Learning Populations of Preferences via Pairwise Comparison Queries

Gokcan Tatli
University of Wisconsin-Madison

Yi Chen
University of Wisconsin-Madison

Ramya Korlakai Vinayak
University of Wisconsin-Madison

Abstract

Ideal point based preference learning using pairwise comparisons of type "Do you prefer a or b?" has emerged as a powerful tool for understanding how we make preferences. Existing preference learning approaches assume homogeneity and focus on learning preference on average over the population or require a large number of queries per individual to localize individual preferences. However, in practical scenarios with heterogeneous preferences and limited availability of responses, these approaches are impractical. Therefore, we introduce the problem of *learning the distribution of preferences over a population* via pairwise comparisons using only one response per individual. Due to binary answers from comparison queries, we focus on learning the mass of the underlying distribution in the regions created by the intersection of bisecting hyperplanes between queried item pairs. We investigate this fundamental question in both 1-D and higher dimensional settings with noiseless response to comparison queries. We show that the problem is identifiable in 1-D setting and provide recovery guarantees. We show that the problem is not identifiable for higher dimensional settings in general and establish sufficient condition for identifiability. We propose using a regularized recovery and provide guarantees on the total variation distance between the true mass and the learned distribution. We validate our findings through simulations and experiments on real datasets. We also introduce a new dataset for this task collected on a real crowdsourcing platform.

Proceedings of the 27$^{\text{th}}$ International Conference on Artificial Intelligence and Statistics (AISTATS) 2024, Valencia, Spain. PMLR: Volume 238. Copyright 2024 by the author(s).

1 INTRODUCTION

Learning user preferences via pairwise comparison queries of the type "Do you prefer item a or b?" (Figure 1(a)) is widely used in various applications, such as political science, to model voters' political preferences and predict their voting behavior, and in recommendation systems, to model users' preferences for products or services (Saaty and Vargas, 2012; Fichtner, 1986; Abildtrup et al., 2006; Hopkins and Noel, 2022; Oishi et al., 2005). Let $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^d$ be the known feature representation of concepts (items, objects, images, choices, etc.). Preference learning based on ideal point model (Coombs, 1950; Jamieson and Nowak, 2011; Ding, 2016; Singla et al., 2016; Xu and Davenport, 2020; Canal et al., 2022) assumes that there is an *unknown ideal preference point* $\boldsymbol{u} \in \mathcal{X}$ that represents the reference point people use for their preference judgments based on distances. When presented a preference query, "Do you prefer a or b?", the ideal point model (Coombs, 1950) assumes that a is preferred over b if the individual's preference point $\boldsymbol{u}$ is closer to the representation of item a, $\boldsymbol{x}_a$ than item b, $\boldsymbol{x}_b$ (Figure 1(b)), i.e., $||\boldsymbol{x}_a - \boldsymbol{u}||_2 < ||\boldsymbol{x}_b - \boldsymbol{u}||_2$. Preference learning aims to use the responses to pairwise comparison queries from people and learn the preference point $\boldsymbol{u}$. Once we learn $\boldsymbol{u}$, we can predict people's choices between new unseen pairs.

Many works on preference learning have focused on a universal model, where the data from everyone is pooled together to learn a single preference point on average for the population (Green, 1975; Johnson, 1971; Bhargava et al., 2016). However, different individuals can have different preferences. While one can focus on learning an individual's preference separately, it takes $\mathcal{O}(d \log(d/\varepsilon))$ queries in $\mathbb{R}^d$ to learn an individual's preference point within an ε-ball (Massimino and Davenport, 2021). This can be a prohibitively large number of queries per individual due to cost, cognitive overload or privacy concerns. Therefore, we introduce the problem of *learning the distribution of preferences over a population* via pairwise comparisons using only one response per individual. In this scenario, learning each individual's preference is impossible. In many

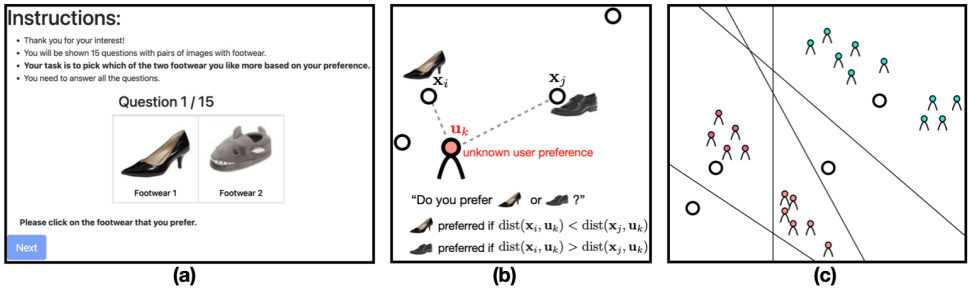

Figure 1: (a) Example of pairwise comparison query. (b) Ideal point model based response to a comparison query. The colorless circles denote the known representation for items being compared and the human denotes the unknown user preference point. (c) Example of the regions formed by the bisecting hyperplanes between pairs queried and mass of user preferences in different regions.

applications, learning the distribution of user preferences (Figure 1(c)) can be useful for many downstream tasks. E.g., if an ice cream company wants to come up with new flavors, knowing which regions of flavor profiles have more mass is beneficial in discovering new ice cream flavors. The learned distribution can thus be helpful in various tasks ranging from understanding the polarization of preferences within a population and testing differences between preferences of different populations to using the distribution as a prior to efficiently learn new user preferences.

We investigate the problem of learning the distribution of user preferences using pairwise comparisons. Since we are limited to binary answers to comparison queries for pairs of items, we focus on learning the mass of the underlying distribution in the regions (polytopes) defined by the intersection of the bisecting hyperplanes pairs of items (see Figure 1(c)). If we could have queried $\tilde{\mathcal{O}}(d)$ per user, we can localize each user preference point to one of the regions. So, if we sample a large number of individuals from the population and query each of them with a sufficiently large number of queries, we can build a histogram of the underlying distribution in these regions. However, querying a large number of comparison pairs per individual can be prohibitive due to privacy issues, limited interaction of individuals with the platform, cognitive overload, and cost.

Goal: Develop a fundamental understanding of what we can learn about the distribution of user preferences with only one comparison query per user.

Our contribution: We study the novel problem of learning the distribution of user preferences over the population via pairwise comparison queries with only one response per individual in the ideal point model. We investigate the fundamental questions of identifiability and recovery guarantees leading to the following contributions:

- We show that the problem is identifiable in 1D setting and is not identifiable in a higher dimensional setting in general. We provide sufficient condition under which the problem is identifiable in higher dimensions.
- For the 1D setting, we provide recovery guarantees for the mass in the regions defined by the intersection of hyperplanes at the mid-point of queried pairs.
- For the higher dimensional setting, we propose to use regularized recovery and provide guarantees on the total variation distance between the true mass in each region and the estimated mass in terms of the regularization parameter and the interplay between the true mass and regularization.
- We provide experiments on synthetic and real datasets that validate our results and observations. We also introduce a new dataset for this task collected on a real crowdsourcing platform [1].

In addition to the above contributions, our work leads to several interesting open questions regarding learning from diverse populations in preference learning.

2 PROBLEM SETUP

Let $\boldsymbol{x} \in \mathcal{X} \subseteq \mathbb{R}^d$ denote known feature representation of items[2]. Under the ideal point model (Coombs, 1950), each individual's preference is also modeled as an *unknown* point in the same space. Let $P^\star$ denote the unknown underlying distribution of user preferences. Each individual l has an unknown preference $\boldsymbol{u}_l \in \mathbb{R}^d$. We assume that $\boldsymbol{u}_l \overset{i.i.d.}{\sim} P^\star$. Let $\mathcal{T}$ denote the set of pairs of items (i, j) that are queried. We consider pairwise comparison queries of the form "do you prefer item i or item j?". We assume that the answer to the pairwise query (i, j) from an individual l is $y_{ij}^{(l)} = 1$

[1] Codes for our methods and datasets are available at `https://github.com/ramyakv/Learning-Population-of-Preferences-AISTATS2024`

[2] This is a reasonable assumption, especially with the availability of large pre-trained foundation models.

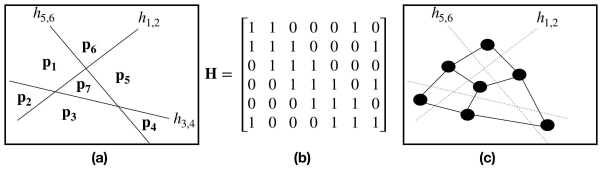

Figure 2: (a) Example of regions (polytopes) formed by the intersection of three hyperplanes. (b) The corresponding matrix $\mathbf{H}$. (c) The corresponding graph where the regions are the nodes.

if $||\boldsymbol{x}_i - \boldsymbol{u}_l||_2 < ||\boldsymbol{x}_j - \boldsymbol{u}_l||_2$ and $y_{ij}^{(l)} = -1$ otherwise. Note that each pair of items (i, j) with $i < j$ in $\mathcal{T}$ is associated with a hyperplane, denoted by h_{ij}, that is perpendicular at the midpoint joining the two items $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. We will slightly abuse the notation and use $\mathcal{T}$ to denote the set of hyperplanes as well as the pairs of items. Note that there is a one-to-one correspondence between each pair and the respective hyperplane. The intersection of these hyperplanes carve out regions in $\mathbb{R}^d$ that are polytopes. Let $\mathcal{H}(\mathcal{T})$ denote the set of partitions of $\mathbb{R}^d$ that is created by the set of all hyperplanes in $\mathcal{T}$. Note that for $|\mathcal{T}|$ hyperplanes in $\mathbb{R}^d$, $|\mathcal{H}(\mathcal{T})| = \mathcal{O}(|\mathcal{T}|^d)$ (Buck, 1943). Given answers to pairwise comparison queries, our goal is to understand what we can learn about the underlying distribution of user preferences $P^\star$. Given a finite number of hyperplanes and only binary answers to pairwise comparison queries, we focus on the fundamental question of whether we can learn the mass induced by $P^\star$ on the regions $\mathcal{H}(\mathcal{T})$, denoted by $\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}$ which is a discrete probability mass function of size $|\mathcal{H}(\mathcal{T})|$.

For each pair of items $(i, j) \in \mathcal{T}$, let $q^\star_{ij}$ denote the mass of $P^\star$ on the side of $\boldsymbol{x}_i$ of the hyperplane h_{ij} and $q^\star_{ji} = 1 - q^\star_{ij}$ is the mass on the other side of h_{ij}. Let $\mathbf{q}^\star \in \mathbb{R}^{2|\mathcal{T}|}$ denote the vector that stacks q_{ij}'s for the ordered pairs $(i, j) \in \mathcal{T}$, followed by the corresponding q_{ji}'s. We note $\mathbf{q}^\star$ denotes the fraction of people on either side of the hyperplanes in $\mathcal{T}$ and can be written as a linear combination of the mass $\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}$ in the regions via the following linear system of equations,

$$\mathbf{H}\ \mathbf{p}^\star_{\mathcal{H}(\mathcal{T})} = \mathbf{q}^\star, \tag{1}$$

where $\mathbf{H}$ is a $2|\mathcal{T}| \times |\mathcal{H}(\mathcal{T})|$ binary matrix where in each row, the 1's indicate the regions that contribute to the side of the hyperplane. Each column of $\mathbf{H}$ corresponds to a region (polytope) created by the intersection of the hyperplanes. Note that $\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}$ *identifiable* if it is the unique probability vector of size $|\mathcal{H}(\mathcal{T})|$ that gives rise to $\mathbf{q}^\star$. So, $\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}$ is *not identifiable* if there exist a valid probability vector $\mathbf{p} \neq \mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}$ such that $\mathbf{Hp} = \mathbf{Hp}^\star_{\mathcal{H}(\mathcal{T})}$.

Figure 2 shows an example of partition of $\mathbb{R}^2$ with 3 hyperplanes $h_{1,2}, h_{3,4}$ and $h_{5,6}$. With the enumeration of the regions shown in the figure, we can construct the binary matrix $\mathbf{H}$, where the first 3 rows represent regions corresponding to $h_{1,2}$ towards the side of item 1, $h_{3,4}$ towards the side of item 3 and $h_{5,6}$ towards the side of item 5 respectively. Similarly, the last 3 rows represent regions corresponding to the other side of each of the hyperplanes. We also note that each column gives positions of the corresponding region $\mathbf{p}_i$ in terms of hyperplanes $h_{1,2}, h_{3,4}$ and $h_{5,6}$.

For each pair $(i, j) \in \mathcal{T}$, we can estimate the mass on either side of the hyperplane h_{ij} by querying the pair to a random sample of people. Given these estimated masses on either side of the hyperplanes in $\mathcal{T}$, the question of interest is: *Can we estimate* $\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}$*, the probability mass in the regions of intersections of hyperplanes in* $\mathcal{T}$ *induced by the underlying distribution of preferences* $P^\star$*?*

3 RELATED WORKS

We briefly review some of the related works here, deferring a more detailed discussion to the Appendix.

Preference learning based on ideal point model (Coombs, 1950; Jamieson and Nowak, 2011; Ding, 2016; Singla et al., 2016; Xu and Davenport, 2020; Massimino and Davenport, 2021; Canal et al., 2022) has been studied by several works. A key limitation of these works is that they either focus on learning an individual preference by making many queries per individual or assume homogeneity and learn a single preference point using data from all the users. A recent work by Tatli et al. (2022) considers a distance query model, i.e. when a user is queried with an item, the response is how far their preference point is from the queried item, and studies the problem of learning the distribution of preferences for the specific case of discrete 1D discrete distributions. In contrast, we consider the more realistic setting of response to pairwise comparison between items and study 1D and higher dimensional settings. The differences in the query model lead to different insights in terms of the nature of identifiability issues as well as the properties of the linear systems that arise with pairwise comparisons. We defer a more detailed discussion to the appendix due to space constraints.

Another line of work in preference learning involves ranking based models, e.g., Bradley-Terry-Luce model (Bradley and Terry, 1952; Luce, 1959), Mallows model (Mallows, 1957), stochastic transitivity models (Shah et al., 2016), that focus on finding a single ranking of m items or finding top-k items by pairwise comparisons (Hunter, 2004; Kenyon-Mathieu and Schudy, 2007; Braverman and Mossel, 2007; Negahban et al., 2012; Eriksson, 2013; Rajkumar and Agarwal,

2014; Shah and Wainwright, 2017). Ranking m items in these settings requires $\mathcal{O}(m \log m)$ queries. Note in contrast, under the ideal point based models, the query complexity for ranking reduces to $\mathcal{O}(d \log m)$, where d is the dimension of the domain of representations which is usually much smaller than the number of items being ranked (Jamieson and Nowak, 2011). This is due to the fact that the geometry induces constraint on the number of possible rankings from $m!$ to $\mathcal{O}(m^d)$. Lu and Boutilier (2014) consider a mixture of K-Mallows model for the ranking setting and proposes an EM algorithm that takes user preferences as input and outputs the estimated model parameters. However, there are no guarantees provided for this algorithm. Other works on ranking models (Awasthi et al., 2014; Mao and Wu, 2022) have noted that even a two component mixture of Mallows model is not identifiable with pairwise comparisons and the focus has been on settings that consider samples that are full or list-wise or group-wise ranking data per user and tensor-based algorithms for recovery. A recent work (Zhang et al., 2022) considers a mixture of two BTL-models with pairwise comparison and shows that it is identifiable except for certain sets of parameters and show similar results for list-wise query setting, but it does not provide algorithmic approaches to estimate them. Our work focuses on preference learning based on ideal point model with pairwise comparison queries and develops fundamental understanding under a nonparametric setting.

4 ONE DIMENSIONAL SETTING

We first study the problem in the 1D setting and provide results on identifiability and recovery guarantees.

Identifiability: In 1D setting, $|\mathcal{T}|$ pairs creates $|\mathcal{T}|+1$ intervals. Measuring the fraction of mass on either side of each of the hyperplanes in $\mathcal{T}$ is equivalent to measuring the cumulative distribution function (CDF) of the distribution $\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}$. The linear system of equations (1) has $|\mathcal{T}|+1$ unknowns and $|\mathcal{T}|+1$ equations. The corresponding binary matrix $\mathbf{H}$ can be written as a concatenation of two triangular matrices where one is a lower triangular and the other is an upper triangular matrix. E.g., 2 pairs create 3 regions, and the corresponding matrix $\mathbf{H} = [1,\ 0,\ 0;\ \ 1,\ 1,\ 0;\ \ 0,\ 1,\ 1;\ \ 0,\ 0,\ 1]$. Any such $\mathbf{H}$ is full column rank by construction. Therefore, the linear system of equations $\mathbf{q}^\star = \mathbf{H}\ \mathbf{p}$ has a unique solution in terms of the true $\mathbf{q}^\star$, given by $\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})} = (\mathbf{H}^T\mathbf{H})^{-1}\mathbf{H}^T\mathbf{q}^\star$. This is summarized in the following proposition.

Proposition 1. *(Identifiability in 1D) In 1D setting, the mass $\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}$ in the regions of the intersection of hyperplanes in $\mathcal{T}$ induced by the underlying distribution of preferences can be uniquely determined by only measuring the fraction of the population on either side of each of the hyperplanes in $\mathcal{T}$.*

Recovery Guarantees: As we do not have access to true $\mathbf{q}^\star$, we have to learn $\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}$ from $\hat{\mathbf{q}}$ estimated by querying pairs in $\mathcal{T}$. We use the following constrained optimization problem:

$$\hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})} := \arg\min_{\mathbf{p}\geq 0, \mathbf{1}^\top\mathbf{p}=1} \quad \frac{1}{2}||\mathbf{H}\mathbf{p} - \hat{\mathbf{q}}||_2^2.$$

The objective function is strongly convex and therefore the above optimization problem is guaranteed to have a unique solution. We provide the following recovery guarantee for the 1D noiseless setting for the above optimization.

Theorem 4.1. *(Recovery in 1D) With probability at least $1-\delta$, the total variation distance between $\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}$ and the recovered mass $\hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})}$ is bounded as follows,*

$$TV\Big(\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})}\Big) \leq \sqrt{\frac{|\mathcal{T}|+1}{2}}\sqrt{\frac{\log(4|\mathcal{T}|/\delta)}{2n_p}},$$

where n_p is the number of users queried per pairwise comparison query.

As the number of users increases, the total variation distance between $\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}$ and $\hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})}$ goes to 0. Proof details are available in the appendix.

5 HIGHER DIMENSIONAL SETTINGS

In this section, we discuss the identifiability results and recovery guarantees for $\mathbb{R}^d$, with $d \geq 2$. The details of the proofs are deferred to the appendix.

Identifiability: Assuming items and users are supported on $\mathbb{R}^d$, with $d \geq 2$, we note that the number of regions, $|\mathcal{H}(\mathcal{T})|$, formed by the hyperplanes in $\mathcal{T}$ is of $\mathcal{O}(|\mathcal{T}|^d)$. So, the linear system of equations (1) has more unknowns than the number of equations. We show the following with regard to identifiability.

Proposition 2. *For $d \geq 2$, the binary matrix $\mathbf{H}$ which of size $2|\mathcal{T}| \times \mathcal{O}(|\mathcal{T}|^d)$ has rank$(\mathbf{H}) = |\mathcal{T}|+1$ and the solution to the linear system of equations (1) is not unique and hence $\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}$ is not identifiable.*

Sparsity: From Proposition 2, for $d \geq 2$, we cannot hope to recover $\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}$ in general. However, $\mathbf{H}$ is a fat matrix, and a natural question that arises is what if $\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}$ is sparse, i.e., if only $k \ll \mathcal{O}(|\mathcal{T}|^d)$ entries of $\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}$ are non-zero? We note that, since rank$(\mathbf{H}) = |\mathcal{T}|+1$, for any $k > |\mathcal{T}|+1$, there exists at least another solution to the equation (1). This follows from noting that the distribution of preferences on either side of each

of the hyperplanes can, in fact, be re-written as a convex combination of a set of vectors in $\mathbb{R}^{2|\mathcal{T}|}$ corresponding to the columns of $\mathbf{H}$, and then using the Carathéodory theorem which guarantees that there always exists a solution which can be expressed with at most $|\mathcal{T}|+1$ columns (see appendix for a detailed discussion of this). We further note that the results from sparse signal recovery literature(Candes and Wakin, 2008; Baraniuk, 2007; Elad, 2010), in particular, from Theorem 2.13 by Foucart and Rauhut (2013), any k-sparse solution of an underdetermined linear system of equations is unique if and only if every set of $2k$ columns of measurement matrix is linearly independent. Given the structure of our matrix $\mathbf{H}$ in equation (1), we show the following.

Proposition 3. *For the problem setting in (1), we can always find linearly dependent ℓ columns of $\mathbf{H}$ if $\ell \geq 4$.*

As a consequence, even with a lot of sparsity, uniqueness cannot be guaranteed for all k-sparse distributions for $k \geq 2$. For example, consider the partition in the Figure 2(a). Then, suppose that the solution is 2-sparse and $\mathbf{q}^\star = [0.5,\ 1,\ 0.5,\ 0.5,\ 0,\ 0.5]^\top$. We note that $\mathbf{q} = \mathbf{H}\mathbf{p}$ holds for both $\mathbf{p} = [0.5,\ 0,\ 0.5,\ 0,\ 0,\ 0,\ 0]^\top$ and $\mathbf{p} = [0,\ 0.5,\ 0,\ 0,\ 0,\ 0,\ 0.5]^\top$. Similarly, we suppose that the solution is 4-sparse and $\mathbf{p}^\star = [0.3,\ 0,\ 0,\ 0,\ 0.4,\ 0.1,\ 0.2]^\top$. Therefore, we have $\mathbf{H}\mathbf{p}^\star = \mathbf{q}$, where $\mathbf{q} = [0.4,\ 0.5,\ 0,\ 0.6,\ 0.5,\ 1]^\top$. However, $\mathbf{q} = \mathbf{H}\mathbf{p}$ holds also for $\mathbf{p} = [0.2,\ 0,\ 0,\ 0,\ 0.3,\ 0.2,\ 0.3]^\top$.

While not all k-sparse solutions are unique for $k \geq 2$, we characterize a sufficient condition under which we can guarantee uniqueness for k-sparse solutions. First, we recall that the convex hull of a set of affinely independent points is called a *simplex*, and a face of a convex set that is itself a simplex is called a *simplicial face*. Noting that $\mathbf{p}$ is a probability vector, from the equation 1, we see that $\mathbf{q}^\star$ is a convex combination of the columns of $\mathbf{H}$. Therefore, the set of all possible vectors $\mathbf{q}^\star$ is a convex polytope. We show the following sufficient condition for identifiability when $k \leq |\mathcal{T}|+1$.

Theorem 5.1. *(Restricted identifiability under sparsity) If $\mathbf{q}^\star$ belongs to the relative interior of a simplicial face of conv($\mathbf{H}$), the system of linear equations $\mathbf{q}^\star = \mathbf{H}\mathbf{p}$ has a unique solution $\mathbf{p}^\star$, where conv($\mathbf{H}$) refers to the convex hull of columns of $\mathbf{H}$.*

In $\mathbb{R}^d$, we can have at most $d+1$ affinely independent set of points. Therefore, the above theorem only applies to $k \leq |\mathcal{T}|+1$. To illustrate the implication of Theorem 5.1, we consider the example from the partition in Figure 2(a) with 2-sparse cases. As we observed before, $\mathbf{p}^\star = [0.5,\ 0,\ 0.5,\ 0,\ 0,\ 0,\ 0]^\top$ is not a unique solution for $\mathbf{q}^\star = [0.5,\ 1,\ 0.5,\ 0.5,\ 0,\ 0.5]^\top$. Here, we note that conv($\mathbf{H}_{:,1}, \mathbf{H}_{:,3}$) does not form a simplicial face of conv($\mathbf{H}$), where $\mathbf{H}_{:,j}$ denotes j-th column of $\mathbf{H}$. On the other hand, any $\mathbf{q}^\star$ that is formed by a convex combination of $\mathbf{H}_{:,1}$ and $\mathbf{H}_{:,6}$ will have the unique solution $\mathbf{p}^\star = [\mathbf{q}_2^\star, 0, ..., 0,\ 1-\mathbf{q}_2^\star]^\top$, since conv($\mathbf{H}_{:,1}, \mathbf{H}_{:,6}$) is a simplicial face of conv($\mathbf{H}$). We provide more examples and details of the proof for Theorem 5.1 in the appendix.

Bounds on the mass in the regions: Given the restricted identifiability of $\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}$ in higher dimensions, we cannot always hope to recover it from binary answers to pairwise comparison queries. Here, we show that we can obtain lower and upper bounds for each entry of $\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}$ from the estimated $\hat{\mathbf{q}}$ without requiring any additional assumptions. To state these bounds, we need some notations. Let $\mathbf{H}_{i,:}$ be the i-th row of $\mathbf{H}$ (corresponding to the i-th hyperplane) and let (a_i, b_i) denote the pair queried corresponding to this row. Let $\hat{\mathbf{q}}_{a_i,b_i}$ denote the estimated mass on the side of a_i of the i-th hyperplane. Let $\mathcal{K}_j$ denote the position of rows of $\mathbf{H}$ whose j-th column entry is 1 and $\hat{\mathbf{Q}}_0^j := [\min_{i\in\mathcal{K}_1} \hat{\mathbf{q}}_{a_ib_i}, \dots \min_{i\in\mathcal{K}_{j-1}} \hat{\mathbf{q}}_{a_ib_i},\ 0,\ \min_{i\in\mathcal{K}_{j+1}} \hat{\mathbf{q}}_{a_ib_i}, \dots \min_{i\in\mathcal{K}_{|\mathcal{H}(\mathcal{T})|}} \hat{\mathbf{q}}_{a_ib_i}]^T$.

Proposition 4. *With probability at least $1-\delta$, each entry of $\mathbf{p}_{\mathcal{H}(\mathcal{T})}$ can be bounded below and above as follows:*

$$\max\{0, L_j\} \leq \mathbf{p}^\star_{\mathcal{H}(\mathcal{T})_j} \leq U_j, \tag{2}$$

where $L_j := \max_{i\in\mathcal{K}_j} \hat{\mathbf{q}}_{a_ib_i} - \mathbf{H}_{i,:}^T\hat{\mathbf{Q}}_0^j - (|\mathbf{H}_{i,:}|_1+1)\gamma$, $U_j := \min_{i\in\mathcal{K}_j} \hat{\mathbf{q}}_{a_ib_i} + \gamma$, $\gamma = \sqrt{\frac{\log(4|\mathcal{T}|/\delta)}{2n_p}}$ and n_p is the number of people answering each pairwise query.

Proposition 4 provides most general tight bounds when there is no side information about the setting. See the appendix for examples and details of the proof.

Graph Regularization: In the face of non-identifiability, additional structural assumptions are needed for learning $\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}$. While $\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}$ is a $\mathcal{O}(|\mathcal{T}|^d)$-dimensional probability vector, the entries corresponding to mass in regions have a geometry in the space $\mathcal{X} \subseteq \mathbb{R}^d$ (recall Figure 2(a)) that gives a notion of *near-by* and *far-away* regions. We construct a connected undirected graph with the polytopes as the nodes and two nodes are connected by an edge if they share a $(d-1)$-dimensional face between them (see Figure 2(c)). We propose using a graph regularizer (normalized by volume to account for differences in the sizes of the regions) to recover $\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}$. Intuitively, this means that we expect preferences to accumulate in spatially nearby regions (Figure 1(c)). Several works in signal recovery have used graph regularization to exploit local invariance in data as side information and find a locally

invariant representation of the data (Belkin and Niyogi, 2001; Cai et al., 2011; Hadsell et al., 2006).

We note that this proposed graph structure can be constructed using the matrix $\mathbf{H}$. Recall that the rows of $\mathbf{H}$ correspond to hyperplanes and the columns correspond to the regions (polytopes) in $\mathcal{H}(\mathcal{T})$, providing a binary encoding for them by construction. That is, each entry of a given column of $\mathbf{H}$ determines which side of a hyperplane the corresponding region is located on. Therefore, there exists an edge between nodes corresponding to the regions that have only two different entries in their hyperplane coordinates, i.e., only if one pairwise comparison yields opposite results. Accordingly, neighboring regions have common $(d-1)$-dimensional faces in between. We define the weight matrix $\mathbf{W}$ for the graph regularization as follows,

$$\mathbf{W}_{i,j} = \|\mathbf{H}_{:,i} - \mathbf{H}_{:,j}\|_1^{-1}/(\alpha_i \alpha_j), \tag{3}$$

where $\alpha = [\alpha_1, \ldots, \alpha_{|\mathcal{H}(\mathcal{T})|}]^T$ represent the volumes of regions with corresponding mass $\mathbf{p} = [\mathbf{p}_1, \ldots, \mathbf{p}_{|\mathcal{H}(\mathcal{T})|}]^T$. Each entry of $\mathbf{W}$ is the weighted inverse of the Hamming distance between corresponding nodes i and j, where $\mathbf{H}_{:,i}$ is the i-th column of the matrix $\mathbf{H}$. Furthermore, since the regions in $\mathcal{H}(\mathcal{T})$ are not equal in size, we normalize with the volumes of the regions. One can similarly construct different weight matrices for regularization as long as the entries are inversely proportional to the distances between nodes. Heat kernel weighting (Belkin and Niyogi, 2001), 0-1 weighting (Cai et al., 2011) are some of the widely used ones in the literature. We use $\mathbf{W}$ defined in equation (3) and form following graph Laplacian regularizer:

$$\begin{aligned}\frac{1}{2}\sum_{i=1}^{|\mathcal{H}(\mathcal{T})|}\sum_{j=1}^{|\mathcal{H}(\mathcal{T})|} |\mathbf{p}_i - \mathbf{p}_j|^2 \mathbf{W}_{i,j} &=: \mathbf{p}^T\mathbf{D}\mathbf{p} - \mathbf{p}^T\mathbf{W}\mathbf{p} \\ &=: \mathbf{p}^T\mathbf{L}\mathbf{p}, \end{aligned} \tag{4}$$

where $\mathbf{D}_{i,i} = \sum_{j=1}^{|\mathcal{H}(\mathcal{T})|} \mathbf{W}_{i,j}$, $\mathbf{D}_{i,j} = 0$ when $i \neq j$ and $\mathbf{L} = \mathbf{D} - \mathbf{W}$. Using this regularizer, we propose the following optimization problem for recovering $\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}$:

$$\begin{aligned}\hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})} &:= \arg\min_{\mathbf{p}\geq 0, \mathbf{1}^\top\mathbf{p}=1} \ \frac{1}{2}||\mathbf{H}\mathbf{p} - \hat{\mathbf{q}}||_2^2 + \frac{\lambda}{2}\mathbf{p}^T\mathbf{L}\mathbf{p} \\ &:= \arg\min_{\mathbf{p}\geq 0, \mathbf{1}^\top\mathbf{p}=1} \ \frac{1}{2}||\mathbf{R}\mathbf{p} - \hat{\mathbf{b}}||_2^2, \end{aligned} \tag{5}$$

where $\mathbf{R}^T\mathbf{R} = \mathbf{H}^T\mathbf{H} + \lambda\mathbf{L}$ by Cholesky decomposition and $\hat{\mathbf{b}} = \mathbf{R}^{-T}\mathbf{H}^T\hat{\mathbf{q}}$. The regularizer in equation (4) encourages the changes in nearby regions to be smooth, which is similar to the local invariance property considered by Belkin and Niyogi (2001); Cai et al. (2011); Hadsell et al. (2006). Weighted Laplacian regularizer $\mathbf{L}$ imposes a penalty on $\mathbf{p}$ in such a way that potential values correlated with eigenvectors of $\mathbf{L}$ are diminished. Therefore, eigenvectors corresponding to larger eigenvalues cause more penalty. Note that the eigenvectors of $\mathbf{L}$ are mutually orthogonal by spectral theorem. So, we conclude that orthogonal eigenvectors of nonzero eigenvalues force the potential solution to be close to the distribution $\overline{\alpha}$ by diminishing possible directions other than α, where $\overline{\alpha}$ is the normalized α. We provide the following recovery guarantee using the solution to the proposed regularized optimization problem.

Theorem 5.2. *The convex optimization problem in (5) has a unique solution. Furthermore, with probability at least $1-\delta$, the total variation distance between $\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}$ and the recovered mass $\hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})}$ is bounded as follows,*

$$\begin{aligned}TV(\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})}) \leq & \frac{\lambda}{2}\sqrt{|\mathcal{H}(\mathcal{T})|}\|\mathbf{R}^{-1}\|_2^2\|\mathbf{L}\|\left\|\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})} - \overline{\alpha}\right\| \\ & + \frac{|\mathcal{T}||\mathcal{H}(\mathcal{T})|}{\sqrt{2}}\|\mathbf{R}^{-1}\|_2^2\sqrt{\frac{\log(4|\mathcal{T}|/\delta)}{2n_p}}, \end{aligned}$$

where n_p is the number of users queried per pairwise comparison query.

The maximum singular value of $\mathbf{L}$ and the minimum singular value of $\mathbf{R}$ play an important role in determining the first component of the bound. On the other hand, the second component tends towards 0 as the number of users increases.

L2 Regularization: Using L2 regularizer, we can obtain the following optimization problem:

$$\begin{aligned}\hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})} &:= \arg\min_{\mathbf{p}\geq 0, \mathbf{1}^\top\mathbf{p}=1} \ \frac{1}{2}||\mathbf{H}\mathbf{p} - \hat{\mathbf{q}}||_2^2 + \frac{\lambda}{2}\mathbf{p}^T\mathbf{p} \\ &:= \arg\min_{\mathbf{p}\geq 0, \mathbf{1}^\top\mathbf{p}=1} \ \frac{1}{2}||\mathbf{R}\mathbf{p} - \hat{\mathbf{b}}||_2^2, \end{aligned} \tag{6}$$

This can be cast as a specific version of graph regularizer with a special Laplacian regularizer $\mathbf{I}$, where $\mathbf{R}^T\mathbf{R} = \mathbf{H}^T\mathbf{H} + \lambda\mathbf{I}$ and $\hat{\mathbf{b}} = \mathbf{R}^{-T}\mathbf{H}^T\hat{\mathbf{q}}$. Similarly, we can write the following recovery guarantees.

$$\begin{aligned}\mathrm{TV}(\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})}) \leq & \frac{\lambda}{2}\sqrt{|\mathcal{H}(\mathcal{T})|}\|\mathbf{R}^{-1}\|_2^2\left\|\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})} - \overline{\alpha}\right\|_2 \\ & + \frac{|\mathcal{T}||\mathcal{H}(\mathcal{T})|}{\sqrt{2}}\|\mathbf{R}^{-1}\|_2^2\sqrt{\frac{\log(4|\mathcal{T}|/\delta)}{2n_p}}, \end{aligned}$$

6 EXPERIMENTAL RESULTS

We evaluate the proposed approaches for both simulated and real datasets [3]. We quantify the total

[3]Code for our methods and our datasets are available at `https://github.com/ramyakv/Learning-Population-of-Preferences-AISTATS2024`. This repoistory contains a dataset we have developed for preference learning, which is based on pairwise comparisons and was curated through crowdsourcing.

variation distance (TV) and Wasserstein distance between $\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}$ and the recovered mass in partitions $\mathcal{H}(\mathcal{T})$. For 1D setting, we use Wasserstein-1 distance; and for higher dimensional settings, we use the graph Wasserstein distance with normalized cost matrix written as follows,

$$\mathrm{W}_{\mathcal{G}}(\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})}) := \min_{\mathbf{K}\in\mathcal{M}(\mathbf{K})} \sum_{i=1}^{|\mathcal{H}(\mathcal{T})|}\sum_{j=1}^{|\mathcal{H}(\mathcal{T})|} \mathbf{K}_{i,j}\mathbf{C}_{i,j},$$

where $\mathbf{C}_{i,j}$ is the ratio of distance between nodes i and j to the maximum length on the graph induced by matrix $\mathbf{H}$; and $\mathcal{M}(\mathbf{K}) := \{\mathbf{K} : \mathbf{K} \geq 0,\ \mathbf{K}\mathbf{1} = \mathbf{p}^\star_{\mathcal{H}(\mathcal{T})},\ \mathbf{K}^T\mathbf{1} = \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})}\}$. Note that the total variation distance does not differentiate between whether the mass is moved between neighbor regions or any faraway region. Whereas Wasserstein distances take into account the geometry and hence distinguish between these scenarios.

For simulations, we consider four true user distributions: uniform, Gaussian, a mixture of two Gaussians, and a mixture of three Gaussians. We present simulation results with a mixture of three Gaussians here and defer the rest to the appendix. We also consider two types of noises. (a) Bernoulli(p_{flip}) that flips a simulated user's answer with probability p_{flip}. (b) flipping the answer of user $\boldsymbol{u}$ for pair $\boldsymbol{x}_a, \boldsymbol{x}_b$ with probability $\frac{1}{1+e^{-cd_{\text{diff}}}}$, where c is a scaling factor and $d_{\text{diff}} = -|(\text{dist}(\boldsymbol{x}_a, \boldsymbol{u}) - \text{dist}(\boldsymbol{x}_b, \boldsymbol{u}))|$. Here, we provide results only for $p_{\text{flip}} = 0.01$ and $c = 500$, and defer a more comprehensive analysis to the appendix. We sample $m = 5$ items uniformly at random from $[-1, 1]^d$. For each pairwise comparison among the items, we sample users from the underlying distribution, repeating 10 times. We use CVXPY (Diamond and Boyd, 2016; Agrawal et al., 2018), ECOS (Domahidi et al., 2013), and Gurobi (Gurobi Optimization, LLC, 2023) to solve optimization problems and run all simulations on Python 3.11. Parallel computations are done using GNU Parallel (Tange, 2018), CHTC (Center for High Throughput Computing, 2006), and OSG Consortium (Pordes et al., 2007; Sfiligoi et al., 2009; OSG, 2006).

1D Simulations: Figure 3 (a-b) show the relationship between the number of people asked per query, n_p, and the error, by varying $n_p \in \{10^2, 10^3, 10^4, 10^5\}$. As shown in our analysis, the recovery gets better as the number of users increases.

Colors dataset: Colors dataset (Palmer and Schloss, 2010; Palmer et al., 2013) consists of answers to pairwise queries from 48 different users and 37 colors. Each person was asked all $\binom{37}{2}$ pairwise comparisons. Each color is considered as a 3-dimensional vector in CIELAB color space (lightness, red vs. green, blue vs. yellow).

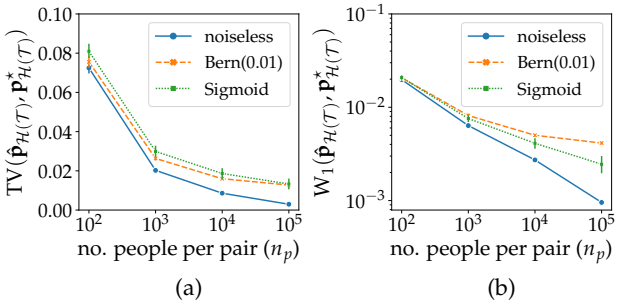

Figure 3: 1D setting: (a) $\mathrm{TV}(\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})})$ and (b) $\mathrm{W}_1(\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})})$ for mixture of 3 Gaussians.

For our experiment, we use the 1D user embedding of the colors dataset learned by Canal et al. (2022). We then project the CIELAB color space onto the user embedding space.

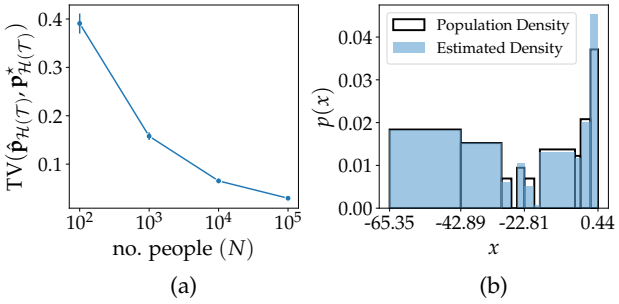

Figure 4: (a) $\mathrm{TV}(\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})})$ by varying number of people (b) $\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}$ and $\hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})}$ for Colors dataset.

We consider a subset of $m = 5$ colors sampled from this space and use all 10 pairs for comparison. Then, we uniformly sample $\{10^2, 10^3, 10^4, 10^5\}$ users from all 48 user preference points with replacement for each pair to estimate $\hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})}$ and report $\mathrm{TV}(\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})})$ in Figure 4 (a). Figure 4 (b) shows the true preference distribution for the population (computed using multiple queries per user on a separate set of users) and the distribution recovered via our method.

Simulations for $d \geq 2$: Figure 5 (a-b) shows the relationship between the number of people asked per query, n_p, and the error for $d = 2$. We use $\lambda = 1$ here and defer results with varying regularization parameter λ to the appendix.

We also provide simulation results using l_1, l_2-norm regularization, and maximum likelihood estimate (KL) as baselines in Figure 6. Additionally, to compare the performance between our method and Mallows model-based method, we implemented the EM algorithm described by Lu and Boutilier (2014) that learns

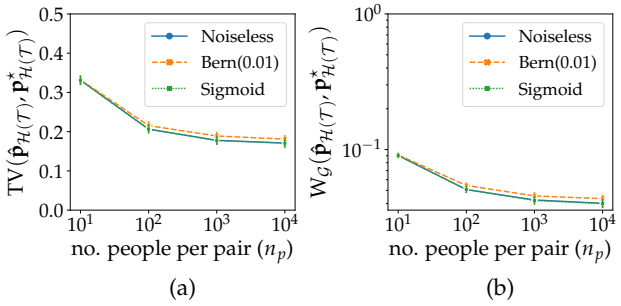

Figure 5: (a) $\mathrm{TV}(\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})})$, (b) $\mathrm{W}_{\mathcal{G}}(\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})})$ for mixture of 3 Gaussians in 2D ($\hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})}$ is recovered using graph regularization).

a mixture of Mallows model. Details regarding the EM algorithm are deferred to the appendix. Figure 6 includes the result with 2 mixtures.

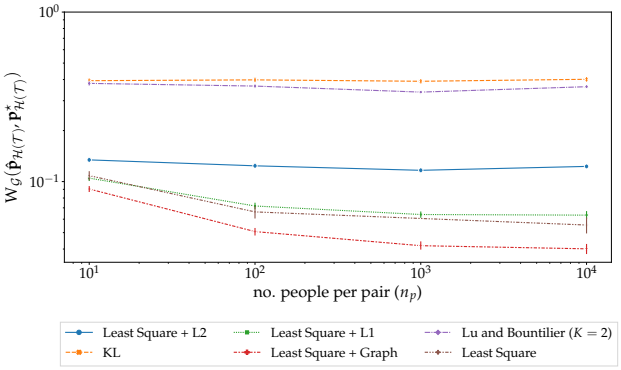

Figure 6: $\mathrm{W}_{\mathcal{G}}(\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})})$ for different objective functions with varying n_p.

Bounds on the Mass: We generate the true underlying preferences from a mixture of 3 Gaussians in 2D. We query for 5 pairs of items and 10,000 users per pair.

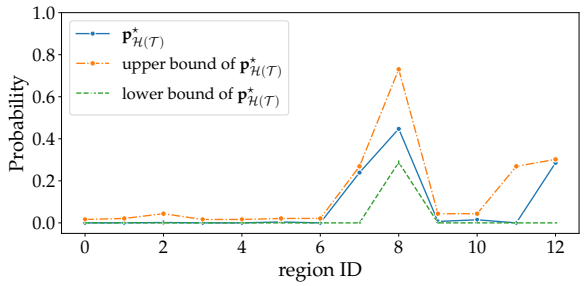

Figure 7: Lower and upper bounds with the true underlying distribution, a mixture of 3 Gaussians.

Figure 7 shows the upper and lower bounds on the mass in each of the regions in the intersection of the 5 hyperplanes using equations (2). We also show the true mass induced by the underlying distribution in these regions which highlights the efficacy of our bounds.

Zappos: UT Zappos50K (Yu and Grauman, 2014, 2017) is a large dataset with 50,025 catalog images of shoes in different categories, such as shoes, sandals, slippers, and boots. We manually pick five shoes from this dataset and collect responses from 6000 Amazon Mechanical Turk (AMT) workers for each possible pairwise query. With a subset of workers' answers to each possible pair, we estimate $\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}$ and use the remaining workers to answer pairwise comparison queries using only one response per worker to estimate $\hat{\mathbf{p}}$. We also use the method of Lu and Boutilier (2014) to estimate $\hat{\mathbf{p}}$. Details of the setting are deferred to the appendix. Figure 8 shows the results of our experiments on this dataset.

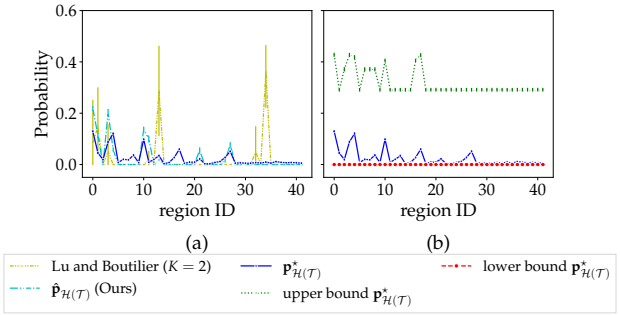

Figure 8: (a) True $\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}$ and $\hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})}$ recovered by our method in comparison to that by Lu and Boutilier (2014) for Zappos dataset. (b) Estimated upper and lower bounds for $\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}$ in Zappos dataset.

Movies: We create a new dataset comprising 4,266 movies from different countries, produced between 2013 and 2022, inclusive. Each movie is associated with its plot and info scrapped from Wikipedia (2023). We utilize *text-embedding-ada-002* model from OpenAI (2023) to generate an embedding for each movie. Then, we train a regression neural network with these embedding as input, and the target is each movie's average IMDB (2023) rating. We obtain the 100-dimensional output from the penultimate layer and reduce it to 2D using PaCMAP (Wang et al., 2021a). In the subsequent experiment, we consider the 2D embedding as the coordinates for the movies. We scrape the ratings of critics and audiences from Tomatoes (2023) and use them to create answers to pairwise comparison queries. We ran our experiment on a set of 13 DC Comics superhero movies. We consider 9 pairwise comparison questions and assign each pair to 50 reviewers. Our results are presented in Figure 9.

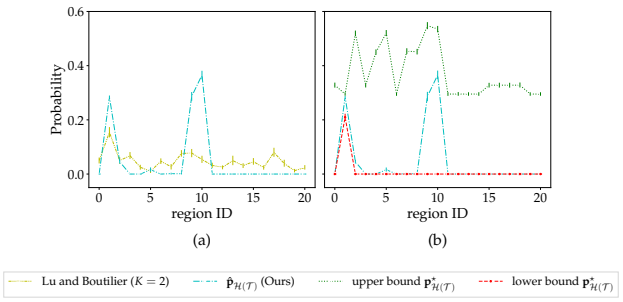

Figure 9: (a) $\hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})}$ recovered by our method in comparison to that by Lu and Boutilier (2014) for Movies dataset (b) Estimated upper and lower bounds for $\mathbf{p}^{\star}_{\mathcal{H}(\mathcal{T})}$ in movies dataset.

7 CONCLUSIONS AND FUTURE WORK

We propose a novel problem of learning distribution of user preferences from pairwise comparison queries. We focus on fundamental questions regarding what we can learn about the underlying distribution from a single query per user. We show that the problem is identifiable in 1D setting and provide recovery guarantees under the total variation distance. We show that this problem is not identifiable in dimensions $d \geq 2$. We provide upper and lower bounds on the masses in the regions formed by the intersecting hyperplanes corresponding to the queried pairs. We proposed using graph regularization for recovery of the masses in these regions and provide bound on the total variation distance between the true distribution and the estimated distribution. We validate these fundamental results on extensive numerical simulations. Furthermore, we show the efficacy of the proposed methods on real datasets. As a byproduct of this work, we introduce two new datasets for learning distribution of user preferences. In the future, we would like to mathematically characterize how large the set of underlying preference distribution that lead to the same answers to pairwise queries in terms of the TV and Wasserstein distances. We would also like to further explore what other structures on the underlying distributions make it amenable to overcome non-identifiability and develop recovery guarantees under the graph Wasserstein distance which takes into account the geometry of the feature space.

Acknowledgements

This work was partially supported by NSF grants NCS-FO 2219903 and NSF CAREER Award CCF 2238876. We thank Harit Vishwakarma for his valuable feedback and proofreading. We thank the anonymous reviewers for their valuable comments and constructive feedback on our work.

References

J. Abildtrup, E. Audsley, M. Fekete-Farkas, C. Giupponi, M. Gylling, P. Rosato, and M. Rounsevell. Socio-economic scenario development for the assessment of climate change impacts on agricultural land use: a pairwise comparison approach. *Environmental science & policy*, 9(2):101–115, 2006.

A. Agrawal, R. Verschueren, S. Diamond, and S. Boyd. A rewriting system for convex optimization problems. *Journal of Control and Decision*, 5(1):42–60, 2018.

AMT. Amazon mechanical turk. `https://www.mturk.com/`. Accessed: 2023-04-27.

P. Awasthi, A. Blum, O. Sheffet, and A. Vijayaraghavan. Learning mixtures of ranking models. *Advances in Neural Information Processing Systems*, 27, 2014.

R. G. Baraniuk. Compressive sensing [lecture notes]. *IEEE Signal Processing Magazine*, 24(4):118–121, 2007. doi: 10.1109/MSP.2007.4286571.

M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In T. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems*, volume 14. MIT Press, 2001. URL `https://proceedings.neurips.cc/paper/2001/file/f106b7f99d2cb30c3db1c3cc0fde9ccb-Paper.pdf`.

A. Bellet, A. Habrard, and M. Sebban. *Metric Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, 2015. doi: 10.2200/S00626ED1V01Y201501AIM030. URL `https://doi.org/10.2200/S00626ED1V01Y201501AIM030`.

A. Bhargava, R. Ganti, and R. D. Nowak. Active algorithms for preference learning problems with multiple populations. *arXiv: Machine Learning*, 2016.

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

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

M. Braverman and E. Mossel. Noisy sorting without resampling. *arXiv preprint arXiv:0707.1051*, 2007.

R. C. Buck. Partition of space. *The American Mathematical Monthly*, 50(9):541–544, 1943. doi: 10.1080/00029890.1943.11991447. URL `https://doi.org/10.1080/00029890.1943.11991447`.

D. Cai, X. He, J. Han, and T. S. Huang. Graph regularized nonnegative matrix factorization for data representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1548–1560, 2011. doi: 10.1109/TPAMI.2010.231.

G. Canal, B. Mason, R. Korlakai Vinayak, and R. Nowak. One for all: Simultaneous metric and preference learning over multiple users. *Advances in Neural Information Processing Systems*, 35:4943–4956, 2022.

E. J. Candes and M. B. Wakin. An introduction to compressive sampling. *IEEE Signal Processing Magazine*, 25(2):21–30, 2008. doi: 10.1109/MSP.2007.914731.

Center for High Throughput Computing. Center for high throughput computing, 2006. URL `https://chtc.cs.wisc.edu/`.

L. Condat. Least-squares on the simplex for multispectral unmixing. *Res. Rep, GIPSA-Lab, Univ. Grenoble Alpes, Grenoble, France*, 2017.

C. H. Coombs. Psychological scaling without a unit of measurement. *Psychological review*, 57(3):145, 1950.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

S. Diamond and S. Boyd. CVXPY: A Python-embedded modeling language for convex optimization. *Journal of Machine Learning Research*, 17(83):1–5, 2016.

C. Ding. Evaluating change in behavioral preferences: Multidimensional scaling single-ideal point model. *Measurement and Evaluation in Counseling and Development*, 49(1):77–88, 2016.

A. Domahidi, E. Chu, and S. Boyd. ECOS: An SOCP solver for embedded systems. In *European Control Conference (ECC)*, pages 3071–3076, 2013.

M. Elad. Sparse and redundant representations: From theory to applications in signal and image processing, 2010.

B. Eriksson. Learning to top-k search using pairwise comparisons. In *Artificial Intelligence and Statistics*, pages 265–273. PMLR, 2013.

J. Fichtner. On deriving priority vectors from matrices of pairwise comparisons. *Socio-Economic Planning Sciences*, 20(6):341–345, 1986.

S. Foucart. Stability and robustness of $\ell 1$-minimizations with weibull matrices and redundant dictionaries. *Linear Algebra and its Applications*, 441:4–21, 2014. ISSN 0024-3795. doi: https://doi.org/10.1016/j.laa.2012.10.003. URL `https://www.sciencedirect.com/science/article/pii/S0024379512007008`. Special Issue on Sparse Approximate Solution of Linear Systems.

S. Foucart and H. Rauhut. A mathematical introduction to compressive sensing. In *Applied and Numerical Harmonic Analysis*, 2013.

P. E. Green. Marketing applications of mds: Assessment and outlook. *Journal of Marketing*, 39(1):24–31, 1975. ISSN 00222429. URL `http://www.jstor.org/stable/1250799`.

Gurobi Optimization, LLC. Gurobi Optimizer Reference Manual, 2023. URL `https://www.gurobi.com`.

R. Hadsell, S. Chopra, and Y. LeCun. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742, 2006. doi: 10.1109/CVPR.2006.100.

W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. ISSN 01621459. URL `http://www.jstor.org/stable/2282952`.

D. J. Hopkins and H. Noel. Trump and the shifting meaning of "conservative": Using activists' pairwise comparisons to measure politicians' perceived ideologies. *American Political Science Review*, 116(3):1133–1140, 2022.

D. R. Hunter. Mm algorithms for generalized bradley-terry models. *The annals of statistics*, 32(1):384–406, 2004.

IMDB. Imdb non-commercial dataset. `https://developer.imdb.com/non-commercial-datasets/`, 2023. URL `https://developer.imdb.com/non-commercial-datasets/`. Accessed: 2023-5-12.

L. Jain, B. Mason, and R. Nowak. Learning low-dimensional metrics, 2017. URL `https://arxiv.org/abs/1709.06171`.

K. G. Jamieson and R. Nowak. Active ranking using pairwise comparisons. *Advances in neural information processing systems*, 24, 2011.

R. M. Johnson. Market segmentation: A strategic management tool. *Journal of Marketing Research*, 8 (1):13–18, 1971. ISSN 00222437. URL `http://www.jstor.org/stable/3149720`.

C. Kenyon-Mathieu and W. Schudy. How to rank with few errors. In *Proceedings of the thirty-ninth annual ACM symposium on Theory of computing*, pages 95–103, 2007.

R. Kueng and J. A. Tropp. Binary component decomposition part i: the positive-semidefinite case. *SIAM Journal on Mathematics of Data Science*, 3 (2):544–572, 2021.

B. Kulis. Metric learning: A survey. *Foundations and Trends in Machine Learning*, 5, 01 2012. doi: 10.1561/2200000019.

A. Liu and A. Moitra. Efficiently learning mixtures of mallows models. In *2018 IEEE 59th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 627–638. IEEE, 2018.

M. Lotfi and M. Vidyasagar. Compressed sensing using binary matrices of nearly optimal dimensions. *IEEE Transactions on Signal Processing*, 68:3008–3021, 2020. doi: 10.1109/TSP.2020.2990154.

T. Lu and C. Boutilier. Effective sampling and learning for mallows models with pairwise-preference data. *J. Mach. Learn. Res.*, 15(1):3783–3829, 2014.

R. D. Luce. *Individual choice behavior: A theoretical analysis*. New York: Wiley, 1959.

C. L. Mallows. Non-null ranking models. i. *Biometrika*, 44(1/2):114–130, 1957. ISSN 00063444. URL `http://www.jstor.org/stable/2333244`.

C. Mao and Y. Wu. Learning mixtures of permutations: Groups of pairwise comparisons and combinatorial method of moments. *The Annals of Statistics*, 50(4): 2231–2255, 2022.

A. K. Massimino and M. A. Davenport. As you like it: Localization via paired comparisons. *J. Mach. Learn. Res.*, 22:186–1, 2021.

S. Negahban, S. Oh, and D. Shah. Iterative ranking from pair-wise comparisons. *Advances in neural information processing systems*, 25, 2012.

S. Oishi, J. Hahn, U. Schimmack, P. Radhakrishan, V. Dzokoto, and S. Ahadi. The measurement of values across cultures: A pairwise comparison approach. *Journal of research in Personality*, 39(2):299–305, 2005.

OpenAI. text-embedding-ada-002. `https://openai.com/blog/new-and-improved-embedding-model`, 2023. URL `https://openai.com/blog/new-and-improved-embedding-model`. Accessed: 2023-5-12.

OSG. Ospool, 2006. URL `https://osg-htc.org/services/open_science_pool.html`.

S. E. Palmer and K. B. Schloss. An ecological valence theory of human color preference. *Proceedings of the National Academy of Sciences*, 107(19):8877–8882, 2010.

S. E. Palmer, K. B. Schloss, and J. Sammartino. Visual aesthetics and human preference. *Annual review of psychology*, 64:77–107, 2013.

C. Poignard, T. Pereira, and J. P. Pade. Spectra of laplacian matrices of weighted graphs: Structural genericity properties. *SIAM Journal on Applied Mathematics*, 78(1):372–394, 2018. doi: 10.1137/17M1124474. URL `https://doi.org/10.1137/17M1124474`.

R. Pordes, D. Petravick, B. Kramer, D. Olson, M. Livny, A. Roy, P. Avery, K. Blackburn, T. Wenaus, F. Würthwein, I. Foster, R. Gardner, M. Wilde, A. Blatecky, J. McGee, and R. Quick. The open science grid. In *J. Phys. Conf. Ser.*, volume 78 of *78*, page 012057, 2007. doi: 10.1088/1742-6596/78/1/012057.

A. Rajkumar and S. Agarwal. A statistical convergence perspective of algorithms for rank aggregation from pairwise data. In *International conference on machine learning*, pages 118–126. PMLR, 2014.

T. L. Saaty and L. G. Vargas. The possibility of group choice: pairwise comparisons and merging functions. *Social Choice and Welfare*, 38(3):481–496, 2012.

R. Schneider. *Convex Bodies: The Brunn–Minkowski Theory*. Encyclopedia of Mathematics and its Applications. Cambridge University Press, 2 edition, 2013. doi: 10.1017/CBO9781139003858.

I. Sfiligoi, D. C. Bradley, B. Holzman, P. Mhashilkar, S. Padhi, and F. Wurthwein. The pilot way to grid resources using glideinwms. In *2009 WRI World Congress on Computer Science and Information Engineering*, volume 2 of *2*, pages 428–432, 2009. doi: 10.1109/CSIE.2009.950.

N. Shah, S. Balakrishnan, A. Guntuboyina, and M. Wainwright. Stochastically transitive models for pairwise comparisons: Statistical and computational issues. In *International Conference on Machine Learning*, pages 11–20. PMLR, 2016.

N. B. Shah and M. J. Wainwright. Simple, robust and optimal ranking from pairwise comparisons. *The Journal of Machine Learning Research*, 18(1):7246–7283, 2017.

R. Shepard. The analysis of proximities: Multidimensional scaling with an unknown distance function. I. *Psychometrika*, 27(2):125–140, June 1962a. doi: 10.1007/BF02289630. URL `https://ideas.repec.org/a/spr/psycho/v27y1962i2p125-140.html`.

R. N. Shepard. The analysis of proximities: Multidimensional scaling with an unknown distance function. ii. *Psychometrika*, 27:219–246, 1962b.

R. N. Shepard. Metric structures in ordinal data. *Journal of Mathematical Psychology*, 3:287–315, 1966. URL `https://api.semanticscholar.org/CorpusID:120906276`.

K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

A. Singla, S. Tschiatschek, and A. Krause. Actively learning hemimetrics with applications to eliciting user preferences. In *International Conference on Machine Learning*, pages 412–420. PMLR, 2016.

O. Tange. *GNU parallel 2018*. Lulu. com, 2018.

G. Tatli, R. Nowak, and R. K. Vinayak. Learning preference distributions from distance measurements. *58th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2022.

R. Tomatoes. Rotten tomatoes. `https://www.rottentomatoes.com/`, 2023. URL `https://www.rottentomatoes.com/`. Accessed: 2023-5-12.

Y. Wang, H. Huang, C. Rudin, and Y. Shaposhnik. Understanding how dimension reduction tools work: An empirical approach to deciphering t-sne, umap, trimap, and pacmap for data visualization. *Journal of Machine Learning Research*, 22(201):1–73, 2021a. URL `http://jmlr.org/papers/v22/20-1061.html`.

Y. Wang, H. Huang, C. Rudin, and Y. Shaposhnik. Understanding how dimension reduction tools work: an empirical approach to deciphering t-sne, umap, trimap, and pacmap for data visualization. *The Journal of Machine Learning Research*, 22(1):9129–9201, 2021b.

Wikipedia. Wikipedia. `https://en.wikipedia.org/`, 2023. URL `https://en.wikipedia.org/`. Accessed: 2023-5-12.

A. Xu and M. Davenport. Simultaneous preference and metric learning from paired comparisons. *Advances in Neural Information Processing Systems*, 33:454–465, 2020.

A. Yu and K. Grauman. Fine-grained visual comparisons with local learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 192–199, 2014.

A. Yu and K. Grauman. Semantic jitter: Dense supervision for visual comparisons via synthetic images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 5570–5579, 2017.

X. Zhang, X. Zhang, P.-L. Loh, and Y. Liang. On the identifiability of mixtures of ranking models. *arXiv preprint arXiv:2201.13132*, 2022.

Checklist

1. For all models and algorithms presented, check if you include:

   (a) A clear description of the mathematical setting, assumptions, algorithm, and/or model. [Yes] We have included key descriptions in the main paper and provided additional details of the algorithms in the supplementary material.

   (b) An analysis of the properties and complexity (time, space, sample size) of any algorithm. [Yes] We have included short analysis for algorithms in the main paper regarding sample complexity and provided details in the supplementary material.

   (c) (Optional) Anonymized source code, with specification of all dependencies, including external libraries. [Yes] We will upload the source code with the supplementary material.

2. For any theoretical claim, check if you include:

   (a) Statements of the full set of assumptions of all theoretical results. [Yes] We provide the details of the settings and assumptions in sections 2, 4 and 5.

   (b) Complete proofs of all theoretical results. [Yes] We have included details of proofs in the supplementary material.

   (c) Clear explanations of any assumptions. [Yes] We have explained any assumption that we had in the main paper and supplementary material.

3. For all figures and tables that present empirical results, check if you include:

   (a) The code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL). [Yes] We have included them in the supplementary material.

   (b) All the training details (e.g., data splits, hyperparameters, how they were chosen). [Yes] We have included training details of each experiment in the supplementary material.

   (c) A clear definition of the specific measure or statistics and error bars (e.g., with respect to the random seed after running experiments multiple times). [Yes] They are included within details of simulations and experiments

   (d) A description of the computing infrastructure used. (e.g., type of GPUs, internal cluster, or cloud provider). [Yes] They are included in both main paper and the supplementary material, when it is necessary.

4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets, check if you include:

   (a) Citations of the creator If your work uses

existing assets. [Yes] We have referred any existing assets that we used and cited their source.

(b) The license information of the assets, if applicable. [Not applicable]

(c) New assets either in the supplemental material or as a URL, if applicable. [Yes]

(d) Information about consent from data providers/curators. [Not Applicable]

(e) Discussion of sensible content if applicable, e.g., personally identifiable information or offensive content. [Not Applicable]

5. If you used crowdsourcing or conducted research with human subjects, check if you include:

(a) The full text of instructions given to participants and screenshots. [Yes] See Appendix Section D.4.

(b) Descriptions of potential participant risks, with links to Institutional Review Board (IRB) approvals if applicable. [Not Applicable]

(c) The estimated hourly wage paid to participants and the total amount spent on participant compensation. [Yes] See Appendix Section D.4.

Learning Populations of Preferences via Pairwise Comparison Queries: Supplementary Materials

A Related Works

Preference learning based on ideal point model (Coombs, 1950; Jamieson and Nowak, 2011; Ding, 2016; Singla et al., 2016; Xu and Davenport, 2020; Massimino and Davenport, 2021; Canal et al., 2022) has been studied by several works. Learning a user preference point up to ε-error with pairwise comparisons requires $\mathcal{O}(d/\log(d/\varepsilon))$ queries under mild assumptions of coverage of query items on the space of preferences (Massimino and Davenport, 2021). A recent work by Xu and Davenport (2020) considered the problem of simultaneously learning an unknown metric and a user preference point and proposed an alternating minimization algorithm. A key limitation of these works is that they either focus on learning an individual preference by making many queries per individual or assume homogeneity and learn a single preference point using data from all the users. Another recent work by Canal et al. (2022) considers a setting with multiple users with different preference points, but the focus there is to learn each individual preference point while also learning an unknown metric that is common across the users and the pairwise queries per individual scales as $\tilde{\mathcal{O}}(d)$ which is needed to learn the individual preferences (where $\tilde{\mathcal{O}}(.)$ hides log factors). However, querying multiple pairs per user is an hurdle in various situations, e.g. privacy concerns due to tracking a user over time, cognitive overload, and cost in obtaining answers to multiple queries, especially in larger dimensional spaces. When we have only one query per users these methods are not applicable unless all users are homogeneous in which case one can pool in all the data to learn a single preference point.

A recent work by Tatli et al. (2022) considers a *distance query model*, i.e. when a user is queried with an item, the response is how far their preference point is from the queried item, and studies the problem of learning the distribution of preferences for the specific case of discrete 1D discrete distributions. Under this setting, they provide sufficient condition in terms of number of items in the query set and their locations on the 1D grid that makes the problem identifiable – just one item in the query set is sufficient if it is at the edge and two items in the query set are sufficient otherwise to identify the mass of preferences on all the locations on the 1D grid. They also use a linear system formulation to represent their setting and provide guarantees on recovering the discrete preference distribution on the 1D grid by solving constrained least square optimization. In contrast, in our setting the responses are for pairwise comparison between items and we study both 1D and higher dimensional settings and not restricted to discrete distributions for user preferences. Instead, we focus on learning $\mathbf{p}^*_{\mathcal{H}(\mathcal{T})}$, i.e., the mass of user preferences induced in the regions created by the intersection of the hyperplanes that correspond to pairwise comparisons. The differences in the query model and the geometry lead to different insights in terms of the nature of identifiability issues as well as the properties of the linear systems that arise with pairwise comparisons. For example, for the 1D case in our setting, depending on the number of pairs being compared, we get different number of intervals. If we have just two items in the query set, we only have one pairwise comparison, leading to two intervals and we could only hope to learn mass on either side of the midpoint between the two items. Instead, if we have 3 items, we can have 3 possible pairwise comparisons and 4 intervals and we can learn the mass in these 4 intervals and so on. So, with more pairs leading to more intervals, we can learn more refined information about the underlying continuous distribution of preferences. In higher dimensional case, we show that the problem of learning $\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}$ is not identifiable in our setting (Proposition 2). We further show that if the true solution $\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}$ is sparse and satisfies certain geometric property, then the problem is identifiable (Theorem 5.1).

Another line of work in preference learning involves ranking based models, e.g., Bradley-Terry-Luce (BTL) model (Bradley and Terry, 1952; Luce, 1959), Mallows model (Mallows, 1957), stochastic transitivity models (Shah et al., 2016), that focus on finding a single ranking of m items or finding top-k items by pairwise comparisons (Hunter, 2004; Kenyon-Mathieu and Schudy, 2007; Braverman and Mossel, 2007; Negahban et al., 2012; Eriksson, 2013; Rajkumar and Agarwal, 2014; Shah and Wainwright, 2017). Ranking m items in these settings requires $\mathcal{O}(m \log m)$ queries. Note in contrast, under the ideal point based models, the query complexity

for ranking reduces to $\mathcal{O}(d \log m)$, where d is the dimension of the domain of representations which is usually much smaller than the number of items being ranked (Jamieson and Nowak, 2011). This is due to the fact that the geometry induces constraint on the number of possible rankings from $m!$ to $\mathcal{O}(m^{2d})$. Lu and Boutilier (2014) consider a mixture of K-Mallows model for the ranking setting and propose an EM algorithm that takes user preferences as input and outputs the estimated model parameters. However, there are no guarantees provided for convergence of this algorithm. Other works on ranking models (Awasthi et al., 2014; Mao and Wu, 2022) have noted that even a two component mixture of Mallows model is not identifiable with pairwise comparisons and the focus has been on settings that consider samples that are full or list-wise or group-wise ranking data per user and tensor-based algorithms for recovery. A recent work (Zhang et al., 2022) considers a mixture of two BTL-models with pairwise comparison and shows that it is identifiable except for certain sets of parameters and show similar results for list-wise query setting, but it does not provide algorithmic approaches to estimate them. Our work focuses on preference learning based on ideal point model with pairwise comparison queries and develops fundamental understanding under a nonparametric setting.

B Limitations

We study the novel problem of *learning populations of preferences via pairwise comparison queries* when we are limited to making one query per individual. We show that the problem is identifiable in 1D setting and provide recovery guarantees. Further, we show that the problem is not identifiable in dimensions $d \geq 2$. Linear system of equations in (1) is underdetermined in dimensions $d \geq 2$. So, we cannot recover $\mathbf{p}^{\star}_{\mathcal{H}(\mathcal{T})}$ in dimensions $d \geq 2$, unless the true solution $\mathbf{p}^{\star}_{\mathcal{H}(\mathcal{T})}$ is sparse and satisfies certain geometric property. Therefore, we propose using graph regularization for recovery of masses in $\mathcal{H}(\mathcal{T})$ and provide recovery guarantees. Our recovery guarantees are limited to the noiseless setting. For noisy settings, we show simulation results that are promising. Furthermore, the suitability of the regularizer depends on the property of underlying distribution of preferences. We have explored one such regularization technique in this work. Theoretical analysis of noisy setting and other regularizers suited for different properties would be interesting to study in the future.

In this work, we focus on the case where we can only make one comparison query per individual. On the other end, if we can make $\tilde{\mathcal{O}}(d)$ queries per individual, we can estimate individual preferences. We expect there is a trade-off between these two regimes, that is, single query per individual to enough queries to learn individual preference points, in terms of information gain regarding the underlying distribution of preferences, which is left to future work for further investigation.

In our problem setup, we assume that item representations are known and learn the distribution of preferences. In the era of large foundational models, we presume that we can learn feature representation of items as we did in our experiments in the paper. On the other hand, euclidean distance in the item representation space might not reflect the correct embeddings for the judgements of the population. So, one can try to learn the distance function or tune the embeddings further to reflect human judgements. In our experiments, we use other regression or classification tasks to refine the pre-trained embeddings to lower dimensional feature space. However, there is a rich line of research about learning distance metric using triplet queries of type "Is a closer to b or c?", and often low-dimensional metrics seem to capture human preferences well (Shepard, 1962b,a, 1966; Kulis, 2012; Bellet et al., 2015; Jain et al., 2017). Note that learning metric from triplet queries is essentially akin to contrastive learning in the deep learning literature. Recent works by Xu and Davenport (2020); Canal et al. (2022) have proposed learning metric and preferences simultaneously from pairwise comparison queries where both unknown metric and unknown user preferences are learned simultaneously. So, these methods need learning user preferences as well and therefore need at least $\tilde{\mathcal{O}}(d)$ queries per individual. We leave it as a future direction to explore the problem of learning distribution of preferences for an unknown metric without learning user preferences exactly.

C Proofs

C.1 Proof of Theorem 4.1

We recall that $\hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})}$ is the solution to the constrained least square optimization problem with unit simplex constraint in Section 4. Then, as discussed in pages 301-302 by Boyd and Vandenberghe (2004) and mentioned by Condat (2017), and also observed in the proof of Theorem 2 by Tatli et al. (2022), we note that $\mathbf{H}\hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})}$ is the projection of $\hat{\mathbf{q}}$ onto the closed convex set $C_{\mathbf{H}}$ under ℓ_2 distance, which we call $\mathrm{P}_{C_{\mathbf{H}}}(\hat{\mathbf{q}})$, where

$$C_{\mathbf{H}} := \mathrm{conv}(\mathbf{H}e_1, \ldots, \mathbf{H}e_m).$$

Therefore, we can write

$$\begin{aligned} ||\mathbf{p}^*_{\mathcal{H}(\mathcal{T})} - \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})}||_2 &= ||\mathbf{H}^\dagger(\mathbf{q}^\star - \mathrm{P}_{C_\mathbf{H}}(\hat{\mathbf{q}}))||_2 \\ &\leq ||\mathbf{H}^\dagger||_2||\mathbf{q}^\star - \mathrm{P}_{C_\mathbf{H}}(\hat{\mathbf{q}})||_2 \\ &\leq ||\mathbf{H}^\dagger||_2||\mathbf{q}^\star - \hat{\mathbf{q}}||_2 \\ &\leq ||\mathbf{H}^\dagger||_2||\mathbf{q}^\star - \hat{\mathbf{q}}||_1, \end{aligned} \tag{7}$$

where the inequality (7) is due the property that the projection onto closed convex sets is contracting (Thm. 1.2.2. by Schneider (2013)). Then, we note that $2\ \mathrm{TV}(\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})}) = ||\mathbf{p}^*_{\mathcal{H}(\mathcal{T})} - \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})}||_1$, and use $l_1 - l_2$ norm inequality to obtain the following from (7),

$$\begin{aligned} \mathrm{TV}(\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})}) = \frac{1}{2}||\mathbf{p}^*_{\mathcal{H}(\mathcal{T})} - \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})}||_1 &\leq \frac{1}{2}\sqrt{|\mathcal{T}|+1}||\mathbf{p}^*_{\mathcal{H}(\mathcal{T})} - \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})}||_2 \\ &\leq \frac{1}{2}\sqrt{|\mathcal{T}|+1}||\mathbf{H}^\dagger||_2||\hat{\mathbf{q}} - \mathbf{q}^\star||_1. \end{aligned}$$

Recall from Section 4, $\mathbf{H}$ can be written as the concatenation of one lower triangular and one upper triangular binary matrix, e.g., 3 pairs create 4 regions, and the corresponding matrix

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Each possible $\mathbf{H}$ matrix from Theorem 4.1 has a similar structure with $|\mathcal{T}|$ 1's in each column with shifted positions as we pass through columns. This very specific structure allows us to express a left inverse $\mathbf{H}^\dagger = \frac{1}{|\mathcal{T}|}\mathbf{1}_{|\mathcal{T}|+1}\mathbf{1}^T_{2|\mathcal{T}|} + \mathbf{S} = (-\frac{1}{|\mathcal{T}|}\mathbf{1}_{|\mathcal{T}|+1}\mathbf{1}^T_{|\mathcal{T}|+1} + \mathbf{I})\mathbf{S}$, where $\mathbf{S}$ is $(|\mathcal{T}|+1) \times 2|\mathcal{T}|$ with $\mathbf{S}_{i,j} = -1$ when $j = i-1$ and $j = i + |\mathcal{T}|$, and $\mathbf{S}_{i,j} = 0$ everywhere else. For the given example above,

$$\mathbf{H}^\dagger = \begin{bmatrix} 1/3 & 1/3 & 1/3 & -2/3 & 1/3 & 1/3 \\ -2/3 & 1/3 & 1/3 & 1/3 & -2/3 & 1/3 \\ 1/3 & -2/3 & 1/3 & 1/3 & 1/3 & -2/3 \\ 1/3 & 1/3 & -2/3 & 1/3 & 1/3 & 1/3 \end{bmatrix} \text{ with } \mathbf{S} = \begin{bmatrix} 0 & 0 & 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 0 & -1 & 0 \\ 0 & -1 & 0 & 0 & 0 & -1 \\ 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix}.$$

Note that, by construction, $\mathbf{S}$ has only one non-zero entry in each column and 2 non-zero entries in each row except first and second rows which have 1 nonzero entry. Therefore, $\mathbf{S}\mathbf{S}^T$ is a diagonal matrix such that $\mathbf{S}_{1,1} = \mathbf{S}_{|\mathcal{T}|+1,|\mathcal{T}|+1} = 1$ and the rest is 2. It easily follows that $\|\mathbf{S}\|_2 = \sqrt{2}$, since $\|\mathbf{S}\mathbf{S}^T\|_2 = 2$. We also note that

$$\|\mathbf{H}^\dagger\|_2 = \|(-\frac{1}{|\mathcal{T}|}\mathbf{1}\mathbf{1}^T + \mathbf{I})\mathbf{S}\|_2 \leq \| - \frac{1}{|\mathcal{T}|}\mathbf{1}\mathbf{1}^T + \mathbf{I}\|_2\|\mathbf{S}\|_2 = 1 * \sqrt{2} = \sqrt{2}. \tag{8}$$

Therefore, we can write

$$\mathrm{TV}(\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})}) \leq \sqrt{\frac{|\mathcal{T}|+1}{2}}\|\hat{\mathbf{q}} - \mathbf{q}^\star\|_1. \tag{9}$$

Now, note that the term $||\hat{\mathbf{q}} - \mathbf{q}^\star||_1$ is the sum of l_1- distances between the empirical and the true fraction of people on either side of hyperplanes in $\mathcal{T}$. Therefore, any entry $\hat{\mathbf{q}}_i$ can be considered a summation of n_p binary random variables with mean $\mathbf{q}^\star_i$, i.e., $\mathbb{E}[\hat{\mathbf{q}}_i] = \mathbf{q}^\star_i$, where n_p is the number of people queried per pairwise comparison query. We, then, can apply Hoeffding's inequality (Hoeffding, 1963) (which we have restated below) for each pair queried (i.e., for each hyperplane) to obtain a bound for $|\hat{\mathbf{q}}_i - \mathbf{q}^\star_i|$ and write that,

$$|\hat{\mathbf{q}}_i - \mathbf{q}^\star_i| \leq \sqrt{\frac{\log(2/\delta')}{2n_p}} \tag{10}$$

holds with probability at least $1-\delta'$. We then use union bound bound over all the $|\mathcal{T}|$ pairs to obtain the bound on $\|\hat{\mathbf{q}}-\mathbf{q}^\star\|_1$.

Hoeffding's Inequality (Hoeffding, 1963): Let $X_1, X_2 \ldots, X_N$ be independent random variables such that $a_i \le X_i \le b_i$ and let $S_N := \sum_{i=1}^N X_i$, then for all $t > 0$,

$$\Pr(|S_N - \mathbb{E}(S_N)| \ge t) \le 2\exp\left(-\frac{2t^2}{\sum_{i=1}^N (b_i-a_i)^2}\right). \tag{11}$$

C.2 Proof of Proposition 2

We first note that the entry-wise product, i.e., Hadamard product, of the rows of $\mathbf{H}$ that correspond to the columns with 1's in the j-th position leads to a standard vector with 1 in the j-th entry. So, we can write the following,

$$\mathbf{e}_j = \prod_{i\in\mathcal{K}_j}^{\odot} \mathbf{H}_{i,:}, \quad j = 1,\ldots,|\mathcal{H}(\mathcal{T})|, \tag{12}$$

where $\mathbf{e}_j$ is the standard basis vector with j-th entry is 1, $\odot$ represents Hadamard product operation and $\mathcal{K}_j$ denotes the set of rows of $\mathbf{H}$ whose j-th entry is 1. Then, considering the structure of matrix $\mathbf{H}$, we note that

$$\sum_{i=1}^{2|\mathcal{T}|} \lambda_i \mathbf{H}_{i,:} = \sum_{i=1}^{|\mathcal{T}|} (\lambda_i - \lambda_{|\mathcal{T}|+i})\mathbf{H}_{i,:} + \left(\sum_{i=1}^{|\mathcal{T}|} \lambda_{|\mathcal{T}|+i}\right)\mathbf{1}.$$

If $\sum_{i=1}^{|\mathcal{T}|}(\lambda_i - \lambda_{|\mathcal{T}|+i})\mathbf{H}_{i,:} + \left(\sum_{i=1}^{|\mathcal{T}|}\lambda_{|\mathcal{T}|+i}\right)\mathbf{1} = 0$ holds only when $\lambda_i - \lambda_{|\mathcal{T}|+i} = 0$ for all $i = 1,\ldots,|\mathcal{T}|$ and $\sum_{i=1}^{|\mathcal{T}|}\lambda_{|\mathcal{T}|+i} = 0$, we can claim that $\mathbf{1}$ and $\mathbf{H}_{i,:}$'s for $i = 1,\ldots,|\mathcal{T}|$ are linearly independent. Therefore, we suppose that

$$\sum_{i=1}^{2|\mathcal{T}|} \lambda_i \mathbf{H}_{i,:} = \sum_{i=1}^{|\mathcal{T}|} (\lambda_i - \lambda_{|\mathcal{T}|+i})\mathbf{H}_{i,:} + \left(\sum_{i=1}^{|\mathcal{T}|} \lambda_{|\mathcal{T}|+i}\right)\mathbf{1} = 0.$$

Now, we take $|\mathcal{T}|$-th power of the left-hand side with respect to Hadamard product and write it as follows:

$$\left(\sum_{i=1}^{2|\mathcal{T}|} \lambda_i \mathbf{H}_{i,:}\right)^{\odot|\mathcal{T}|} = 0 \tag{13}$$

Considering results of all products in given expression, we can write following Lemma.

Lemma 1. *Given the binary matrix $\mathbf{H} \in \{0,1\}^{2|\mathcal{T}|\times|\mathcal{H}(\mathcal{T}))|}$ in (1) and real coefficients λ_i's, we can write following*

$$\left(\sum_{i=1}^{2|\mathcal{T}|} \lambda_i \mathbf{H}_{i,:}\right)^{\odot|\mathcal{T}|} = \sum_{j=1}^{2|\mathcal{T}|} \left(\sum_{i\in\mathcal{K}_j} \lambda_i\right)^{|\mathcal{T}|} \mathbf{e}_j,$$

where $\mathcal{K}_j$ is the position of rows of $\mathbf{H}$ whose j-th entry is 1 and $\mathbf{e}_j$'s are standard basis vectors.

Lemma 2. *Given the binary matrix $\mathbf{H}$ in Section 2, for any $j \le |\mathcal{T}|$, we can find two columns $\mathbf{H}_{:,j_1}$ and $\mathbf{H}_{:,j_2}$ such that only j-th and $(|\mathcal{T}|+j)$-th entries of $\mathbf{H}_{:,j_1}$ and $\mathbf{H}_{:,j_2}$ differ.*

Proof: *Each hyperplane has to form neighboring regions by construction. Therefore, there exists two columns $\mathbf{H}_{:,j_1}$ and $\mathbf{H}_{:,j_2}$ such that only j-th and $(|\mathcal{T}|+j)$-th entries differ. To understand it better, we can consider a scenario where we delete j-th row of the matrix $\mathbf{H}$ and call $\mathbf{H}^j$ to this new matrix. $\mathbf{H}^j$ has to have a pair of same columns. Otherwise, we would conclude that j-th hyperplane does not form new regions, which is not possible by construction, when we consider adding one hyperplane at a time to end up with final partition. We can also refer to the fact that each hyperplane has to divide at least one previous region into two, when that specific hyperplane is added.*

Then, from Lemma 1, (13) yields that

$$\sum_{j=1}^{2|\mathcal{T}|} \left(\sum_{i \in \mathcal{K}_j} \lambda_i \right)^{|\mathcal{T}|} \mathbf{e}_j = 0,$$

which happens only if

$$\sum_{i \in \mathcal{K}_j} \lambda_i = 0, \quad j = 1, \dots, |\mathcal{H}(\mathcal{T})|,$$

since standard basis vectors are linearly independent. From Lemma 2, it follows that we can find two numbers j_1 and j_2 for all $j = 1, \dots, |\mathcal{H}(\mathcal{T})|$ such that

$$\sum_{i \in \mathcal{K}_{j_1}} \lambda_i = \sum_{i \in \mathcal{K}_{j_2}} \lambda_i = 0,$$

where $j \in \mathcal{K}_{j_1}$, $|\mathcal{T}| + j \in \mathcal{K}_{j_2}$ and $\mathcal{K}_{j_1} \backslash \{j\} = \mathcal{K}_{j_2} \backslash \{|\mathcal{T}| + j\}$. Therefore, we conclude that $\lambda_j = \lambda_{|\mathcal{T}|+j}$ for all $j = 1, \dots, |\mathcal{H}(\mathcal{T})|$. Now, (13) implies $\sum_{i=1}^{|\mathcal{T}|} \lambda_{|\mathcal{T}|+i} = 0$, which confirms the claim that $\text{rank}(\mathbf{H}) = |\mathcal{T}| + 1$.

For the nonuniquness of the solution to the linear system of equations (1), we refer to the discussion below(Section C.3), where we argue that the solution is not unique even for some sparse cases, and complete the proof of Proposition 2.

C.3 Sparsity

We first recall that half of the rows among $2|\mathcal{T}|$ rows of $\mathbf{H}$ reflect the mass on the other side of each hyperplane. Basically, adding a row of all ones makes half of the rows redundant, since the rows representing the mass on the other side of each hyperplane are just flipped versions of rows representing the mass on the first side, i.e., $\mathbf{H}_{i+|\mathcal{T}|,:} = \mathbf{1}^T - \mathbf{H}_{i,:}$. We call $\mathbf{H}_{\text{half}}$ to the simplified version of $\mathbf{H}$. Then, we note that $\text{rank}(\mathbf{H}_{\text{half}}) = \text{rank}(\mathbf{H}) = |\mathcal{T}| + 1$ from Proposition 2. Therefore, we cannot make further simplifications on $\mathbf{H}$ to get redundant rows.

Now, we consider the simplified version $\mathbf{H}_{\text{half}}$ and recall that any solution $\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}$ to the problem setting in (1) has to be in the probability simplex. Therefore, all possible $\hat{\mathbf{q}}_{\text{half}}$ vectors belong to the convex hull of columns of matrix $\mathbf{H}_{\text{half}}$, which we call $\text{conv}(\mathbf{H}_{\text{half}})$. Then, we apply Carathéodory's Theorem and write following expression. *Each element in conv*$(\mathbf{H}_{half})$ *can be written as a convex combination of at most* $|\mathcal{T}| + 1$ *columns of* $\mathbf{H}_{half}$. We can easily observe that the same property also applies to $\text{conv}(\mathbf{H})$ and $\hat{\mathbf{q}}$, as they share a one-to-one correspondence with $\mathbf{H}_{\text{half}}$ and $\hat{\mathbf{q}}_{\text{half}}$, respectively.

C.4 Proof of Proposition 3

Given any two neighboring regions in the partition $\mathcal{H}(\mathcal{T})$, we suppose that $\mathbf{H}_{:,i_1}$ and $\mathbf{H}_{:,i_2}$ are corresponding columns to those regions, where only j-th hyperplane differ in between. We observe that it is possible to find another pair of columns, satisfying the same condition, separated solely by the j-th hyperplane as long as there exists another hyperplane that intersects with j-th hyperplane. Therefore, we can find linearly dependent 4 columns except the trivial case when all hyperplanes are parallel to each other. The problem setting boils down to the 1D setting, when all hyperplanes are parallel. The binary measurement matrix $\mathbf{H}$ becomes full rank and we can uniquely recover underlying distribution of regions in the partition $\mathcal{H}(\mathcal{T})$ separated by parallel hyperplanes.

Before going into the discussion of Theorem 5.1, we provide following remark based on the fact that $\mathbf{M}$ is a column-regular matrix, i.e. each column of $\mathbf{M}$ has exactly the same number of 1's.

Remark 1. *Robust Null Space Property (RNSP) has been proposed as a sufficient condition for basis pursuit approach (a popular recovery algorithm in compressed sensing literature) (Foucart and Rauhut, 2013; Foucart, 2014). Recently, Lotfi and Vidyasagar (2020) proposed sufficient conditions for a column-regular binary matrix to achieve RNSP, which are the best sufficient conditions for column-regular binary matrices to the best of our knowledge. According to Theorem 9 by Lotfi and Vidyasagar (2020), a column-regular binary matrix satisfies RNSP when* $k < d_L/\rho$, *where* d_L *is the number of 1's in each column and* ρ *is the maximum inner product among columns. Our binary matrix* $\mathbf{M}$ *is column-regular binary matrix with* $|\mathcal{T}|$ *1's in each column. Since there are neighboring regions, i.e., regions that has only one different coordinate, maximum inner product among columns is* $|\mathcal{T}| - 1$. *Therefore, RNSP is achieved when* $k = 1$.

Remark 1 shows that current results from compressed sensing literature for binary matrices can only guarantee unique solution for trivial case where the solution is assumed to be 1-sparse.

C.5 Discussion of Theorem 5.1

We first start by recalling some definitions from convex geometry that will be useful in the proof of Theorem 5.1.

Affinely Independent Set: A set of vectors $\{\mathbf{v}_1, \mathbf{v}_2 \dots \mathbf{v}_n\}$ is said to be affinely independent if $\{\mathbf{v}_2 - \mathbf{v}_1, \mathbf{v}_3 - \mathbf{v}_1, \dots \mathbf{v}_n - \mathbf{v}_1\}$ are linearly independent.

Simplex: Convex hull of affinely independent set of vectors is called simplex.

Face of a Convex Polytope: A face of a convex polytope is defined as the intersection of the convex set with its supporting hyperplanes. Each face is also a polytope and contains a subset of vertices of the convex polytope.

Simplicial face: If the set of vertices contained in a face of a convex set are affinely independent, this face is called a simplicial face.

Convex Polytope: There are 2 equivalent ways of describing a convex polytope $\mathcal{P}$ by the Farkas–Minkowski–Weyl Theorem. We can define $\mathcal{P}$ as the convex hull of a finite set of vertices, i.e., $\mathcal{P} = \text{conv}(\mathbf{v}_1, \mathbf{v}_2 \dots \mathbf{v}_n)$, which is known as *V-representation* of polytope $\mathcal{P}$. We can also define $\mathcal{P}$ as the intersection of a finite set of half spaces, i.e.,

$$\mathcal{P} = \{\mathbf{x} \in \mathbb{R}^d | \langle \mathbf{a}_i, \mathbf{x} \rangle \leq b_i, i = 1, \dots, f\},$$

which is also called *H-representation* of polytope $\mathcal{P}$. Here, f is the number of facets, i.e., $d-1$ dimensional faces and $\mathbf{a}_i$'s are corresponding face normals. Each $\langle \mathbf{a}_i, \mathbf{x} \rangle \leq b_i$ defines a supporting hyperplane with the boundary condition $\langle \mathbf{a}_i, \mathbf{x} \rangle = b_i$. Recall that *faces* of the polytope $\mathcal{P}$ are defined as the intersections of supporting hyperplanes with $\mathcal{P}$. For any point $\mathbf{x}$ belong to a face of convex polytope $\mathcal{P}$, at least one of the f inequalities has to satisfy as an equality. We recall the fact that a simplicial face is a face formed by affinely independent set of vertices and provide the proof of Theorem 5.1.

Proof of Theorem 5.1: We suppose that $\mathbf{q}^\star$ belongs to interior of a simplicial face $\mathcal{F}_\mathcal{S}$ of the polytope $\text{conv}(\mathbf{H})$. Then the uniquness of the solution $\mathbf{p}^\star$ follows from Proposition 4.6 by Kueng and Tropp (2021), which is stated below.

Proposition 5. *(Proposition 4.6 by Kueng and Tropp (2021)): Given a compact convex set $\mathcal{K}$ and a point $\mathbf{x} \in \mathcal{K}$, $\mathbf{x}$ has a unique proper convex combination of extreme points of $\mathcal{K}$ with participating extreme points if and only if $\mathbf{x}$ belongs to a relative interior of a simplicial face of $\mathcal{K}$, where proper convex combination refers to the convex combination with nonzero convex coefficients.*

For $\mathcal{K} = \text{conv}(\mathbf{H})$, we can conclude that $\mathbf{q}^\star$ can be written as a unique convex combination of a subset of extreme points of $\text{conv}(\mathbf{H})$, i.e., there exists a unique probability vector $\mathbf{p}^\star$ such that $\mathbf{H}\mathbf{p}^\star = \mathbf{q}^\star$.

We consider another example from the partition in the Figure 2(a) with 3-sparse cases. First, we suppose that $\mathbf{q} = [0.1,\ 0.7,\ 0.2,\ 0.9,\ 0.3,\ 0.8]$. $\mathbf{q} = \mathbf{H}\mathbf{p}$ holds for both $\mathbf{p} = [0.7,\ 0,\ 0.2,\ 0,\ 0,\ 0,\ 0.1]$ and $\mathbf{p} = [0.5,\ 0.2,\ 0,\ 0,\ 0,\ 0,\ 0.3]$. Here, we note that $\text{conv}(\mathbf{H}_{:,1}, \mathbf{H}_{:,3}, \mathbf{H}_{:,7})$ and $\text{conv}(\mathbf{H}_{:,1}, \mathbf{H}_{:,2}, \mathbf{H}_{:,7})$ are not even faces of $\text{conv}(\mathbf{H})$. They both belong to the face intersecting with the hyperplane defined as $[\mathbf{e}_3,\ \mathbf{e}_6]^T\mathbf{p} = [1,\ 0]^T$. On the other hand, we can easily observe that $\text{conv}(\mathbf{H}_{:,1}, \mathbf{H}_{:,2}, \mathbf{H}_{:,6})$ is a simplicial face of $\text{conv}(\mathbf{H})$, since they are affinely independent and $\text{conv}(\mathbf{H}_{:,1}, \mathbf{H}_{:,2}, \mathbf{H}_{:,6})$ is a face of $\text{conv}(\mathbf{H})$ formed by the intersection of the boundary of half space corresponding to $[\mathbf{e}_1,\ \mathbf{e}_4]^T\mathbf{x} \leq [1,\ 0]^T$. Then, any $\mathbf{q}$ that is formed by a convex combination of $\mathbf{H}_{:,1}, \mathbf{H}_{:,2}$ and $\mathbf{H}_{:,6}$ will have a unique solution $\mathbf{p}^\star = [\mathbf{q}_2 = \mathbf{q}_3,\ \mathbf{q}_3,\ 0,\ 0,\ 0,\ 1 - \mathbf{q}_2,\ 0]^T$, which confirms that $\text{conv}(\mathbf{H}_{:,1}, \mathbf{H}_{:,2}, \mathbf{H}_{:,6})$ is a simplicial face of $\text{conv}(\mathbf{H})$.

C.6 Proof of Proposition 4

Recall that $\mathbf{H}_{i,:}$ and (a_i, b_i) are the row and the item pair corresponding to i-th hyperplane. $\mathbf{q}^\star_{a_i,b_i}$ denotes the true mass on the side of a_i of the i-th hyperplane and $\mathcal{K}_j$ is the position of rows of $\mathbf{H}$ whose j-th column entry is 1. $\mathbf{q}^\star_{a_i,b_i}$ has $\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})_j}$, j-th entry of $\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}$, as a nonnegative summand when $i \in \mathcal{K}_j$. Therefore, we can write

following:

$$\mathbf{p}^{\star}_{\mathcal{H}(\mathcal{T})_j} \leq \min_{i \in \mathcal{K}_j} \mathbf{q}^{\star}_{a_i b_i}, \quad j = 1, \ldots, |\mathcal{H}(\mathcal{T})|. \tag{14}$$

Using those upper bounds and nonnegativity of entries of matrix $\mathbf{H}$, we can write following set of inequalities:

$$\mathbf{H} \underbrace{\begin{bmatrix} \min_{i \in \mathcal{K}_1} \mathbf{q}^{\star}_{a_i b_i} \\ \vdots \\ \min_{i \in \mathcal{K}_{j-1}} \mathbf{q}^{\star}_{a_i b_i} \\ \mathbf{p}^{\star}_{\mathcal{H}(\mathcal{T})_j} \\ \min_{i \in \mathcal{K}_{j+1}} \mathbf{q}^{\star}_{a_i b_i} \\ \vdots \\ \min_{i \in \mathcal{K}_l} \mathbf{q}^{\star}_{a_i b_i} \end{bmatrix}}_{\mathbf{Q}^j} \geq \mathbf{q}^{\star}, \quad j = 1, \ldots, |\mathcal{H}(\mathcal{T})|, \tag{15}$$

which enables us to lower bound each entry $\mathbf{p}^{\star}_{\mathcal{H}(\mathcal{T})_j}$ for $j = 1, \ldots, |\mathcal{H}(\mathcal{T})|$. Here, $\mathbf{Q}^j$ represents the vector constructed with minimum $\mathbf{q}^{\star}_{a_i b_i}$'s over different sets and $\mathbf{p}^{\star}_{\mathcal{H}(\mathcal{T})_j}$. We also define $\mathbf{Q}^j_0$ as the vector that jth entry of $\mathbf{Q}^j$ is replaced with 0. Note that each inequality in (15) can be rewritten as follows

$$\mathbf{H}^T_{k,:} \mathbf{Q}^j \geq \mathbf{q}^{\star}_{a_k b_k}, \quad k = 1, \ldots, |\mathcal{T}|.$$

We can also write an alternative expression by using standard basis vectors, i.e., $\mathbf{e}_j$'s,:

$$\mathbf{p}^{\star}_{\mathcal{H}(\mathcal{T})_j} \mathbf{H}^T_{k,:} \mathbf{e}_j \geq \mathbf{q}^{\star}_{a_k b_k} - \mathbf{H}^T_{k,:} \mathbf{Q}^j_0, \quad k = 1, \ldots, |\mathcal{T}|,$$

which provides us following bound

$$\mathbf{p}^{\star}_{\mathcal{H}(\mathcal{T})_j} \geq \max\{\max_{i \in \mathcal{K}_j} \mathbf{q}^{\star}_{a_i b_i} - \mathbf{H}^T_{i,:} \mathbf{Q}^j_0, 0\}, \quad j = 1, \ldots, |\mathcal{H}(\mathcal{T})|. \tag{16}$$

Combining (14) and (16), we obtain following expression

$$\max_{i \in \mathcal{K}_j} \mathbf{q}^{\star}_{a_i b_i} - \mathbf{H}^T_{i,:} \mathbf{Q}^j_0 \leq \mathbf{p}^{\star}_{\mathcal{H}(\mathcal{T})_j} \leq \min_{i \in \mathcal{K}_j} \mathbf{q}^{\star}_{a_i b_i}. \tag{17}$$

Below, we expand on estimation errors to replace $\mathbf{q}^{\star}_{a_i b_i}$'s with corresponding estimates. For any $\mathbf{q}^{\star}_{a_i b_i}$, we can say that

$$|\hat{\mathbf{q}}_{a_i b_i} - \mathbf{q}^{\star}_{a_i b_i}| \leq \sqrt{\frac{\log(2/\delta')}{2n_p}} \tag{18}$$

holds with probability at least $1 - \delta'$ by Hoeffding's Inequality (see 11). Therefore, we want to bound the probability that

$$|\hat{\mathbf{q}}_{a_i b_i} - \mathbf{q}^{\star}_{a_i b_i}| \geq \sqrt{\frac{\log(2/\delta')}{2n_p}}$$

holds at least for one i, where n_p is the number of people answering each pairwise query. Therefore, we write following expression

$$\Pr\left(\bigcup_i \left\{ |\hat{\mathbf{q}}_{a_i b_i} - \mathbf{q}^{\star}_{a_i b_i}| \geq \sqrt{\frac{\log(2/\delta')}{2n_i}} \right\}\right) \leq \sum_i \Pr\left(\left\{ |\hat{\mathbf{q}}_{a_i b_i} - \mathbf{q}^{\star}_{a_i b_i}| \geq \sqrt{\frac{\log(2/\delta')}{2n_i}} \right\}\right) \tag{19}$$

$$\leq 2|\mathcal{T}|\delta', \tag{20}$$

where (19) is from union bound and (20) is due to (18). Picking $\delta = 2|\mathcal{T}|\delta'$, we conclude that

$$|\hat{\mathbf{q}} - \mathbf{q}^\star|_1 = \sum_i |\hat{\mathbf{q}}_{a_i b_i} - \mathbf{q}^\star_{a_i b_i}| \leq \sqrt{\frac{\log\left(4|\mathcal{T}|/\delta\right)}{2n_p}} \tag{21}$$

holds with probability at least $1 - \delta$. Inserting it to the result in 17, we complete the proof of Proposition 4.

Additionally, below, we provide a simple example with two hyperplanes to highlight the tightness of the bounds on mass in each region given in Proposition 4.

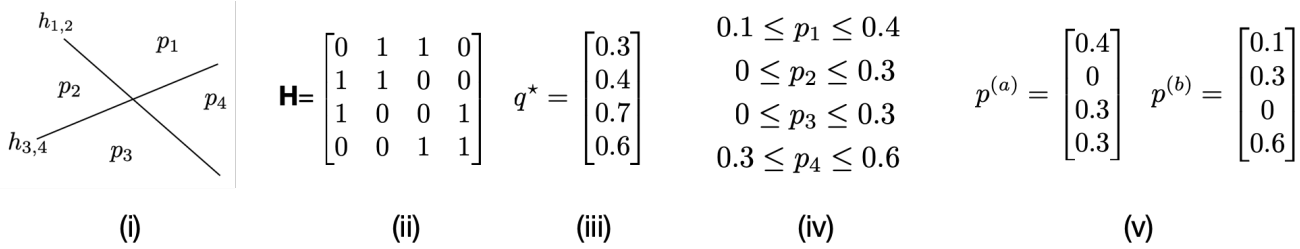

Figure 10: Example for bounds on mass in each region in 2D with 2 hyperplanes with two solutions that give rise to same true $\mathbf{q}^\star$ highlighting the tightness of the bounds on mass in each region provided in Proposition 4.

For the given value of $\mathbf{q}^\star$ in Figure 10, Proposition 4 provides following bounds

$$\max\left\{0, \max_{i \in \mathcal{K}_j} \mathbf{q}^\star_{a_i b_i} - \mathbf{H}^T_{i,:}\mathbf{Q}_0^{\star j}\right\} \leq \mathbf{p}^\star_{\mathcal{H}(\mathcal{T})_j} \leq \min_{i \in \mathcal{K}_j} \mathbf{q}^\star_{a_i b_i}. \tag{22}$$

Recall that $\mathcal{K}_j$ represents the set of all rows of H where the j-th column entry is 1. For this example,$\mathcal{K}_1 = \{2, 3\}$, $\mathcal{K}_2 = \{1, 2\}$, $\mathcal{K}_3 = \{1, 4\}$ and $\mathcal{K}_4 = \{3, 4\}$. Applying the upper and lower bounds provided in the above equation, we obtain the bounds shown Figure 10(iv). Furthermore, we note the $\mathbf{p}^{(a)}$ and $\mathbf{p}^{(b)}$ in Figure 10(v), are both solutions to the set of linear equation $\mathbf{q}^\star = \mathbf{H}\mathbf{p}$ in this example and these two reach the bounds obtained in Figure 10(iv) illustrating the general tightnes of these bounds.

C.7 Graph Regularization

In this section, we discuss about the graph regularization that we proposed using in Section 5. We provide a standard graph regularizer without using volume weighting here to give a better intuition about graph regularizers and why we used volume weighting in Section 5. We start by defining following weight matrix $\mathbf{W}^{\text{unif}}$:

$$\mathbf{W}^{\text{unif}}_{i,j} = \|\mathbf{H}_{:,i} - \mathbf{H}_{:,j}\|_1^{-1}, \tag{23}$$

which is the inverse of the Hamming distance between nodes i and j. Accordingly, we can write following graph Laplacian regularizer:

$$\begin{aligned} R &= \frac{1}{2}\sum_{i=1}^n \sum_{j=1}^n |\mathbf{p}_i - \mathbf{p}_j|^2 \mathbf{W}^{\text{unif}}_{i,j} \\ &= \sum_{i=1}^n \mathbf{p}_i \mathbf{p}_i \mathbf{D}^{\text{unif}}_{i,i} - \sum_{i=1}^n \sum_{j=1}^n \mathbf{p}_i \mathbf{p}_j \mathbf{W}^{\text{unif}}_{i,j} = \mathbf{p}^T \mathbf{D}^{\text{unif}} \mathbf{p} - \mathbf{p}^T \mathbf{W}^{\text{unif}} \mathbf{p} = \mathbf{p}^T \mathbf{L}^{\text{unif}} \mathbf{p}, \end{aligned}$$

where $\mathbf{D}^{\text{unif}}_{i,i} = \sum_{j=1}^n \mathbf{W}^{\text{unif}}_{i,j}$, $\mathbf{D}^{\text{unif}}_{i,j} = 0$ when $i \neq j$ and $\mathbf{L}^{\text{unif}} = \mathbf{D}^{\text{unif}} - \mathbf{W}^{\text{unif}}$. Now, suppose that the spectral decomposition of $\mathbf{L}^{\text{unif}}$ can be written as $\mathbf{L}^{\text{unif}} = \sum_{i=1}^l \mu_i \mathbf{v}_i \mathbf{v}_i^T$, where $\mathbf{v}_i$'s are eigenvectors and μ_i's are the corresponding eigenvalues. We now further elaborate on spectral properties of Laplacian matrices and use following Lemma.

Lemma 3. *Graph Laplacian matrices are positive semi-definite by the Gershgorin circle theorem. Furthermore, the eigenvectors of the Laplacian matrix $\mathbf{L}^{unif}$ corresponding to zero eigenvalues are spanned by $\mathbf{1}$, which is referred as constant vectors by Poignard et al. (2018).*

Then, we can rewrite Laplacian regularizer in (4) as

$$\mathbf{p}^T\mathbf{A}^T\mathbf{L}^{\text{unif}}\mathbf{A}\mathbf{p} = \mathbf{p}^T\sum_{i=1}^{l}\mu_i\mathbf{A}^T\mathbf{v}_i\mathbf{v}_i^T\mathbf{A}\mathbf{p} = \sum_{i=1}^{l}\mu_i(\mathbf{p}^T(\mathbf{A}^T\mathbf{v}_i))^2,$$

where $\mathbf{A}$ is a diagonal matrix with the entries $\mathbf{A}_{i,i} = \frac{1}{\alpha_i}$ and $\sum_i \mathbf{A}_{i,i} = 1$. Laplacian regularizer $\mathbf{L} = \mathbf{A}^T\mathbf{L}^{\text{unif}}\mathbf{A}$ penalizes $\mathbf{p}$ so that potential $\mathbf{p}$ values correlated to vectors $\mathbf{A}^T\mathbf{v}_i$'s are diminished. We can rephrase it as follows: regularizer penalizes $\mathbf{p}$ so that potential $\mathbf{A}^{-1}\mathbf{p}$ values correlated to eigenvectors $\mathbf{v}_i$'s are diminished. Therefore, $\mathbf{v}_i$'s corresponding to larger eigenvalues cause more penalty. From Lemma 3, it follows that Laplacian matrix $\mathbf{L}$ corresponding to zero eigenvalues are spanned by $\mathbf{A}^{-1}\mathbf{1}$. Poignard et al. (2018) also point out that the multiplicity of the eigenvalue is equal to the number of connected components in the graph, which is clearly 1 in our graph structure induced by $\mathbf{H}$, since the regions in $\mathcal{H}(\mathcal{T})$ are connected. We note that the eigenvectors of $\mathbf{L}^{\text{unif}}$ are mutually orthogonal by spectral theory. We observe that orthogonal eigenvectors of nonzero eigenvalues would force the candidate of the solution $\mathbf{p}$ to be similar to uniform distribution by punishing possible directions other than $\mathbf{1}$. However, we note that regions in $\mathcal{H}(\mathcal{S}_m)$ are not similar to an equally spaced grid. Therefore, we use a weighted version of the regularizer in (4) with respect to the volumes of the regions in $\mathcal{H}(\mathcal{T})$ instead of $\mathbf{L}^{\text{unif}}$ and punish possible directions other than $\mathbf{A}^{-1}\mathbf{1}$, i.e. $\overline{\alpha}$.

C.8 Proof of Theorem 5.2

We first show that the solution to the convex optimization problem in (5) is unique. Let $f(\mathbf{p})$ be the objective function $\frac{1}{2}||\mathbf{H}\mathbf{p} - \hat{\mathbf{q}}||_2^2 + \frac{\lambda}{2}\mathbf{p}^T\mathbf{L}\mathbf{p}$. If we can guarantee that

$$\frac{\partial^2 f}{\partial \mathbf{p}^2} = 2\mathbf{H}^T\mathbf{H} + 2\lambda\mathbf{L} \succ 0, \tag{24}$$

we deduce that solution to the convex optimization problem in (5) is unique. Therefore, we first focus on matrix $\mathbf{L}$. From Lemma 3, null space of $\mathbf{L}^{\text{unif}}$ is spanned by $\mathbf{1}$. Since $\mathbf{A}$ is a full rank matrix, null space of $\mathbf{L} = \mathbf{A}^T\mathbf{L}^{\text{unif}}\mathbf{A}$ is spanned by $\mathbf{A}^{-1}\mathbf{1}$. All entries of $\mathbf{A}^{-1}\mathbf{1}$ are nonnegative since $\mathbf{A}^{-1}$ is a diagonal matrix with nonnegative entries. Now, we have following

$$\begin{aligned}\mathbf{H}^T\mathbf{H} &\succeq 0,\\ \mathbf{L} &\succeq 0,\\ \mathbf{H}^T\mathbf{H} + \lambda\mathbf{L} &\succeq 0.\end{aligned}$$

If $ker(\mathbf{H}^T\mathbf{H}) \neq ker(\mathbf{L})$, we can guarantee that $\mathbf{H}^T\mathbf{H} + \lambda\mathbf{L} \succ 0$. $\mathbf{H}^T\mathbf{H}$ is already positive semidefinite and $\mathbf{A}^{-1}\mathbf{1}$ cannot be an eigenvector for $\mathbf{H}^T\mathbf{H}$, since all nonzero entries of $\mathbf{H}^T\mathbf{H}$ have same sign. Therefore, $\mathbf{H}^T\mathbf{H} + \lambda\mathbf{L}$ is always positive definite.

Now, we recall that $\mathbf{R}^T\mathbf{R} = \mathbf{H}^T\mathbf{H} + \lambda\mathbf{L}$ and note that multiplication of each element in the unit simplex with matrix $\mathbf{R}$ defines following closed convex set,

$$C_{\mathbf{R}} := \text{conv}(\mathbf{R}\mathbf{e}_1, \mathbf{R}\mathbf{e}_2 \ldots, \mathbf{R}\mathbf{e}_{|\mathcal{H}(\mathcal{T})|}).$$

Then, the unique solution $\hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})}$ to the optimization setting in (5) can be expressed as

$$\hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})} = \mathbf{R}^{-1}\text{P}_{C_{\mathbf{R}}}(\mathbf{b}), \tag{25}$$

where $\mathbf{b} = \mathbf{R}^{-T}\mathbf{H}^T\mathbf{H}\mathbf{p}^*$. Therefore,

$$\mathbf{R}\hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})} = \text{P}_{C_{\mathbf{R}}}(\mathbf{R}^{-T}\mathbf{H}^T\hat{\mathbf{q}}).$$

We start with bounding ℓ_2 norm error and write

$$\begin{aligned}
\|\hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})} - \mathbf{p}^*_{\mathcal{H}(\mathcal{T})}\|_2 &\leq \|\mathbf{R}^{-1}\|_2 \|\mathbf{R}\hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})} - \mathbf{R}\mathbf{p}^*_{\mathcal{H}(\mathcal{T})}\|_2 \\
&= \|\mathbf{R}^{-1}\|_2 \|\mathbf{R}\mathbf{p}^*_{\mathcal{H}(\mathcal{T})} - \mathrm{P}_{C_{\mathbf{R}}}(\mathbf{R}^{-T}\mathbf{H}^T\hat{\mathbf{q}})\|_2 \\
&\leq \|\mathbf{R}^{-1}\|_2 \|\mathbf{R}\mathbf{p}^*_{\mathcal{H}(\mathcal{T})} - \mathbf{R}^{-T}\mathbf{H}^T\hat{\mathbf{q}}\|_2 \qquad (26) \\
&\leq \|\mathbf{R}^{-1}\|_2 (\|\mathbf{R}\mathbf{p}^*_{\mathcal{H}(\mathcal{T})} - \mathbf{R}^{-T}\mathbf{H}^T\mathbf{H}\mathbf{p}^*_{\mathcal{H}(\mathcal{T})} + \mathbf{R}^{-T}\mathbf{H}^T\mathbf{H}\mathbf{p}^*_{\mathcal{H}(\mathcal{T})} - \mathbf{R}^{-T}\mathbf{H}^T\hat{\mathbf{q}}\|_2) \\
&\leq \|\mathbf{R}^{-1}\|_2 (\|\mathbf{R}\mathbf{p}^*_{\mathcal{H}(\mathcal{T})} - \mathbf{R}^{-T}\mathbf{H}^T\mathbf{H}\mathbf{p}^*_{\mathcal{H}(\mathcal{T})}\|_2 + \|\mathbf{R}^{-T}\mathbf{H}^T(\mathbf{H}\mathbf{p}^*_{\mathcal{H}(\mathcal{T})} - \hat{\mathbf{q}})\|_2) \\
&\leq \|\mathbf{R}^{-1}\|_2 (\|\lambda\mathbf{R}^{-T}\mathbf{L}\mathbf{p}^*_{\mathcal{H}(\mathcal{T})}\|_2 + \|\mathbf{R}^{-T}\mathbf{H}^T(\mathbf{H}\mathbf{p}^*_{\mathcal{H}(\mathcal{T})} - \hat{\mathbf{q}})\|_2) \qquad (27) \\
&\leq \|\mathbf{R}^{-1}\|_2^2 (\lambda\|\mathbf{L}(\mathbf{p}^*_{\mathcal{H}(\mathcal{T})} - \overline{\alpha})\|_2 + \|\mathbf{H}^T\|_2\|\mathbf{q}^\star - \hat{\mathbf{q}}\|_2) \qquad (28) \\
&\leq \|\mathbf{R}^{-1}\|_2^2 (\lambda\|\mathbf{L}\|_2 \left\|\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})} - \overline{\alpha}\right\|_2 + \|\mathbf{H}^T\|_2\|\mathbf{q}^\star - \hat{\mathbf{q}}\|_2), \qquad (29)
\end{aligned}$$

where (26) is due to contracting property of projection operator onto closed convex sets, (27) is because $\mathbf{H}^T\mathbf{H} = \mathbf{R}^T\mathbf{R} - \lambda\mathbf{L}$, and (28) follows from $\mathbf{H}\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})} = \mathbf{q}^\star$ and $\mathbf{L}\overline{\alpha} = 0$. Then, by using $\ell_1 - \ell_2$ norm inequality, we can simply write following inequalities

$$\begin{aligned}
\mathrm{TV}(\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}, \|\hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})}) &= \frac{1}{2}||\mathbf{p}^*_{\mathcal{H}(\mathcal{T})} - \|\hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})}||_1 \\
&\leq \frac{\sqrt{|\mathcal{H}(\mathcal{T})|}}{2}||\mathbf{p}^*_{\mathcal{H}(\mathcal{T})} - \|\hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})}||_2 \\
&\leq \frac{\sqrt{|\mathcal{H}(\mathcal{T})|}}{2}\|\mathbf{R}^{-1}\|_2^2(\lambda\|\mathbf{L}\|_2 \left\|\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})} - \overline{\alpha}\right\|_2 + \|\mathbf{H}^T\|_2\|\mathbf{q}^\star - \hat{\mathbf{q}}\|_2) \\
&\leq \frac{\lambda}{2}\sqrt{|\mathcal{H}(\mathcal{T})|}\|\mathbf{R}^{-1}\|_2^2\|\mathbf{L}\|_2 \left\|\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})} - \overline{\alpha}\right\|_2 + \frac{\sqrt{|\mathcal{H}(\mathcal{T})|}}{2}\|\mathbf{R}^{-1}\|_2^2\|\mathbf{H}^T\|_2\|\mathbf{q}^\star - \hat{\mathbf{q}}\|_2 \\
&\leq \frac{\lambda}{2}\sqrt{|\mathcal{H}(\mathcal{T})|}\|\mathbf{R}^{-1}\|_2^2\|\mathbf{L}\|_2 \left\|\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})} - \overline{\alpha}\right\|_2 + \frac{\sqrt{|\mathcal{H}(\mathcal{T})|}}{2}\|\mathbf{R}^{-1}\|_2^2\|\mathbf{H}^T\|_2\|\mathbf{q}^\star - \hat{\mathbf{q}}\|_1 \\
&\leq \frac{\lambda}{2}\sqrt{|\mathcal{H}(\mathcal{T})|}\|\mathbf{R}^{-1}\|_2^2\|\mathbf{L}\|_2 \left\|\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})} - \overline{\alpha}\right\|_2 + \frac{1}{\sqrt{2}}|\mathcal{T}||\mathcal{H}(\mathcal{T})|\|\mathbf{R}^{-1}\|_2^2\|\mathbf{q}^\star - \hat{\mathbf{q}}\|_1.
\end{aligned}$$

Lastly, we use (21) to bound $\|\mathbf{q}^\star - \hat{\mathbf{q}}\|_1$ and complete the proof.

D Additional Simulations and Experimental Details

D.1 Simulations for $d = 1$

We provide simulation results for following group of user distributions: uniform, Gaussian, a mixture of 2 Gaussians, and a mixture of 3 Gaussians. We also present simulation results for varying amount of noises in 2 different noise models that we defined in Section 6.

Figure 11-14 show the relationship between the number of hyperplanes, n_h, and the error in recovered mass in the partition $\mathcal{H}(\mathcal{T})$ while varying $n_h \in \{1, \ldots, 10\}$, as well as the relationship between the number of people asked per query, n_p, and the error, while varying $n_p \in \{10^2, 10^3, 10^4, 10^5\}$, under the four user distributions and different noise levels. Our analysis demonstrates that the recovery gets better as the number of users increases. It is important to note that as the number of pairs, (equivalently, the number of hyperplanes) increases, the size of $\mathbf{p}$ also increases, which leads to an expected increase in the total variation (TV).

D.2 Construction of H in dimensions $d \geq 2$

Unlike 1D setting, the algorithmic construction of binary matrix $\mathbf{H}$ is not straightforward in dimensions $d \geq 2$. We need to figure out which polytopes, i.e., regions, are on the left side of a given hyperplane. We recall that these polytopes are defined by the halfspaces induced by the bisecting hyperplanes of item pairs in $\mathcal{T}$. Hence, our problem can be formally described as follows: Given a set of halfspaces $\mathbb{H}_s = \{\boldsymbol{a}_{ij}^\top\boldsymbol{x} + b_{ij} < 0 : h_{ij} = \boldsymbol{a}_{ij}^\top\boldsymbol{x} + b_{ij} = 0, i < j\}$, where h_{ij} is the bisecting hyperplanes of pair $(\boldsymbol{x}_i, \boldsymbol{x}_j) \in \mathcal{T}$, we want to find all polytopes in $\mathcal{H}(\mathcal{T})$ that are in the halfspace h_s, for each $h_s \in \mathbb{H}_s$. To simplify our analysis, we define a bounding box $[-1, 1]^d$, so that we can only look at the polytopes within this box and avoid unbounded polytopes. For simplicity, we use the vector $[\boldsymbol{a}_{ij} \quad b_{ij}]$ to represent a halfspace.

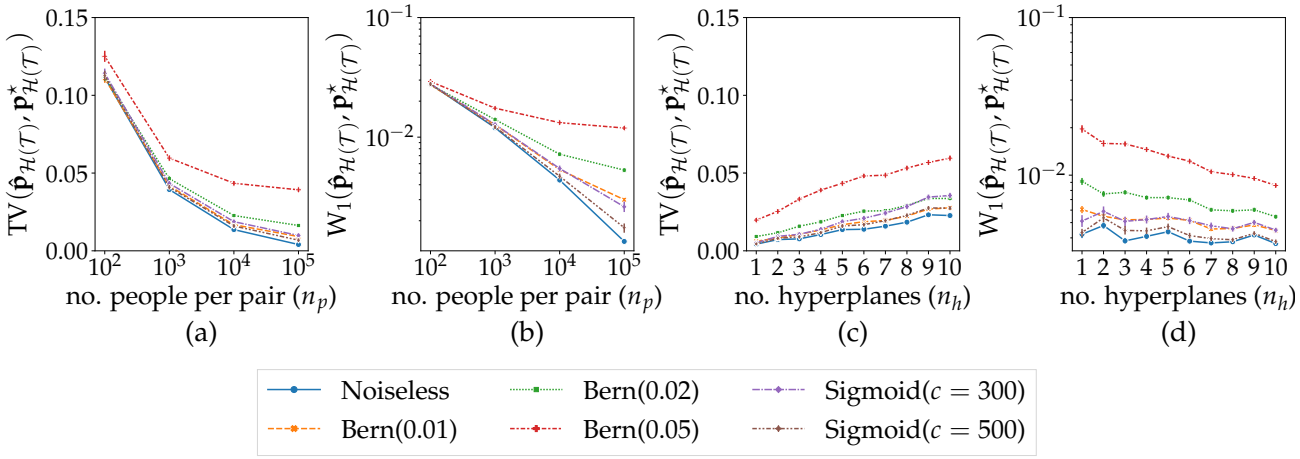

Figure 11: $\mathrm{TV}(\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})})$ and $\mathrm{W}_1(\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})})$ for uniform user distribution in 1D.

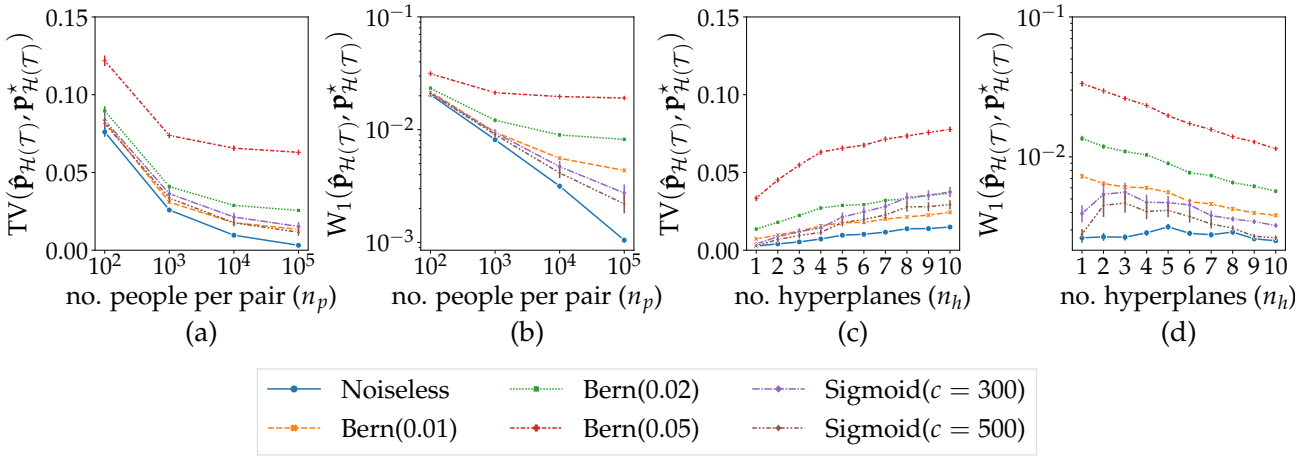

Figure 12: $\mathrm{TV}(\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})})$ and $\mathrm{W}_1(\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})})$ for Gaussian user distribution in 1D.

Let $\mathbb{B}_s$ denote the set of halfspaces that defines the bounding box $[-1, 1]^d$. Let $\mathbb{P}_t$ denote the set of polytopes that we have discovered. Let $\mathbb{H}_s^u$ denote the set of halfspaces we have not explored yet. Our algorithm works as follows:

$\mathbb{P}_t \leftarrow \{\mathbb{B}_s\}$
for $h_s \in \mathbb{H}_s \setminus \mathbb{H}_s^o$ **do**
  for $p_t \in \mathbb{P}_t$ **do**
    if h_s intersects with p_t **then**
      $p_t^l \leftarrow p_t \cup \{h_s\}$
      $p_t^r \leftarrow p_t \cup \{-h_s\}$
      $\mathbb{P}_t \leftarrow \mathbb{P}_t \setminus p_t$
      $\mathbb{P}_t \leftarrow \mathbb{P}_t \cup \{p_t^l, p_t^r\}$
    end if
  end for
end for

To check if h_s intersects with p_t, we first assume that h_s splits p_t into two polytopes, namely, $p_t^l := p_t \cup \{h_s\}$ and $p_t^r := p_t \cup \{-h_s\}$. If p_t^l or p_t^r is degenerate, the assumption does not hold. Therefore h_s does not intersect with p_t. To verify whether p_t^l or p_t^r is degenerate, we check whether they have a Chebyshev center, which can be found by

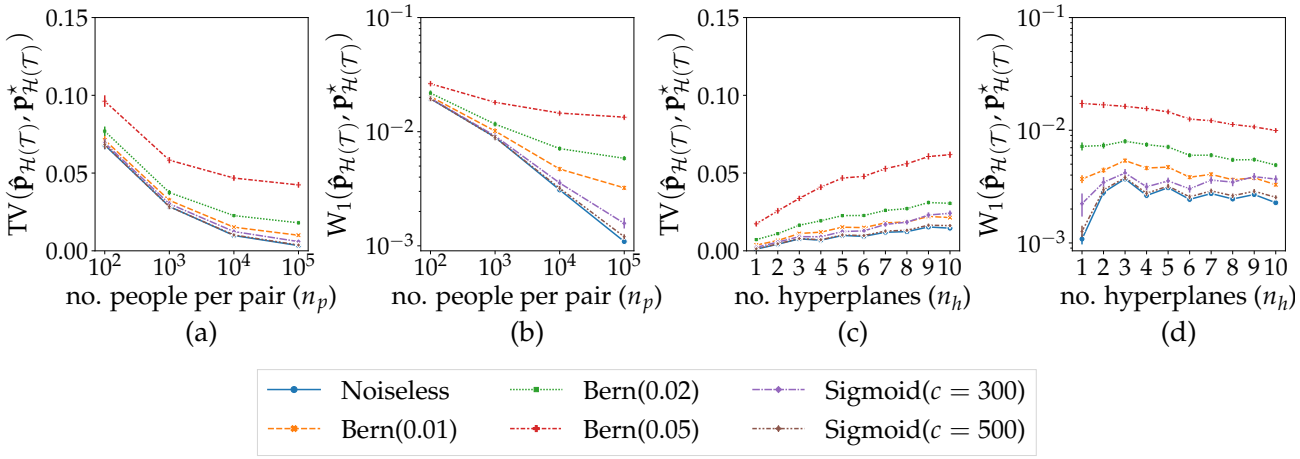

Figure 13: $\mathrm{TV}(\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})})$ and $\mathrm{W}_1(\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})})$ for a mixture of 2 Gaussians user distribution in 1D.

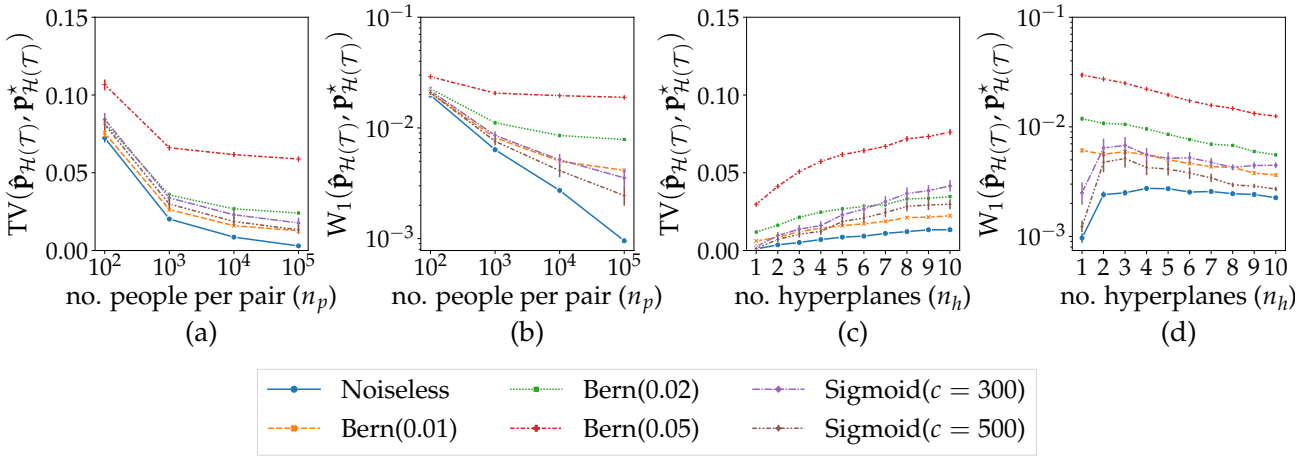

Figure 14: $\mathrm{TV}(\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})})$ and $\mathrm{W}_1(\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})})$ for a mixture of 3 Gaussians user distribution in 1D.

solving the following linear program twice:

$$\begin{aligned} \max_{\boldsymbol{y}, r} \quad & r \\ \text{subject to} \quad & \boldsymbol{a}_i^T \boldsymbol{y} + ||a_i|| r \leq b_i, \quad \forall i \in [|p_t| + 1] \end{aligned}$$

where $[\boldsymbol{a}_i \quad b_i]$ is the i^{th} halfspace in p_t^l(or p_t^r) and $\boldsymbol{y}$ is the Chebyshev center (when solved). If the two linear programs have (bounded) solutions and $\boldsymbol{y}$ is in p_t, we can say that p_t^l and p_t^r have Chebyshev centers. Consequently, h_s intersects with p_t. Otherwise, we can conclude that h_s does not intersect with p_t. To determine the position of any polytope with respect to hyperlanes (halfspaces), we check whether the Chebyshev center of that polytope is on the left or right of the hyperplane (is in the halfspace).

D.3 Simulations for $d \geq 2$

We first present the results with varying regularization parameter λ. Figure 15 and 16 present the behavior of TV and $W_{\mathcal{G}}$ under the four user distributions while we vary λ when $n_p = 10,000$, $m = 5$, $n_h = 10$, and $d = 2$. No noise model is introduced in this set of simulation. Four different colored lines in Figures refer to the four different objectives we used. Least Square + Graph means that the objective is least square with graph regularization; Least Square + L1 means that the objective is least square with ℓ_1 regularization, Least Square + L2 means that the objective is least square with ℓ_2 regularization, and KL means that the objective is the KL divergence of $\hat{\mathbf{q}}$ from $\mathbf{Hp}$, $D_{\mathrm{KL}}(\hat{\mathbf{q}}, \mathbf{Hp})$, where the solution is maximum likelihood estimate.

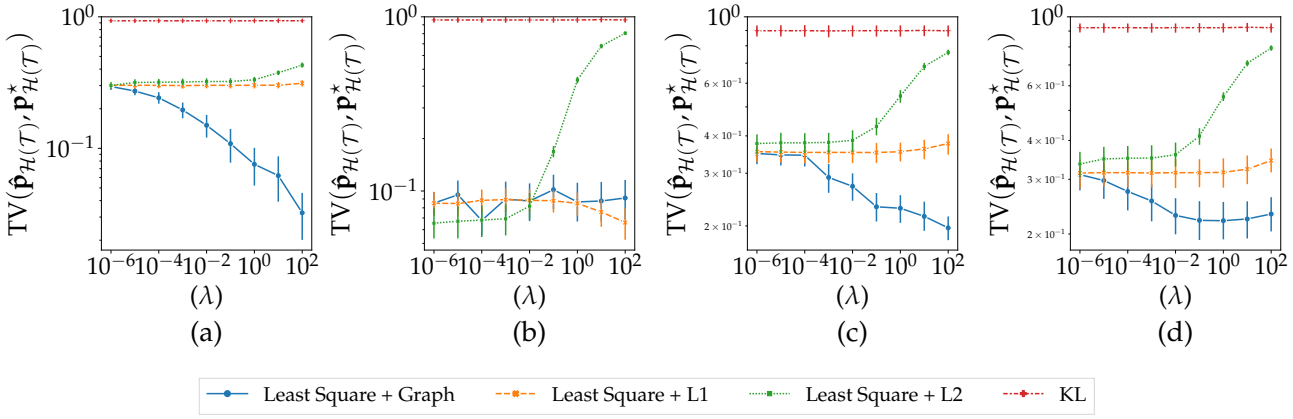

Figure 15: $\mathrm{TV}(\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})})$ for (a) uniform (b) Gaussian (c) mixture of 2 Gaussians (d) mixture of 3 Gaussians user distribution while varying the regularization parameter λ.

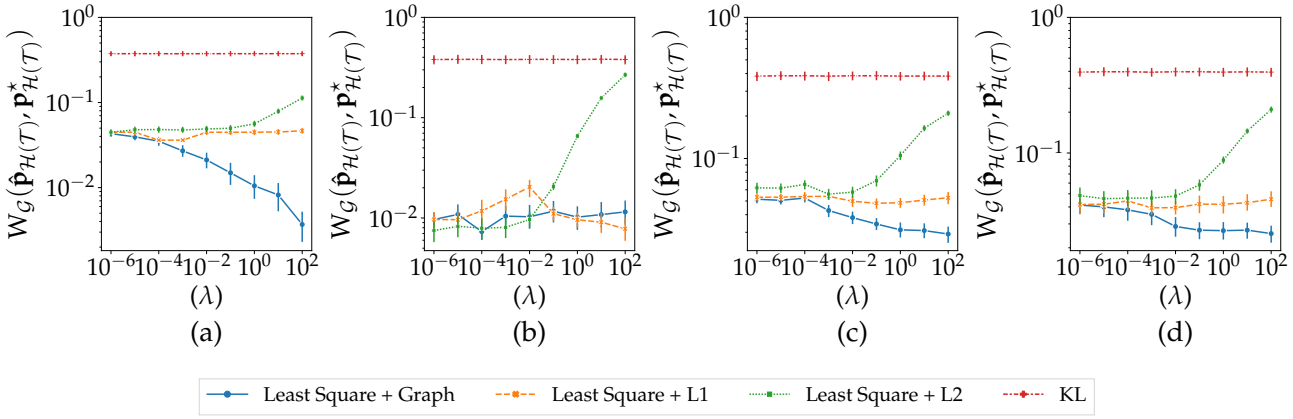

Figure 16: $\mathrm{W}_{\mathcal{G}}(\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})})$ for (a) uniform (b) Gaussian (c) mixture of 2 Gaussians (d) mixture of 3 Gaussians user distribution while varying the regularization parameter λ.

We also present the behavior of TV and $W_{\mathcal{G}}$ when we use a different formulation of the optimization problem, where the regularization term in the original optimization is the sole objective, and $||\mathbf{Hp} - \hat{\mathbf{q}}||_2^2 \leq \varepsilon$ ($D_{\mathrm{KL}}(\hat{\mathbf{q}}, \mathbf{Hp}) \leq \varepsilon$) is an additional constraint. We set $\varepsilon = 10^{-5}$ in simulations. Figure 17 and 18 present the results with the new formulation of the optimization problem under the same setting as above.

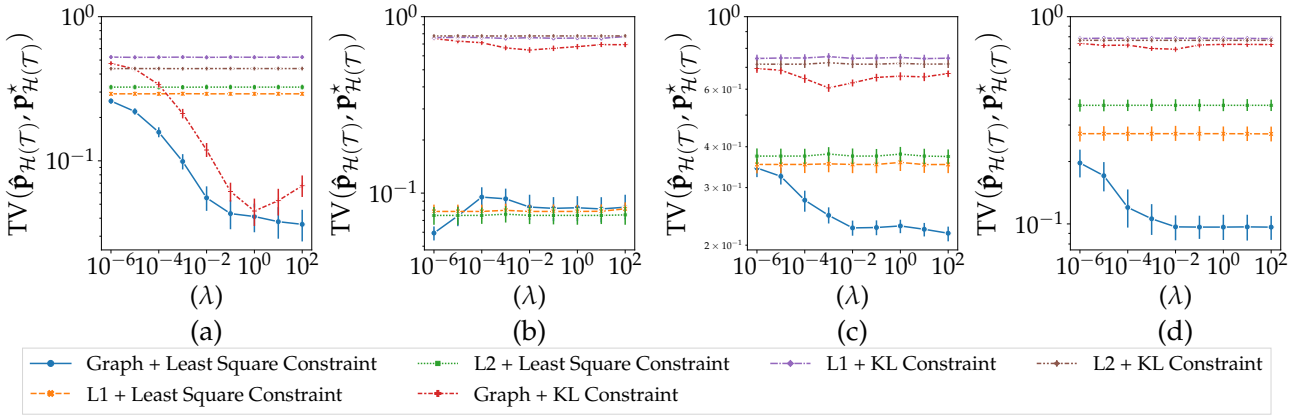

Figure 17: $\mathrm{TV}(\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})})$ for (a) uniform (b) Gaussian (c) mixture of 2 Gaussians (d) mixture of 3 Gaussians user distribution while varying the regularization parameter λ.

In the subsequent simulation, we fix $\lambda = 1$. We now provide simulation results for following group of users distributions: uniform, Gaussian, a mixture of 2 Gaussians, and a mixture of 3 Gaussians. We also present

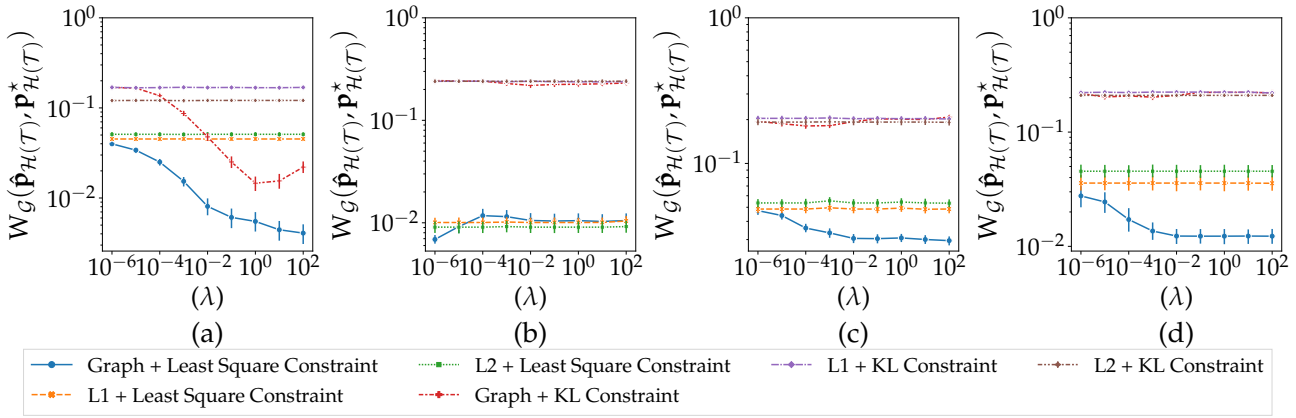

Figure 18: $W_{\mathcal{G}}(\mathbf{p}^{\star}_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})})$ for (a) uniform (b) Gaussian (c) mixture of 2 Gaussians (d) mixture of 3 Gaussians user distribution while varying the regularization parameter λ.

simulations results for varying amount of noises in both noise models. Figure 19-22 show the relationship between the number of hyperplanes, n_h, and the error in recovered mass in the partition $\mathcal{H}(\mathcal{T})$, as well as the relationship between the number of people asked per query, n_p, and the error for $d = 2$, with the four user distributions and different noise models.

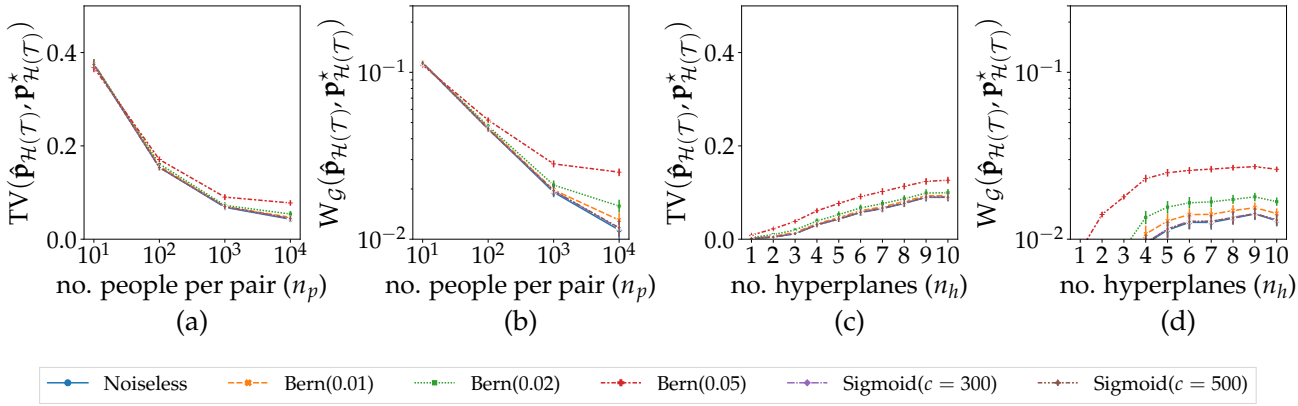

Figure 19: $TV(\mathbf{p}^{\star}_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})})$ and $W_{\mathcal{G}}(\mathbf{p}^{\star}_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})})$ for uniform user distribution in 2D.

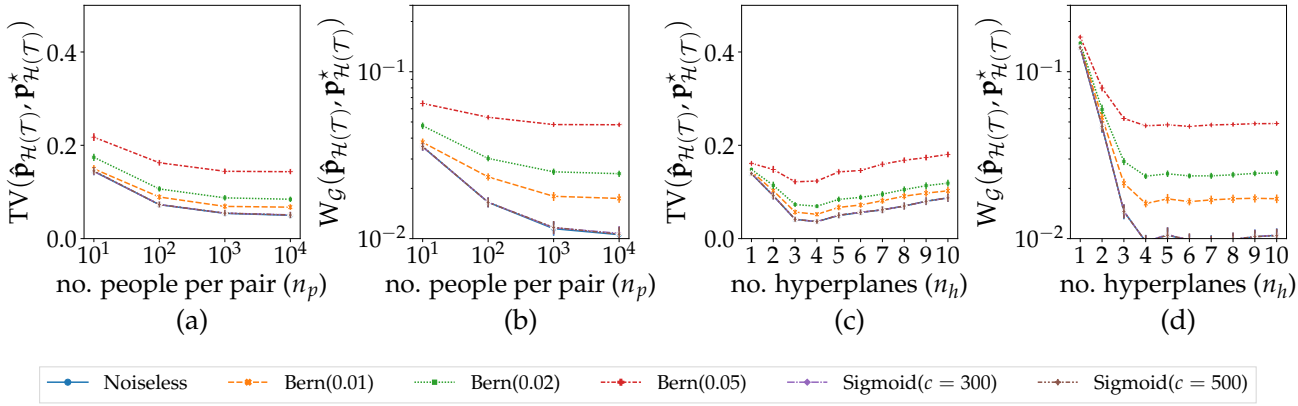

Figure 20: $TV(\mathbf{p}^{\star}_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})})$ and $W_{\mathcal{G}}(\mathbf{p}^{\star}_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})})$ for Gaussian user distribution in 2D.

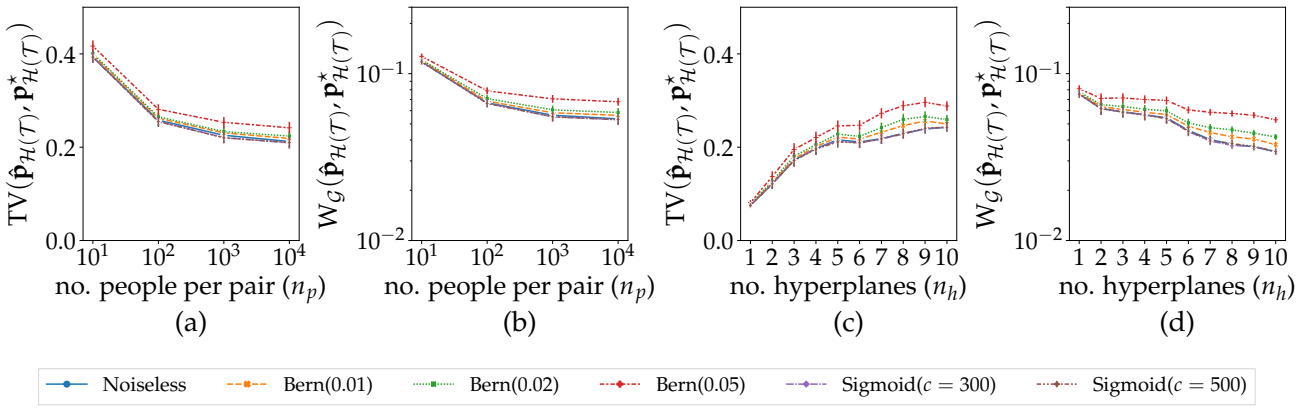

Figure 21: $\mathrm{TV}(\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})})$ and $\mathrm{W}_{\mathcal{G}}(\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})})$ for a mixture of 2 Gaussian user distribution in 2D.

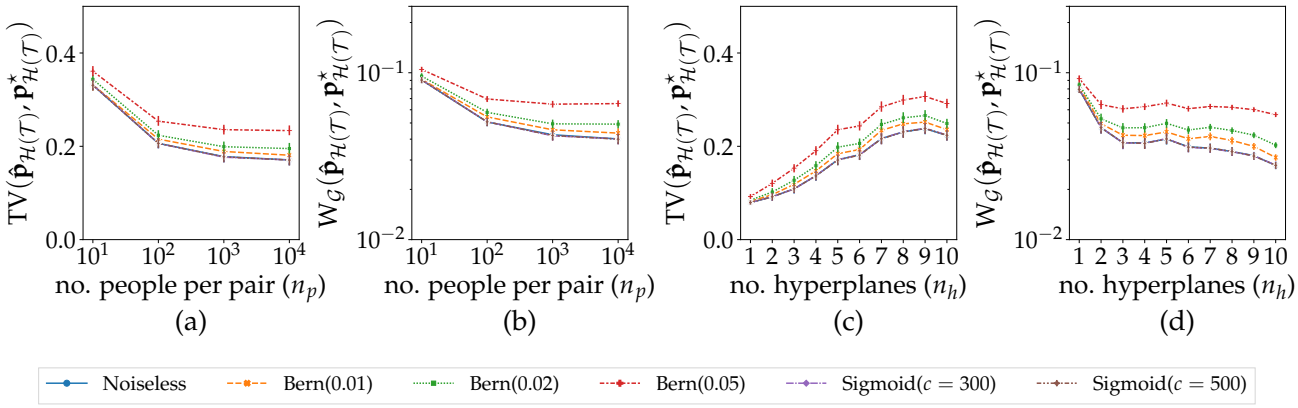

Figure 22: $\mathrm{TV}(\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})})$ and $\mathrm{W}_{\mathcal{G}}(\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})})$ for a mixture of 3 Gaussian user distribution in 2D.

Additionally, Figure 23- 26 show the relationship between the feature dimension d and the error in recovered mass in the partition $\mathcal{H}(\mathcal{T})$.

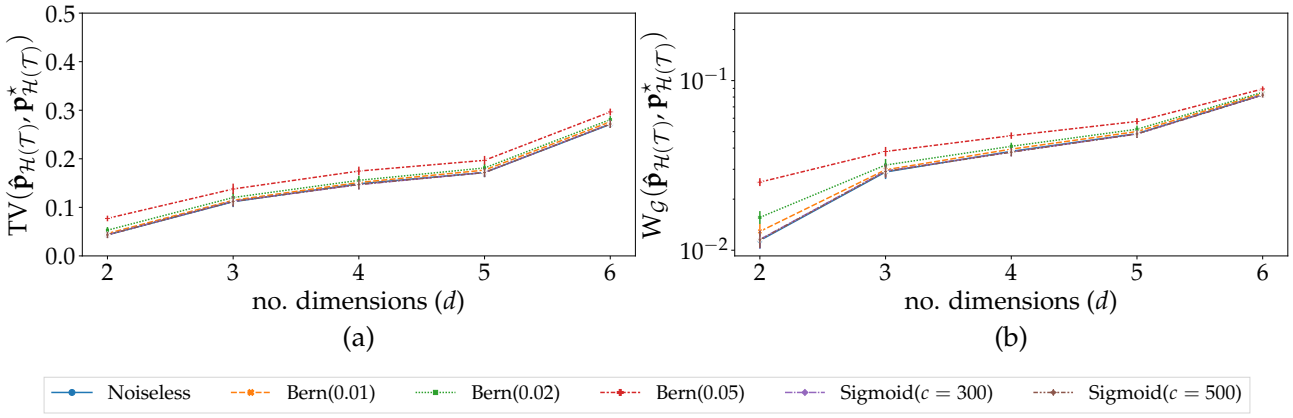

Figure 23: $\mathrm{TV}(\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})})$ and $\mathrm{W}_{\mathcal{G}}(\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})})$ for uniform user distribution in 2D with varying d.

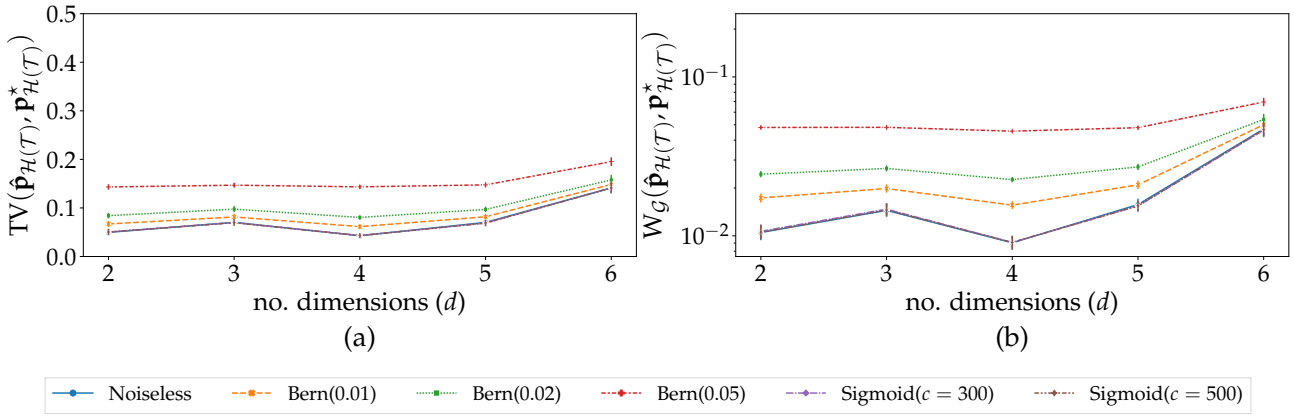

Figure 24: $\mathrm{TV}(\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})})$ and $\mathrm{W}_{\mathcal{G}}(\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})})$ for Gaussian user distribution in 2D with varying d.

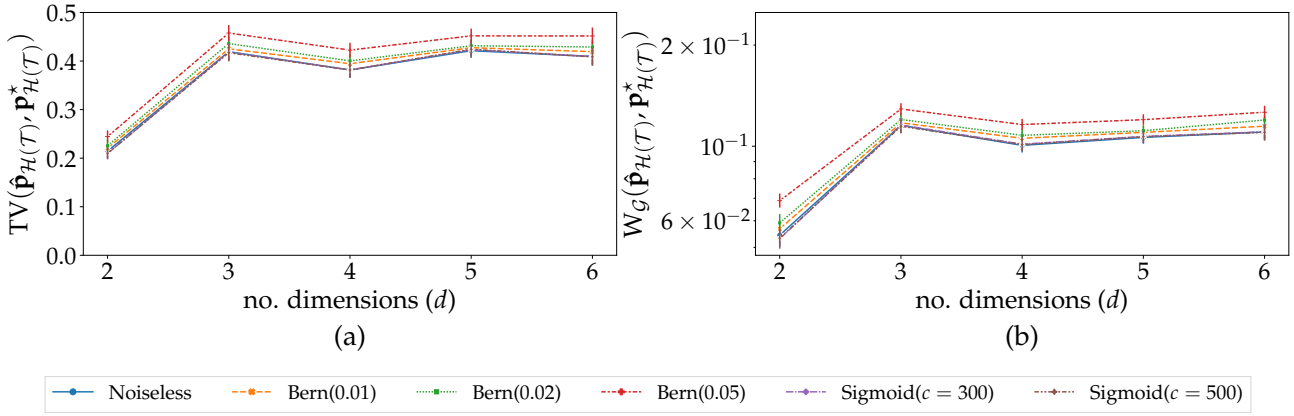

Figure 25: $\mathrm{TV}(\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})})$ and $\mathrm{W}_{\mathcal{G}}(\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})})$ for a mixture of 2 Gaussian user distribution in 2D with varying d.

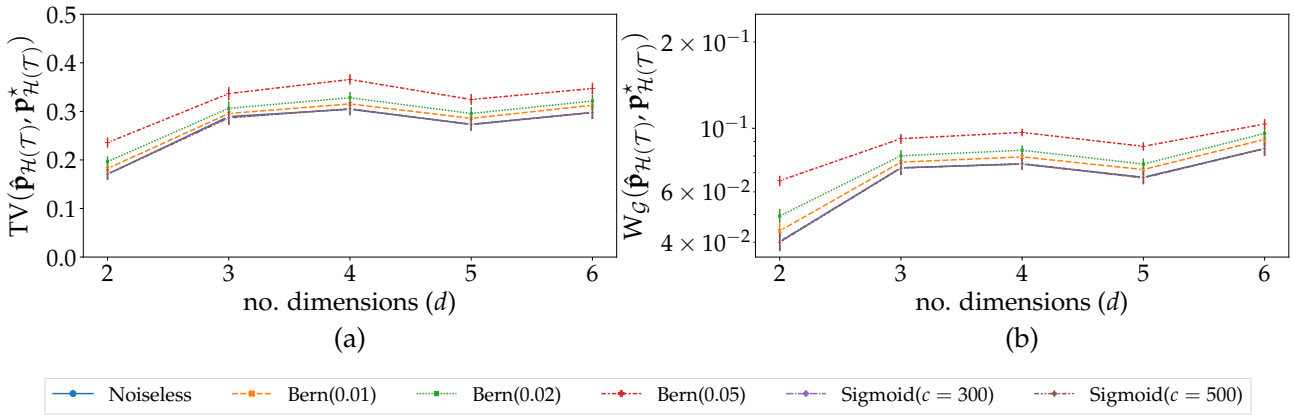

Figure 26: $\mathrm{TV}(\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})})$ and $\mathrm{W}_{\mathcal{G}}(\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})})$ for a mixture of 3 Gaussian user distribution in 2D with varying d.

Figure 27- 30 provide simulation results in terms of $W_{\mathcal{G}}$ using all optimization methods while varying the number of people per pair, n_p, under the four user distributions with $d = 2$ and $n_h = 5$, and no noise model introduced.

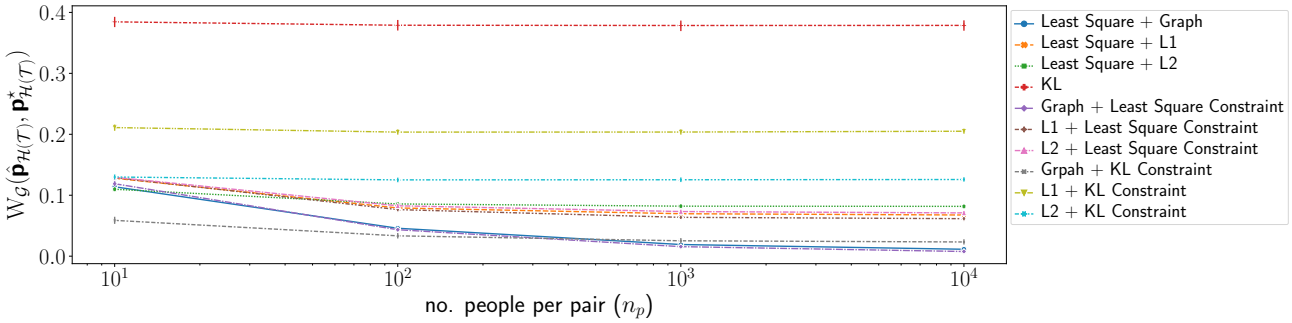

Figure 27: $\mathrm{W}_{\mathcal{G}}(\mathbf{p}^{\star}_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})})$ for uniform user distribution in 2D with varying n_p using all optimization methods.

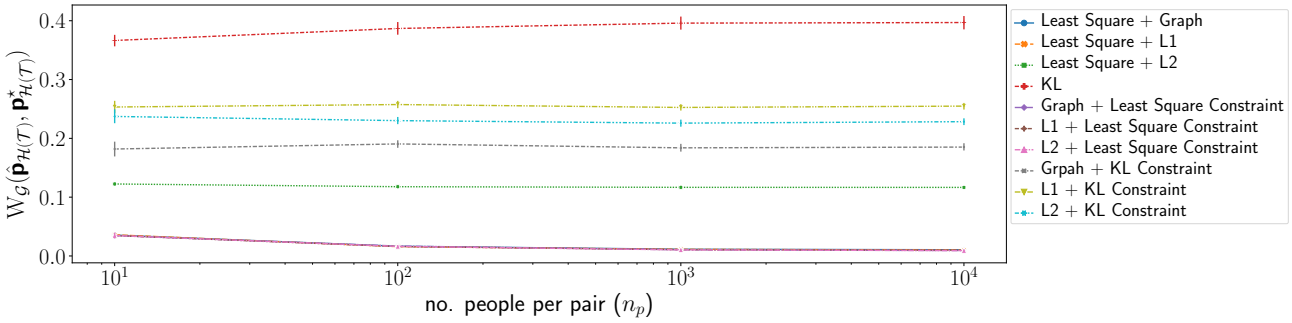

Figure 28: $\mathrm{W}_{\mathcal{G}}(\mathbf{p}^{\star}_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})})$ for Gaussian user distribution in 2D with varying n_p using all optimization methods.

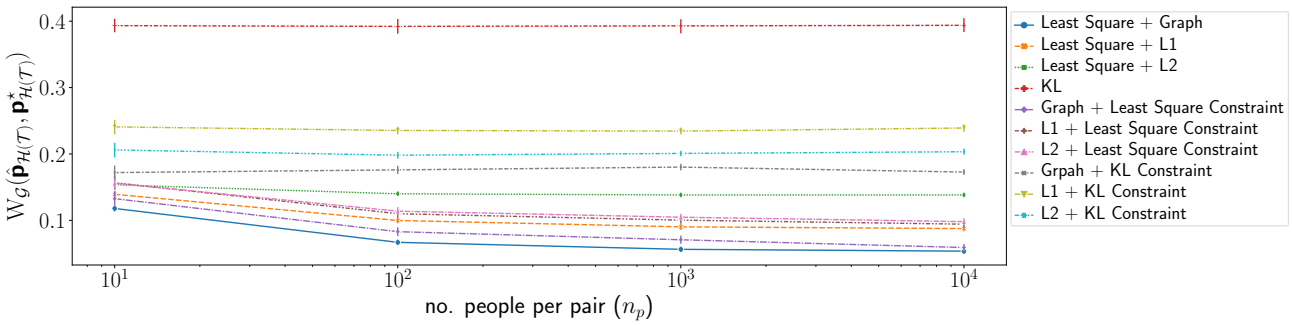

Figure 29: $\mathrm{W}_{\mathcal{G}}(\mathbf{p}^{\star}_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})})$ for a mixture of 2 Gaussian user distribution in 2D with varying n_p using all optimization methods.

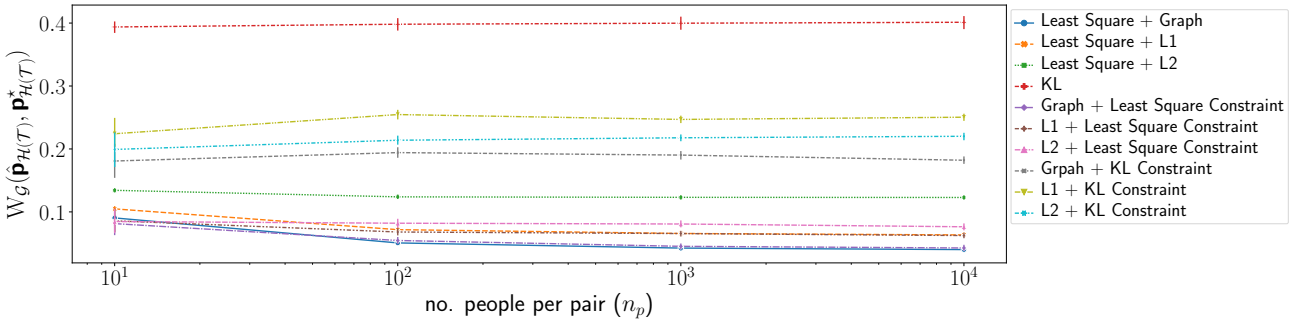

Figure 30: $\mathrm{W}_{\mathcal{G}}(\mathbf{p}^{\star}_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})})$ for a mixture of 3 Gaussian user distribution in 2D with varying n_p using all optimization methods.

Lastly, Figure 31- 34 illustrate simulation results in terms of $W_{\mathcal{G}}$ using ordinary least square, least square with graph, ℓ_1, and ℓ_2 regularizations, and EM algorithm (see below for implementation details) by Lu and Boutilier (2014) with $K \in \{2, 3, 4, 5, 6\}$. We vary the number of people per pair, n_p, under the four user distributions with $d = 2$ and $n_h = 5$, and no noise model introduced.

EM algorithm described by Lu and Boutilier (2014): We implemented the EM algorithm described by Lu and Boutilier (2014), that learns a mixture of Mallows model, to compare the performance of our method and Mallows model-based method. They model the mixture of Mallows model as

$$p(r) = \sum_{k=1}^{K} \pi_k \frac{1}{Z} \phi_k^{d(r,\sigma_k)}$$

where K is the number of components, π_k is the weight parameter for the k-th component, ϕ_k is the dispersion parameter of the k-th component, r is the consensus ranking parameter of the k-th component, and Z is a normalization constant.

In the E-step, the algorithm samples ranking from the mixture according to users' preferences and the parameters for each component. Then, in the M-step, the algorithm updates the parameters accordingly. Specifically, it updates the dispersion parameter using gradient ascent. Therefore, the algorithm takes the following hyperparameters: the number of components K, the number of ranking samples generated for each user in the E-step, the learning rate and the number of steps for the gradient ascent. In all of our experiments, we vary K from 2 to 6 and fix the number of ranking samples per user to be 10, the learning rate for gradient ascent to be 10^{-8}, and the number of steps for gradient ascent to be 10. We end the EM when the consensus ranking for each component at the current step is equal to the one from the previous step.

We begin by generating all rankings that are possible under the ideal point model to determine the mass on polytope regions via the mixture of Mallows model. For each of the region, we assume there exists a user preference point. We find the ranking induced by that point according to the ideal point model. Then, we compare rankings we found for each region with the consensus rankings in the mixture of Mallows model. If the consensus ranking for the j-th component matches the ranking induced by the preference point in the i-th region, the probability mass for the i-th region is assigned to be the weight for the j-th component in the mixture of Mallows. It is possible that none of the rankings induced by the preference points in these regions matches the consensus rankings in the mixture of Mallows model. In such a case, we assign the weight for the j-th component as the probability mass of the region whose ranking is closest to the consensus ranking of the j-th component, measured by Kendall's tau distance.

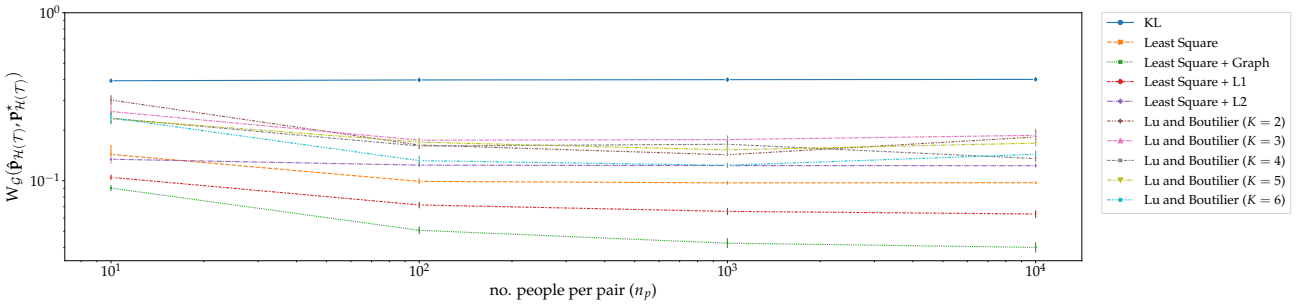

Figure 31: $W_{\mathcal{G}}(\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})})$ for uniform user distribution in 2D with varying n_p using least square with out regularization, least square with graph, ℓ_1, and ℓ_2 regularization, and EM algorithm of Liu and Moitra (2018) with $K \in \{2, 3, 4, 5, 6\}$.

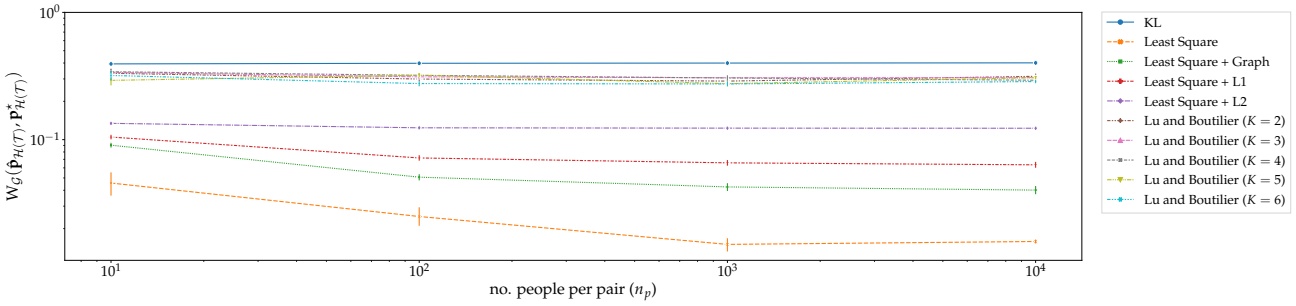

Figure 32: $W_{\mathcal{G}}(\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})})$ for Gaussian user distribution in 2D with varying n_p using least square with out regularization, least square with graph, ℓ_1, and ℓ_2 regularization, and EM algorithm of Liu and Moitra (2018) with $K \in \{2, 3, 4, 5, 6\}$.

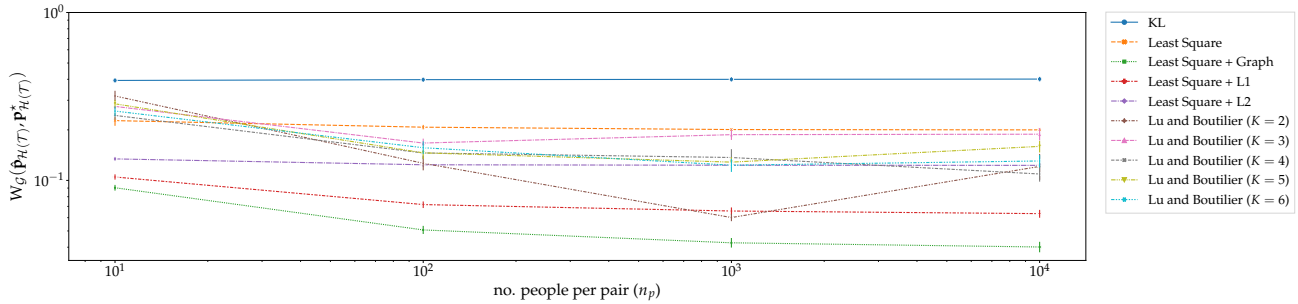

Figure 33: $\mathrm{W}_{\mathcal{G}}(\mathbf{p}^{\star}_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})})$ for a mixture of 2 Gaussian user distribution in 2D with varying n_p using least square with out regularization, least square with graph, ℓ_1, and ℓ_2 regularization, and EM algorithm of Liu and Moitra (2018) with $K \in \{2, 3, 4, 5, 6\}$.

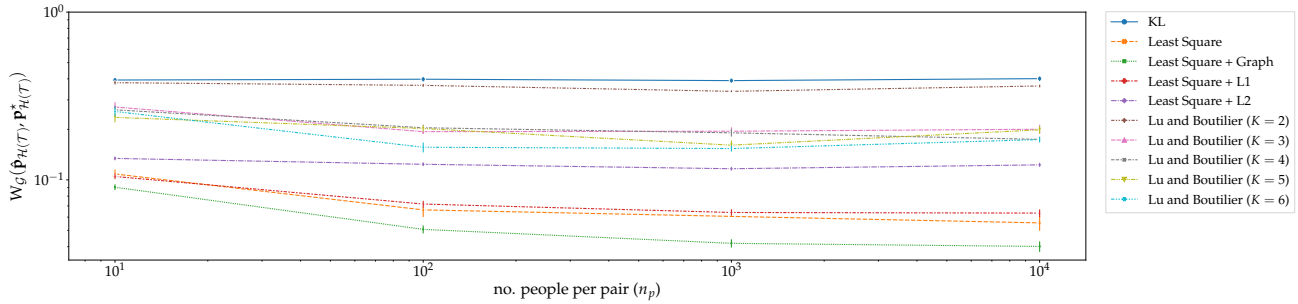

Figure 34: $\mathrm{W}_{\mathcal{G}}(\mathbf{p}^{\star}_{\mathcal{H}(\mathcal{T})}, \hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})})$ for a mixture of 3 Gaussian user distribution in 2D with varying n_p using least square with out regularization, least square with graph, ℓ_1, and ℓ_2 regularization, and EM algorithm of Liu and Moitra (2018) with $K \in \{2, 3, 4, 5, 6\}$.

Computation runtime: We conduct experiments to demonstrate how the optimization runtime varies while we change the dimension of our dataset. It can be seen that the convex optimization solver is able to compute the solution well under 1 second for the number of dimensions ranging from 2 to 5.

dimension	time (in seconds)
2	0.1472 ± 0.1284
3	0.2084 ± 0.3547
4	0.2034 ± 0.3984
5	0.1150 ± 0.2596

Table 1: Average CVX runtime $\pm$ standard deviation for each dimension. The simulations are conducted on AWS EC2 c5.metal instance, with 96 vCPU, 192 GiB or memory. The solver used with CVXPY is ECOS. The underlying user distribution is mixture of 3 Gaussians. We vary d from 2 to 5. For the same d, we use 10 different initialization of users and 10 different initialization of items. We randomly selected 5 out of the 10 possible hyperplanes, where each is queried with 10,000 users.

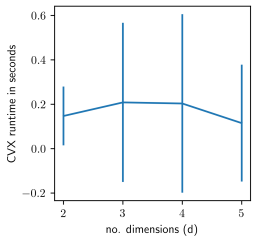

Figure 35: CVX runtime for each dimension (complementary to the Table 1).

Major Category	Minor Category
Boots	Ankle, Knee High, Mid-Calf, Over the Knee, Prewalker Boots
Sandals	Athletic, Flat, Heel
Shoes	Boat Shoes, Clogs and Mules, Crib Shoes, Firstwalker, Flats, Heels, Loafers, Oxfords, Prewalker, Sneakers and Atheletic Shoes
Slippers	Boot, Slipper Flats, Slipper Heels

Table 2: Major and minor categories in the Zappos dataset.

D.4 Zappos

The Zappos dataset (UT Zappos50K) (Yu and Grauman, 2014, 2017) comprises 4 major categories of shoes: Boots, Sandals, Shoes, and Slippers. Each major category includes several minor categories. For instance, within the Boots category, you can find Ankle, Knee High, Mid-Calf, Over the Knee, and Prewalker Boots. Table 2 shows the major and minor categories in the Zappos dataset.

Data Preprocessing: We consider minor category, that has a cardinality of 21, as the label space. To ensure that all the images have the same dimension, we use the Zappos image square dataset. Then, we resize them to 135×135. Lastly, we convert the images into grey scale.

We train a modified VGG11 convolutional neural network (Simonyan and Zisserman, 2014) on the Zappos dataset. VGG11 is intended to be trained on ImageNet (Deng et al., 2009), which has 1000 classes. We modify the last layer of the network so that it works with 21 classes. We insert a new layer as the penultimate layer of the network. This is because the original penultimate layer has an output of dimension 4096, which is too large. By reducing it to 512, we can employ the output of this penultimate layer as the embedding for the Zappos dataset.

Training: We use 80% of the dataset as the training set and the rest as the test set, both with a batch size of 64. We use SGD optimizer with learning rate 0.01, momentum 0.9, and weight decay 0.0005. After 12 epochs, we achieve a training accuracy of 94.05% and a test accuracy of 86.81%. To generate an embedding for the Zappos dataset, we feed the entire dataset into the trained network and extract the output from the penultimate layer, resulting in a matrix of dimensions 50066×512. We use PaCMAP (Wang et al., 2021b) with default parameters to obtain the 2D embedding of shoes as shown in Figure 37.

Data Collection via Crowdsourcing: We pick 5 shoes as our query item set (Figure 36).

Figure 36: The 5 shoes we pick for pairwise comparison task on Amazon Mechanical Turk.

We posted this task on Amazon Mechanical Turk (AMT). Each task has 15 pairwise comparison queries (10 pairs and 5 repeats). The median time taken per query is around $2.58s$ and for the task (15 pair comparisons) is $\sim 47s$. Each worker is paid 15 cents per task. This is roughly $\sim$ \$7 per hour. We did not restrict the task to the master workers. The task was open to all those who had at least 500 HITs approved and 95% approval rate.

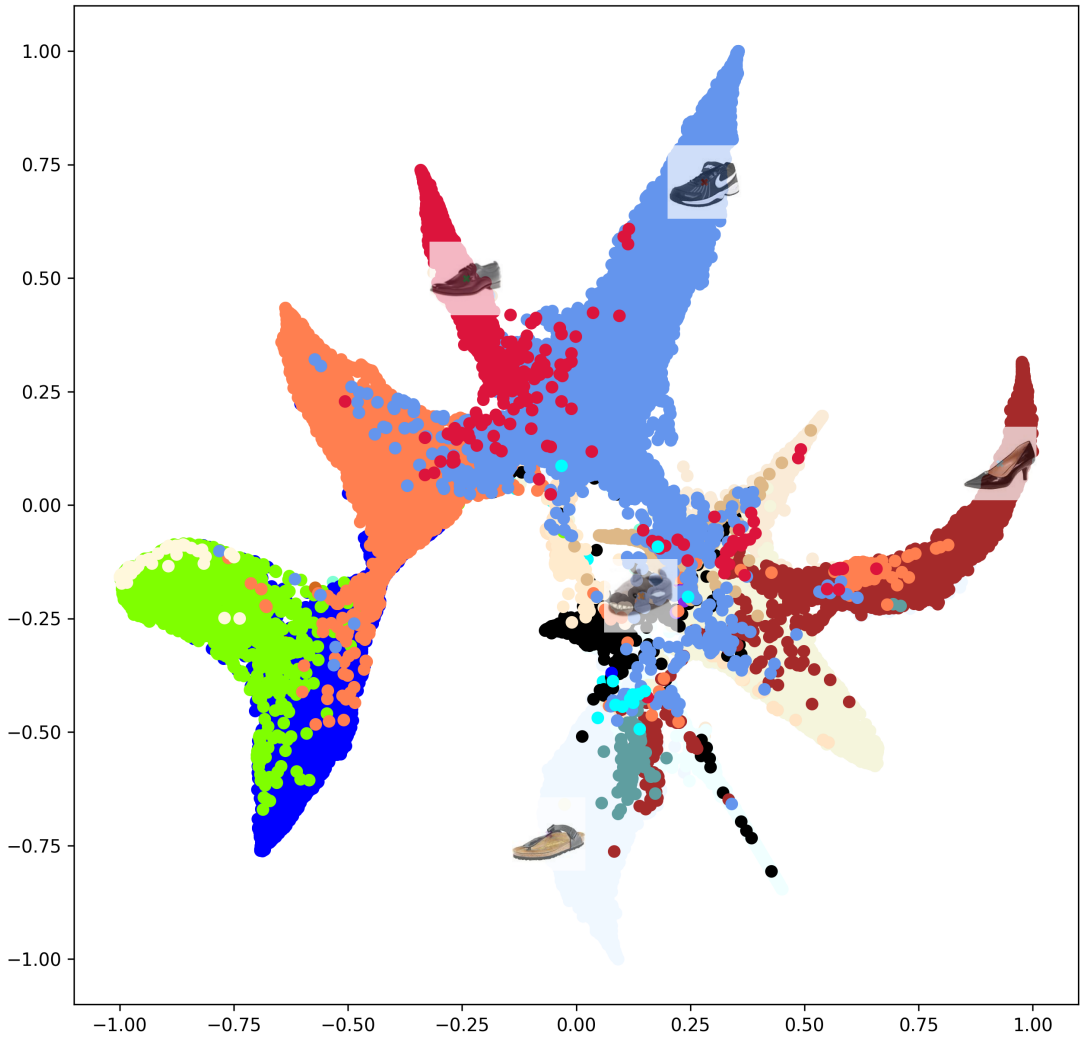

Figure 37: 2D embedding of the 5 shoes obtained using penultimate layer of modified VGG11 and PaCMAP. Each color represents a minor category. 5 shoes we used for experiments are also located.

Figure 38 shows the instructions provided and the interface for answering pairwise comparison queries.

We first bootstrap $10\%, 20\%, 30\%, 40\%, 50\%$ of all workers, repeating the process 100 times for each percentage. Then, we use answers to all possible queries from these workers to estimate the true mass with $n_h = 5$ and $n_h = 10$. We create a global bin (initialized to 0) whose size equals to the number of regions formed by the hyperplanes. Each worker has its own local bin (initialized to 0) that has the same size as the global bin. For each pairwise comparison query, a worker can only be on one side of the corresponding hyperplane. Consider all polytopes on the side of the hyperplane related to worker's answer. We increase the corresponding entries of these polytopes in the bin by 1. After we examine all queries, a set of entries has the maximum value among all entries in the bin. Ideally, this set has a cardinality of 1. However, due to noises and worker's inconsistency, it is

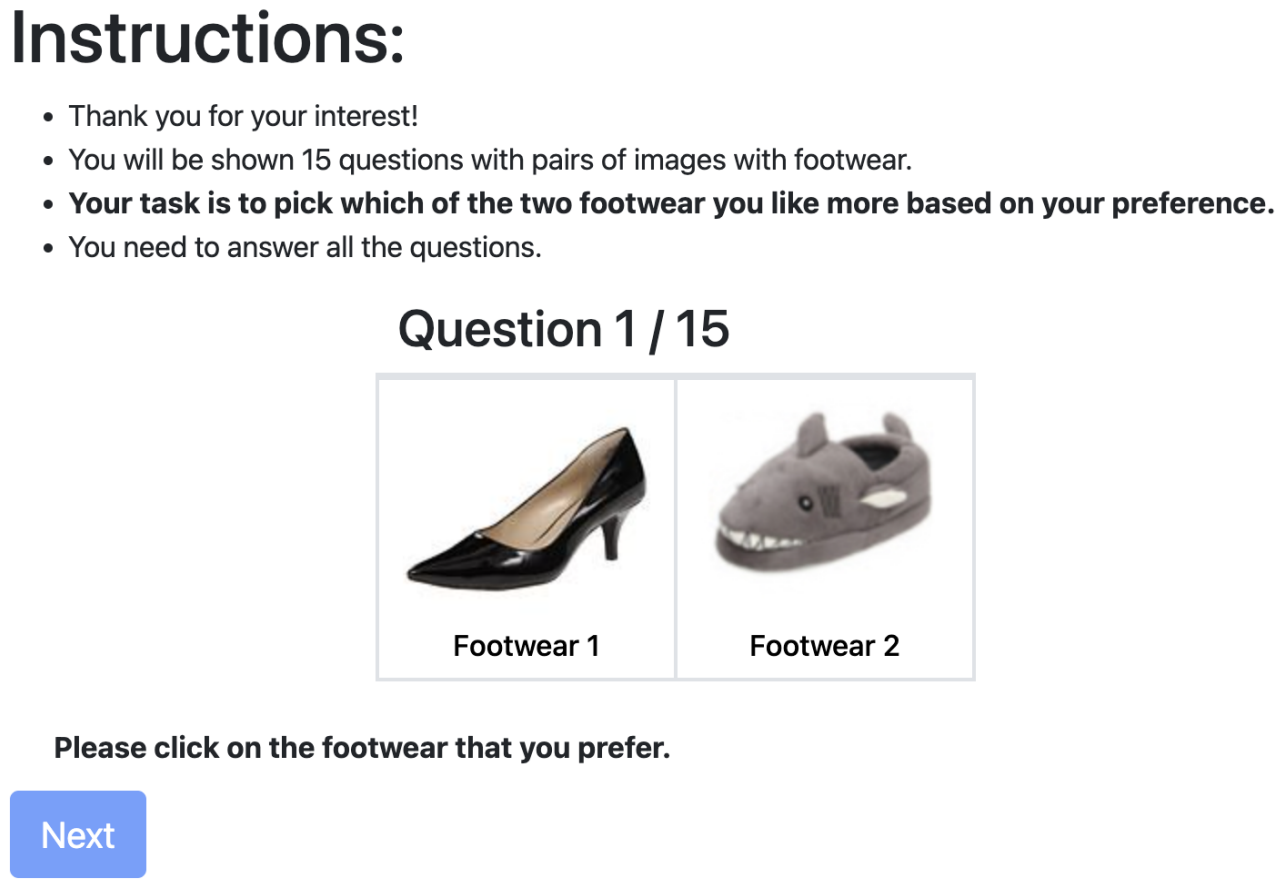

Figure 38: Amazon Mechanical Turk Task interface

possible for the cardinality to be greater than 1. We increase the corresponding entries of this set in the global bin by $\frac{1}{\text{cardinality of the set}}$. After we examine all workers we bootstrapped, we normalize the global bin and obtain a probability vector, which is our estimate of the $\mathbf{p}^\star$.

Figure 39 and 40 illustrate the $\mathbf{p}^\star$ we estimated via bootstrapping. It can be seen that the true distribution $\mathbf{p}^\star$ that we estimated is stable across different bootstrap settings.

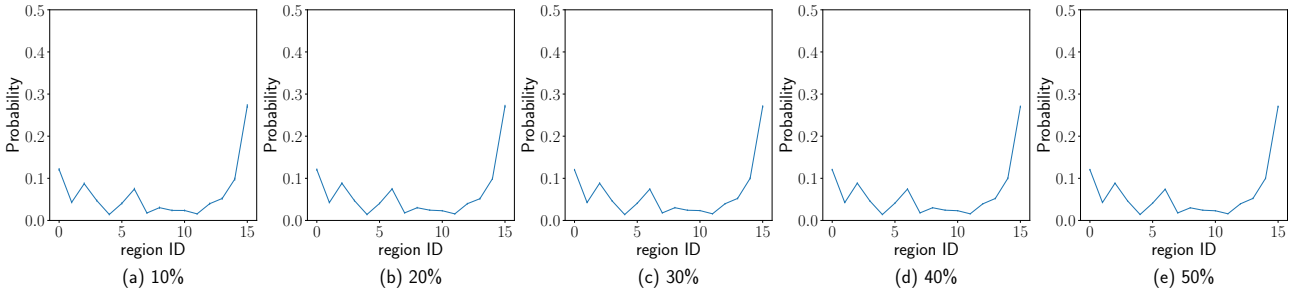

Figure 39: $\mathbf{p}^\star$ obtained via bootstrap (a) 10% (b) 20% (c) 30% (d) 40% (e) 50% of all crowdworkers when $n_h = 5$.

We also present $\hat{\mathbf{p}}$ that we obtain via our method. We first use 20% of crowdworkers to estimate $\mathbf{p}^\star$. Then, use the remaining 80% of crowdworkers to answer the pairwise comparisons and estimate $\hat{\mathbf{p}}$ using our method. We shuffle the remaining 80% of crowdworkers 100 times to obtain 100 different $\hat{\mathbf{q}}$ and hence 100 different $\hat{\mathbf{p}}$. We repeat the above process 5 times (each time with different 20% of crowdworkers to estimate $\mathbf{p}^\star$) for both $n_h = 5$ and $n_h = 10$. The results are presented in Figure 41 and 42. The bounds for $\mathbf{p}^\star$ are presented in Figure 43 and 44 for $n_h = 5$ and $n_h = 10$, respectively.

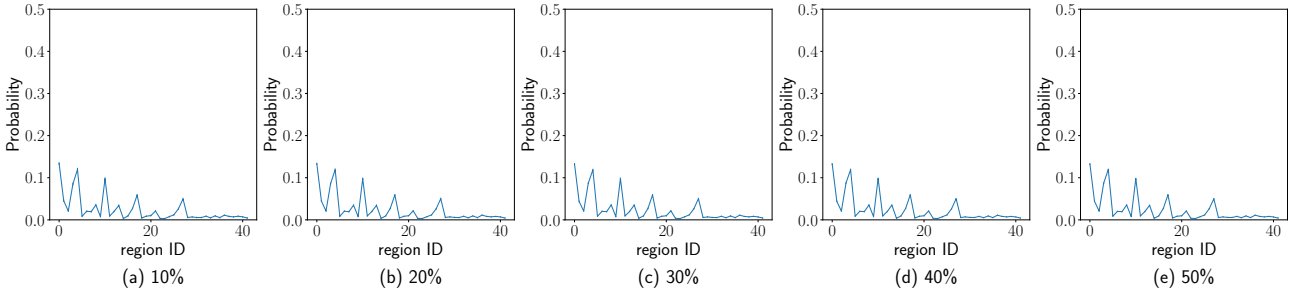

Figure 40: $\mathbf{p}^\star$ obtained via bootstrap (a) 10% (b) 20% (c) 30% (d) 40% (e) 50% of all crowdworkers when $n_h = 10$.

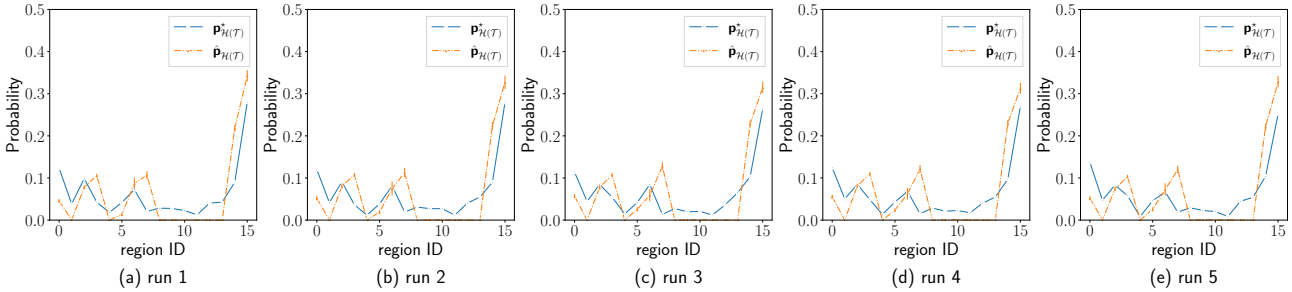

Figure 41: $\mathbf{p}^\star$ obtained using 20% of crowdworkers and $\hat{\mathbf{p}}$ obtained using the remaining 80% of the crowdworkers 100 times. Each of the (a)-(e) uses different set of 20% of all crowdworkers to obtain $\mathbf{p}^\star$, when $n_h = 5$.

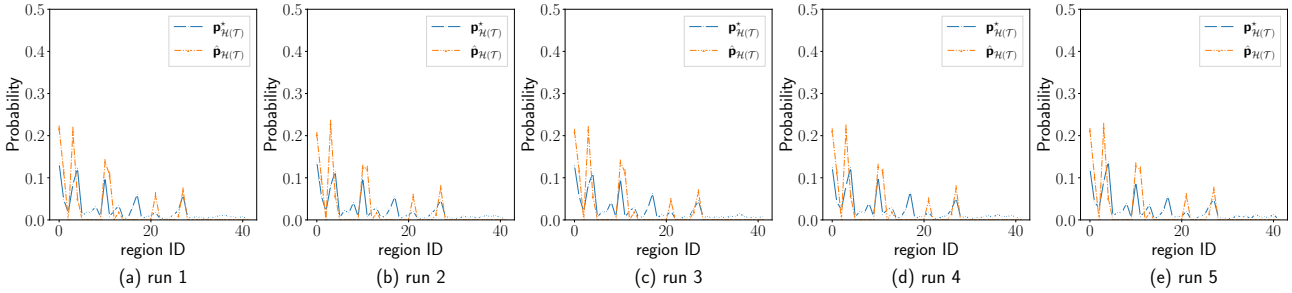

Figure 42: $\mathbf{p}^\star$ obtained using 20% of crowdworkers and $\hat{\mathbf{p}}$ obtained using the remaining 80% of the crowdworkers 100 times. Each of the (a)-(e) uses different set of 20% of all crowdworkers to obtain $\mathbf{p}^\star$, when $n_h = 10$.

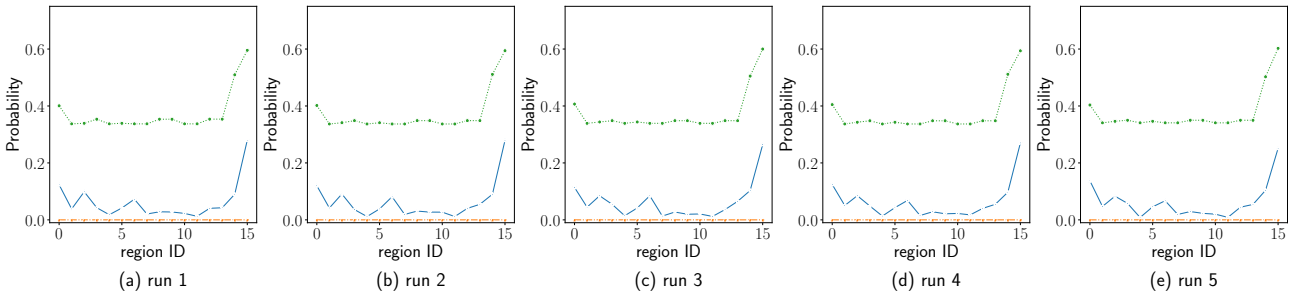

Figure 43: Upper and lower bound for $\mathbf{p}^\star$ when $n_h = 5$.

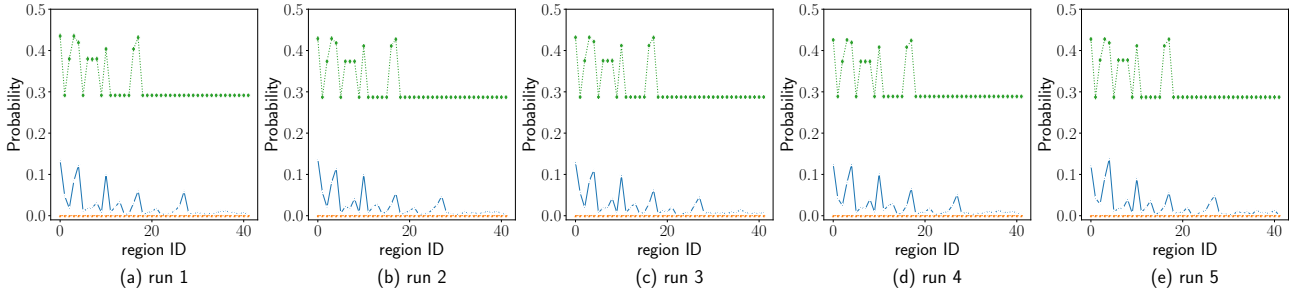

Figure 44: Upper and lower bound for $\mathbf{p}^\star$ when $n_h = 10$.

Figure 45 shows the TV and $W_{\mathcal{G}}$ between $\mathbf{p}^\star$ and $\hat{\mathbf{p}}$ while we vary n_h from 1 to 10.

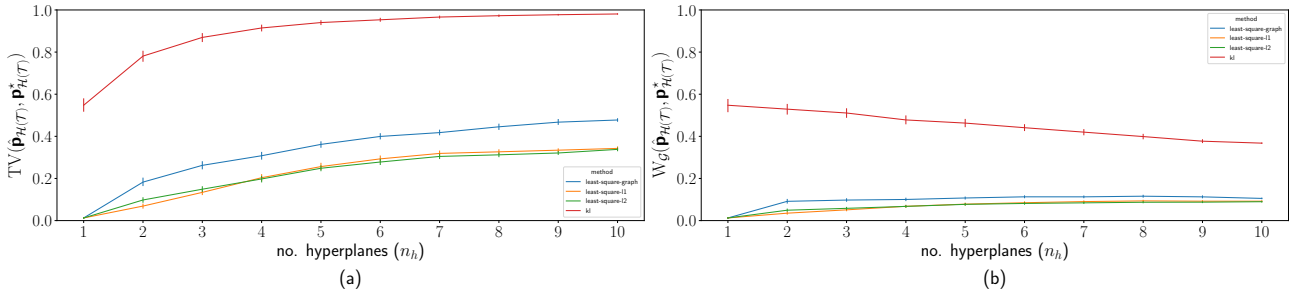

Figure 45: TV and $W_{\mathcal{G}}$ when we vary n_h.

Figure 46 illustrates the probability mass recovered by our algorithm and Lu and Boutilier, with $K \in \{2, 3, 4, 5, 6\}$.

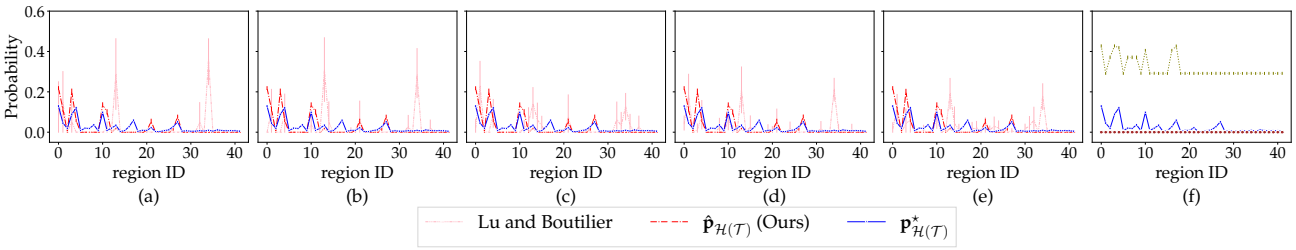

Figure 46: (a)-(e) Our $\hat{\mathbf{p}}$ and the probability mass recovered by Lu and Boutilier, with $K = \{2, 3, 4, 5, 6\}$, respectively. (f) Upper and lower bound for $\mathbf{p}^\star$

Lastly, Figure 47-56 illustrate the polytopes formed by the hyperplanes as well as $\mathbf{p}^\star$ and $\hat{\mathbf{p}}$ while we vary n_h from 1 to 10.

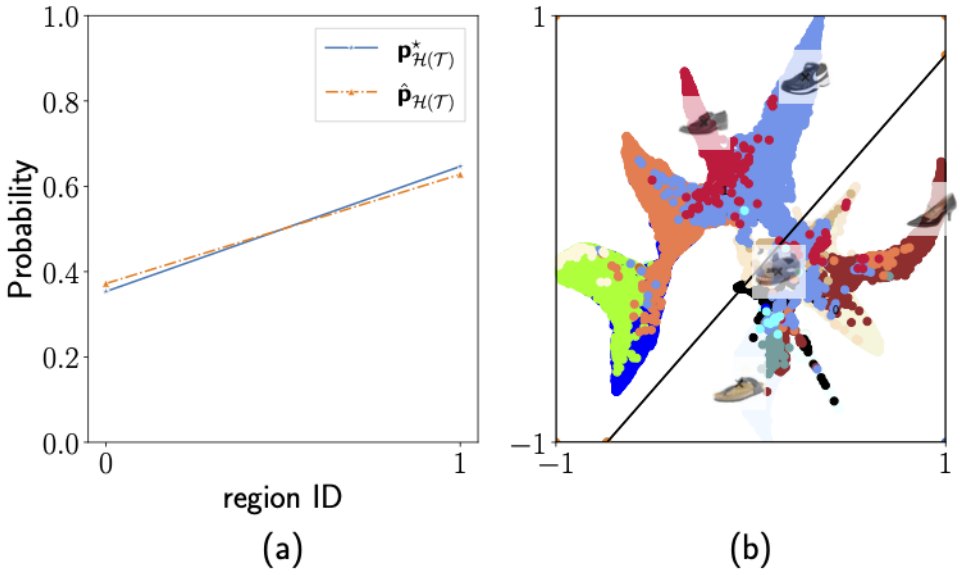

Figure 47: (a) $\mathbf{p}^{\star}$ and $\hat{\mathbf{p}}$ recovered by our algorithm when $n_h = 1$. (b) regions formed by 1 hyperplane. The numbers in each region corresponds to the region ID.

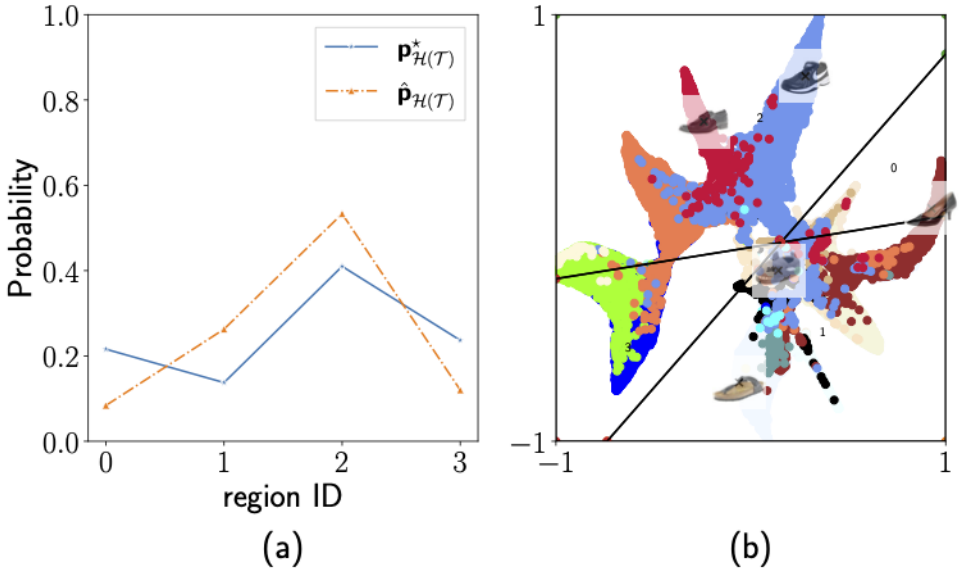

Figure 48: (a) $\mathbf{p}^{\star}$ and $\hat{\mathbf{p}}$ recovered by our algorithm when $n_h = 2$. (b) regions formed by the 2 hyperplanes. The numbers in each region corresponds to the region ID.

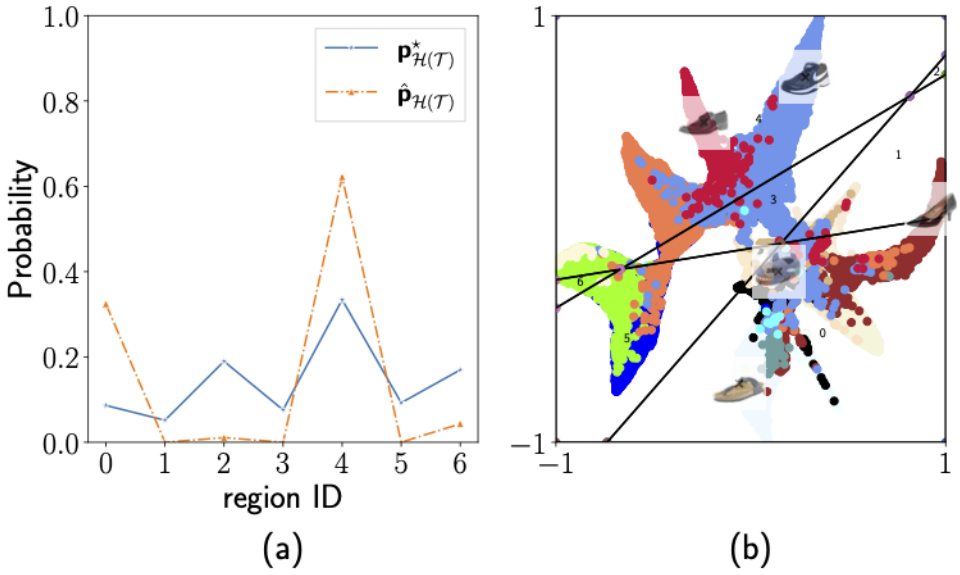

Figure 49: (a) $\mathbf{p}^\star$ and $\hat{\mathbf{p}}$ recovered by our algorithm when $n_h = 3$. (b) regions formed by the 3 hyperplanes. The numbers in each region corresponds to the region ID.

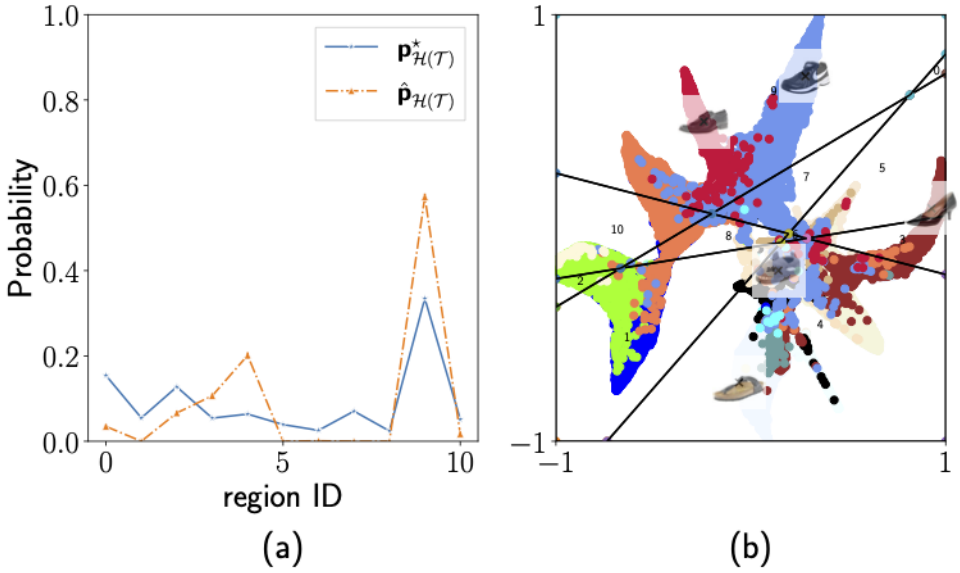

Figure 50: (a) $\mathbf{p}^\star$ and $\hat{\mathbf{p}}$ recovered by our algorithm when $n_h = 4$. (b) regions formed by the 4 hyperplanes. The numbers in each region corresponds to the region ID.

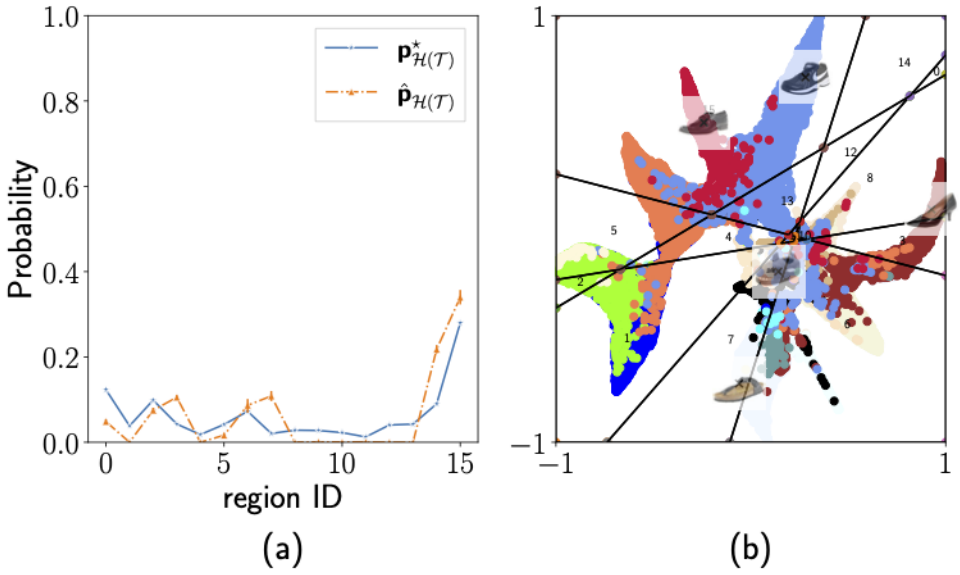

Figure 51: (a) $\mathbf{p}^{\star}$ and $\hat{\mathbf{p}}$ recovered by our algorithm when $n_h = 5$. (b) regions formed by the 5 hyperplanes. The numbers in each region corresponds to the region ID.

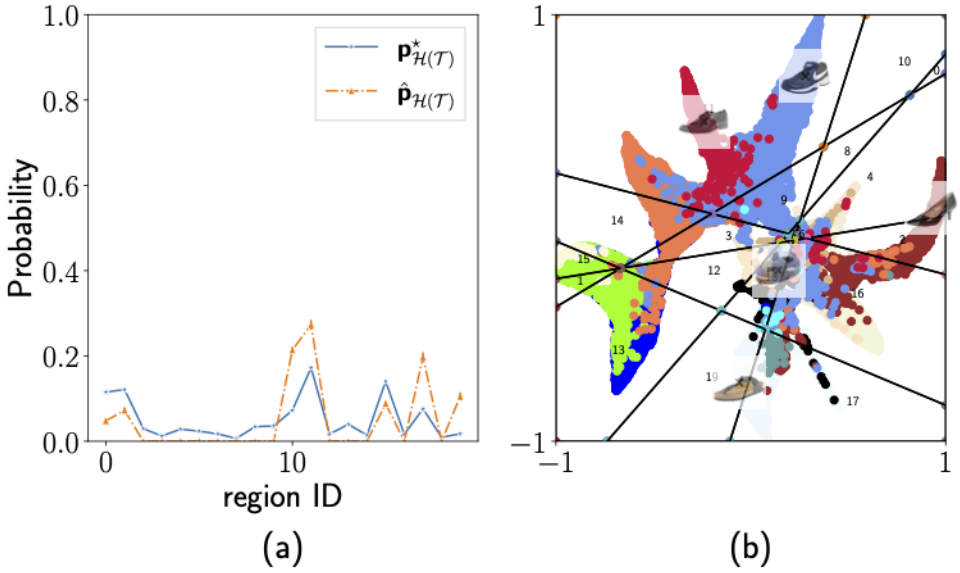

Figure 52: (a) $\mathbf{p}^{\star}$ and $\hat{\mathbf{p}}$ recovered by our algorithm when $n_h = 6$. (b) regions formed by the 6 hyperplanes. The numbers in each region corresponds to the region ID.

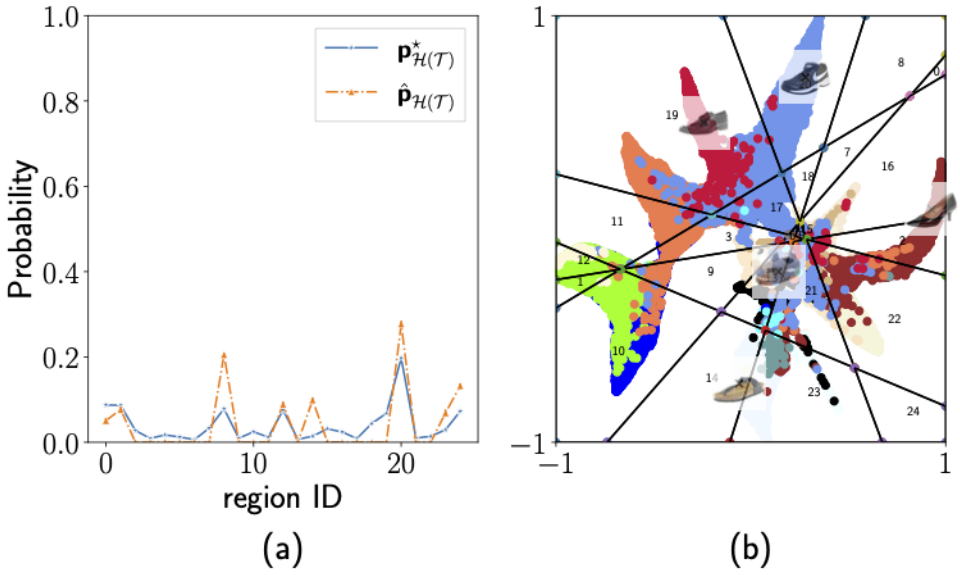

Figure 53: (a) $\mathbf{p}^\star$ and $\hat{\mathbf{p}}$ recovered by our algorithm when $n_h = 7$. (b) regions formed by the 7 hyperplanes. The numbers in each region corresponds to the region ID.

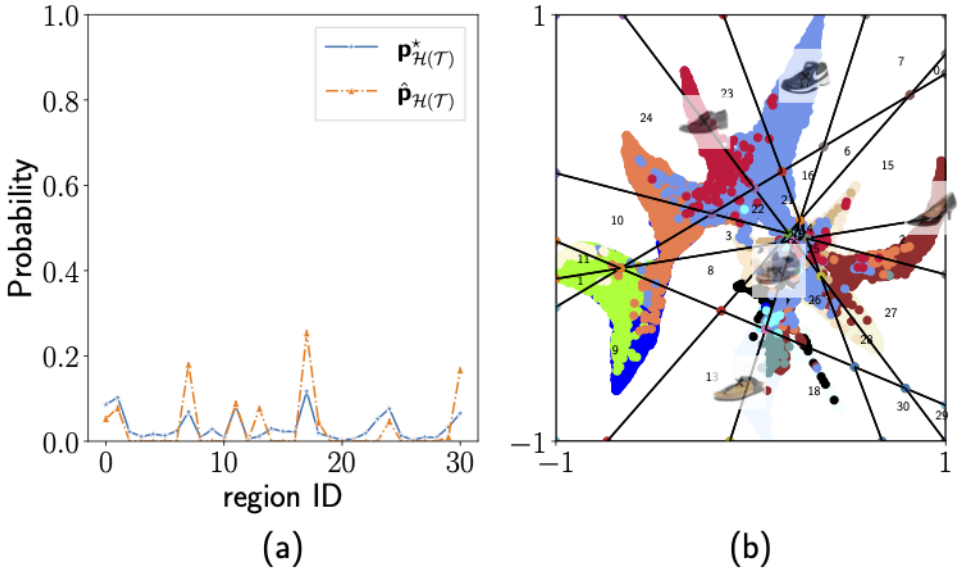

Figure 54: (a) $\mathbf{p}^\star$ and $\hat{\mathbf{p}}$ recovered by our algorithm when $n_h = 8$. (b) regions formed by the 8 hyperplanes. The numbers in each region corresponds to the region ID.

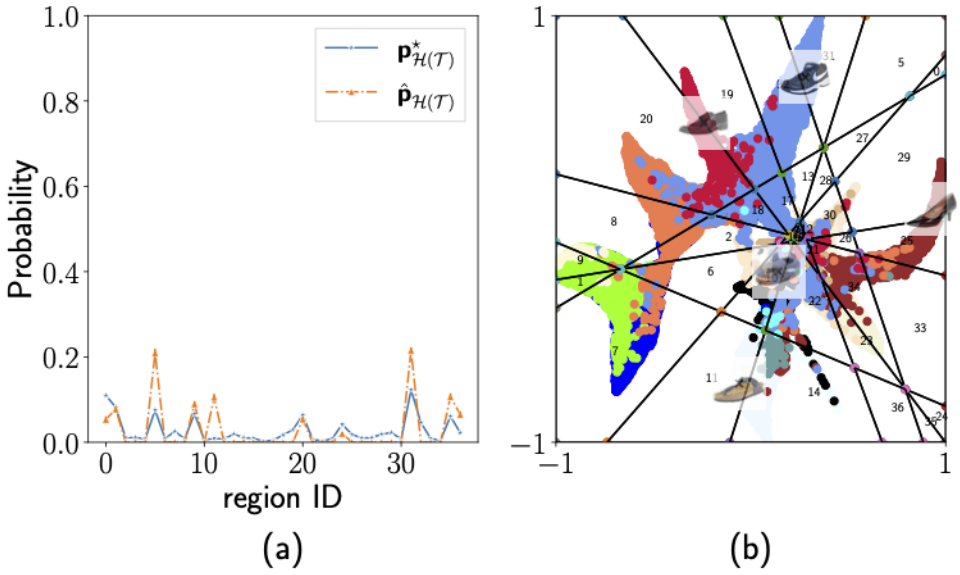

Figure 55: (a) $\mathbf{p}^\star$ and $\hat{\mathbf{p}}$ recovered by our algorithm when $n_h = 9$. (b) regions formed by the 9 hyperplanes. The numbers in each region corresponds to the region ID.

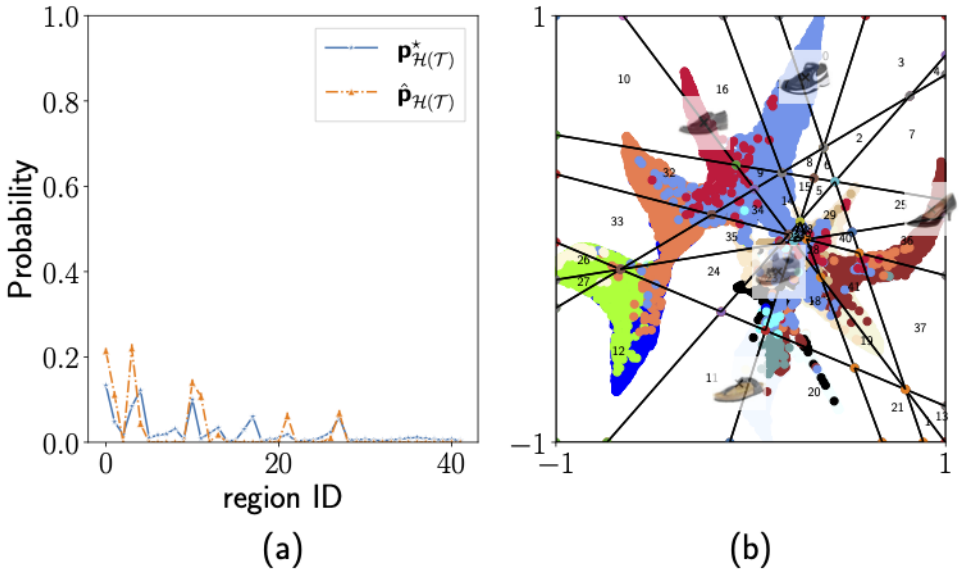

Figure 56: (a) $\mathbf{p}^\star$ and $\hat{\mathbf{p}}$ recovered by our algorithm when $n_h = 10$. (b) regions formed by the 10 hyperplanes. The numbers in each region corresponds to the region ID.

D.4.1 Movies

We create a new dataset of 4266 movies. We use OpenAI's *text-embedding-ada-002* model to generate an 1536 dimensional embedding for each movie. Then, we train a regression neural network, where the target is each movie's average IMDB rating. For this, we use 80% of the movies as the training set and the rest as the test set, where batch size 4. We use SGD optimization with learning rate 0.0001, momentum 0.9, Huber loss. After 250 epochs, we reach a mean average error of 0.63 on the test set.

For pairwise comparisons task, we pick the following 2 sets of movies in a way that most of the movies in those 2 sets have unbalanced opinion in terms of critics by general audience:

- DCEU superheroes (12 DC superhero movies)
- Movie2 (7 movies from US, China, and South Korea)

We scrap critics and audience ratings for selected movies from Rotten Tomatoes. Then, we construct a set of users for each movie from its reviewers. We look for intersections of user sets for each pair of movies. If the size of intersection is small, we discard the corresponding pair. Since movies in DCEU have the similar type and are from the same franchise, it is more likely that we encounter common reviewers. Hence, the definition of small size of intersection is different for the 2 sets. For DCEU, we discard pairs whose size of intersection is less than 200. For Movie2, we discard pairs whose size of intersection is less than 50. This process leaves us with 9 pairs of movies for DCEU and 3 pairs of movies for Movie 2.

For a given pairwise comparison query based on movie pairs, a randomly selected reviewer picks the one that has a higher rating (rated by the same reviewer). If both movies in a pair have the same rating, we pick the movie on the left of the pair. After a reviewer answered one query, we are done with this reviewer. We perform 100 repetitions of calculating $\hat{\mathbf{p}}_{\mathcal{H}(\mathcal{T})}$ by reshuffling users each time, where $n_p = 50$ for DCEU and $n_p = 25$ for Movie2. Since we do not query any user more than once, we do not have enough information to estimate $\mathbf{p}^\star_{\mathcal{H}(\mathcal{T})}$, unlike our experimental work on the Zappos dataset.

Figure 57 (a) (b) show the $\hat{\mathbf{p}}$ recovered using our method and the bounds for $\mathbf{p}^\star$ for the DCEU movie set. Figure 57 (c) (d) show the $\hat{\mathbf{p}}$ recovered using our method and the bounds for $\mathbf{p}^\star$ for the Movie2 movie set.

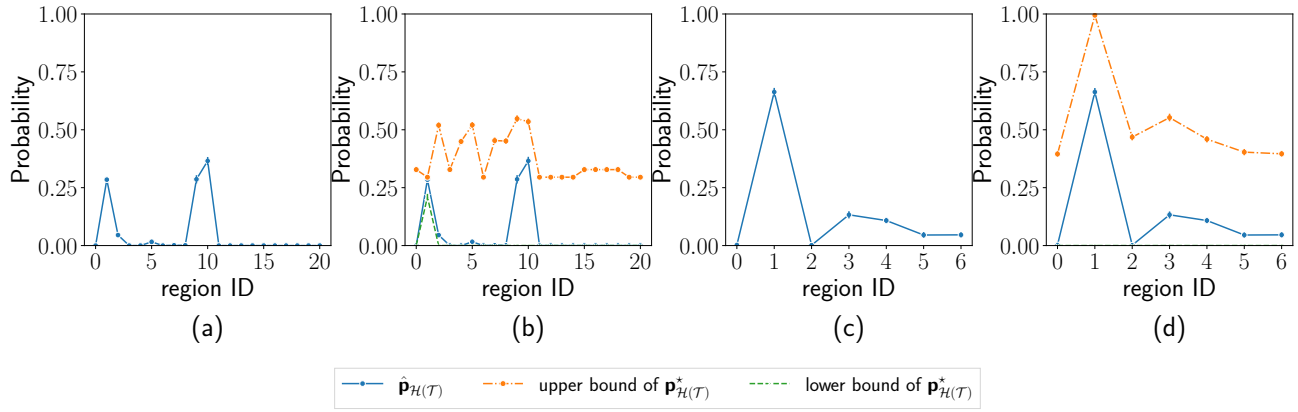

Figure 57: (a) $\hat{\mathbf{p}}$ recovered for the DCEU movie set (b) Bounds for $\mathbf{p}^\star$ for the DCEU movie set (c) $\hat{\mathbf{p}}$ recovered for the Movie2 movie set (d) Bounds for $\mathbf{p}^\star$ for the Movie2 movie set.

Figure 58 and 59 illustrates the probability mass recovered by our algorithm and EM algorithm of Lu and Boutilier (2014), with $K \in \{2, 3, 4, 5, 6\}$, respectively.

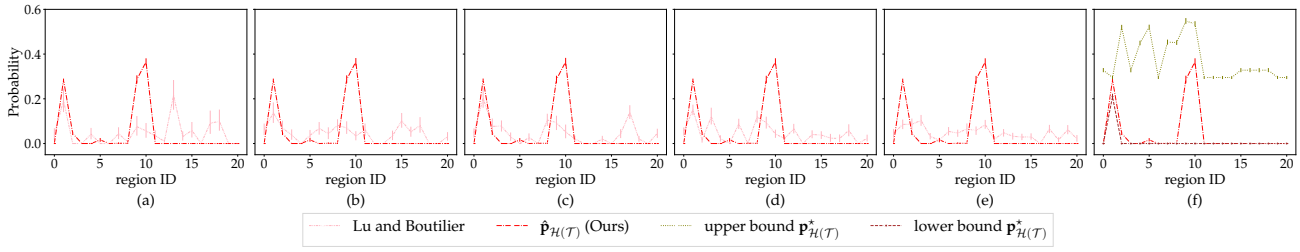

Figure 58: (a)-(e) Our $\hat{\mathbf{p}}$ and the probability mass recovered by Lu and Boutilier, with $K = \{2, 3, 4, 5, 6\}$, respectively. (f) Upper and lower bound for $\mathbf{p}^\star$

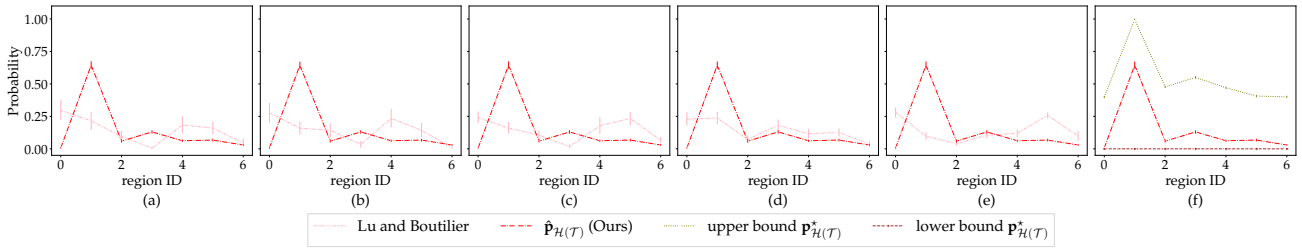

Figure 59: (a)-(e) Our $\hat{\mathbf{p}}$ and the probability mass recovered by Lu and Boutilier, with $K = \{2, 3, 4, 5, 6\}$, respectively. (f) Upper and lower bound for $\mathbf{p}^{\star}$

Figure 60 and 61 show the regions formed by the hyperplanes and the movies' location in the embedding space for DCEU and Movie2 movie set, respectively.

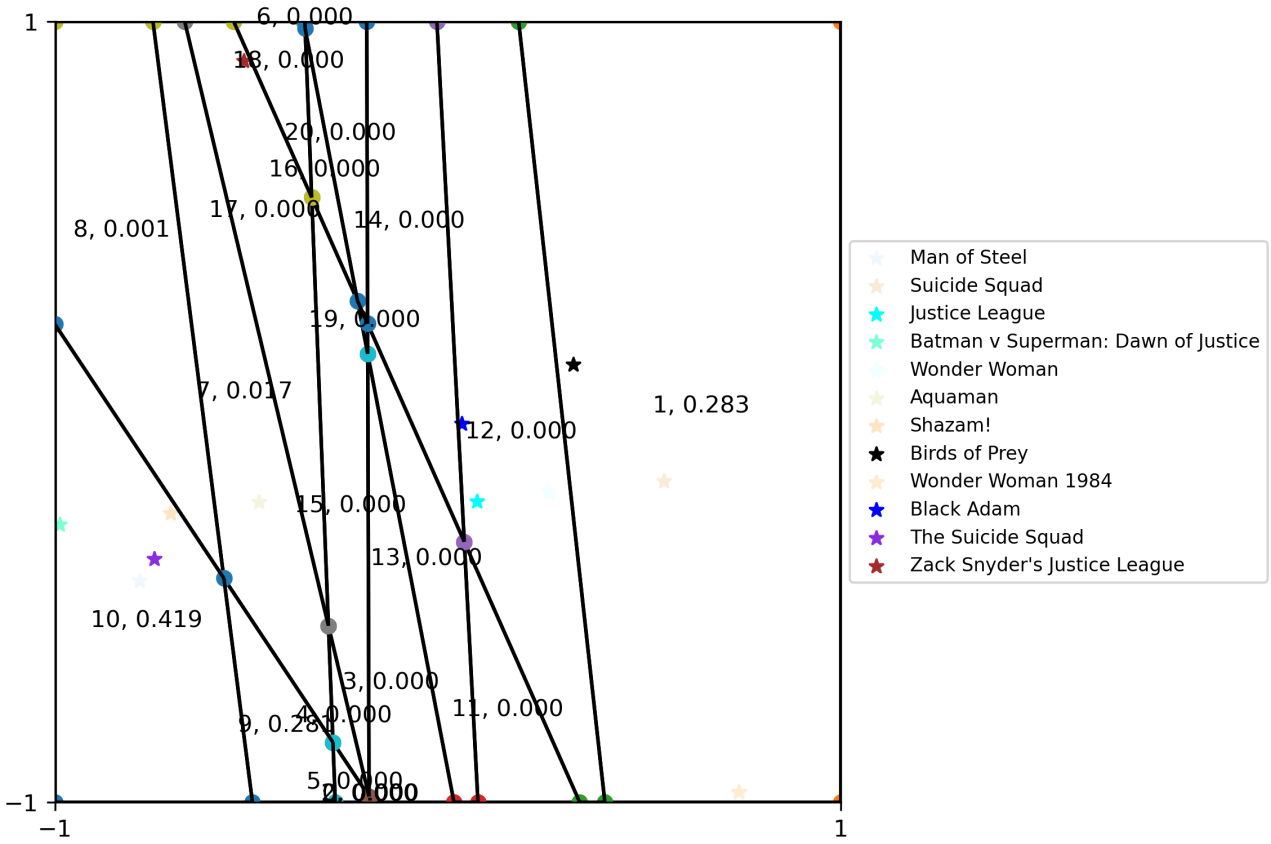

Figure 60: Regions formed by the hyperplanes from DCEU movie set. The numbers in each region represent the region ID as well as the probability mass recovered by our method in that region. Movies in the DCEU movie set are also labeled using their corresponding embedding.

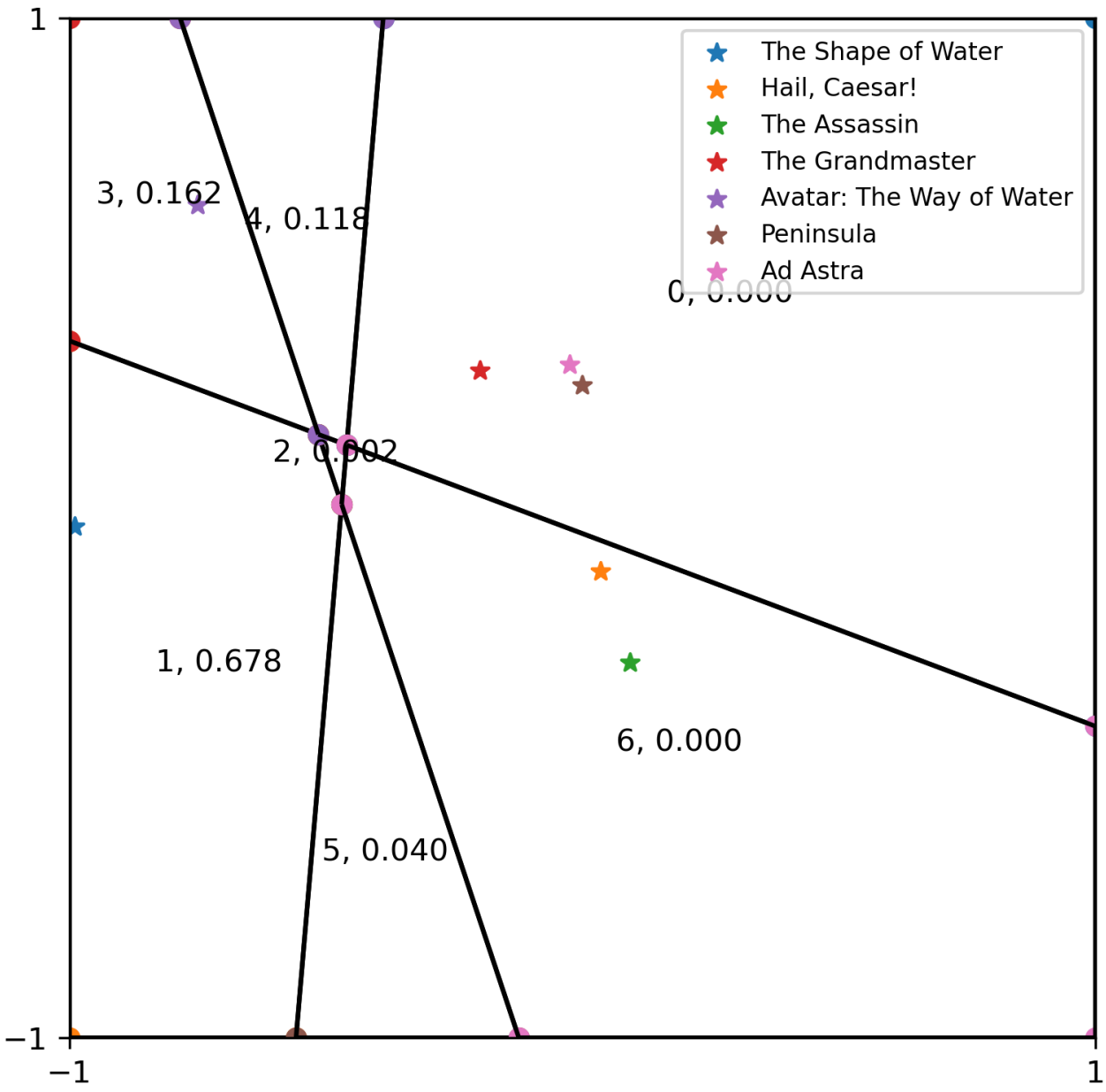

Figure 61: Regions formed by the hyperplanes from Movie2 movie set. The numbers in each region represent the region ID as well as the probability mass recovered by our method in that region. Movies in the Movie2 movie set are also labeled using their corresponding embedding.