackett-Luce	Gamma	Gamma
Robust	MM	sample	fractional likelihood	—	—	—	—
	CrowdBT	distribution	Dawid-Skene model	—	BT/TH	Gaussian	—
	ROPAL	distribution	Dawid-Skene model	—	PL	Gaussian	—
	PeerGrader	distribution	fractional likelihood	—	BT/PL/TH	Gaussian	—
Coarsen	CoarsenTH	distribution	—	exponential prior	TH	Gaussian	—
	CoarsenBT	distribution	—	exponential prior	BT	Gamma	Gamma
	CoarsenPL	distribution	—	exponential prior	PL	Gamma	Gamma

4.4 Noisy assumption and rank model assumption

In this section, we summarize the differences between our CoarsenRank and other baselines in terms of distance measure, noisy data assumption and ranking model assumption. A detailed comparison is listed in Table 4.1.

Specifically, we classify the rank aggregation methods into three categories: Vanilla RA, Robust RA, and Coarsen RA. Vanilla RA, such as Thurstone model [Thurstone, 1927b] Bradley-Terry (BT) model [Bradley and Terry, 1952] Plackett-Luce (PL) model [Plackett, 1975, Luce, 1959], assumes all ranking lists are generated from the same distribution, under some parameterized ranking model, e.g., TH, BT, and PL. Further, prior assumptions are usually introduced for the parameter to avoid overfitting.

The formulation of MM is similar to the fractional likelihood [Bhattacharya et al., 2019] where the tunable spread parameter in MM has the same function as the exponential variable in a fractional likelihood. This is the reason we classify MM as a Robust RA. See section 4.2.3 for more detailed discussion.

Robust RA extends vanilla RA by considering extra noisy perturbations incurred during the data collection. Representative robust RAs are CorwdBT [Chen et al., 2013] and ROPAL [Han et al., 2018], which recover the clean ranking lists by some predefined perturbation mechanisms, following the Dawid-Skene model. Meanwhile, PeerGrader [Raman and Joachims, 2014] enhances the robustness of vanilla RA following the principle of fractional likelihood. In particular, PeerGrader introduces an extra parameter for each user, which

would decrease the effects of noisy ranking lists from an unreliable user. Above Robust RAs aim at capturing the corruption pattern w.r.t. to each user, which requires sufficient available preferences for each user. This constraint is too strong while only one preference from each user is usually available in real application.

Our CoarsenRank is motivated by the Coarsen mechanism, where we perform regular RA over the neighborhood of original ranking lists. We only introduce one extra parameter, representing the size of the neighborhood. A simple formulation of CoarsenRank is further derived if we adopt relative entropy as the distance measure and assign an exponential prior for the unknown neighborhood size. Three variants of CoarsenRank are introduced by instantiating with the vanilla TH, BT and PL model, and closed-form EM solution could be derived under the same assumption as the vanilla RAs.

4.5 Experimental evaluation

In this section, I verify the efficacy of the proposed CoarsenRank algorithm on noisy rank aggregation with the start-of-the-art approaches. The results are carried on four real-world noisy ranking datasets.

4.5.1 Experimental setting

Performance metric: Regarding the performance metric, I considered the Kendall tau correlation, which is one of the most common measures of similarity between rankings, namely

$$S(\sigma_1,\sigma_2) = 1 - \frac{1}{\overline{M}}\sum_{i<j}(\mathbb{1}[(\sigma_1^i > \sigma_1^j) \wedge (\sigma_2^i < \sigma_2^j)] + \mathbb{1}[(\sigma_1^i < \sigma_1^j) \wedge (\sigma_2^i > \sigma_2^j)]).$$

$S(\sigma_1,\sigma_2)$ counts the pairwise agreements between items from two rankings σ_1 and σ_2. $\overline{M} = \frac{1}{2}M(M-1)$ denotes total number of pairs. $S(\sigma_1,\sigma_2)$ will be equal to 1 if the two ranking lists are identical and 0 if one list is the reverse of the other.

Baselines: As for baselines, I first considered the vanilla Plackett-Luce model [Plackett, 1975, Luce, 1959]. For the sake of fair comparison, I optimized it with two optimization approaches, i.e., gradient descent (PL) [Boyd and Vandenberghe, 2004] and EM using data augmentation (PL-EM) [Caron and Doucet, 2012]. Then, I compared the results with PeerGrader [Raman and Joachims, 2014], which is a variation of the Plackett-Luce model for partial preferences while incorporating the user reliability estimation module. I also compared

with the popular noisy ranking model CrowdBT [Chen et al., 2013]. Since CrowdBT was originally designed for pairwise preferences, I generalized CrowdBT to partial preferences following rank-breaking [Negahban et al., 2016]. Namely, I first broke each partial preference into a set of pairwise comparisons and then applied CrowdBT to each pairwise comparison independently. Note that ROPAL requires more ground truth preferences from each user for initializing the parameters, compared to CrowdBT. It constraints ROPAL to a crowdsourcing setting, where multiple preferences provided by each user are available. Therefore, we do not consider comparing with ROPAL here, since we are considering a more general setting in this chapter.

Calibration for baselines: Similar to CoarsenRank, I proposed to calibrate the baseline to avoid overfitting and guarantee a unique solution. In terms of PL-EM, I adopted the same calibration method as CoarsenRank. In terms of other baselines whose formulation is a little different from mine, my calibration method cannot be applied. Following CrowdBT, I used virtual node regularization [Chen et al., 2013]. Specifically, it augments the original dataset $\mathscr{R}_N$ with $\mathscr{R}_o$, which consists of the pairwise comparisons between all items and a virtual item θ_0, namely $\mathscr{R}_o = \{\theta_m > \theta_0, \theta_m < \theta_0, m = 1,2,\ldots,M\}$.

4.5.2 Detailed descriptions of datasets

I conducted my experiment on four real-world datasets introduced in previous research. The detailed descriptions of the datasets are introduced in the following.

The *Readlevel* dataset [Chen et al., 2013] contains English text excerpts whose reading difficulty level is annotated by workers from a crowdsourcing platform. This dataset consists of 490 excerpts from 624 workers, resulting in a total of $12,728$ pairwise comparisons. A total order for 490 excerpts provided by the domain expert is regarded as ground truth. The number of annotation varies significantly for different works, ranging from 1 to 42.

The *SUSHI* dataset is introduced in [Kamishima, 2003], which consists of partial preferences over 100 types of sushi from $n = 5,000$ customers. Following [Khetan and Oh, 2016], I generated the total order as ground truth using the vanilla PL over the entire $5,000$ preferences. To create training data, I randomly replaced $2,000$ preferences in the original *SUSHI* dataset with another $2,000$ random generated preferences. The number of preference provided by each customer is fixed to one.

The *BabyFace* dataset [Han et al., 2018] consists of the evaluations of workers from Amazon Mechanical Turk on images of children's facial microexpressions from happy to angry, which yields a collection of $3,074$ trinary preferences from 41 workers. A total order

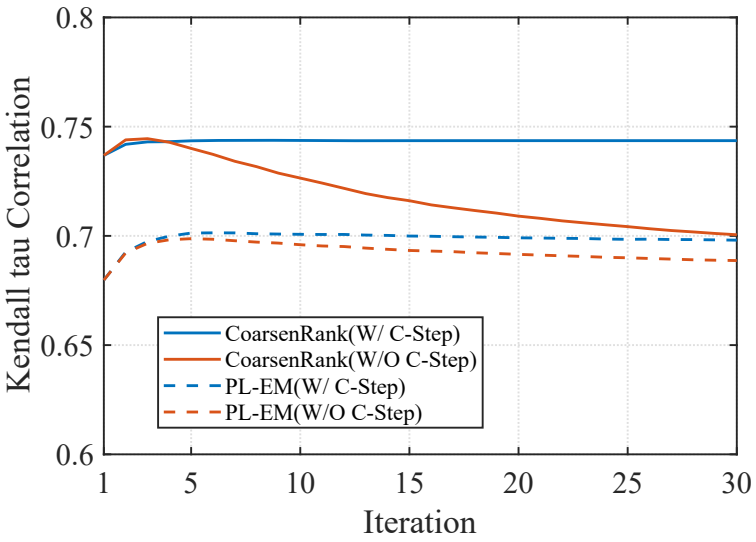


Figure 4.2 The performance comparison of CoarsenRank and PL-EM algorithm under the case of with or without calibration step. "W/" denotes "with" while "W/O" denotes "without".

over all microexpressions is provided as ground truth by the agreement of most workers after the experiment. Each worker provides at least 60 annotations.

The *PeerGrading* dataset [Sajjadi et al., 2016] consists of assessments, i.e., Self grading and Peer grading, from 219 students over 79 group submissions. I then created the ordinal gradings by merging the Self grading and Peer grading regarding the same assignment provided by each student, which results in a total of $3,619$ preferences with each containing $2-3$ items. Further, the TA gradings (following a linear order) provided by six teaching assistants over all submissions are considered as ground truth. The number of annotation from different students ranges from 2 to 26.

The statistics of four datasets are summarized in Table 4.2.

Table 4.2 The statistics of four real ranking datasets

Dataset	#items (M)	#users	#preferences (N)	length of preferences (k)	#annotations per user
Readlevel	490	624	$12,728$	2	$1-42$
SUSHI	100	$5,000$	$5,000$	10	1
BabyFace	18	41	$3,074$	3	≥ 60
PeerGrading	219	79	$3,619$	$2/3$	$2-26$

4.5.3 Exploring the efficacy of the calibration step

In section 4.3.2, I introduced a calibration step to the vanilla EM algorithm to avoid overfitting. I claim that the calibration step is necessary when the collected preferences do not evenly cover the items. In particular, the calibration step serves as regularization, which is helpful to avoid overfitting and leads to a unique solution.

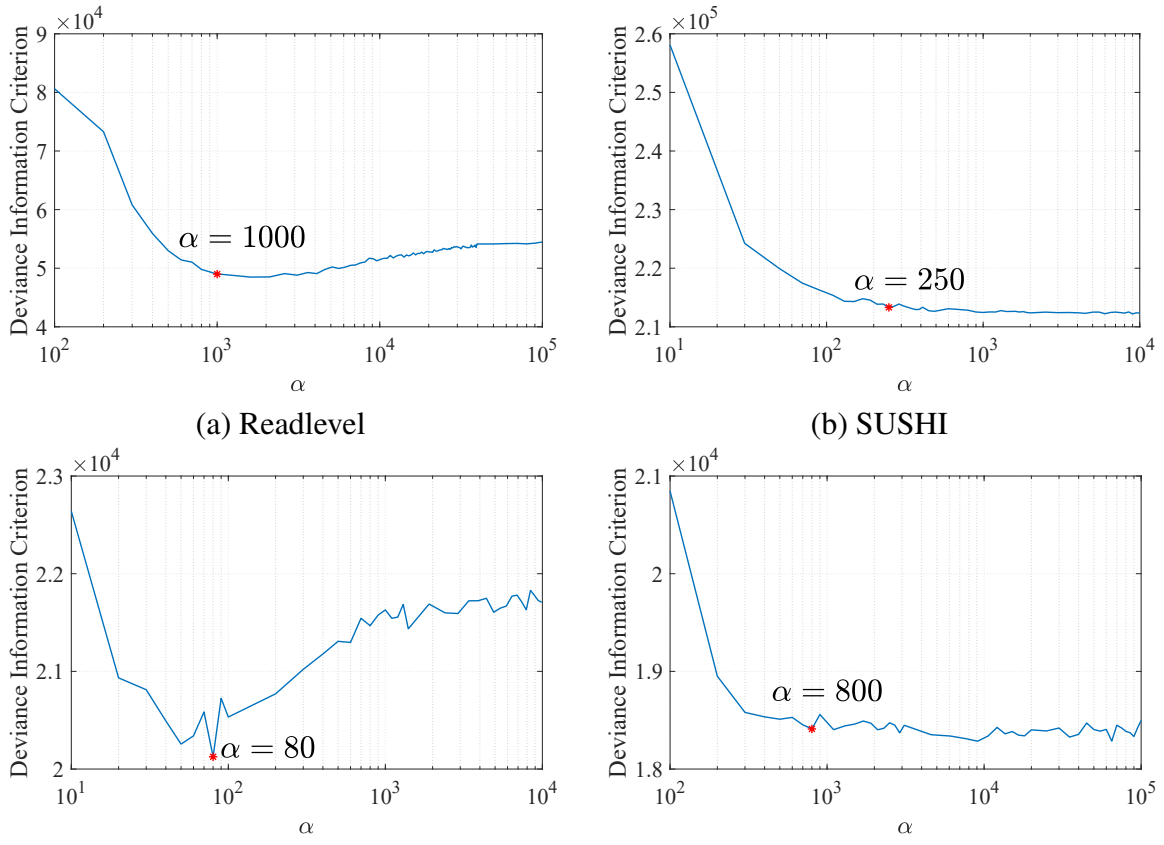
To verify the efficacy of the calibration step, I conducted rank aggregation using CoarsenRank and PL-EM on the *Readlevel* dataset. Then I collected the Kendall tau correlation of these two methods in Figure 4.2, under the case of with or without calibration step, respectively.

As I can see in Figure 4.2:(1) without the calibration step, CoarsenRank and PL-EM are all prone to overfitting. The Kendall tau correlation of CoarsenRank and PL-EM can reach their optima at the first few iterations but start to decrease in the later iterations. (2) With the help of the calibration step, the overfitting problem is avoided. The Kendall tau correlation of two methods remains stable after reaching their optima, respectively. (3) With or without the calibration, The optimum Kendall tau correlation of CoarsenRank is similar. The same is true for PL-EM. It implies that the calibration step does not change the result of rank aggregation methods but just avoids overfitting.

4.5.4 Deviance Information Criterion (DIC) for choosing the hyperparameter α

Following Section 4.3.4, I adopted the DIC to choose the hyperparameter α for different datasets. Since it is intractable to analytically calculate the posterior expectation in DIC, I implemented a Gibbs Sampling procedure in Algorithm 3. Then, I collected the samplings from $P(\theta_m|\mathscr{R}_N, \Xi)$ (Equation 4.29) and calculate the Monte Carlo estimation of DIC Equation 4.31 for different α. The number of samplings is set to 50 in the experiment. The diagnostic curves of α on four datasets are plotted in Figure 4.3, respectively.

The results show that the DIC decreases dramatically at first when α is small, then the curve reaches a cusp and levels off, with more modest increases/decreases in fit when α becomes larger. α is chosen at the point with the lowest DIC or where DIC levels off in the experiment, marked as " $*$ " in each figure.



(a) Readlevel (b) SUSHI

(c) BabyFace (d) PeerGrading

Figure 4.3 (a)-(d) The diagnostic plot of DIC VS. α on four datasets, respectively. The α used in the experiment are marked as " $*$ " in each figure.

4.5.5 The Kendall tau correlation of CoarsenRank in four real applications

I conducted the experiments on four real datasets and collected the empirical results of various rank aggregation methods in Table 4.3. For better comparison, I further showed the performance improvement of all methods over PL-EM on four datasets in Figure 4.4.

It can be observed that: (1) CoarsenRank achieves consistent improvement over other rank aggregation baselines. It demonstrates the great potential of CoarsenRank in real applications, where model misspecification widely exists. (2) The Kendall tau correlation of PL is comparable with PL-EM on all datasets, which rules out the possibility that the EM algorithm would lead to performance improvement. (3) CrowdBT and PeerGrader get superior performance on *BabyFace* because of sufficient annotations (over 60) from each user and the trinary preferences setting in BabyFace. (4) The improvement of CrowdBT and PeerGrader vary significantly on different datasets. The reason is that their pre-assumed

Table 4.3 Experiment results of various rank aggregation methods on four real datasets. Best results are marked in bold.

Baselines	Vanilla		Robust		Ours
	PL	PL-EM	CrowdBT	PeerGrader	CoarsenRank
Readlevel	0.6853	0.6980	0.6944	0.6965	**0.7436**
SUSHI	0.8554	0.8578	0.8765	0.8588	**0.8970**
BabyFace	0.8824	0.8824	0.9085	**0.9150**	0.9020
PeerGrading	0.8023	0.8014	0.8060	0.8040	**0.8130**

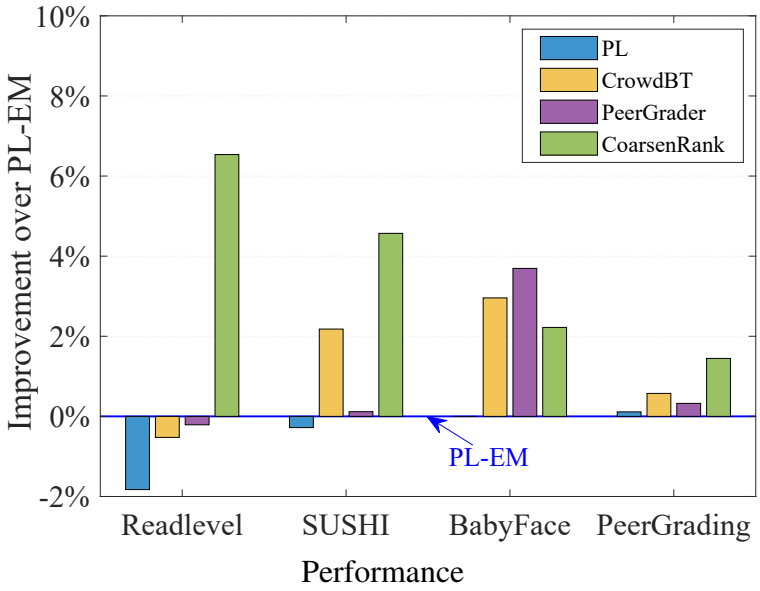


Figure 4.4 Performance improvement of various methods over PL-EM on four datasets, following $\frac{S_*-S_0}{S_0}$. S_0 is the correlation of PL-EM in Kendall tau correlation.

perturbation patterns may not be consistent with noise agnostic perturbations in each dataset. (5) Marginal improvement is achieved by CrowdBT and PeerGrader on *Readlevel*, *SUSHI* and *PeerGrading* where each user provides almost one preference. Points 4 & 5 are model misspecification cases which CoarsenRank is intended to address.

4.5.6 The computational cost compassion of all methods

I independently ran each baseline 50 times and collected the computation cost (s) in Table 4.4 and Figure 4.5. The computation cost is represented by the mean with the standard deviation. For the sake of fair comparison, the inner iteration is fixed to 15 for all methods. Empirical

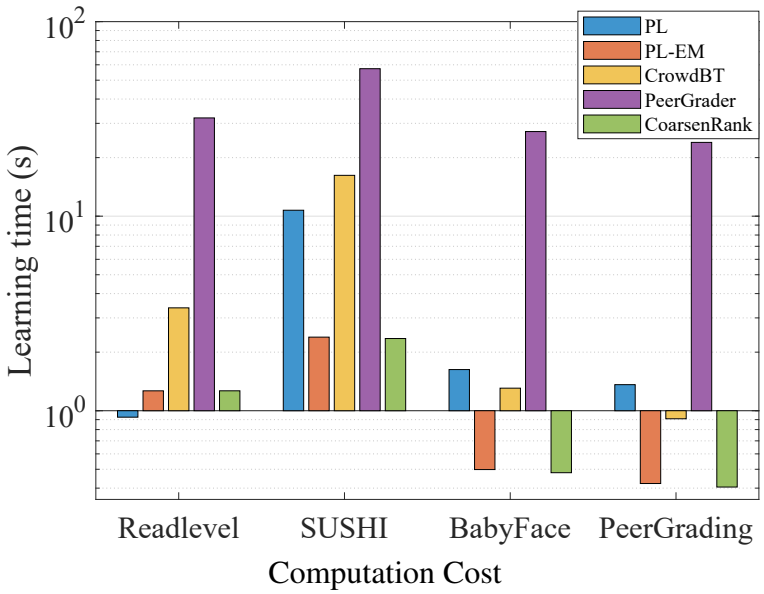


Figure 4.5 The computation cost of all baselines on four datasets, respectively.

results were implemented with an Intel i5 processor(2.30 GHz) and 8 GB random-access memory (RAM).

Table 4.4 The computation cost (s) of all baselines on four datasets, respectively. Best results are marked in bold.

Baselines	Vanilla		Robust		Ours
	PL	PL-EM	CrowdBT	PeerGrader	CoarsenRank
Readlevel	$\mathbf{0.9263 \pm 0.1247}$	1.265 ± 0.2015	3.3788 ± 0.2542	31.9912 ± 0.8596	1.2658 ± 0.2385
SUSHI	10.7290 ± 1.95	$\mathbf{2.3894 \pm 0.4537}$	16.2149 ± 1.9536	57.2892 ± 1.3125	$\mathbf{2.3503 \pm 0.4943}$
BabyFace	1.6276 ± 0.1256	$\mathbf{0.4983 \pm 0.0366}$	1.3060 ± 0.0432	27.2255 ± 0.9180	$\mathbf{0.4801 \pm 0.0242}$
PeerGrading	1.3613 ± 0.1355	$\mathbf{0.4224 \pm 0.0456}$	0.9091 ± 0.0729	23.9479 ± 0.1328	$\mathbf{0.4052 \pm 0.0259}$

It shows that: (1) CoarsenRank achieves a much lower computation compared to other robust ranking aggregation baselines. It shows that CoarsenRank is promising for deploying in a large-scale environment, where reliability and efficiency are both required. (2) The computation costs of CoarsenRank and PL-EM are comparable because of the only difference between CoarsenRank and PL-EM lying at the choosing of parameter τ (See Equation 4.27). (3) PeerGrader suffers from significant inefficiencies since it needs to optimize parameters alternatively. (4) CrowdBT replaces the inefficient alternative optimization with the online Bayesian moment matching and achieves lower computation compared to PeerGrader. However, it is still inefficient on *SUSHI* dataset because of the lack of an efficient rank-break method for long preferences.

4.6 Summary of This Chapter

Our CoarsenRank performs imprecise inference conditioning on a neighborhood of the ranking dataset, which opens a new door to the robust rank aggregation against model misspecification. Experiments on four real applications demonstrate imprecise inference on the neighborhood of the preferences, instead of the original dataset, can improve the model reliability. It shows that CoarsenRank has great potential in real applications, e.g., social choice, information retrieval, recommender system, etc, where model misspecification widely exists. A promising direction for future research is to explore other divergence metrics for other statistical properties of rank aggregation.

Chapter 5

SWORE: Online Bayesian Ranking for Real-time Mental Fatigue Monitoring

Mental fatigue is a common physiological phenomenon [Borghini et al., 2014], which induces sub-optimal functioning, and may even lead to accidents with severe consequences [Van Cutsem et al., 2017]. The National Highway Traffic Safety Administration[1] estimates that about 100,000 official reports of crashes are the direct result of driver mental fatigue each year that results in an estimated 1,550 deaths, 71,000 injuries, and 12.5 billion in monetary losses. In response to these critical issues of mental fatigue, several algorithms have been developed using electro-cardio signal (ECG), functional Near Infrared Spectroscopy (fNIRS) [Nguyen et al., 2017], electroencephalographic (EEG) [Yu et al., 2010, Wang et al., 2015], etc. Among these signals, EEG signals are assumed to be most accurate and valid to fetch the information related to driver's mental fatigue due to the availability of a vast variety of methods to process signals accurately [Sahayadhas et al., 2012, Palanivel Rajan and Dinesh, 2015].

Some of previous work derived from linear [Resalat and Saba, 2015, Lin et al., 2010] and non-linear [Liu et al., 2016] methods show that it is possible to detect the mental fatigue with high accuracy. It is impressive but it is rather blind to the wealth of the dynamics and behavioral variability [Müller et al., 2008, Ratcliff et al., 2009, Yarkoni et al., 2009] available only to offline analysis methods with sufficient training samples. Therefore, previous offline analysis methods often result in poor generalization performance due to insufficient available training data in real-time applications, and also suffer from high re-training cost for sequentially coming data due to lack of efficient calibration strategy.

[1] https://www.nhtsa.gov/press-releases/usdot-releases-2016-fatal-traffic-crash-data

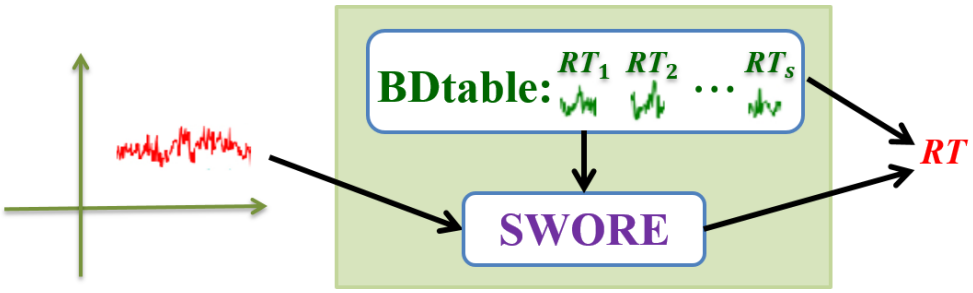


Figure 5.1 Real-Time Mental Fatigue Evaluation. First, I maintain a fixed size Brain Dynamics table (BDtable) as the reference, which consists of the representative EEG signals and the corresponding reaction times (RTs). For a new collected EEG signal x_t, the Self-Weight Ordinal REgression (SWORE) model could give a coarse estimation of the reaction time RT_t by interpolating it among the bunch of maintained RTs using the brain dynamics preferences.

Another important factor among existing proposed methods is lack of efficient aggregation mechanism to aggregate the predictions from multiple noisy channels, while simple majority voting would incur significant reliability degradation for the final results; simple concatenation easily leads to overfitting and poor generalization performance because the features from different channels are highly correlated. Furthermore, since I target at proposing a methodology that works in real-time, the major concern behind chosen methods is the computational resources. Deep learning [Goodfellow et al., 2016] methods require massive training data while Riemannian methods [Barachant et al., 2012, Congedo et al., 2017] would lead to high computation cost, which let them fail to meet the harsh requirement in online setting.

Based on the above discussion, it is clear that there is requirement of the learning model which can work in online fashion while being calibrated in real-time. In this paper, a novel real-time mental fatigue evaluation framework (see Figure 5.1) has been proposed, which significantly overcomes the above mentioned issues in the context of mental fatigue of drivers by utilizing the brain dynamics related preferences. This framework consists of two components, a Self-Weight Ordinal REgression (SWORE) model and a Brain Dynamics table (BDtable). SWORE model learns from brain dynamics preferences from multiple noisy channels by learning the reliability of each channel explicitly within the aggregation process; while BDtable maintains the landmark EEG signals and the corresponding RTs as the reference in real-time (online). Whenever a new EEG signal x_t comes at time t, SWORE model could give a coarse estimation of its reaction time RT_t by interpolating it among the bunch of maintained RTs using the brain dynamics preferences. An online generalized Bayesian moment matching (OGMM) algorithm is further proposed for Bayesian posterior

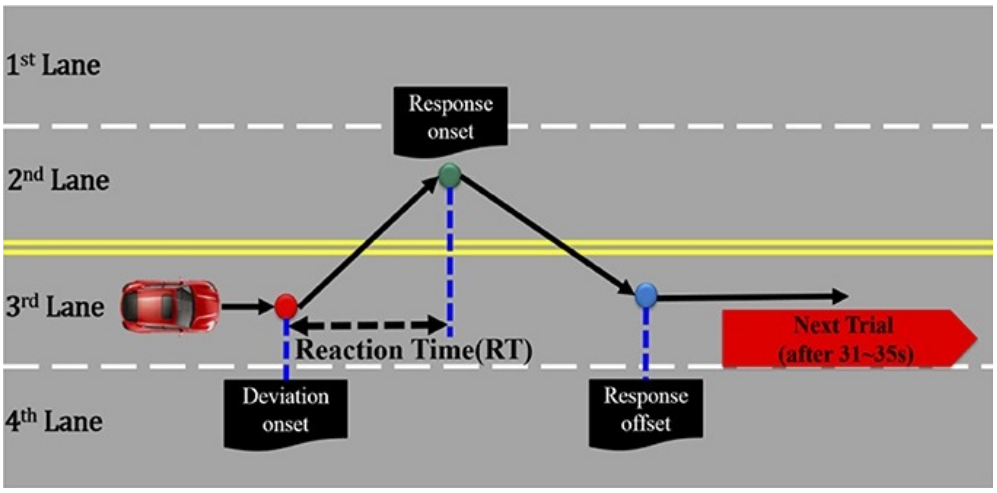


Figure 5.2 Event-related lane-departure driving paradigm

updating. Once the real reaction time RT_t is available, BDtable would help online calibrating the SWORE model by utilizing the simple analytic update rules introduced in the OGMM algorithm.

5.1 Setup and Problem Statement

5.1.1 Mental Fatigue Monitoring

This paper uses the EEG data introduced by Huang et al. [2009]. This EEG data was recorded during the lane-departure driving paradigm in a virtual driving simulating environment (Figure 5.2) with 40 adult participants. In this experiment, participants were required to drive on the four-lane highway and steer back to the middle of road from the random deviation towards the side of road, called as lan-keeping task (LKT). Each participant was instructed to perform the LKT during the total 1h in a continuous driving experiment while performing the task. A complete trial in this study includes deviation onset, response onset and response offset. Every LKT during the whole driving experiment called as trial which occurs within an interval of 31-35s after finishing the current trial. Each participant completed T trials within 1h. For each trial t, the 10s EEG signals $\{x_{n,t}\}_{n=1}^{N}$ before the deviation onset from $N(=32)$ different EEG channels were recorded simultaneously and the corresponding reaction time RT_t was also collected afterwards. Before this 1h driving experiment, the participants were instructed to maintain high level of alerts and required to concentrate on the task, even if they felt fatigue.

5.1.2 Real-time Mental Fatigue Evaluation

First of all, same to previous mental fatigue evaluation attempts, the 10s EEG signals as the feature vector is adopted, which is assumed to be long enough to detect any significant changes in brain activity [Zhang, 2000]. Further, the reaction time is generally accepted as the most intuitive and resourceful metric to evaluate mental fatigue. A common practice for mental fatigue monitoring is to forecast the reaction time using the 10s EEG signals recorded before the deviation onset [Soon et al., 2008].

Although a variety of regression methods have been proposed for mental fatigue evaluations, most conventional methods suffer from three major drawbacks. First of all, regression tasks usually perform well when the response variable is smooth. The underlying assumption may not be satisfied for human reaction time where extreme values commonly exist. Regression models often result in poor generalization for those extreme values. Second, different regions perform different functions in the human brain. Accordingly, different EEG channels contribute differently to human reaction time. Therefore, simple majority voting which makes no distinctions about the channel contributions would incur significant reliability degradation for the final result. Last but not least, brain dynamics are non-stationary, which is characterized by significant trial-by-trial and subject-by-subject variability [Ratcliff et al., 2009, Yarkoni et al., 2009]. Due to this variability, traditional batch learning models suffer from repeated training and updating cost with respect to every new coming data. Apart from that in real-time settings such EEG data usually arrives sequentially, which brings additional scalability challenges.

To build a robust and real-time mental fatigue evaluation model, following sub-goals will be discussed in this paper:

- How to reliably estimate the reaction time for new coming EEG signals?
- How to learn the relative contribution of different channels within the learning model?
- How to effectively calibrate the learning model with an EEG signal when its truth RT is available?
- How to get better performance for high dimensional EEG signals with limited sample size?

5.2 Proposed Approach

In this section, I focus on answering the first two questions. Please refer to Section 5.3.2 and Section 5.3.2 for my solutions to the third and forth questions.

Considering the non-smooth property of the reaction time, it is usually difficult for a learning model to get the exact estimation of the reaction time with relatively few samples, especially in the online setting. Therefore, I no longer obsess with estimating the exact value of the reaction time, which may not be the most important problem. A rough but reliable estimation is also considered acceptable in real world situations of mental fatigue monitoring [Colosio et al., 2017]. Here comes the problem, how to obtain a good shape estimate of RT with insufficiently available EEG signals?

5.2.1 Brain Dynamic Preferences

Let's revisit the prediction of RT in the perspective of ordinal regression. RT is actually defined in the complete ordered field $\mathbb{R}$ which owns its structure meanings. The relative structure information is entirely preserved among the pairwise comparisons of RTs. Therefore, if there exists a learning model which could maximally preserve the whole structure information, a new trial could find its own position (a rough estimation of RT) by its pairwise comparisons with previous recorded EEG signals.

Instead of modelling the global pattern of the brain dynamics within a regression model, I consider the local discrepancy between the current and next brains dynamics state. First, the difference between RTs is leveraged as the indicator for each comparison,

$$y : RT_1 > RT_0, \tag{5.1}$$

where RT_0 and RT_1 denotes the current and next response time, respectively. y has two cases: 1 denotes an up ($RT_1 > RT_0$) and -1 denotes a down ($RT_1 < RT_0$). Further, the brain dynamics preference[2] $(x_0^n, x_1^n), n = 1, 2, \cdots, N$ (typically a pair of d-dimensional feature vector) for each comparison could be constructed with the corresponding pairwise EEG signals recorded from each channel. Note that every EEG sensor used for recording is assumed to record independently from scalp without influencing other sensors [Homan et al., 1987, Teplan et al., 2002], so the brain dynamics preferences are constructed for each channel separately. Namely, there are totally N brain dynamics preferences constructed for each comparison.

[2]We used the term "preference" intentionally to show that brain dynamics keep changing w.r.t. human behaviour and it happens because the human brain prefers one decision over others.

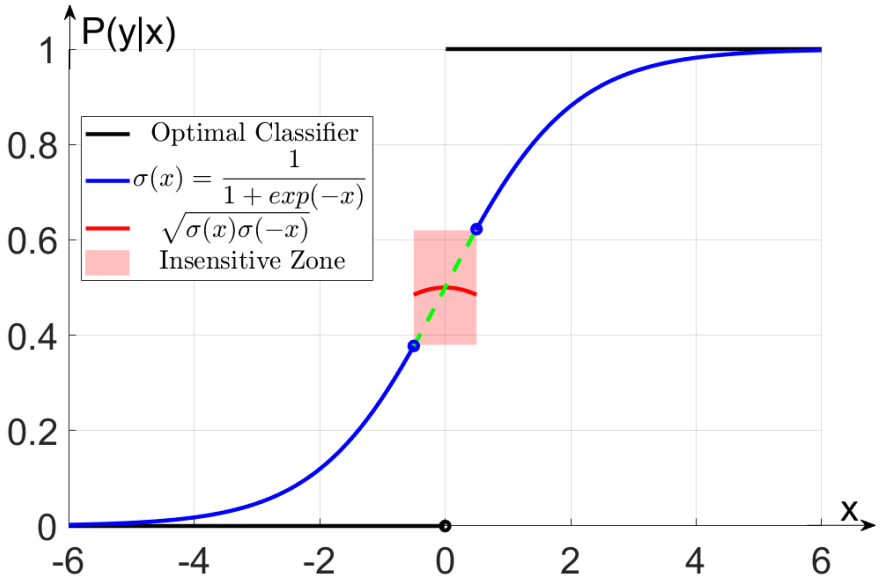


Figure 5.3 Gradient flattening w.r.t. sigmoid function. The dash line represents the original sigmoid function.

5.2.2 Heterogeneous Brain Dynamic Preferences

The study of brain dynamic preferences can be transformed as a pairwise learning to rank problem, in which my main goal is to estimate the optimal classifier in Figure 5.3. Many binary classification models can be adopted to model this problem, for example, the logistic ordinal regression $P(y|w,x_0^n,x_1^n) = \sigma(w^T\Delta x_n)$, where $\sigma(z) = 1/(1+e^{-z})$ is the sigmoid function and $\Delta x_n = x_1^n - x_0^n$. However, the vanilla logistic classification model suffers from reliability issue (a subtle discrepancy around classification boundary $P(y|x) = 0.5$ leads to the steepest gradient, See Figure 5.3) when applied to brain dynamics, since the subtle difference between the RTs may not be caused by the intrinsic difference between two brain dynamics but the unknown noise. To improve the model reliability, I introduce the "insensitive zone", which flattens the steepest gradient and therefore enables the classification model less sensitive to the subtle difference of the responses.

According to my analysis, two states are considered to describe the discrepancy between the brain dynamics, i.e., shaking state ($\mathscr{Y}_1$), where the discrepancy between the brain dynamics are significant, and steady state ($\mathscr{Y}_2$), where the brain dynamics remain stable.

$$y : \begin{cases} |RT_0 - RT_1| > \tau & y \in \mathscr{Y}_1 \\ |RT_0 - RT_1| \leq \tau & y \in \mathscr{Y}_2 \end{cases}, \tag{5.2}$$

where τ is the parameter controlling the model sensitivity.

Shaking State:

In terms of the shaking state $y \in \mathscr{Y}_1$, it has two cases: an up ($RT_1 > RT_0$) and a down ($RT_1 < RT_0$), which could be formulated into the learning to rank problem as mentioned above. However, considering the different functions of different regions in the human brain, the relative contributions of different channels to human reaction time may vary a lot. If I simply aggregate the N brain dynamics preferences recorded in different channels without making any distinctions about the channel reliability, the performance of the learning model would inevitably degrade. Raykar et al. [2010] proposed a robust aggregation model, which aggregates the noisy annotations from multiple crowdworkers by learning the reliability of each worker explicitly within the aggregation process. Inspired by their work, a robust pairwise learning to rank model can be formulated as follows,

$$P(y|w,\pi_{1:N},x_0^{1:N},x_1^{1:N}) = \prod_{n=1}^{N} \left[\pi_n \sigma(w^T \Delta x_n) + (1-\pi_n)\sigma(-w^T \Delta x_n)\right]. \tag{5.3}$$

Note that Equation 5.3 actually models each brain dynamics preference as the weighted arithmetic mean of two cases. The weight $\pi_n \in [0,1]$, learned from the data, denotes the relative contribution from positive ($\pi_n = 1$) to negative ($\pi_n = 0$) of the n-th channel w.r.t to the learning task, $\forall n = 1,2,\cdots,N$.

Remark 10 *From the perspective of EEG channel analysis, the novel mechanism introduced in Equation 5.3 provides a new perspective to combine the information from different channels. Different from majority voting which simply categorizes the channels into reliable ones and noisy ones, Equation 5.3 performs a fine-grained analysis and further categorizes the noisy channels into non-relevant ones and negative reliable ones. Therefore, three types of channels can be recognized with the channel reliability π_n, namely positive reliable ones ($\pi_n \to 1^-$)*[3]*, non-relevant ones ($\pi_n \approx 0.5$) and negative reliable ones ($\pi_n \to 0^+$), $\forall n = 1,2,\cdots,N$.* ■

Steady State:

In terms of the steady state $y \in \mathscr{Y}_2$, it denotes the brain dynamics preferences with comparable RTs. To improve the robustness of the learning model w.r.t the easy corrupted brain dynamics, a technique called gradient flattening is introduced to model the insensitive zone in Figure 5.3. Namely, it flattens the steepest gradient at classification boundary $P(y|x) = 0.5$, enabling the

[3] $\pi_n \to 1^-$ denotes π_n is up to approximate 1. Similar notation applies to $\pi_n \to 0^+$ as well.

learning model to be less sensitive to subtle noisy. One approach for modelling the brain dynamics preferences at steady state is

$$P(y|w,x_0^n,x_1^n) = \sqrt{\sigma(w^T\Delta x_n)\sigma(-w^T\Delta x_n)} \quad y \in \mathscr{Y}_2, \tag{5.4}$$

which is the geometric mean of an up ($RT_1 > RT_0$) and a down ($RT_1 < RT_0$). Refer to [Zhou et al., 2008, Weng and Lin, 2011] for more options. Note that Equation 5.4 can be also extended to robust aggregation model while considering the channel reliability π_n. Due to the symmetric of Equation 5.4, I have $P(y|w,\pi_n,x_1^n,x_2^n) \equiv P(y|w,x_1^n,x_2^n),\ \forall y \in \mathscr{Y}_2$.

Remark 11 *From the perspective of model robustness, the gradient flattening used in Equation 5.4 could be understood as a regularization. It enables the SWORE model to be robust to the subtle fluctuation between RTs. Actually, many applications for example, Crowd counting [Liu et al., 2018] can be modeled with such steady state to enhance the model robustness w.r.t. the response variable.* ■

5.2.3 Self-Weighted Ordinal Regression Model

In summary, the Self-Weighted Ordinal REgression (SWORE) for heterogeneous brain dynamic preferences can be uniformly formulated as follows,

$$P(y|w,\pi_{1:N},x_0^{1:N},x_1^{1:N}) = \begin{cases} \prod_{n=1}^N \left[\pi_n\sigma(w^T\Delta x_n) + (1-\pi_n)\sigma(-w^T\Delta x_n)\right] & y \in \mathscr{Y}_1 \\ \prod_{n=1}^N \sqrt{\sigma(w^T\Delta x_n)\sigma(-w^T\Delta x_n)} & y \in \mathscr{Y}_2 \end{cases}. \tag{5.5}$$

Remark 12 *The superiority of the SWORE model lies in two reasons:* **(1) Reliability:** *SWORE model only trusts the brain dynamics preferences from (positive and negative) reliable channels. Since SWORE model trains a mixture of two complementary classifiers with shared parameter w, it categorizes the channels into positive channels ($\pi_n \to 1^-$), negative channels ($\pi_n \to 0^+$) and non-relevant channels ($\pi_n \approx 0.5$). Based on channel reliability π, SWORE could automatically choose the suitable classifier to extract correct information from the positive and negative channels and update the shared parameter w accordingly. Further, SWORE resists the non-relevant channels by assigning a constant likelihood (i.e.,* 0.5*) to each brain dynamics preference from the non-relevant channels.*

(2) Regularization: *SWORE model only extracts task-related information from each brain dynamics preference. Since the probability of the steady state $y \in \mathscr{Y}_2$ (Equation 5.4) does not depend on the channel reliability π_n, gradient flattening actually performs as a*

Algorithm 4 Online Reservoir Sampling

while $t < S$ **do**
  1) store the t-th EEG signals to the BDtable.
end while
for $t > S$ **to** $+\infty$ **do**
  2) save the t-th EEG signal to the BDtable with probability S/t, else discard it.
  3) use it to replace one EEG signals randomly sampled from the BDtable.
end for

regularization on the regression weight w and enables SWORE model being robust to the random fluctuation, which widely exists in brain dynamics. ■

5.3 Efficient Online Updating Strategy

In this section, I introduce my strategy for (1) online maintaining the BDtable and (2) online updating the SWORE model. In terms of BDtable, it should be a good summarization of previous seen EEG signals, which can help to calibrate the evaluations and guide the model update process. In terms of the online updating procedure for SWORE, it should update SWORE in good approximation and meanwhile introduce low computation cost.

5.3.1 Online Reservoir Sampling for BDtable

Considering the requirement of real-time applications, different strategies can be adopted to maintain the BDtable: (1) fixed BDtable: it is constructed by the predefined EEG signals. For example, the initial S EEG signals is considered to be most representative [Huang et al., 2015, Soon et al., 2008]; (2) dynamic BDtable with prior S EEG signals: considering the time series properties of EEG signals, the S EEG signal extracted before time t are considered to be the most relevant to EEG signals extracted at time t; (3) dynamic BDtable with random sampling S EEG signals: each element of the BDtable is uniformly sampled from the EEG signals seen so far.

To better demonstrate the efficacy of the SWORE model, I adopt the third also the most challenging strategies since the first two strategies are kind of depending on the learning tasks. Specifically, the reservoir sampling [Vitter, 1985] is carried out according to Algorithm 4.

5.3.2 Online Generalized Bayesian Moment Matching for SWORE

Bayesian Moment Matching

Bayesian moment matching (BMM) is a Bayesian approach used to estimate the model parameters. Specifically, it estimates the parameters of the approximated posterior by matching a set of sufficient moments of the exact complex posterior. Moreover, BMM can be further extended to the sequential update paradigm for large-scale or streaming datasets, for example, OnlineBMM [Jaini et al., 2017]. That is, the approximated posterior is updated after each sample instead of the whole dataset. However, previous BMM-based methods cannot handle the non-conjugate likelihood function (for example, Equation 5.5). Inspired by the work from Weng and Lin [2011], a generalized BMM method is introduced to online update the SWORE model with analytic update rules.

Note that I only consider the posterior distribution w.r.t brain dynamic preferences from single channel since different channels are modeled independently. The following equations can be easily extended to the posterior distribution w.r.t the brain dynamic preferences from all channels.

First, SWORE is extended to its Bayesian version. Specifically, a Gaussian prior is introduced for weight vector w, i.e., $w \sim N(\mu, \Sigma)$, while a Beta prior is introduced for each channel reliability π_n, namely, $p_0(\pi) = \prod_{n=1}^{N} \text{Beta}(\pi_n|\alpha_n, \beta_n)$. Given a brain dynamics preference (x_0, x_1)[4] recorded in the n-th channel with its ordinal supervision y, the posterior of the model parameters can be represented as:

$$P(w, \pi|y, \Delta x) = \frac{P(y|w, \pi, \Delta x)p_0(w)p_0(\pi)}{P(y|\Delta x)}. \tag{5.6}$$

The main issue with Equation 5.6 is that the joint posterior distribution $P(w, \pi|y, \Delta x)$ is usually complicated or even intractable. To keep the computation tractable, I adopt the mean-filed assumption and project the posterior into the same form with the prior (product of a Normal with Betas, i.e., $P(w, \pi|y, \Delta x) \approx q(w)q(\pi) = N(w|\mu, \Sigma)\prod_{n=1}^{N} \text{Beta}(\pi_n|\alpha_n, \beta_n)$). Then the posterior parameters are estimated by matching a set of sufficient moments of the approximate posterior with the exact posterior:

1. Match the moments between $q(w)$ and $P(w|y, \Delta x)$: $\int wq(w)dw = \int wP(w|y, \Delta x)dw$ and $\int ww^T q(w)dw = \int ww^T P(w|y, \Delta x)dw$. Due to the non-conjugation between the marginalized likelihood $P(y|w, \Delta x)$[5] and the normal prior $N(w|\mu, \Sigma)$, the posterior $P(w|y, \Delta x)$

[4] I omitted the superscript n for simplicity.

[5] $P(y|w, \Delta x) = E_{\text{Beta}(\pi|\alpha,\beta)}[P(y|w, \pi, \Delta x)]$.

is complex. Therefore, the posterior parameters $(\mu^{new}, \Sigma^{new})$ cannot be computed analytically because of the intractability of the integrals in the moment constrains.

2. Match the moments between $q(\pi)$ and $P(\pi|y,\Delta x)$: $\int \pi_n q(\pi)d\pi = \int \pi_n P(\pi|y,\Delta x)d\pi$ and $\int \pi_n^2 q(\pi)d\pi = \int \pi_n^2 P(\pi|y,\Delta x)d\pi$, $n = 1,2,\cdots,N$. Fortunately, I can solve the moment constrains with closed-form integrals, and get the posterior parameters $(\alpha_n^{new}, \beta_n^{new}), \forall n = 1,2,\cdots,N$ accordingly.

Generalized Bayesian Moment Matching

Based on the Bayesian approximation method proposed by Weng and Lin [2011], which is extended from the Stein's Lemma [Woodroofe, 1989], I can estimate the posterior parameters $(\mu^{new}, \Sigma^{new})$ of approximate posterior $q(w)$ by differential operations instead of integral operations. Therefore, the BMM algorithm is extended to a general situation where the likelihood function is twice differentiable.

Theorem 2 *Assume $f(w)$ is the marginalized likelihood of one brain dynamics preference and almost twice differentiable. Upon updating this preference, the posterior parameters $(\mu^{new}, \Sigma^{new})$ of weight w can be estimated as:*

$$\mu^{new} \approx \mu + \Sigma \times \frac{d\log f(w)}{dw}\Big|_{w=\mu}, \tag{5.7a}$$

$$\Sigma^{new} \approx \Sigma + \Sigma \times \frac{d^2\log f(w)}{dwdw^T}\Big|_{w=\mu} \times \Sigma. \tag{5.7b}$$

I set $w = \mu$ as I expect that the posterior density of w to be concentrated on μ [Weng and Lin, 2011]. The detailed proof can be found in the Appendix.

Online Generalized Bayesian Moment Matching

In the following, I resort to Generalized Bayesian Moment Matching (GBMM) method to estimate the posterior parameters. I take the brain dynamic preference (x_0, x_1) at the shaking state $y \in \mathscr{Y}_1$ as an example. The equations can be easily extended to the brain dynamics preference at the steady state. Specifically, I first update the hyperparameters (μ, Σ) of w, then update the hyperparameter (α_n, β_n) of π_n $\forall n = 1,2,\cdots,N$. To update w, I integrate out π_n to obtain the marginalized likelihood function $f(w)$:

$$f(w) = \int P(y|w,\pi,\Delta x)\text{Beta}(\pi|\alpha,\beta)d\pi = \frac{\alpha_n}{\alpha_n+\beta_n}\sigma(w^T\Delta x) + \frac{\beta_n}{\alpha_n+\beta_n}\sigma(-w^T\Delta x). \tag{5.8}$$

According to Equation 5.7a in Theorem 2, I can update μ as follows:

$$\mu^{new} \approx \mu + \Sigma \times \frac{d \log f(w)}{dw}\Big|_{w=\mu} = \mu + (A-a) \times \Sigma \times \Delta x. \tag{5.9}$$

where $A = \frac{\alpha_n}{\alpha_n + \beta_n e^{-\mu^T \Delta x}}$ and $a = \frac{1}{1+e^{-\mu^T \Delta x}}$. According to Equation 5.7b in Theorem 2, I can update Σ as follows:

$$\begin{aligned} \Sigma^{new} &\approx \Sigma + \Sigma \times \frac{d^2 \log f(w)}{dw dw^T}\Big|_{w=\mu} \times \Sigma \\ &\approx \Sigma + \kappa \mathbf{I} + [A(1-A) - a(1-a)] \times \Sigma \times \Delta x \Delta x^T \times \Sigma. \end{aligned} \tag{5.10}$$

where κ is a small positive value to ensure a positive definite variance matrix. $\mathbf{I}$ is an identity matrix.

To update π, I first integrate out w to obtain the marginalized likelihood $f(\pi_n)$ for each preference:

$$\begin{aligned} f(\pi_n) &= \int P(y|w, \pi_n, \Delta x) N(w|\mu, \Sigma) dw \\ &= \pi_n \times E_{N(w|\mu,\Sigma)}[\sigma(w^T \Delta x)] + (1-\pi_n) \times E_{N(w|\mu,\Sigma)}[\sigma(-w^T \Delta x)]. \end{aligned} \tag{5.11}$$

Let $R_1 = E_{N(w|\mu,\Sigma)}[\sigma(w^T \Delta x)]$ and calculate R_1 by the second order Taylor approximation of $\sigma(w^T \Delta x)$ at μ. Namely,

$$R_1 = E_{\mathcal{N}(w|\mu,\Sigma)} \sigma(w^T \Delta x) = \sigma(\mu^T \Delta x) + \frac{1}{2} \sigma(\mu^T \Delta x)[1 - \sigma(\mu^T \Delta x)][1 - 2\sigma(\mu^T \Delta x)] \Delta x^T \Sigma \Delta x,$$

where I set $R_1 = \max(R_1, \kappa_2)$, where κ_2 is a small positive value to ensure a positive R_1.

Then I have $R_2 = E_{N(w|\mu,\Sigma)}[\sigma(-w^T \Delta x)] = 1 - R_1$ and the normalization constant $P(y|\Delta x)$ can be represented as follows,

$$R = P(y|\Delta x) = \int f(\pi_n) \text{Beta}(\pi_n|\alpha_n, \beta_n) d\pi = \frac{\alpha_n R_1 + \beta_n R_2}{\alpha_n + \beta_n}.$$

According to Bayesian Theorem, the posterior distribution of π is

$$P(\pi_n|y, \Delta x) = \frac{f(\pi_n) \text{Beta}(\pi_n|\alpha_n, \beta_n)}{R},$$

the moments $E[\pi_n]$ and $E[\pi_n^2]$ w.r.t to $P(\pi_n|y,\Delta x)$ can be computed as follows:

$$\begin{aligned} E_{P(\pi_n|y,\Delta x)}[\pi_n] =& \frac{R_1(\alpha_n+1)\alpha_n + R_2\alpha_n\beta_n}{R(\alpha_n+\beta_n+1)(\alpha_n+\beta_n)}, \\ E_{P(\pi_n|y,\Delta x)}[\pi_n^2] =& \frac{\alpha_n(\alpha_n+1)[R_1(\alpha_n+2)+R_2\beta_n]}{R(\alpha_n+\beta_n+2)(\alpha_n+\beta_n+1)(\alpha_n+\beta_n)}, \end{aligned} \tag{5.12}$$

where $n = \{1,2,\cdots,N\}$. The detailed derivation can be found in the Appendix for the sake of completeness.

In terms of the sufficient moments with regard to the posterior distribution $q(\pi_n|\alpha_n^{new},\beta_n^{new})$, I have

$$\begin{aligned} E[\pi_n] &= \int \pi_n q(\pi_n|\alpha_n^{new},\beta_n^{new})d\pi_n = \frac{\alpha_n^{new}}{\alpha_n^{new}+\beta_n^{new}}, \\ E[\pi_n^2] &= \int \pi_n^2 q(\pi_n|\alpha_n^{new},\beta_n^{new})d\pi_n = \frac{\alpha_n^{new}(\alpha_n^{new}+1)}{(\alpha_n^{new}+\beta_n^{new}+1)(\alpha_n^{new}+\beta_n^{new})}. \end{aligned}$$

According to the above equations, I have

$$\begin{aligned} E[\pi_n] - E[\pi_n^2] &= \frac{\alpha_n^{new}\beta_n^{new}}{(\alpha_n^{new}+\beta_n^{new}+1)(\alpha_n^{new}+\beta_n^{new})}, \\ E[\pi_n^2] - (E[\pi_n])^2 &= \frac{\alpha_n^{new}\beta_n^{new}}{(\alpha_n^{new}+\beta_n^{new}+1)(\alpha_n^{new}+\beta_n^{new})^2}. \end{aligned}$$

Then I can update the hyperparameter (α_n,β_n) of π_n as follow:

$$\alpha_n^{new} = \frac{(E[\pi_n]-E[\pi_n^2])E[\pi_n]}{E[\pi_n^2]-(E[\pi_n])^2}, \tag{5.13a}$$

$$\beta_n^{new} = \frac{(E[\pi_n]-E[\pi_n^2])(1-E[\pi_n])}{E[\pi_n^2]-(E[\pi_n])^2}. \tag{5.13b}$$

According to my above analysis, I summarize my OGMM implementation for SWORE model in Figure 5.4. It is notable that the weight update and channel reliability update can all be completed with analytic rules (Equation 5.9, 5.10, 5.13a, 5.13b). As a result of the efficient posterior updating procedure, OGMM leads SWORE naturally to handle streaming preferences in real-time.

OGMM with Data Augmentation

Due to the limited size of available trials, the learning model is prone to be overfitting during the training process. Therefore, I adopt the data augmentation trick in this paper. Data

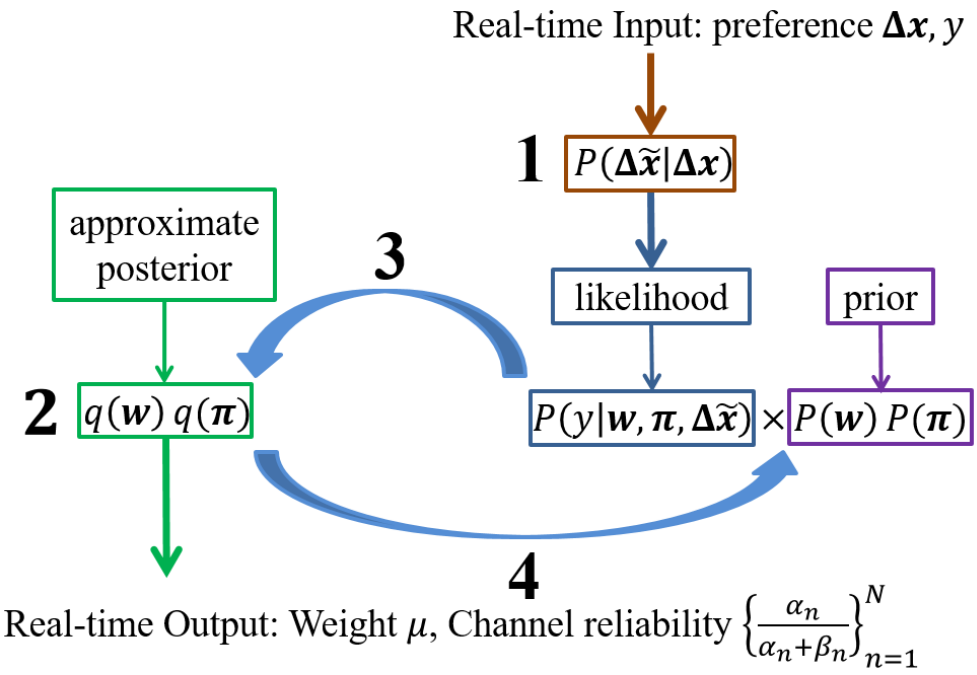


Figure 5.4 OGMM with Data Augmentation. Note that: (1) sample the corrupted EEG signal $\Delta\tilde{x}$ from the predefined corrupting distribution $P(\Delta\tilde{x}|\Delta x)$; (2) define $q(w)q(\pi)$ in the same form as the prior (product of a Normal with Betas); (3) estimate $q(w)q(\pi)$ with generalized Bayesian moment matching; (4) replace prior $P(w)P(\pi)$ with approximate posterior $q(w)q(\pi)$.

augmentation trick[6] replaces the original EEG signals with T corrupted versions from the predefined corrupting distribution $P(\Delta\tilde{x}|\Delta x)$. For simplicity, I focuse on the blank-out noise model (a.k.a dropout) as the corrupting distribution, which randomly omitting subsets of neurons (or features). More precisely,

$$P(\Delta\tilde{x}_l|\Delta x_l;\theta)=\begin{cases}\theta & \Delta\tilde{x}_l=0\\ 1-\theta & \Delta\tilde{x}_l=\Delta x_l\end{cases}, \tag{5.14}$$

where $\Delta\tilde{x}_l\in\{0,\Delta x_l\}\ \forall l=1,2,\cdots,L$ and L is the feature dimension.

Note that each dimension of the input Δx is corrupted independently. Equation 5.14 is also a promising technique to break up the complex co-adaptations, caused by high correlation among different dimensions of the EEG signals (either in time domain or frequency domain). Since the presence of any particular dimension is unreliable, each dimension cannot rely

[6] Although data augmentation procedure generates a corrupted dataset with a larger size (scaling linearly with T), the incurred computational cost is acceptable benefiting from the efficient Bayesian updating rules.

on other specific dimensions to correct its mistakes. It must perform well in a wide variety of different contexts provided by the other dimensions. Therefore, OGMM equipped with Equation 5.14 could improve the generalization of BDrank for more complex situations (See Figure 5.7).

5.4 Real-time Mental Fatigue Evaluation

In this section, I apply the SWORE model (Equation 5.5) to perform real-time mental fatigue monitoring. Specifically, following the updating strategies introduced in Section 5.3, the SWORE model $\{w, \pi_{1:N}\}$ and the BDtable $\{x_i^{1:N}, RT_i\}_{i=1:S}$ are all updated to time $t-1$[7]. Then, the real-time mental fatigue monitoring refers to predict the response time RT_t with the EEG signals $x_t^{1:N}$ extracted at time t using the up-to-date SWORE model and BDtable.

As stated in Section 5.2.1, the relative ordinal structure of RT is revealed through the pairwise comparisons, i.e., Brain Dynamic Preferences, which are maximally preserved by my SWORE model $\{w, \pi_{1:N}\}$. Particularly, the relative ordinal structure of RT is consistent with that of $w^T x$ according to the definition of my SWORE model, namely

$$\begin{aligned} RT_i > RT_j \Longleftrightarrow & \begin{cases} w^T x_i^n > w^T x_j^n & \text{positive channel } (\pi_n > 0.5) \\ w^T x_i^n < w^T x_j^n & \text{negative channel } (\pi_n < 0.5) \end{cases} \\ \text{or} \quad RT_i > RT_j \Longleftrightarrow & \operatorname{sgn}(\pi_n - 0.5) w^T x_i^n > \operatorname{sgn}(\pi_n - 0.5) w^T x_j^n, \end{aligned} \tag{5.15}$$

where i, j denote the index of different trials, and n represents the index of different channels.

Then, I compare the newly recorded EEG signals $x_t^{1:N}$ to EEG signals $\{x_i^{1:N}\}_{i=1:S}$ stored in the BDtable following Equation 5.15 and derive N full ranking lists over $S+1$ trials regarding each channel, respectively.

$$\begin{cases} \operatorname{sgn}(\pi_1 - 0.5) w^T \times \{x_t^1, x_1^1, x_2^1, \ldots, x_S^1\} & n = 1 \\ \operatorname{sgn}(\pi_2 - 0.5) w^T \times \{x_t^2, x_1^2, x_2^2, \ldots, x_S^2\} & n = 2 \\ \qquad\qquad \cdots & \\ \operatorname{sgn}(\pi_N - 0.5) w^T \times \{x_t^N, x_1^N, x_2^N, \ldots, x_S^N\} & n = N \end{cases}. \tag{5.16}$$

Let $\operatorname{Sort}(x_t^n)$ output the ranking position of the estimated RT_t for the EEG signals x_t^n in terms of the n-th channel, compared to EEG signals stored in the BDtable. The ranking position of the estimated RT_t over all N channels could be derived by aggregating the results

[7]The model parameters $\{w, \pi_{1:N}\}$ are used to represent the SWORE model, since Equation 5.5 is fully determined by $\{w, \pi_{1:N}\}$. Meanwhile, we omit the subscript $t-1$ for convenience.

from all channels while considering the channel reliability. Namely

$$\text{Sort}(x_t^{1:N}) = \sum_{n=1}^{N} \frac{|2\pi_n - 1| \times \text{Sort}(x_t^n)}{\sum_{k=1}^{N} |2\pi_k - 1|}. \tag{5.17}$$

Correspondingly, I sort the S response times $\{RT_i\}_{i=1:S}$ stored in the BDtable and derive the full ranking list. Following the consistency of the relative ordinal structure between RTs and Brain Dynamic Preferences (Equation 5.15), I can find the neighboring RTs with the ranking position being close to $\text{Sort}(x_t^{1:N})$. Therefore, the corresponding RT_t for the newly recorded EEG signals $x_t^{1:N}$ can be estimated by the interpolation of the detected neighboring RTs.

5.5 Numerical Experiments

In this section, I first explore the reliability of SWORE in real-time mental fatigue evaluation tasks. Then, I analyze the parameter sensitivity and model uncertainty of the SWORE w.r.t the proposed OGMM algorithm.

5.5.1 Experiment Setup

The EEG signals were collected from forty participants. Considering the time delay among the channels in the time domain, Fourier transform [Welch, 1967] has been applied to EEG signals to transform time-series into frequency domain. Further, to avoid overhead computation, EEG power within 0-30Hz has been selected, which is considered to be the most relevant to the RTs [Huang et al., 2015].

The shaking state preferences $\mathscr{Y}_1$ were constructed with RT comparisons $(RT_{m,1}, RT_{m,2})$, which satisfies $RT_{m,2} < \min(RT_{m,2} + \tau_1, \tau_2 * RT_{m,2}) < RT_{m,1}$; the steady state preferences $\mathscr{Y}_2$ were constructed with RT comparisons $(RT_{m',1}, RT_{m',2})$, which satisfies $RT_{m',2} < RT_{m',1} < \min(RT_{m',2} + \tau_3, \tau_4 * RT_{m',2})$. It is notable that $\tau_1 > \tau_3 > 0$ and $\tau_2 > \tau_4 > 1$ control the difference in the RT comparisons simultaneously, I empirically set $\tau_1 = 1; \tau_2 = 1.5; \tau_3 = 0.8; \tau_4 = 1.3$ for all participants in the experiment. The dropout rate θ in corrupting distribution (Equation 5.14) is set to 0.5, which is a widely adopted setting in deep learning literatures.

Here, I adapted the Wilcoxon-Mann-Whitney statistics [Yan et al., 2003] to measure the performance of SWORE model. First, I estimated the ordinal supervision of each RT comparison by aggregating the predictions from the reliable channels using a voting scheme. Namely $\hat{y}_m = sign\big(\sum_{n=1}^{N} \hat{y}_m^{(n)} \big[\mathbf{1}(\pi_n > \kappa) - \mathbf{1}(\pi_n < 1 - \kappa)\big]\big)$, where m is the RT comparison index. $\hat{y}_m^{(n)}$ is the predicted order for the brain dynamic preference $(x_0^n, x_1^n)_m$ from the n-th channel: 1 denotes an up and -1 denotes a down. $\mathbf{1}()$ is an indicator that returns one if the

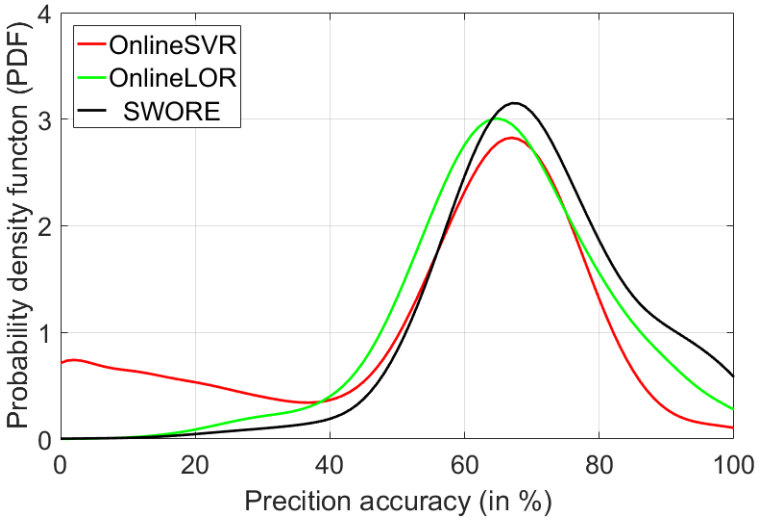


Figure 5.5 The PDF of prediction accuracy.

argument is valid and returns zero otherwise. Note that a channel is recognized as a reliable channel if it satisfies $\pi_n > \kappa$ or $\pi_n < 1-\kappa$. κ is set to 0.85 in this paper. Then, I calculated the accuracy (in %, higher is better) over all pairs, namely $\frac{1}{M}\sum_{m=1}^{M}\mathbf{1}(y_m = \hat{y}_m)$, where y_m is the ground truth for the RT comparison $\mathbf{1}(RT_1 > RT_0)_m$.

5.5.2 Online Mental Fatigue Evaluation

Following the real-time mental fatigue evaluation framework proposed in Figure 5.1, I explored the reliability of SWORE in real-time monitoring task. Specifically, I leveraged the prerecorded 25 trials to construct the brain dynamic preferences and to train an embryonic SWORE model accordingly. I randomly initialized μ in $[-10^{-2}, 10^{-2}]$, Σ in $[\mathbf{0}, 10^{-4}\times\mathbf{I}]$ and set $\alpha_n = \beta_n = 5$ according to my parameter sensitivity analysis in Section 5.5.3. The data augmentation size T is set to 5 according to Figure 5.7 since I encountered the insufficient training samples situation here. The brain dynamics table size is fixed to 10. Therefore, I could sequentially get the coarse estimation of RT for each new trail, collect the prediction accuracy and update the SWORE model when the truth RT is available. I ran the SWORE model following the procedure in Figure 5.1 for 100 times and estimated the probability density function (PDF) of prediction accuracy for each EEG signal (See Figure 5.5).

We compared SWORE with online Support Vector Regression (OnlineSVR) [Sahoo et al., 2014] and online Logistics Ordinal Regression (OnlineLOR)[8]. The parameter for

[8]OnlineLOR is a special case of SWORE. A simple OnlineLOR can be implemented by fixing (α_n, β_n) to $(1,0)$ $\forall n = 1,2,\cdots,N$ and abandoning the gradient flattening trick, while others remain the same as SWORE.

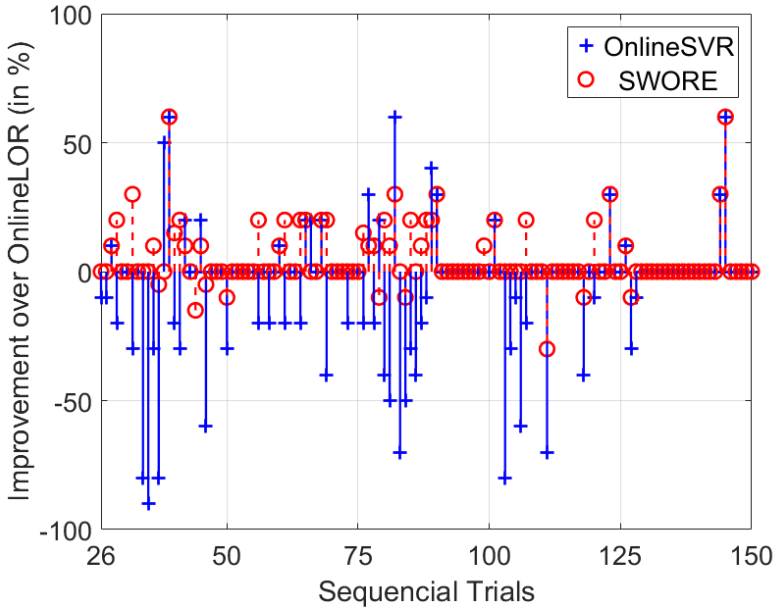


Figure 5.6 Real-Time Mental Fatigue Evaluation.

OnlineSVR is chosen by five-fold cross-validation in the offline setting. Same to SWORE, the prediction for each pair is calculated using a majority voting scheme over all channels and the final accuracy is averaged over all pairs. Since there is no mechanism for OnlineSVR and OnlineLOR to evaluate the channel state beforehand, they trust all the channels by default.

From Figure 5.5, I can observe that: (1) SWORE could give a reliable evaluation ($\geq 70\%$ prediction accuracy) for any new EEG signal with higher confidence ($P(X \geq 70\%) = 52.3\%$), comparing to OnlineLOR (41.6%) and OnlineSVR (33.7%). (2) OnlineLOR shows inferior performance compared to SWORE. Its prediction accuracy for a new EEG signal are mainly distributed within $[55\%, 80\%]$ ($P(55\% \leq X \leq 80\%) = 65\%$). (3) The performance of OnlineSVR is quite unstable, since it would give serious wrong estimations (ACC $\leq 40\%$) for some new EEG signals with high probability ($P(X \leq 40\%) = 21.7\%$), while SWORE (2.5%) and OninelOR (5.1%) are not.

To give an intuitive comparison of online setting, I calculated the prediction accuracy of each trails for one random experiment with three methods and then ploted performance improvement of SWORE and OnlineSVR compared to OnlineLOR (See Figure 5.6). It is notable that SWORE could consistently achieve superior or at least comparable performance compared to OnlineSVR, which demonstrates my claim that channel reliability indeed affects the efficacy of the learning model. In terms of (regression-based) OnlineSVR, it suffers from high generalization errors for new EEG signals compared to (classification-based) SWORE

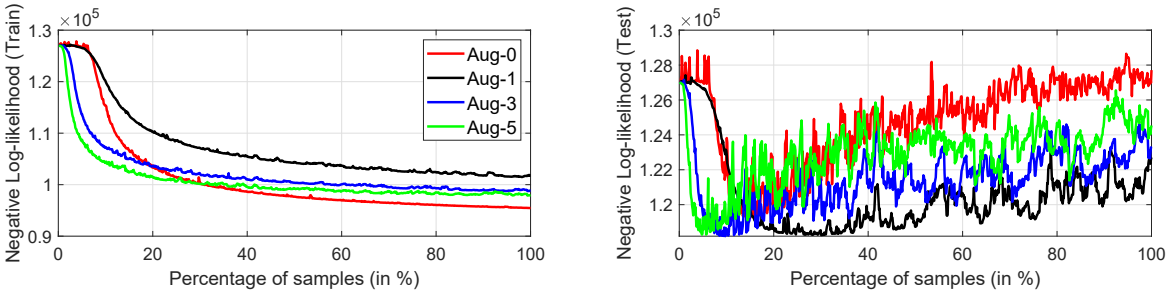


Figure 5.7 The negative Log-likelihood of brain dynamic preferences on training and test dataset w.r.t different level of data augmentation size. Note that "Aug-n" denotes the data augmentation size T is set to n.

and OnlineSVR. Moreover, the computation cost for OnlineSVR is considerable high due to the absence of appropriate updating rules for the support vectors [Sahoo et al., 2014].

5.5.3 Parameter Sensitivity and Model Uncertainty

In this section, I explored the parameter sensitivity of SWORE w.r.t the hyperparameters (μ, Σ) and (α, β), respectively. In particular, I generated the offline brain dynamic preferences as follows: (1) the trials of each participant were randomly divided into two parts: 50% for training and 50% for test; (2) Offline Brain dynamic preferences were constructed according to the pairwise comparisons between the RTs regardless of their sequential property.

Sensitivity analysis w.r.t hyperparameters (μ, Σ) For the sake of simplicity, I considered diagonal covariance matrix here. Specifically, I randomly initialized μ in $[-a, a]$ and Σ in $[\mathbf{0}, b\mathbf{I}]$. The value of a and b are set within $\{1, 10^{-2}, 10^{-4}\}$, respectively. Further, I adopted a non-informative prior for π_n, namely $\alpha_n = \beta_n = 5$ to eliminate the effects of noisy channels. The data augmentation size T is set to 1 since the size of the training data is sufficient. The testing performance of SWORE under different parameter settings is presented in Table 5.1. The detailed experimental results for all forty participants can be found in the Appendix.

Table 5.1 shows that: (1) SWORE consistently performs very well with testing accuracy greater than 70% on all participants under small initialization ($10^{-4} < a, b < 10^{-2}$) for (μ, Σ). Because SWORE model suffers from spurious overflow/underflow problems with large initialization at each updating step, due to the high dimension feature ($L = 492$) and the exponential operator (within the sigmoid function). (2) Although the performance of SWORE has minor differences for different participants, SWORE is robust to the small

initialization and shows comparable performance for the same participant under different initialization.

Sensitivity analysis w.r.t hyperparameters (α, β) To explore the effects of hyperparameter (α_n, β_n) w.r.t SWORE model, I randomly initialized (α_n, β_n) in $\{1,3,5\}$. I randomly initialized μ in $[-10^{-2}, 10^{-2}]$ and Σ in $[\mathbf{0}, 10^{-4} \times \mathbf{I}]$. The corrupting size T was set to 1 as previous. The performance of SWORE on the testing data are reported in Table 5.2. The detailed experimental results for all forty participants can be found in the supplementary.

It is worth to note that SWORE is insensitive to the initialization of hyperparameters (α, β). Particularly, SWORE could achieve comparable performance for each participant under different initialization of (α, β). SWORE consistently performs very well on all forty participants, regardless of different initialization for (α, β).

Sensitivity analysis w.r.t data augmentation size T According to Table 5.1 and Table 5.2, I set $a = 10^{-2}$ and $b = 10^{-4}$ w.r.t hyperparameters (μ, Σ), and initialize hyperparameters (α, β) to $(5,5)$. Then, I collected the negative Log-likelihood of brain dynamic preferences on training and test dataset (See Figure 5.7) with data augmentation size T being set to $\{0,1,3,5\}$, respectively. Note that I only showed the results of the first participant due to the limited space.

From Figure 5.7, I can observe that: (1) the SWORE model is prone to be overfitting on the original EEG signal, since the dimensions of the EEG signals (either in time domain or frequency domain) are closely related to each other. See Section 5.3.2 for detailed explanations. (2) The feature corruption trick $(T = 1)$ achieves the best performance comparing to other setting, including the data augmentation methods $(T > 1)$, the larger the data augmentation size T, the worse the generalization performance of SWORE. (3) It is interesting to note that SWORE with data augmentation methods $(T > 1)$ performance extremely good with only a few samples (less than 20% training data), but SWORE starts to overfitting when updated with more samples.

Stability analysis of OGMM algorithm Here, I empirically analyzed the stability of OnlinGBMM algorithm. According to my sensitivity analysis w.r.t hyperparameters (μ, Σ) and (α, β) (See Table 5.1, Table 5.2), I randomly initialized μ in $[-10^{-2}, 10^{-2}]$, Σ in $[\mathbf{0}, 10^{-4} \times \mathbf{I}]$. Further, I initialized hyperparameters (α, β) to $(5,5)$. The corrupting size T is set to 1. Then, I repeated the OGMM algorithm on the training data for 20 times and summarized the prediction accuracy on the test data Figure 5.8.

Table 5.1 Test accuracy (in %, the larger the better) w.r.t hyperparameter (μ, Σ) with hyperparameter (α, β) fixed to $(5,5)$, dropout rate $\theta = 0.5$, data augmentation number $T = 1$. The best parameter settings are marked in gray. Some parameter settings do not consistently perform very well and may fail on some participants (marked in bold).

Test ACC	SVR	LOR	(μ,Σ) with (α,β) fixed to $(5,5)$								
			$(1,1)$	$(10^{-2},1)$	$(10^{-4},1)$	$(1,10^{-2})$	$(10^{-2},10^{-2})$	$(10^{-4},10^{-2})$	$(1,10^{-4})$	$(10^{-2},10^{-4})$	$(10^{-4},10^{-4})$
P1	69.49	76.24	50.00	50.00	79.89	77.23	78.29	78.75	50.00	78.71	79.28
P2	76.35	83.56	50.00	50.00	50.00	79.55	81.48	83.00	50.00	82.03	81.85
P3	79.65	79.56	50.00	50.00	50.00	85.32	84.54	84.29	70.07	82.89	83.14
P4	65.68	70.33	50.00	50.00	50.00	72.16	76.45	75.72	50.00	73.26	73.26
P5	85.76	84.67	50.00	50.00	50.00	85.34	85.52	85.21	50.00	85.07	85.00
P6	74.84	64.54	50.00	50.00	50.00	86.76	83.25	84.76	50.00	84.56	84.31
P7	72.48	66.36	50.00	50.00	50.00	75.44	75.21	75.10	50.00	75.18	75.31
P8	78.60	78.60	50.00	50.00	50.00	84.38	84.06	84.10	50.00	84.70	84.62
P9	73.67	67.71	50.00	50.00	50.00	83.12	83.01	83.12	50.00	83.26	83.46
P10	92.55	91.93	50.00	50.00	50.00	**79.17**	88.00	88.51	50.00	89.50	89.47
P11	42.91	56.95	50.00	50.00	50.00	80.45	75.99	76.39	50.00	77.49	77.42
P12	70.18	78.28	50.00	50.00	50.00	79.97	80.18	80.28	50.00	80.09	80.09
P13	78.06	82.00	50.00	50.00	50.00	81.09	80.75	80.75	50.00	81.52	81.52
P14	77.26	77.40	50.00	50.00	50.00	**50.00**	78.58	78.65	50.00	80.03	80.07
P15	90.34	85.18	50.00	50.00	50.00	89.58	89.92	89.86	50.00	89.93	89.97
P16	76.03	72.76	50.00	50.00	50.00	73.02	72.75	72.59	50.00	72.41	72.17
P17	76.19	77.10	50.00	50.00	50.00	**50.00**	76.99	77.63	50.00	78.05	78.09
P18	61.09	77.12	50.00	50.00	50.00	**50.00**	**78.03**	85.38	50.00	89.36	93.52
P19	71.60	68.54	50.00	50.00	50.00	77.97	77.94	77.80	50.00	77.92	77.89
P20	77.60	74.44	50.00	50.00	50.00	80.78	79.96	79.80	50.00	80.32	80.48
P21	80.52	79.30	50.00	50.00	50.00	**50.00**	**69.92**	73.51	50.00	75.74	78.23
P22	70.85	68.94	50.00	50.00	50.00	78.42	77.96	78.08	50.00	78.05	78.22
P23	82.47	80.32	50.00	50.00	50.00	84.72	84.34	84.47	50.00	84.66	84.58
P24	73.00	78.67	50.00	50.00	50.00	80.08	80.12	80.11	50.00	80.04	80.09
P25	87.04	58.58	50.00	50.00	50.00	82.12	82.18	82.26	50.00	82.17	82.39
P26	83.23	77.58	50.00	50.00	50.00	86.71	86.52	86.55	50.00	86.61	86.63
P27	83.24	76.48	50.00	50.00	50.00	81.17	**77.35**	82.60	50.00	82.71	83.20
P28	80.98	80.87	50.00	50.00	50.00	85.36	**64.69**	85.40	50.00	85.34	83.73
P29	80.81	85.06	50.00	50.00	50.00	84.12	84.06	84.13	50.00	83.87	83.86
P30	90.07	84.40	50.00	50.00	50.00	**50.00**	82.30	84.32	50.00	84.27	84.40
P31	86.21	80.91	50.00	50.00	50.00	82.28	83.80	83.33	50.00	83.55	83.60
P32	83.91	87.09	50.00	50.00	50.00	84.54	86.02	85.19	50.00	85.69	86.69
P33	76.62	75.50	50.00	65.27	50.00	80.05	80.59	80.62	50.00	80.90	81.34
P34	87.50	86.00	50.00	50.00	69.92	87.27	86.98	87.37	50.00	87.65	87.65
P35	66.72	70.62	50.00	50.00	50.00	74.24	75.32	74.28	50.00	74.77	74.95
P36	79.96	82.41	50.00	50.00	50.00	86.17	85.58	85.42	50.00	85.55	85.58
P37	90.42	85.43	50.00	50.00	50.00	90.96	89.81	90.25	50.00	90.20	90.64
P38	87.59	88.84	50.00	50.00	50.00	90.30	90.06	90.14	50.00	90.52	90.40
P39	79.93	83.36	50.00	50.00	50.00	85.09	84.65	84.65	50.00	84.90	84.98
P40	67.55	73.63	50.00	50.00	50.00	75.80	75.90	75.86	50.00	75.93	75.96

Table 5.2 Test accuracy (in %, the larger the better) w.r.t hyperparameter (α,β) with hyperparameter (μ,Σ) fixed to $(10^{-2},10^{-4})$, dropout rate $\theta = 0.5$, data augmentation number $T = 1$. The best parameter settings are marked in gray. Some parameter settings do not consistently perform very well and may fail on some participants (marked in bold).

Test ACC	SVR	LOR	(α,β) with (μ,Σ) fixed to $(10^{-2},10^{-4})$								
			(1,1)	(1,3)	(1,5)	(3,1)	(3,3)	(3,5)	(5,1)	(5,3)	(5,5)
P1	69.49	76.24	78.48	78.26	78.29	78.71	78.52	78.37	78.64	78.83	78.71
P2	76.35	83.56	81.62	82.13	82.09	82.03	81.75	82.17	82.04	81.98	82.03
P3	79.65	79.56	83.47	83.47	83.47	82.89	83.02	83.55	83.55	83.72	82.89
P4	65.68	70.33	73.34	73.65	73.68	73.26	73.42	73.65	73.38	73.46	73.26
P5	85.76	84.67	85.17	85.14	85.13	85.07	85.06	85.17	85.06	85.07	85.07
P6	74.84	64.54	84.23	84.31	84.31	84.56	84.64	84.40	85.13	85.05	84.56
P7	72.48	66.36	75.19	75.12	75.19	75.18	75.22	75.10	75.19	75.23	75.18
P8	78.60	78.60	84.30	84.54	84.5	84.70	84.66	84.46	84.61	84.53	84.70
P9	73.67	67.71	83.25	82.86	82.87	83.26	83.22	83.00	82.85	83.14	83.26
P10	92.55	91.93	89.14	88.56	88.56	89.50	89.21	88.44	88.43	88.41	89.50
P11	42.91	56.95	77.06	76.14	76.14	77.49	77.29	76.26	77.11	77.09	77.49
P12	70.18	78.28	80.13	80.09	80.13	80.09	80.13	80.09	79.94	79.94	80.09
P13	78.06	82.00	81.33	81.23	81.23	81.52	81.33	81.14	81.04	81.18	81.52
P14	77.26	77.40	79.70	80.00	80.03	80.03	79.70	80.03	79.46	79.49	80.03
P15	90.34	85.18	89.95	89.95	89.95	89.93	89.90	89.95	90.04	89.97	89.93
P16	76.03	72.76	72.44	72.37	72.37	72.41	72.42	72.33	72.50	72.42	72.41
P17	76.19	77.10	78.09	77.94	77.79	78.05	78.24	77.45	77.86	77.75	78.05
P18	61.09	77.12	88.38	87.82	87.79	89.36	88.88	88.00	87.31	87.92	89.36
P19	71.60	68.54	77.69	77.62	77.61	77.92	77.92	77.83	77.68	77.89	77.92
P20	77.60	74.44	80.29	80.29	80.28	80.32	80.29	80.30	80.33	80.34	80.32
P21	80.52	79.30	79.13	78.89	78.90	75.74	74.38	79.05	79.29	79.37	75.74
P22	70.85	68.94	77.96	77.95	77.96	78.05	77.99	77.98	77.95	77.96	78.05
P23	82.47	80.32	84.55	84.55	84.55	84.66	84.64	84.55	84.80	84.69	84.66
P24	73.00	78.67	79.98	79.99	79.99	80.04	80.01	79.98	80.12	80.09	80.04
P25	87.04	82.39	82.18	82.20	82.21	82.17	82.31	81.99	82.18	81.99	82.17
P26	83.23	77.58	86.61	86.61	86.61	86.61	86.60	86.61	86.59	86.60	86.61
P27	83.24	76.48	82.79	82.94	82.93	82.71	82.77	82.86	82.97	83.00	82.71
P28	80.98	80.87	85.37	85.34	85.29	85.34	85.31	85.33	85.23	85.32	85.34
P29	80.81	85.06	83.83	83.85	83.85	83.87	83.85	83.86	83.85	83.84	83.87
P30	90.07	84.40	84.07	84.08	84.14	84.27	84.19	84.03	84.08	84.08	84.27
P31	86.21	80.91	83.55	83.53	83.52	83.55	83.57	83.52	83.51	83.53	83.55
P32	83.91	87.09	85.66	86.62	86.56	85.69	85.65	86.54	86.46	86.51	85.69
P33	76.62	75.50	80.83	81.00	80.98	80.90	80.83	80.79	81.26	81.18	80.90
P34	87.50	86.00	87.47	87.65	87.65	87.65	87.59	87.62	87.40	87.47	87.65
P35	66.72	70.62	74.81	74.77	74.74	74.77	74.76	74.69	74.90	74.82	74.77
P36	79.96	82.41	85.50	85.55	85.55	85.55	85.52	85.50	85.47	85.52	85.55
P37	90.42	85.43	89.81	89.43	89.43	90.20	90.03	89.49	89.98	89.92	90.20
P38	87.59	88.84	90.48	90.28	90.28	90.52	90.52	90.28	90.44	90.48	90.52
P39	79.93	83.36	84.94	84.90	84.90	84.90	84.98	84.98	84.68	84.79	84.90
P40	67.55	73.63	75.93	75.93	75.92	75.93	75.92	75.91	75.86	75.89	75.93

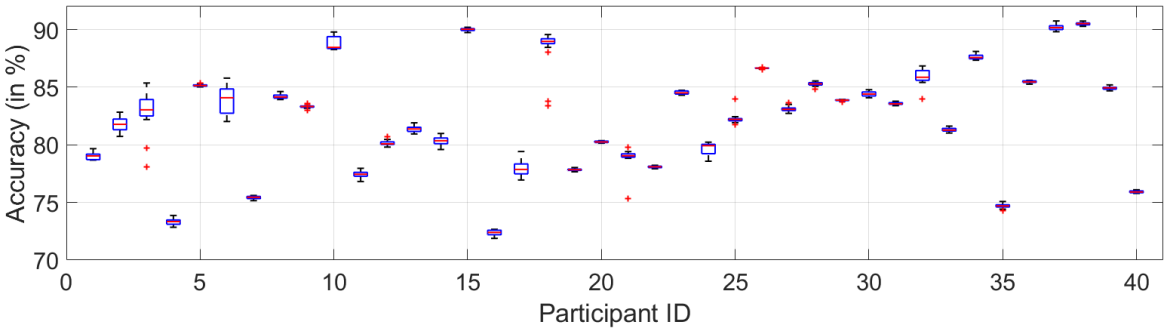


Figure 5.8 Box plot of the prediction accuracy on the test dataset. The symbol "+" denotes the outliers.

It can be observed from Figure 5.8 that: (1) the test accuracies of each participant are quite stable for different runs. (2) SWORE could consistently achieve high generalization performance (test accuracy above 80%) on 26 over forty participants with 95% confidence. Note that the performance of each specific participant can be further improved by tailor-designed brain dynamic preferences for each participant.

5.6 Summary of This Chapter

In this chapter, I propose a Self-Weight Ordinal REgression (SWORE) model with Brain Dynamics table (BDtable) for real-time mental fatigue monitoring. SWORE could aggregate the information from multiple noisy channels based on the brain dynamic preferences; while BDtable is used to online calibrate the SWORE model utilizing a generalized Bayesian moment matching algorithm. Empirical results demonstrate that the proposed framework achieves significantly better performance than baseline approaches like SVR, LOR, online SVR and online LOR. As a direction for future research, I am committed to assess and evaluate the feasibility of performing online mental-fatigue monitoring system directly with raw EEG signals in the real-world environment. Moreover, it remains open to introduce domain knowledge to tailor-design task guided mechanism for maintaining the BDtable online.

Chapter 6

Conclusion and Future Work

In this chapter, I first conclude the entire thesis and then elaborate on possible trends for future research.

6.1 Conclusion

Originating in social choice theory [Lijphart, 1994, Saari, 1999], RA has become a popular machine learning paradigm and has significantly attracted the attention of researchers as a result of its capability for modeling structured data. Furthermore, the preferences could arise not only by explicitly querying users, but also through passive data collection, that is, by observing user purchasing behavior [Baltrunas et al., 2010], clicks on search engine results [Dwork et al., 2001], etc. The flexible collection of preferences enables successful application of RA in various fields, from image rating [Liang and Grauman, 2014] to peer grading [Raman and Joachims, 2014], and bioinformatics [Kim et al., 2014].

Meanwhile, a basic assumption underlying the vanilla RA is that all preferences are provided by homogeneous users, sharing the same annotation accuracy and agreeing with the single ground truth ranking. Due to the flexible data construction, the homogeneity assumption underlying the vanilla RA is hardly satisfied in real applications. Therefore, vanilla RAs usually suffer from two important problems:

- Reliability. Typical preference annotation tasks are tedious and annotators usually come from a diverse pool, including genuine experts, amateurs, biased workers, and malicious annotators. Annotation generated by the crowd suffer from low quality.
- Scalability. RA usually involves massive or even an unlimited number of items in real applications. It would further lead to an exponential volume of preferences for aggregation.

To answer the above problems, Chapter 3 first considered RA in a crowdsourcing scenario, where sufficient annotations from each crowd worker are available. This enables me to explore the heterogeneity of users to enhance the model reliability. To address these challenges, I proposed a reliable CrowdsOUrced Plackett-LucE (COUPLE) model with an efficient Bayesian learning inference method. In particular, COUPLE models the annotation process of a partial preference as a series of sequential comparison stages. In each stage, compared to all the remaining alternatives, one object, regarded as the "local winner", was selected preferentially (without replacement). However, due to crowdsourced workers' limited expertise, stagewise strategy was adversely affected by workers' hesitation at each stage. To ensure reliability, I proposed a robust stagewise learning strategy, which revealed their hesitation to select the local winner at each stage and helped to correct (in expectation) the misordered objects in the noisy preferences. To ensure the scalability of COUPLE, I proposed an efficient Bayesian moment matching method, which leads COUPLE to naturally online update. Specifically, I projected the intractable Bayesian posterior onto a family of tractable distributions after each observation, by matching a set of sufficient moments. Empirical results showed that COUPLE with the Bayesian moment matching method achieved substantial improvements in reliability over current approaches.

In Chapter 4, I discussed RA under model misspecification. The term "model misspecification" here refers to the mismatch between the ranking model assumption and the ranking dataset, namely the collected user preferences do not strictly satisfy the user homogeneity assumption of the ranking model. Therefore, the model misspecification issue still arises in previous model attempts because:

- Each user usually provides one preference in real applications, which would cause overfitting since I need to estimate the reliability of each preference.
- The above attempt amounts to convolving the original ranking model with some pre-assumed perturbation mechanism. This leads to a new model with a few more parameters but is just as bound to be misspecified with regard to other overlooked perturbations.
- The perturbation patterns leading to model misspecification vary from setting to setting, it is impossible to design a universal practice that can be generalized to most settings.

Therefore, I considered RA against model misspecification in Chapter 4 from another perspective. In particular, I presented a novel RA approach, called CoarsenRank. The main idea of CoarsenRank is to perform regular RA over a neighborhood of the collected inconsistent preferences, which enables CoarsenRank against most potential perturbations

within the defined neighborhood. However, it is usually intractable to infer directly over the neighborhood of the ranking data because of the unlimited samples involved. Furthermore, it also prohibits sampling-based stochastic gradient solutions adopted for distributional robustness in the optimization community, due to the particularity of the ranking data. For the sake of tractability, the relative entropy is adopted as the divergences metric to define the neighborhood. I further introduced a prior distribution for the unknown size of the neighborhood, to avoid parameter tuning, and derive a much-simplified formula for CoarsenRank. Furthermore, CoarsenRank is instantiated using three popular probability ranking models, followed by the corresponding optimization strategies. Empirical results on four real-world datasets demonstrate the superior reliability and efficiency of CoarsenRank over other baselines.

Mental fatigue is a common physiological phenomenon, which induces sub-optimal functioning and may even lead to accidents with severe consequences. Some previous works, derived from linear and non-linear regression methods, show that it is possible to detect mental fatigue with high accuracy. This is impressive, but it is rather blind to the wealth of the dynamics and behavioral variability available only to offline analysis methods with sufficient training samples. In particular, previous attempts would suffer from three major drawbacks for Real-time Mental Fatigue Monitoring:

- poor generalization when extreme values commonly exist for the response variable;
- lacking an efficient calibration strategy for online updating of the learning model online; and
- lacking efficient aggregation mechanism to aggregate the predictions from multiple noisy channels.

Based on the above discussion, it is clear that there is a requirement of the learning model which can work in online fashion while being calibrated in real-time. In Chapter 5, I applies RA to real-time mental fatigue monitoring. Particularly, I consider the RT as the item score to construct brain dynamics preferences and viewed each EEG channel as a crowd worker. The mental fatigue monitoring task could then be formulated as RA under model misspecification, while involving the EEG signal as the features. To address this problem, a Self-Weight Ordinal REgression (SWORE) model with Brain Dynamics table (BDtable) was proposed. The SWORE model learns from brain dynamics preferences from multiple noisy channels by learning the reliability of each channel explicitly within the aggregation process; while the BDtable maintains the landmark EEG signals and the corresponding RTs as the reference in real-time (online). Whenever a new EEG signal comes at time t, the

SWORE model could give a coarse estimation of its reaction time by interpolating it among the bunch of maintained RTs using the brain dynamics preferences. An online generalized Bayesian moment matching (OGMM) algorithm is further proposed for Bayesian posterior updating. Once the real reaction time is available, the BDtable would help online calibrating of the SWORE model by utilizing the simple analytic update rules introduced in the OGMM algorithm. I explored the reliability of SWORE in a real-time mental fatigue evaluation task and analyzed the parameter sensitivity and model uncertainty of the SWORE with regard to the OGMM algorithm.

6.2 Future Work

In this thesis, several methods and algorithms are proposed to address RA under model misspecification from different perspectives. Some other perspectives are also very interesting and should be further investigated:

- **Adversarial Rank Aggregation:** Adversarial training, intentionally injecting adversarial examples into training data to mislead deep neural networks, have attracted significant attention in the past few years. Specifically, adversarial training solves a min-max optimization problem, with the inner maximization generating adversarial examples by maximizing the loss, and the outer minimization finding model parameters by minimizing the loss on adversarial examples generated from the inner maximization. In particular, I discussed the connection between CoarsenRank and the min-max optimization problem in Remark 4. Meanwhile, since CoarsenRank needs to perform analytic integration over the neighborhood, it poses a strong requirement for the choice of distance measure. For easy tractability concerns, I adopted relative entropy, as the distribution distance to define the distribution-level neighborhood. However, the min-max optimization problem does not suffer from this issue, because it calculates the worst cases directly over the defined neighborhood. Regarding other distance measures, for example, Kendall tau distance, where CoarsenRank would be difficult to infer, or even intractable, adversarial training could be a promising alternative for RA to enhance distributional robustness.

- **Online CoarsenRank and convergence analysis:** Online Ranking is a very challenging but practical problem in real applications, where preferences come into the ranking system in a sequential style and the global rankings are updated accordingly. Elo [Elo, 1978] and Glicko [Glickman, 1999] are famous online ranking systems. Herbrich et al. [2007] developed TrueSkill, which constructs a graphical model and

inferences with approximate message passing. Weng and Lin [2011] introduced a Bayesian approximation method to derive simple analytic rules for inference in RA, such as OnlineBT and OnlinePL. However, current online ranking methods fail to consider the model misspecification issue, which would lead to inferior performance when aggregating low-quality preferences. I proposed CoarsenRank and resorted to a closed-form EM algorithm to do infer. Although I solved CoarsenRank with a batch optimization method, the closed-form EM solution denotes that the closed-form Bayesian updating rules are also available for online ranking under the mean-field assumption. This would greatly enlarge the scope of application for CoarsenRank. Meanwhile, the convergence analysis of online CoarsenRank would also become an important topic for further study.

Appendix A

Appendix

A.1 Proof for Proposition 1

Before I give the proof for Proposition 1, I first introduce the preliminary results proved in Weng and Lin [2011].

Lemma 2 *Let $Z = (Z_1, Z_2, \ldots, Z_L)^T$ be a random vector, where each entry is independent and $Z_r \sim N(0,1)$, $r = 1,2,\ldots,L$. Suppose that $f(Z)$ is the likelihood function and almost twice differentiable. Then, the mean and the variance of the posterior distribution can be approximated as*

$$E[Z] = E\left[\frac{\nabla f(Z)}{f(Z)}\right], \tag{A.1a}$$

$$E[Z_p Z_q] = \mathbf{1}_{pq} + E\left[\frac{\nabla^2 f(Z)}{f(Z)}\right]_{pq}, \quad p,q = 1,\ldots,L \tag{A.1b}$$

where $\mathbf{1}_{pq} = 1$ if $p = q$ and 0 otherwise, and $\left[.\right]_{pq}$ denotes the (p,q) component of a matrix. ■

In cases with a general normal prior, I instantiate Lemma 2 with COUPLE, and introduce Proposition 1 to deal with more general situations.

Proposition 1 Let $Z = (Z_1, Z_2, \ldots, Z_L)^T$, where $Z_r = \frac{\theta_r - \mu_r}{\sigma_r} \sim N(0,1)$, $r = 1,2,\ldots,L$. Assume $l(Z)$ is the likelihood $P(\tilde{X} = \rho^i|\theta)$ and almost twice differentiable. Upon the completion

of stage i, the posterior parameters $(\mu_r^{new}, (\sigma_r^2)^{new})$ of score θ_r can be estimated as:

$$\mu_r^{new} = \mu_r + \sigma_r E[\frac{\partial l(Z)/\partial Z_r}{l(Z)}], \tag{A.2a}$$

$$(\sigma_r^2)^{new} = \sigma_r^2(1 + E[\frac{\partial^2 l(Z)/\partial^2 Z_r}{l(Z)}]_{rr} - E[\frac{\partial l(Z)/\partial Z_r}{l(Z)}]^2), \tag{A.2b}$$

where $r = 1, 2, \ldots, L$.

Proof: Substituting Z_r in Lemma A.1a with general form $\theta_r = \mu_r + \sigma_r * Z_r$ and replace the likelihood function, will result in Proposition A.2a after simplifying. Similarly, I can result in Proposition A.2b with the same procedure. ■

A.2 Detailed derivations for Equation 3.8

$$\begin{aligned}
E[\eta_w^t] &= \int \eta_w^t P(\eta_w|\tilde{X} = \rho^i) d\eta_w = \int \eta_w^t \frac{P(\tilde{X} = \rho^i|\eta_w)\mathrm{Dir}(\eta_w|\alpha_w)}{R} d\eta_w \\
&= \frac{1}{R}\int \eta_w^t \sum_{v=1}^{\tilde{k}} (\eta_w^v \times R_v)\mathrm{Dir}(\eta_w|\alpha_w) d\eta_w = \frac{1}{R}\sum_{v=1}^{\tilde{k}} R_v \int \eta_w^t \eta_w^v \mathrm{Dir}(\eta_w|\alpha_w) d\eta_w \\
&= \frac{1}{R}(\frac{\sum_{v=1}^{\tilde{k}} R_v \alpha_w^t \alpha_w^v + R_v \alpha_w^t}{(\sum_{m=1}^{\tilde{k}} \alpha_w^m + 1)(\sum_{m=1}^{\tilde{k}} \alpha_w^m)}) = \frac{\alpha_w^t(\sum_{v=1}^{\tilde{k}}(R_v \times \alpha_w^v) + R_t)}{R(\sum_{v=1}^{\tilde{k}} \alpha_w^v + 1)(\sum_{v=1}^{\tilde{k}} \alpha_w^v)}. \\
E[(\eta_w^t)^2] &= \int (\eta_w^t)^2 P(\eta_w|\tilde{X} = \rho^i) d\eta_w = \int (\eta_w^t)^2 \frac{P(\tilde{X} = \rho^i|\eta_w)\mathrm{Dir}(\eta_w|\alpha_w)}{R} d\eta_w \\
&= \frac{1}{R}\int (\eta_w^t)^2 \sum_{v=1}^{\tilde{k}} (\eta_w^v \times R_v)\mathrm{Dir}(\eta_w|\alpha_w) d\eta_w = \frac{1}{R}\sum_{v=1}^{\tilde{k}} R_v \int (\eta_w^t)^2 \eta_w^v \mathrm{Dir}(\eta_w|\alpha_w) d\eta_w \\
&= \frac{1}{R}(\frac{\sum_{v=1}^{\tilde{k}} R_v(\alpha_w^t + 1)\alpha_w^t \alpha_w^v + 2R_v(\alpha_w^t + 1)\alpha_w^t}{(\sum_{m=1}^{\tilde{k}} \alpha_w^m + 2)(\sum_{m=1}^{\tilde{k}} \alpha_w^m + 1)(\sum_{m=1}^{\tilde{k}} \alpha_w^m)}) \\
&= \frac{\alpha_w^t(\alpha_w^t + 1)(\sum_{v=1}^{\tilde{k}}(R_v \times \alpha_w^v) + 2R_t)}{R(\sum_{v=1}^{\tilde{k}} \alpha_w^v + 2)(\sum_{v=1}^{\tilde{k}} \alpha_w^v + 1)(\sum_{v=1}^{\tilde{k}} \alpha_w^v)}.
\end{aligned}$$

A.3 Proof for Theorem 1

Theorem 1 Suppose $D(\mathscr{R}_N, \mathfrak{R}_N)$ is an almost surely (a.s.)-consistent estimator[1] of $D(P_o, P_\theta)$, namely $D(\mathscr{R}_N, \mathfrak{R}_N) \xrightarrow[N\to+\infty]{\text{a.s.}} D(P_o, P_\theta)$, where $F_N(\mathscr{R}_N) \to P_o$ and $F_N(\mathfrak{R}_N) \to P_\theta$ when $N \to$

[1] In probability theory, an event happens almost surely (a.s.) if it happens with probability one.

$+\infty$. Assume $\mathbb{P}(D(P_o,P_\theta)=\epsilon)=0$ and $\mathbb{P}(D(P_o,P_\theta)<\epsilon)>0$, then I have

$$\mathbb{P}(\theta|D(\mathscr{R}_N,\mathfrak{R}_N)<\epsilon) \xrightarrow[N\to+\infty]{\text{a.s.}} \mathbb{P}(\theta|D(P_o,P_\theta)<\epsilon),$$

for any $\theta\in\Theta$ such that $\int|\theta|\mathbb{P}(d\theta)<\infty$.

Proof: Since $\mathbb{P}(D(P_o,P_\theta)=\epsilon)=0$, I have $\mathbb{1}(D(\mathscr{R}_N,\mathfrak{R}_N)<\epsilon)\xrightarrow[N\to+\infty]{\text{a.s.}}\mathbb{1}(D(P_o,P_\theta)<\epsilon)$[2]. Then I have $|\mathbb{1}(D(\mathscr{R}_N,\mathfrak{R}_N)<\epsilon)|<1$ hold $\forall N>0$ since $0\le\mathbb{1}(\cdot)\le 1$. According to the dominated convergence theorem [Billingsley, 2013], I have $\mathbb{P}(D(\mathscr{R}_N,\mathfrak{R}_N)<\epsilon)\xrightarrow[N\to+\infty]{\text{a.s.}}\mathbb{P}(D(P_o,P_\theta)<\epsilon)$.

Similarly, I have $\theta\mathbb{1}(D(\mathscr{R}_N,\mathfrak{R}_N)<\epsilon))\xrightarrow[N\to+\infty]{\text{a.s.}}\theta\mathbb{1}(D(P_o,P_\theta)<\epsilon))$. Since $0\le\mathbb{1}(\cdot)\le 1$, $|\theta\mathbb{1}(D(\mathscr{R}_N,\mathfrak{R}_N)<\epsilon))|\le|\theta|$, $\forall N>0$. Therefore, I have $\mathbb{P}(\theta\mathbb{1}(D(\mathscr{R}_N,\mathfrak{R}_N)<\epsilon))\xrightarrow[N\to+\infty]{\text{a.s.}}\mathbb{P}(\theta\mathbb{1}(D(P_o,P_\theta)<\epsilon))$, according to the dominated convergence theorem and $\int|\theta|\mathbb{P}(d\theta)<\infty$. Above all, I have

$$\begin{aligned}\mathbb{P}(\theta|D(\mathscr{R}_N,\mathfrak{R}_N)<\epsilon)&=\frac{\mathbb{P}(\theta\mathbb{1}(D(\mathscr{R}_N,\mathfrak{R}_N)<\epsilon))}{\mathbb{P}(D(\mathscr{R}_N,\mathfrak{R}_N)<\epsilon)}\\&\xrightarrow[N\to+\infty]{\text{a.s.}}\frac{\mathbb{P}(\theta\mathbb{1}(D(P_o,P_\theta)<\epsilon))}{\mathbb{P}(D(P_o,P_\theta)<\epsilon)}=\mathbb{P}(\theta|D(P_o,P_\theta)<\epsilon).\end{aligned}$$ ■

A.4 Proof for Corollary 2

Before introducing Corollary 2, I first introduce Lemma 3 which contains some preliminary results from Miller and Dunson [2018].

Lemma 3 *Let $\Delta_k=\{p\in\mathbb{R}^k:\sum_i^k p_i=1,p_i>0\ \forall i\}$. Let $q\in\Delta_k$. I argue that if $X_1,\dots,X_N$ i.i.d. $\sim q$ and $F_N(X_{1:N})=\frac{1}{N}\sum_{n=1}^N\delta_{X_n}(x)$, then for $p\in\Delta_k$ near q,*

$$\mathbb{E}_{X_{1:N}\sim q}[\exp(-\alpha\mathscr{D}_{\mathrm{KL}}(p\|F_N(X_{1:N})))]\ \approx\ \left(\frac{N\tau_N}{\alpha}\right)^{\frac{k-1}{2}}\exp(-N\tau_N\mathscr{D}_{\mathrm{KL}}(p\|q)),$$

where $\tau_N=\frac{1/N}{1/N+1/\alpha}$. ■

Corollary 2 If $D(\mathscr{R}_N,\mathfrak{R}_N)$ is an almost surely (a.s.)-consistent estimator of $\mathscr{D}_{\mathrm{KL}}(P_o\|P_\theta)=\int P_o\log\frac{P_o}{P_\theta}$, and ϵ is subject to an exponential prior, i.e., $\epsilon\sim\mathrm{Exp}(\alpha)$, I obtain the following

[2] $\mathbb{1}(x)$ denotes the indicator function, which returns one when x is true and zero, otherwise.

approximation to the posterior:

$$\mathbb{P}(\theta|D(\mathscr{R}_N,\mathfrak{R}_N)<\epsilon) \;\approx\; \Pi(\theta)\prod_{n=1}^{N} P_\theta^{\tau_N}(\rho_n),$$

where $\approx$ denotes that the term on the left is approximately equal to a term, which is proportional to the expression on the right, and $\tau_N = \frac{1/N}{1/N+1/\alpha}$.

Proof: According to Bayesian theory, I have

$$\mathbb{P}(\theta|D(\mathscr{R}_N,\mathfrak{R}_N)<\epsilon) \overset{i}{\propto} \mathbb{P}(D(\mathscr{R}_N,\mathfrak{R}_N)<\epsilon)|\theta)\Pi(\theta),$$

where *i* holds because of the omission of the normalization constant with respect to θ.

Further, I have

$$\begin{aligned}\mathbb{P}(D(\mathscr{R}_N,\mathfrak{R}_N)<\epsilon\,|\theta) &\overset{i}{=} \mathbb{E}_{\mathfrak{R}_N\sim P_\theta}(\exp(-\alpha\mathscr{D}_{\mathrm{KL}}(F_N(\mathscr{R}_N)\|F_N(\mathfrak{R}_N)))|\theta)\\ &\overset{ii}{\approx} \exp(-N\tau_N\mathscr{D}_{\mathrm{KL}}(F_N(\mathscr{R}_N)\|P_\theta))\\ &\overset{iii}{\propto} \exp\left(N\tau_N\int F_N(\mathscr{R}_N)\log P_\theta\right) \overset{iv}{=} \prod_{n=1}^{N} P_\theta^{\tau_N}(\rho_n).\end{aligned}$$

where $\tau_N = \frac{1/N}{1/N+1/\alpha}$. *i* follows Corollary 1. *ii* follows Lemma 3. *iii* holds due to the removal of the constant entropy term $\int F_N(\mathscr{R}_N)\log F_N(\mathscr{R}_N)$, which is a constant w.r.t. the model parameter θ. *iv* holds according to the definition of the empirical data distribution $F_N(\mathscr{R}_N) = \frac{1}{N}\sum_{n=1}^{N}\delta_{\rho_n}(x)$. ■

A.5 Proof for Theorem 2

Before introducing Theorem 2, I first introduce the Lemma 4 as a building block.

Lemma 4 *([Weng and Lin, 2011]) Let z be a random vector, where each entry is independent and $z_i \sim \mathscr{N}(0,1)$, $i = 1,2,\cdots,L$. Suppose that $f(z)$ is the likelihood function and almost twice differentiable. Then the first and second order moments of the posterior distribution can be estimated as*

$$E[z] = E[\frac{\nabla f(z)}{f(z)}], \tag{A.3a}$$

$$E[z_i z_j] = I_{ij} + E[\frac{\nabla^2 f(z)}{f(z)}]_{ij}, \quad i,j = 1,\cdots,L \tag{A.3b}$$

where $I_{ij} = 1$ *if* $i = j$ *and* 0 *otherwise, and* $\left[.\right]_{ij}$ *indicates the* (i, j) *component of a matrix.*

Theorem 2 Assume $f(w)$ is the marginalized likelihood of one brain dynamics preference and almost twice differentiable. Upon updating this preference, the posterior parameters $(\mu^{new}, \Sigma^{new})$ of weight w can be estimated as:

$$\mu^{new} \approx \mu + \Sigma \times \frac{d \log f(w)}{dw}\Big|_{w=\mu},$$
$$\Sigma^{new} \approx \Sigma + \Sigma \times \frac{d^2 \log f(w)}{dw dw^T}\Big|_{w=\mu} \times \Sigma.$$

where I set $w = \mu$ as I expect that the posterior density of w to be concentrated on μ [Weng and Lin, 2011].

Proof: In terms of the posterior parameter μ^{new}, I have

$$\begin{aligned}
\mu^{new} = E_w[w] = \mu + \Sigma^{1/2} \times E_z[z] &\overset{i}{=} \mu + \Sigma^{1/2} \times E_z\left[\frac{\nabla f(\mu + \Sigma^{1/2} z)}{f(\mu + \Sigma^{1/2} z)}\right] \\
&\overset{ii}{\approx} \mu + \Sigma^{1/2} \times \frac{d \log f(\mu + \Sigma^{1/2} z)}{dz}\Big|_{z=\mathbf{0}} \overset{iii}{=} \mu + \Sigma^{1/2} \times \frac{dw}{dz} \times \frac{d \log f(\mu + \Sigma^{1/2} z)}{dw}\Big|_{w=\mu} \\
&= \mu + \Sigma \times \frac{d \log f(w)}{dw}\Big|_{w=\mu}.
\end{aligned}$$

where *i* follows the Equation A.3a of Lemma 4. *ii* sets $z = 0$. Such a substitution is reasonable as I expect that the posterior density of z to be concentrated on 0. *iii* follows the chain rule.

In terms of the posterior parameter Σ^{new}, I have

$$
\begin{aligned}
\Sigma^{new} &= \mathrm{Var}(w) = \Sigma^{1/2} \times (E_z[zz^T] - E_z[z]E_z^T[z]) \times \Sigma^{1/2} \\
&\overset{i}{=} \Sigma^{1/2} \times \Big(\mathbf{I} + E_z\big[\frac{\nabla^2 f(\mu+\Sigma^{1/2}z)}{f(\mu+\Sigma^{1/2}z)}\big] - E_z\big[\frac{d\log f(\mu+\Sigma^{1/2}z)}{dz}\frac{d\log f(\mu+\Sigma^{1/2}z)}{dz^T}\big]\Big) \times \Sigma^{1/2} \\
&\overset{ii}{\approx} \Sigma^{1/2} \times \Big(\mathbf{I} + \frac{\nabla^2 f(\mu+\Sigma^{1/2}z)}{f(\mu+\Sigma^{1/2}z)}\Big|_{z=\mathbf{0}} - \frac{d\log f(\mu+\Sigma^{1/2}z)}{dz}\Big|_{z=\mathbf{0}}\frac{d\log f(\mu+\Sigma^{1/2}z)}{dz^T}\Big|_{z=\mathbf{0}}\Big) \times \Sigma^{1/2} \\
&\overset{iii}{=} \Sigma + \Sigma^{1/2} \times \frac{d^2\log f(\mu+\Sigma^{1/2}z)}{dzdz^T}\Big|_{z=\mathbf{0}} \times \Sigma^{1/2} \\
&\overset{iv}{=} \Sigma + \Sigma^{1/2} \times \frac{dw}{dz} \times \frac{d^2\log f(\mu+\Sigma^{1/2}z)}{dwdw^T}\Big|_{w=\mu} \times \frac{dw^T}{dz^T} \times \Sigma^{1/2} \\
&= \Sigma + \Sigma \times \frac{d^2\log f(w)}{dwdw^T}\Big|_{w=\mu} \times \Sigma.
\end{aligned}
$$

where *i* follows the Equation A.3b of Lemma 4. *ii* sets $z=0$. Such a substitution is reasonable as I expect that the posterior density of z to be concentrated on 0. *iv* follows the chain rule. I give the proof for *iii* as follows,

$$
\begin{aligned}
\Big[\frac{d^2\log f(\mu+\Sigma^{1/2}z)}{dzdz^T}\Big]_{ij} &= \frac{\partial}{\partial z_j}\Big(\frac{\partial f(\mu+\Sigma^{1/2}z)/\partial z_i}{f(\mu+\Sigma^{1/2}z)}\Big) \\
&= \frac{\frac{\partial^2 f(\mu+\Sigma^{1/2}z)}{\partial z_i \partial z_j} f(\mu+\Sigma^{1/2}z) - \frac{\partial f(\mu+\Sigma^{1/2}z)}{\partial z_i} \times \frac{\partial f(\mu+\Sigma^{1/2}z)}{\partial z_j}}{f^2(\mu+\Sigma^{1/2}z)} \\
&= \Big[\frac{\nabla^2 f(\mu+\Sigma^{1/2}z)}{f(\mu+\Sigma^{1/2}z)}\Big]_{ij} - \frac{\partial\log f(\mu+\Sigma^{1/2}z)}{\partial z_i} \times \frac{\partial\log f(\mu+\Sigma^{1/2}z)}{\partial z_j} \\
&= \Big[\frac{\nabla^2 f(\mu+\Sigma^{1/2}z)}{f(\mu+\Sigma^{1/2}z)}\Big]_{ij} - \Big[\frac{d\log f(\mu+\Sigma^{1/2}z)}{dz}\Big]_i \times \Big[\frac{d\log f(\mu+\Sigma^{1/2}z)}{dz^T}\Big]_j.
\end{aligned}
$$

■

A.6 Detailed derivations for Equation 5.12

$$
\begin{aligned}
E[\pi_n] &= \int \pi_n P(\pi_n|y,\Delta x)d\pi_n = \int \pi_n \frac{P(y|\pi_n,\Delta x)\text{Beta}(\pi_n|\alpha_n,\beta_n)}{R}d\pi_n \\
&= \frac{1}{R}\int \pi_n[\pi_n R_1 + (1-\pi_n)R_2]\text{Beta}(\pi_n|\alpha_n,\beta_n)d\pi_n \\
&= \frac{R_1-R_2}{R}\int \pi_n^2\text{Beta}(\pi_n|\alpha_n,\beta_n)d\pi_n + \frac{R_2}{R}\int \pi_n\text{Beta}(\pi_n|\alpha_n,\beta_n)d\pi_n \\
&= \frac{R_1-R_2}{R}\frac{(\alpha_n+1)\alpha_n}{(\alpha_n+\beta_n+1)(\alpha_n+\beta_n)} + \frac{R_2}{R}\frac{\alpha_n}{\alpha_n+\beta_n} \\
&= \frac{R_1(\alpha_n+1)\alpha_n + R_2\alpha_n\beta_n}{R(\alpha_n+\beta_n+1)(\alpha_n+\beta_n)}.
\end{aligned}
$$

$$
\begin{aligned}
E[\pi_n^2] &= \int \pi_n^2 P(\pi_n|y,\Delta x)d\pi_n = \int \pi_n^2 \frac{P(y|\pi_n,\Delta x)\text{Beta}(\pi_n|\alpha_n,\beta_n)}{R}d\pi_n \\
&= \frac{1}{R}\int \pi_n^2[\pi_n R_1 + (1-\pi_n)R_2]\text{Beta}(\pi_n|\alpha_n,\beta_n)d\pi_n \\
&= \frac{R_1-R_2}{R}\int \pi_n^3\text{Beta}(\pi_n|\alpha_n,\beta_n)d\pi_n + \frac{R_2}{R}\int \pi_n^2\text{Beta}(\pi_n|\alpha_n,\beta_n)d\pi_n \\
&= \frac{R_1-R_2}{R}\frac{(\alpha_n+2)(\alpha_n+1)\alpha_n}{(\alpha_n+\beta_n+2)(\alpha_n+\beta_n+1)(\alpha_n+\beta_n)} + \frac{R_2}{R}\frac{(\alpha_n+1)\alpha_n}{(\alpha_n+\beta_n+1)(\alpha_n+\beta_n)} \\
&= \frac{\alpha_n(\alpha_n+1)[R_1(\alpha_n+2)+R_2\beta_n]}{R(\alpha_n+\beta_n+2)(\alpha_n+\beta_n+1)(\alpha_n+\beta_n)}.
\end{aligned}
$$

References

Abadeh, S. S., Esfahani, P. M. M., and Kuhn, D. (2015). Distributionally robust logistic regression. In *Advances in Neural Information Processing Systems*, pages 1576–1584.

Aerts, S., Lambrechts, D., Maity, S., Van Loo, P., et al. (2006). Gene prioritization through genomic data fusion. *Nature biotechnology*, 24(5):537.

Ailon, N., Charikar, M., and Newman, A. (2008). Aggregating inconsistent information: ranking and clustering. *Journal of the ACM (JACM)*, 55(5):23.

Ali, S. M. and Silvey, S. D. (1966). A general class of coefficients of divergence of one distribution from another. *Journal of the Royal Statistical Society: Series B (Methodological)*, 28(1):131–142.

Baltrunas, L., Makcinskas, T., and Ricci, F. (2010). Group recommendations with rank aggregation and collaborative filtering. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 119–126. ACM.

Barachant, A., Bonnet, S., Congedo, M., and Jutten, C. (2012). Multiclass brain–computer interface classification by riemannian geometry. *IEEE Transactions on Biomedical Engineering*, 59(4):920–928.

Ben-Tal, A., Den Hertog, D., De Waegenaere, A., Melenberg, B., and Rennen, G. (2013). Robust solutions of optimization problems affected by uncertain probabilities. *Management Science*, 59(2):341–357.

Berrada, G. and Cheney, J. (2019). Aggregating unsupervised provenance anomaly detectors. In *11th International Workshop on Theory and Practice of Provenance (TaPP 2019)*.

Bhattacharya, A., Pati, D., Yang, Y., et al. (2019). Bayesian fractional posteriors. *The Annals of Statistics*, 47(1):39–66.

Billingsley, P. (2013). *Convergence of probability measures*. John Wiley & Sons.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer.

Blanchet, J., Kang, Y., and Murthy, K. (2016). Robust wasserstein profile inference and applications to machine learning. *arXiv preprint arXiv:1610.05627*.

Borghini, G., Astolfi, L., Vecchiato, G., Mattia, D., and Babiloni, F. (2014). Measuring neurophysiological signals in aircraft pilots and car drivers for the assessment of mental workload, fatigue and drowsiness. *Neuroscience & Biobehavioral Reviews*, 44:58–75.

Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

Bradley, R. A. and Terry, M. E. (1952). Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

Caron, F. and Doucet, A. (2012). Efficient Bayesian inference for generalized Bradley–Terry models. *Journal of Computational and Graphical Statistics*, 21(1):174–196.

Chen, R. and Paschalidis, I. (2018a). Outlier detection using robust optimization with uncertainty sets constructed from risk measures. *ACM SIGMETRICS Performance Evaluation Review*, 45(3):174–179.

Chen, R. and Paschalidis, I. C. (2018b). A robust learning approach for regression models based on distributionally robust optimization. *The Journal of Machine Learning Research*, 19(1):517–564.

Chen, X., Bennett, P. N., Collins-Thompson, K., and Horvitz, E. (2013). Pairwise ranking aggregation in a crowdsourced setting. In *Proceedings of the sixth ACM international conference on Web search and data mining*, pages 193–202. ACM.

Colosio, M., Shestakova, A., Nikulin, V. V., Blagovechtchenski, E., and Klucharev, V. (2017). Neural mechanisms of cognitive dissonance (revised): An eeg study. *Journal of Neuroscience*, pages 3209–16.

Congedo, M., Barachant, A., and Bhatia, R. (2017). Riemannian geometry for eeg-based brain-computer interfaces; a primer and a review. *Brain-Computer Interfaces*, 4(3):155–174.

Daniel, W. (1990). *Applied nonparametric statistics*. The Duxbury advanced series in statistics and decision sciences. PWS-Kent Publ.

Daunizeau, J. (2017). Semi-analytical approximations to statistical moments of sigmoid and softmax mappings of normal variables. *arXiv preprint arXiv:1703.00091*.

de Borda, J. C. (1781). Mémoire sur les élections au scrutin.

Desarkar, M. S., Sarkar, S., and Mitra, P. (2016). Preference relations based unsupervised rank aggregation for metasearch. *Expert Systems with Applications*, 49:86–98.

Diaconis, P. (1988). Group representations in probability and statistics. *Lecture Notes–Monograph Series, Hayward, CA: Institute of Mathematical Statistics*, 11:i–192.

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

Domshlak, C., Gal, A., and Roitman, H. (2007). Rank aggregation for automatic schema matching. *IEEE Transactions on Knowledge and Data Engineering*, 19(4):538–553.

Dwork, C., Kumar, R., Naor, M., and Sivakumar, D. (2001). Rank aggregation methods for the web. In *WWW*, pages 613–622. ACM.

Elo, A. E. (1978). *The rating of chessplayers, past and present*. Arco Pub.

Fagin, R., Kumar, R., and Sivakumar, D. (2003). Efficient similarity search and classification via rank aggregation. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*, pages 301–312. ACM.

Fligner, M. A. and Verducci, J. S. (1986). Distance based ranking models. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 359–369.

Fu, Y., Hospedales, T. M., Xiang, T., Gong, S., and Yao, Y. (2014). Interestingness prediction by robust learning to rank. In *European conference on computer vision*, pages 488–503. Springer.

Fu, Y., Hospedales, T. M., Xiang, T., Xiong, J., Gong, S., Wang, Y., and Yao, Y. (2015). Robust subjective visual property prediction from crowdsourced pairwise labels. *IEEE transactions on pattern analysis and machine intelligence*, 38(3):563–577.

Fürnkranz, J. and Hüllermeier, E. (2010). *Preference learning*. Springer.

Gao, R., Chen, X., and Kleywegt, A. J. (2017). Wasserstein distributional robustness and regularization in statistical learning. *arXiv preprint arXiv:1712.06050*.

Gleich, D. F. and Lim, L.-h. (2011). Rank aggregation via nuclear norm minimization. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 60–68. ACM.

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

Goodfellow, I., Bengio, Y., Courville, A., and Bengio, Y. (2016). *Deep learning*, volume 1. MIT press Cambridge.

Gormley, I. C. and Murphy, T. B. (2005). Exploring heterogeneity in irish voting data: A mixture modelling approach. Technical report, Technical Report 05/09, Department of Statistics, Trinity College Dublin.

Han, B., Pan, Y., and Tsang, I. W. (2018). Robust plackett–luce model for k-ary crowdsourced preferences. *Machine Learning*, 107(4):675–702.

Herbrich, R., Minka, T., and Graepel, T. (2007). Trueskill$^{\text{TM}}$: a Bayesian skill rating system. In *NIPS*, pages 569–576.

Homan, R. W., Herman, J., and Purdy, P. (1987). Cerebral location of international 10–20 system electrode placement. *Electroencephalography and clinical neurophysiology*, 66(4):376–382.

Hsueh, P.-Y., Melville, P., and Sindhwani, V. (2009). Data quality from crowdsourcing: a study of annotation selection criteria. In *Proceedings of the NAACL HLT 2009 workshop on active learning for natural language processing*, pages 27–35. Association for Computational Linguistics.

Huang, C.-S., Pal, N. R., Chuang, C.-H., and Lin, C.-T. (2015). Identifying changes in eeg information transfer during drowsy driving by transfer entropy. *Frontiers in human neuroscience*, 9:570.

Huang, R.-S., Jung, T.-P., and Makeig, S. (2009). Tonic changes in eeg power spectra during simulated driving. In *International Conference on Foundations of Augmented Cognition*, pages 394–403, Berlin, Heidelberg. Springer.

Jaini, P., Chen, Z., Carbajal, P., Law, E., Middleton, L., Regan, K., Schaekermann, M., Trimponias, G., Tung, J., and Poupart, P. (2017). Online Bayesian transfer learning for sequential data modeling. In *International Conference on Learning Representations*.

Kamishima, T. (2003). Nantonac collaborative filtering: recommendation based on order responses. In *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 583–588. ACM.

Kendall, M. G. (1938). A new measure of rank correlation. *Biometrika*, 30(1/2):81–93.

Kendall, M. G. (1948). Rank correlation methods. *Charles Griffin & Company Limited*.

Khare, R., Good, B. M., Leaman, R., Su, A. I., and Lu, Z. (2015). Crowdsourcing in biomedicine: challenges and opportunities. *Briefings in bioinformatics*, 17(1):23–32.

Khetan, A. and Oh, S. (2016). Data-driven rank breaking for efficient rank aggregation. *Journal of Machine Learning Research*, 17(193):1–54.

Kim, M., Farnoud, F., and Milenkovic, O. (2014). Hydra: gene prioritization via hybrid distance-score rank aggregation. *Bioinformatics*, 31(7):1034–1043.

Knight, H. and Keith, O. (2005). Ranking facial attractiveness. *The European Journal of Orthodontics*, 27(4):340–348.

Kolde, R., Laur, S., Adler, P., and Vilo, J. (2012). Robust rank aggregation for gene list integration and meta-analysis. *Bioinformatics*, 28(4):573–580.

Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

Li, X., Wang, X., and Xiao, G. (2017). A comparative study of rank aggregation methods for partial and top ranked lists in genomic applications. *Briefings in bioinformatics*, 20(1):178–189.

Liang, L. and Grauman, K. (2014). Beyond comparing image pairs: Setwise active learning for relative attributes. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 208–215.

Lijphart, A. (1994). *Electoral Systems and Party Systems: A Study of Twenty-Seven Democracies, 1945-1990*. Oxford University Press.

Lin, C.-T., Chang, C.-J., Lin, B.-S., Hung, S.-H., Chao, C.-F., and Wang, I.-J. (2010). A real-time wireless brain–computer interface system for drowsiness detection. *IEEE transactions on biomedical circuits and systems*, 4(4):214–222.

Lin, S. (2010). Rank aggregation methods. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(5):555–570.

Liu, T.-Y. (2009). Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*.

Liu, X., van de Weijer, J., and Bagdanov, A. D. (2018). Leveraging unlabeled data for crowd counting by learning to rank. *arXiv preprint arXiv:1803.03095*.

Liu, Y.-T., Lin, Y.-Y., Wu, S.-L., Chuang, C.-H., and Lin, C.-T. (2016). Brain dynamics in predicting driving fatigue using a recurrent self-evolving fuzzy neural network. *IEEE transactions on neural networks and learning systems*, 27(2):347–360.

Luaces, O., Díez, J., Alonso-Betanzos, A., Troncoso, A., and Bahamonde, A. (2015). A factorization approach to evaluate open-response assignments in moocs using preference learning on peer assessments. *Knowledge-Based Systems*.

Luce, R. (1959). Individual choice theory: A theoretical analysis.

Mallows, C. (1957). Non-null ranking models. *Biometrika*.

Maydeu-Olivares, A. (1999). Thurstonian modeling of ranking data via mean and covariance structure analysis. *Psychometrika*.

Miller, J. W. and Dunson, D. B. (2018). Robust Bayesian inference via coarsening. *Journal of the American Statistical Association*, pages 1–13.

Mollica, C. and Tardella, L. (2017). Bayesian plackett–luce mixture models for partially ranked data. *psychometrika*, 82(2):442–458.

Montague, M. and Aslam, J. A. (2002). Condorcet fusion for improved retrieval. In *Proceedings of the eleventh international conference on Information and knowledge management*, pages 538–548. ACM.

Müller, K.-R., Tangermann, M., Dornhege, G., Krauledat, M., Curio, G., and Blankertz, B. (2008). Machine learning for real-time single-trial eeg-analysis: from brain–computer interfacing to mental state monitoring. *Journal of neuroscience methods*, 167(1):82–90.

Namkoong, H. and Duchi, J. C. (2017). Variance-based regularization with convex objectives. In *Advances in Neural Information Processing Systems*, pages 2971–2980.

Negahban, S., Oh, S., and Shah, D. (2016). Rank centrality: Ranking from pairwise comparisons. *Operations Research*, 65(1):266–287.

Nguyen, T., Ahn, S., Jang, H., Jun, S. C., and Kim, J. G. (2017). Utilization of a combined eeg/nirs system to predict driver drowsiness. *Scientific reports*, 7:43933.

Niu, S., Lan, Y., Guo, J., Cheng, X., Yu, L., and Long, G. (2015). Listwise approach for rank aggregation in crowdsourcing. In *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*, pages 253–262. ACM.

Nowak, S. and Rüger, S. (2010). How reliable are annotations via crowdsourcing: a study about inter-annotator agreement for multi-label image annotation. In *Proceedings of the international conference on Multimedia information retrieval*, pages 557–566. ACM.

Palanivel Rajan, S. and Dinesh, T. (2015). Systematic review on wearable driver vigilance system with future research directions. *International Journal of Applied Engineering Research*, 10(1):627–32.

Pan, Y., Han, B., and Tsang, I. W. (2018). Stagewise learning for noisy k-ary preferences. *Machine Learning*, pages 1–29.

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

Qin, T., Geng, X., and Liu, T.-Y. (2010). A new probabilistic model for rank aggregation. In *Advances in neural information processing systems*, pages 1948–1956.

Raman, K. and Joachims, T. (2014). Methods for ordinal peer grading. In *KDD*.

Rashwan, A., Zhao, H., and Poupart, P. (2016). Online and distributed bayesian moment matching for parameter learning in sum-product networks. In *AISTATS*, pages 1469–1477.

Ratcliff, R., Philiastides, M. G., and Sajda, P. (2009). Quality of evidence for perceptual decision making is indexed by trial-to-trial variability of the eeg. *Proceedings of the National Academy of Sciences*, 106(16):6539–6544.

Rayana, S. and Akoglu, L. (2014). An ensemble approach for event detection and characterization in dynamic graphs. In *ACM SIGKDD ODD Workshop*.

Raykar, V. C., Yu, S., Zhao, L. H., Valadez, G. H., Florin, C., Bogoni, L., and Moy, L. (2010). Learning from crowds. *Journal of Machine Learning Research*, 11(Apr):1297–1322.

Resalat, S. N. and Saba, V. (2015). A practical method for driver sleepiness detection by processing the eeg signals stimulated with external flickering light. *Signal, Image and Video Processing*, 9(8):1751–1757.

Saari, D. G. (1999). Explaining all three-alternative voting outcomes. *Journal of Economic Theory*, 87(2):313–355.

Sahayadhas, A., Sundaraj, K., and Murugappan, M. (2012). Detecting driver drowsiness based on sensors: a review. *Sensors*, 12(12):16937–16953.

Sahoo, D., Hoi, S. C., and Li, B. (2014). Online multiple kernel regression. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 293–302, New York. ACM.

Sajjadi, M. S., Alamgir, M., and von Luxburg, U. (2016). Peer grading in a course on algorithms and data structures: Machine learning algorithms do not improve over simple baselines. In *Proceedings of the third (2016) ACM conference on Learning@ Scale*, pages 369–378. ACM.

Sarkar, C., Cooley, S., and Srivastava, J. (2014). Robust feature selection technique using rank aggregation. *Applied Artificial Intelligence*, 28(3):243–257.

Shah, N., Balakrishnan, S., Bradley, J., Parekh, A., Ramchandran, K., and Wainwright, M. (2015). Estimation from pairwise comparisons: sharp minimax bounds with topology dependence. In *AISTATS*.

Shah, N., Bradley, J., Parekh, A., Wainwright, M., and Ramchandran, K. (2013). A case for ordinal peer-evaluation in moocs. In *NIPS Workshop on Data Driven Education*.

Sheng, V. S., Provost, F., and Ipeirotis, P. G. (2008). Get another label? improving data quality and data mining using multiple, noisy labelers. In *KDD*, pages 614–622. ACM.

Sinha, A., Namkoong, H., and Duchi, J. (2017). Certifying some distributional robustness with principled adversarial training. *arXiv preprint arXiv:1710.10571*.

Snow, R., O'Connor, B., Jurafsky, D., and Ng, A. Y. (2008). Cheap and fast—but is it good?: evaluating non-expert annotations for natural language tasks. In *EMNLP*, pages 254–263. Association for Computational Linguistics.

Soon, C. S., Brass, M., Heinze, H.-J., and Haynes, J.-D. (2008). Unconscious determinants of free decisions in the human brain. *Nature neuroscience*, 11(5):543.

Soufiani, H. A., Chen, W., Parkes, D. C., and Xia, L. (2013). Generalized method-of-moments for rank aggregation. In *NIPS*, pages 2706–2714.

Soufiani, H. A., Parkes, D. C., and Xia, L. (2014). Computing parametric ranking models via rank-breaking. In *ICML*, pages 360–368.

Spiegelhalter, D. J., Best, N. G., Carlin, B. P., and Van der Linde, A. (1998). Bayesian deviance, the effective number of parameters, and the comparison of arbitrarily complex models. Technical report, Research Report, 98-009.

Stuart, J. M., Segal, E., Koller, D., and Kim, S. K. (2003). A gene-coexpression network for global discovery of conserved genetic modules. *science*, 302(5643):249–255.

Teplan, M. et al. (2002). Fundamentals of eeg measurement. *Measurement science review*, 2(2):1–11.

Thurstone, L. L. (1927a). A law of comparative judgment. *Psychological review*, 34(4):273.

Thurstone, L. L. (1927b). The method of paired comparisons for social values. *Journal of Abnormal and Social Psychology*.

Tsiporkova, E. and Boeva, V. (2006). Multi-step ranking of alternatives in a multi-criteria and multi-expert decision making environment. *Information Sciences*, 176(18):2673–2697.

Turner, T. L. and Miller, P. M. (2012). Investigating natural variation in drosophila courtship song by the evolve and resequence approach. *Genetics*, 191(2):633–642.

Van Cutsem, J., Marcora, S., De Pauw, K., Bailey, S., Meeusen, R., and Roelands, B. (2017). The effects of mental fatigue on physical performance: a systematic review. *Sports medicine*, 47(8):1569–1588.

Villani, C. (2008). *Optimal Transport: Old and New*. Grundlehren der mathematischen Wissenschaften. Springer Berlin Heidelberg.

Vitelli, V., Sørensen, Ø., Frigessi, A., and Arjas, E. (2014). Probabilistic preference learning with the mallows rank model. *arXiv preprint arXiv:1405.7945*.

Vitter, J. S. (1985). Random sampling with a reservoir. *ACM Transactions on Mathematical Software (TOMS)*, 11(1):37–57.

Volkovs, M. and Zemel, R. (2012). A flexible generative model for preference aggregation. In *WWW*.

Volpi, R., Namkoong, H., Sener, O., Duchi, J. C., Murino, V., and Savarese, S. (2018). Generalizing to unseen domains via adversarial data augmentation. In *Advances in Neural Information Processing Systems*, pages 5334–5344.

Vuurens, J., de Vries, A. P., and Eickhoff, C. (2011). How much spam can you take? an analysis of crowdsourcing results to increase accuracy. In *SIGIR Workshop on CIR*.

Wang, S., Zhang, Y., Wu, C., Darvas, F., and Chaovalitwongse, W. A. (2015). Online prediction of driver distraction based on brain activity patterns. *IEEE Transactions on Intelligent Transportation Systems*, 16(1):136–150.

Weisstein, E. W. (2004). Delta function. *delta*, 29:30.

Welch, P. (1967). The use of fast fourier transform for the estimation of power spectra: a method based on time averaging over short, modified periodograms. *IEEE Transactions on audio and electroacoustics*, 15(2):70–73.

Weng, R. C. and Lin, C.-J. (2011). A bayesian approximation method for online ranking. *Journal of Machine Learning Research*, 12(Jan):267–300.

Woodroofe, M. (1989). Very weak expansions for sequentially designed experiments: linear models. *The annals of statistics*, 17(3):1087–1102.

Wulf, J., Blohm, I., Leimeister, J. M., and Brenner, W. (2014). Massive open online courses. *Business & Information Systems Engineering*, 6(2):111–114.

Xu, Q., Xiong, J., Cao, X., Huang, Q., and Yao, Y. (2018). From social to individuals: a parsimonious path of multi-level models for crowdsourced preference aggregation. *IEEE transactions on pattern analysis and machine intelligence*, 41(4):844–856.

Xu, Q., Xiong, J., Huang, Q., and Yao, Y. (2013). Robust evaluation for quality of experience in crowdsourcing. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 43–52. ACM.

Xu, Q., Yan, M., Huang, C., Xiong, J., Huang, Q., and Yao, Y. (2017). Exploring outliers in crowdsourced ranking for qoe. In *Proceedings of the 25th ACM international conference on Multimedia*, pages 1540–1548. ACM.

Yan, L., Dodier, R. H., Mozer, M., and Wolniewicz, R. H. (2003). Optimizing classifier performance via an approximation to the wilcoxon-mann-whitney statistic. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, pages 848–855, Washington D.C. AAAI Press 2003.

Yarkoni, T., Barch, D. M., Gray, J. R., Conturo, T. E., and Braver, T. S. (2009). Bold correlates of trial-by-trial reaction time variability in gray and white matter: a multi-study fmri analysis. *PLoS One*, 4(1):e4257.

Yu, H., Lu, H., Ouyang, T., Liu, H., and Lu, B.-L. (2010). Vigilance detection based on sparse representation of eeg. In *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, pages 2439–2442, Buenos Aires, Argentina. IEEE.

Zhang, G. P. (2000). Neural networks for classification: a survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 30(4):451–462.

Zhao, Z., Piech, P., and Xia, L. (2016). Learning mixtures of plackett-luce models. In *International Conference on Machine Learning*, pages 2906–2914.

Zhao, Z., Villamil, T., and Xia, L. (2018). Learning mixtures of random utility models. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Zhou, K., Xue, G.-R., Zha, H., and Yu, Y. (2008). Learning to rank with ties. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 275–282, Singapore. ACM.