Near-Neighbor Methods in Random Preference Completion

Ao Liu
Department of Computer Science
Rensselaer Polytechnic Institute
Troy, NY 12180, USA
liu6@rpi.edu

Qiong Wu, Zhenming Liu
Department of Computer Science
College of William and Mary
Williamsburg, VA 23187, USA
qwu05@email.wm.edu and zliu@cs.wm.edu

Lirong Xia
Department of Computer Science
Rensselaer Polytechnic Institute
Troy, NY 12180, USA
xia1@cs.rpi.edu

Abstract
This paper studies a stylized, yet natural, learning-to-rank problem and points out the critical incorrectness of a widely used nearest neighbor algorithm. We consider a model with $n$ agents (users) $\{x_i\}_{i \in [n]}$ and $m$ alternatives (items) $\{y_j\}_{j \in [m]}$, each of which is associated with a latent feature vector. Agents rank items nondeterministically according to the Plackett-Luce model, where the higher the utility of an item to the agent, the more likely this item will be ranked high by the agent. Our goal is to identify near neighbors of an arbitrary agent in the latent space for prediction.

We first show that the Kendall-tau distance based kNN produces incorrect results in our model. Next, we propose a new anchor-based algorithm to find neighbors of an agent. A salient feature of our algorithm is that it leverages the rankings of many other agents (the so-called “anchors”) to determine the closeness/similarities of two agents. We provide a rigorous analysis for one-dimensional latent space, and complement the theoretical results with experiments on synthetic and real datasets. The experiments confirm that the new algorithm is robust and practical.

1 Introduction
In a learning-to-rank problem, there is a set of agents (users) $X = \{x_1, \ldots, x_n\}$ and a set of alternatives (items) $Y = \{y_1, \ldots, y_m\}$. Each agent reveals her preferences over a subset of alternatives. The goal is to infer agents’ preferences over all alternatives, including those that are not rated or ranked. This fundamental machine learning problem has many practical applications. For example, recommenders systems use an agent’s revealed preferences to discover other alternatives she might be interested in; product designers learn from consumers’ past choices to estimate the demand curve of a new product; defenders can predict terrorists’ preferences based on their past behavior; and political parties can evaluate campaign options based on voters’ preferences. See (??) for a recent survey.

Rating vs. ranking. Agents’ preferences can be represented by either a rating for each alternative (e.g., an integer rating in Netflix), or a ranking over the alternatives (i.e., complete ordering). Rating-based approaches have many known drawbacks (??), including (i) agents often have different scales for ratings; and (ii) numeric values are often less robust than ranking-based approaches. In fact, rating data can always be converted to ranking data (e.g., $y_1$ is ranked higher than $y_2$ if $y_1$ has a higher rating) and thus ranking-based models and algorithms are more general. We focus on ranking data.

A common approach to infer an agent’s preference is to first identify near neighbors of the agent in terms of the Kendall-Tau (KT) distance, then aggregate their rankings to produce a prediction. The KT distance is a metric that counts the number of pairwise disagreements between two ranking lists. This approach was proposed by ?? (??), and their algorithm will be referred to as KT-kNN in this paper. Many subsequent work are based on the following assumption (??, ??, ??, ??, ??).

Assumption 1. KT distance is a good measure of similarity between agents.

No theoretical justification for this assumption was known until recently. ?? (??) proposed a latent utility model to justify the assumption. In their model, each agent or alternative is associated with a latent feature. Alternative $j$’s utility to agent $i$ is controlled by a deterministic function of the similarity in their latent features. Under this model, consistency result is established for the KT-kNN algorithms.

However, this model assumes that agents’ preferences are deterministic, which is unrealistic in many settings. For example, an agent can exhibit irrational behavior, or provide only a noisy version of her preferences. In fact, human preferences are often highly non-deterministic. Various statistical models have been built to model such randomness, pioneered by the Nobel Laureate McFadden (??) among many other researchers. Therefore, the following question remains open.

How can we learn an agent’s random preferences from other agents’ random preferences?

This question can be answered by designing algorithms for two closely-related problems: (i) preference completion (PC): given each agent’s preferences over a subset of alternatives, the goal is to estimate its preference over all the alternatives. (ii) near neighbors (NN): given an agent $x_i$, the goal is to find agents $x_j$ close to $x_i$ in the latent space.

Standard techniques exist to use algorithms for the NN problem to solve a PC problem (??, ??); see also Appendix D. Therefore, we focus on the NN problem in this paper.
Our Contributions. Our main conceptual contribution is the combination of a distance-based latent model and random preferences for learning to rank. To the best of our knowledge, while there is a large literature in each component, we are the first to consider both. See related work for more discussions.

Our model is called distance-based random preference model. Let the latent feature of agent \( i \) (alternative \( j \)) be \( x_i, (y_j) \). Agent \( i \)'s preferences are determined by a utility function \( u(x_i, y_j) = \theta(x_i, y_j) + \epsilon_{i,j} \), where \( \theta(x_i, y_j) \) is a deterministic monotonically decreasing distance-based function and \( \epsilon_{i,j} \) is a zero mean independent random variable. Our model captures two pervasive characteristics of ranking datasets: Ch1. Economically meaningful \( \theta(\cdot, \cdot) \) function. \( u(x_i, y_j) \) is high in expectation when \( x_i \) and \( y_j \) are close. An agent is more likely to prefer alternatives with similar latent features to itself. Ch2. Random preference model. The function \( u(x_i, y_j) \) contains a noise term \( \epsilon_{i,j} \) to capture uncertainties in agents’ behaviors.

Our technical contributions are two-fold. First, we prove that Assumption 1 does not hold anymore in our distance-based random preference model. More precisely, we prove that the agents found by the KT-kNN algorithm (7) is far away from the given agents with high probability, even when \( n, m \to \infty \).

Second, we design an “anchor-based” algorithm for finding an agent’s near neighbors under random preferences. The algorithm is based on the following natural idea: if two agents \( i_1 \) and \( i_2 \) are close, then their KT distance to any other agent \( j \) (an anchor) should also be close. The algorithm proceeds by using the KT distance to other agents as an agent’s feature, and measures the closeness between two agents by the \( L_1 \) distance of their features. We prove that asymptotically our algorithm identifies an agent’s near neighbors with high probability when the latent space is 1-dimensional. Many techniques we developed can be generalized to high-dim settings.

Experiments on synthetic data verify our theoretical findings, and demonstrate that our algorithm is robust in high-dim spaces. Experiments on Netflix data shows that our anchor-based algorithm is superior to the KT-kNN algorithm and a standard collaborative filter (using the cosine-similarities to determine neighbors).

Related Work and Discussions. While using random utility models in learning-to-rank problems is not new \((?; \ ?; \ ?; \ ?; \ ?; \ ?; \ ?; \ ?)\), we are not aware of any that simultaneously achieves both Ch1 and Ch2.

Random utility-based ranking algorithms \((?; \ ?; \ ?; \ ?; \ ?; \ ?; \ ?)\) address Ch2, but the function \( \theta(x_i, y_j) \) often does not have an explicit economics interpretation. For example, let \( \Theta \in \mathbb{R}^{n \times m} \) be a matrix such that \( \Theta_{i,j} = \theta(x_i, y_j) \). \((?; \ ?; \ ?)\) assume that \( \Theta \) is low rank. But the low rank assumption does not have explicit economically interpretation.

While recent non-parametric models \((e.g., (7))\) allow one to use economically interpretable functions \( \theta \) (addressing Ch1), they operate only under deterministic utility models.

Parametric preference learning has been extensively studied in machine learning, especially learning to rank \((?; \ ?; \ ?; \ ?; \ ?; \ ?; \ ?)\). These works are different with ours as it is often assumed that agents’ preferences are generated from a parametric model.

2 Preliminaries

Distance-Based Random Preference Model. Let \( \mathcal{X} = \{x_1, \ldots, x_n \} \subset \mathbb{R}^d \) denote the set of agents and let \( \mathcal{Y} = \{y_1, \ldots, y_m \} \subset \mathbb{R}^d \) denote the set of alternatives. We slightly abuse the notation and use \( x_i \) to refer to both agent \( i \) and her latent features. Each agent \( x_i \) has a ranking (preference list) \( R_i = [y_{i_1} \succ \cdots \succ y_{i_m}] \) over \( \mathcal{Y} \), where \( \succ \) means “prefer to”. We observe only a subset of \( R_i \) for each \( i \in [n] \).

Utility functions and the random utility model. Agent \( i \)'s expected utility on alternative \( j \) is determined by a function \( \theta(x_i, y_j) \). Throughout this paper, we use \( \theta(x_i, y_j) = \exp(-\|x_i - y_j\|_2) \), where \( \| \cdot \|_2 \) is the 2-norm.

Agent \( i \)'s ranking \( R_i \) is determined by the widely-used Plackett-Luce model \((?; ?)\). The realized utility of alternative \( y_j \) for agent \( i \) is generated by \( u(x_i, y_j) = \theta(x_i, y_j) + \epsilon_{i,j} \), where \( \epsilon_{i,j} \) is a zero mean independent random variable that follows the Gumbel distribution. Then, agent \( i \) ranks the alternatives in decreasing order of their realized utilities. The density function of the Plackett-Luce model has a closed-form formula. Let \( y_{j_1} \succ y_{j_2} \) represent that \( y_{j_1} \) is ahead of \( y_{j_2} \) in \( R_i \) and let \( j_1, j_2, \ldots, j_m \) be a permutation of \([m]\). We have

\[
\Pr \left[ y_{j_1} \succ \cdots \succ y_{j_m} \right] = \frac{\prod_{i=1}^{m} \theta(x_{j_i}, y_{j_i})}{\sum_{r \vdash \{1, 2, \ldots, m\}} \prod_{i \in r} \theta(x_{j_i}, y_{j_i})}. \tag{1}
\]

The marginal distribution between alternatives \( j_1 \) and \( j_2 \) is

\[
\Pr \left[ y_{j_1} \succ y_{j_2} \right] = \frac{\theta(x_{j_1}, y_{j_1}) + \theta(x_{j_2}, y_{j_2})}{\theta(x_{j_1}, y_{j_1}) + \theta(x_{j_2}, y_{j_2}) + \epsilon_{j_1,j_2}}.
\]

Distributions of \( \mathcal{X} \) and \( \mathcal{Y} \). \( x_i \) and \( y_j \) are i.i.d. generated from distributions \( \mathcal{D}_X \) and \( \mathcal{D}_Y \). The supports of \( \mathcal{D}_X \) and \( \mathcal{D}_Y \) are a cube \( \mathbb{B}(d) \) in \( \mathbb{R}^d \), where \( \mathbb{B}(d) = \{v \in \mathbb{R}^d : \|v\|_\infty \leq c\} \), where \( c \) is a constant. We adopt the standard “near uniform” assumption for \( \mathcal{D}_X \) and \( \mathcal{D}_Y \) \((?; ?; ?; ?)\).

Definition 1. Consider a continuous distribution \( \mathcal{D} \) on \( \mathbb{B}(d) \) with probability density function \( f_\mathcal{D}(x) \). \( \mathcal{D} \) is near-uniform if \( \sup_{x \in \mathbb{B}(d)} f_\mathcal{D}(x) \) is bounded by a constant, where \( x \in \mathbb{B}(d) \).

Let \( f_X \) and \( f_Y \) be the PDFs of \( \mathcal{D}_X \) and \( \mathcal{D}_Y \) respectively. Define 

\[
\epsilon_X = \frac{\sup_{x \in \mathbb{B}(d)} f_X(x)}{\inf_{x \in \mathbb{B}(d)} f_X(x)} \text{ and } \epsilon_Y = \frac{\sup_{x \in \mathbb{B}(d)} f_Y(x)}{\inf_{x \in \mathbb{B}(d)} f_Y(x)}.
\]

Observation model. We observe only agent \( i \)'s ranking over a subset \( \mathcal{O}_i \subseteq [m] \) of alternatives. Each alternative \( j \) is in \( \mathcal{O}_i \) independently with probability \( p \). The \( \mathcal{O}_i \)'s are also independently generated across different agents. Let \( R^O \) be the ordered list over \( \mathcal{O}_i \subseteq \mathcal{Y} \) that is consistent with \( R \) (i.e., \( R^O \) is the partial ranking of \( R \) over \( \mathcal{O}_i \)). For each agent \( i \), we observe \( R^O_i \).

The near neighbor problem. Here, an algorithm needs to find near neighbors of an input agent. An algorithm is a \( k(n, m, \epsilon) \)-NN solver with parameter \( \tau(n) \) if

- for any input agent \( i \), the algorithm outputs \( k \) agents \( i_1, i_2, \ldots, i_k \), and
- with overwhelming probability, \( |x_i - x_{i_k}| \leq \tau(n) \), where \( \tau(n) = o(1) \).
1. Input: \( \{ R^O_j \}_{j \in [n]}, k, \text{ and an agent } x_i \).
2. Output: \( k \) neighbors near agent \( i \) in the latent space.
3. Find \( j_1, j_2, \ldots, j_n \in [n]/\{i\} \) such that
   \[
   \text{NK}(R^O_{j_1}, R^O_{j_2}) \leq \cdots \leq \text{NK}(R^O_{j_{n-1}}, R^O_{j_n})
   \]
4. Return \( x_{\text{KNN}} \) ← \( \{ j_1, \ldots, j_k \} \).

**Algorithm 1: KT-kNN (it produces incorrect results)**

We often write \( k\text{-NN} \) or \( k\text{NN} \) instead of \( k(n,m)\text{-NN} \) when \( k \)'s dependencies on \( m \) and \( n \) are not critical.

**Additional notations and examples.** For an arbitrary ordered list \( R \), we use it calligraphic form \( \mathcal{R} \) to extract the rank of an alternative. For example, suppose \( y_j \) is the top-ranked alternative in \( R \), then \( \mathcal{R}(y_j) = 1 \). Let \( I(v) \) be an indicator that sets to 1 if the argument \( v \) is true; if false, it sets to 0.

Let \( |R| \) be the length of the list \( R \). Let \( R_1 \) and \( R_2 \) be two ordered lists over the same set of alternatives. The normalized Kendall-Tau distance between \( R_1 \) and \( R_2 \) is

\[
\text{NK}(R_1, R_2) = \frac{1}{\binom{|R_1|}{2}} \sum_{j_1 \neq j_2 \in R_1} I(|\mathcal{R}(j_1) - \mathcal{R}(j_2)|)
\]

(2)

When \( R_1 \) and \( R_2 \) do not have the same support, the normalized KT distance is defined as \( \text{NK}(R^O_1, R^O_2) \), where \( \mathcal{O} = R_1 \cap R_2 \).

To facilitate analysis, sometimes we need to introduce new agents outside \( \mathcal{X} \). For a new agent with latent features \( x \), let \( \mathcal{R}_x \) denote its ranking over \( \mathcal{Y} \) and let \( R^O_x \) denote the observed ranking.

**Conditional probability and expectations.** There are multiple levels of randomness for producing the rankings \( R_i \): (i) \( x_i \) and \( y_j \) are random and (ii) \( u(x_i, y_j) \) consists of a random component (i.e., randomness from the Plackett-Luce model). Care must be taken when operating the conditional random variables defined in our process. For example,

- \( \mathbb{E}[\text{NK}(R_{i_1}, R_{i_2}) \mid \mathcal{X}] \) refers to fixing the latent positions of the agents and taking expectations over \( \mathcal{Y} \) and randomness from the Plackett-Luce model.
- \( \mathbb{E}[\text{NK}(R_{i_1}, R_{i_2}) \mid \mathcal{X}, \mathcal{Y}] \) refers to fixing the latent positions of both alternatives and agents and taking expectations over randomness from the Plackett-Luce model.

3. **Inefficacy of KT-kNN**

In this section, we will prove the inefficacy of KT-kNN algorithm by \( ? \) \( (? \) (Algorithm 1) in our distanced-based random preference model. This implies that Assumption 1 does not hold in our model.

Recall that KT-kNN uses KT distances to find an agent’s neighbors based on the intuition that when \( x_i \) and \( x_j \) are close, their “opinion” on alternatives’ utilities should be similar. The next theorem show that this intuition does not hold in our model, by proving that KT-kNN does not return any near neighbors for a large fraction of \( x_i \).

**Theorem 1.** Consider Algorithm KT-kNN under distance-based random preference model in which \( d = 1 \) and \( p = 1 \). Let \( \mathcal{D}_Y \) and \( \mathcal{D}_X \) be uniform distributions on \( [-1, 1] \). For any constant \( \epsilon \), any \( x_i \in [-1 + \epsilon, -0.5] \cup [0.5, 1 - \epsilon] \), and any \( k \leq n/\ln^5 n \), we have

\[
\min_{x \in \text{KNN(R}_i^{O_j \cup k, x_i})} \|x - x_i\|_2 \geq \epsilon = \Theta(1).
\]

with high probability. The probability comes from random \( \mathcal{X}/\{x_i\} \), random \( \mathcal{Y} \), and random preferences.

**Remarks.** Theorem 1 states that KT-kNN fails to work even for the simple case where \( d = 1 \) and \( \mathcal{D}_X = \mathcal{D}_Y = \text{Uniform}([-1, 1]) \). Eq. (3) is a strong result because trivial algorithms exist to find an agent \( x_i \) whose distance to \( x_i \) is \( \Theta(1) \) (just picking up an arbitrary \( x_i \)). In addition, this result continues to hold for large populations (e.g., when \( n, m \rightarrow \infty \) and \( p = 1 \)), suggesting that the limitation of the KT-based approach roots at the structural properties of the NK function. In addition, if we use KT-kNN to solve PC problem by applying standard techniques, it will also produce poor results (see Appendix D and Lemma 8 there).

**Comparison to \( \Omega \).** \( ? \) \( (2008) \) proved that KT-kNN is effective under the deterministic utility model. This suggests that with the presence of uncertainties in the utility function (a more realistic assumption), the algorithmic structure of the NN problem is significantly altered.

**Intuitions behind Theorem 1.** The following example highlights the salient structures of KT-distances.

**Example 1 (Near-neighbors in expectation).** Let \( x_1 = 0, y_1 = -0.5, \) and \( y_2 = 1 \). Let \( x^* = \arg \min_x \mathbb{E}[\text{NK}(R_x, R_1) \mid x_1, x, \mathcal{Y}] \) (e.g., where would we place an agent that minimizes its KT distance to \( x_1 \) ?). One would hope that when \( x^* \) and \( x_1 \) are close, \( R_{x^*} \) and \( R_1 \) is close, but here \( x^* = -0.5 \). Specifically, let \( a \) be the probability \( x_1 \) prefers \( y_1 \) to \( y_2 \) (i.e., \( a = \frac{\theta(x_1, y_1)}{\theta(x_1, y_1) + \theta(x_1, y_2)} \)) and let \( b(x) \) be the probability \( x \) prefers \( y_1 \) to \( y_2 \). Recall that agents’ support is \([1, 1]\). We need to solve

\[
x^* = \arg \min_x \mathbb{E}[\text{NK}(R_x, R_1) \mid x_1, x, \mathcal{Y}]
= \arg \min_x a(1 - b(x)) + (1 - a)b(x).
\]

Here, we aim to minimize the weighted sum of \( a \in [0, 1] \) and \((1 - a) \) via controlling \( b(x) \). The optimal solution has a simple structure: when \( a > (1 - a) \) (equivalently, \( a > 0.5 \)), we need to set the weight associated with \( a \) as small as possible, which means setting \( b(x) \) to the largest possible value. When \( a < (1 - a) \), \( b(x) \) needs to be minimized. Thus, the optimal solution uses the following threshold rules (assume \( a \neq 0.5 \) for simplicity).

\[
x^* \in \begin{cases} (-1, y_1) & \text{if } a > 0.5 \\ (y_2, 1) & \text{if } a < 0.5 \end{cases}
\]

This minimizer is far from \( x_1 \). \( \square \)

See also Example 3 in Appendix B.5 for another small and concrete example, in which KT-kNN produces poor output.
3.1 Proof sketch of Theorem 1

We use intuitions from Example 1 to prove the theorem. Specifically, define

\[ G_i(x) \equiv \mathbb{E}[\text{NK}(R_i, R_x) \mid x, i, x]. \]

First, note that NK(R_i, R_x) concentrates at G_i(x) when m is sufficiently large. This comes from the concentration behavior of the NK function:

**Lemma 1.** Let \( \mu = \mathbb{E}[\text{NK}(R_i, R_y) \mid X] = G_i(x_j) \). We have

\[ \Pr \left[ |\text{NK}(R_i, R_y) - \mu| \geq \delta \mu \mid X \right] \leq 4m \exp \left( - \frac{\delta^2 m \mu}{6} \right). \]

See Appendix B.1 for the proof. The terms in NK are not independent terms so we cannot directly apply Chernoff bounds. Our proof uses the combinatorial structure of the NK function to decouple the dependencies among terms. The technique we develop can be of independent interests.

Let \( x^* = \arg \min_x G_i(x) \). Below is our main lemma:

**Lemma 2.** Let \( D_Y \) be uniform distribution on \([-1, 1]\). Let \( x_i \) be any agent in \([-1, -0.5] \). We have

\[ \arg \min_x G_i(x) = -1. \]

Similarly, when \( x_i \in [0.5, 1] \), \( \arg \min_x G_i(x) = 1. \)

For any \( x_i \in [-1, -0.5] \cup [0.5, 1] \), Lemma 1 and Lemma 2 give us:

\[ \min_{i \neq i} \text{NK}(R_i, R_x) \approx \min_{i \neq i} G_i(x_i, i) \approx G_i(-1), \]

where the first approximation comes from the concentration bound of NK and the second approximation comes from the fact that there must exist one agent close to -1 when the number of agent is large. Therefore, all the neighbors produced by KT-kNN are far from \( x_i \) (see Appendix B.4 for a rigorous analysis).

**Proof of Lemma 2.** By linearity of expectation, we have

\[ G_i(x) = \frac{1}{(\begin{smallmatrix} m \\ 2 \end{smallmatrix})} \sum_{i \neq k \neq l} \mathbb{E} \left[ \text{NK} \left( R_i \left( y_{i,1}, y_{i,2} \right), R_x \left( y_{i,1}, y_{i,2} \right) \right) \mid x, i, x \right] \]

\[ = \mathbb{E} \left[ \text{NK} \left( R_i \left( y_{i,1}, y_{i,2} \right), R_x \left( y_{i,1}, y_{i,2} \right) \right) \mid x, i, x \right]. \]

The last equality holds because \( x_i \)'s are i.i.d. samples from \( D_Y \). Define

\[ p_i(y_1, y_2) \equiv \Pr[y_1 > y_2 \mid y_1, y_2, x_i] = \frac{\theta(x, y_1)}{\theta(x, y_1) + \theta(x, y_2)}, \]

\[ p_x(y_1, y_2) \equiv \Pr[y_1 > y_2 \mid y_1, y_2, x] = \frac{\theta(x, y_1)}{\theta(x, y_1) + \theta(x, y_2)}. \]

When the context is clear, we shall refer to \( p_i(y_1, y_2) \) and \( p_x(y_1, y_2) \) as \( p_i \) and \( p_x \), respectively. We have

\[ G_i(x) = \mathbb{E}_{y_1, y_2} \left[ p_i(1 - p_i) + (1 - p_x) p_i \mid x, i, x \right]. \]

One can see that \( G_i(x) \) is a smooth function (the first derivative exists). Our proof consists of three parts. Part 1. When \( x \in (-1, x_i] \), \( \partial G_i(x)/\partial x > 0 \). Part 2. When \( x \in [x_i, -x_i] \), \( G_i(x) - G_i(x_i) > 0 \), and Part 3. When \( x \in [-x_i, 1) \), \( \partial G_i(x)/\partial x > 0 \).

The proof for part 3 is similar to part 1. Proving part 2 is also simpler. Therefore, we focus only on the proof for part 1. Proof for part 2 and 3 is deferred to Appendix B.2.

We now show that when \( x \in (-1, x_i] \), \( \partial G_i(x)/\partial x > 0 \). We have (see Fact 2 in Appendix B.2):

\[ \frac{\partial G_i(x)}{\partial x} = \mathbb{E} \left[ \Phi(y_1, y_2, x, x_i) \mid x, i, x \right], \]

where

\[ \Phi(y_1, y_2, x, x_i) = \frac{\sum_{i \neq j} \Delta_{ij}^{(i)} - \sum_{i \neq j} \Delta_{ij}^{(i)}}{4 \cosh^2 \left( \frac{\Delta_{ij}^{(i)}}{2} \right)}. \]

Here, \( \Delta_{ij}^{(i)} (\Delta_{ij}^{(i)}) \) measures whether \( x_i (x_j) \) is closer to \( y_2 \) or \( y_1 \). Similar to Example 1, they serve as important quantities determining the structure of \( \partial G_i(x)/\partial x \) and therefore the optimal solution.

One can check that \( \Phi(y_1, y_2, x, x_i) = \Phi(y_2, y_1, x, x_i) \). Therefore,

\[ \frac{\partial G_i(x)}{\partial x} = \mathbb{E}_{y_1, y_2} \left[ \Phi(y_1, y_2, x, x_i) \mid x, x_i, (y_1 \leq y_2) \right]. \]

Central to our analysis is carefully partitioning the event \( y_1 \leq y_2 \) into four disjoint (sub)-events. Under each event, the conditional expectation of \( \Phi \) can be computed in a straightforward manner. Specifically, define

- \( \mathcal{E}_1 \): when \( x \leq y_1 \leq y_2 \) or \( y_1 \leq y_2 \leq x \), thus, \( \Pr[\mathcal{E}_1 \mid y_1 \leq y_2] = \frac{x^2 + 1}{2} \),
- \( \mathcal{E}_2 \): when \( y_1 < x \) and \( y_2 \geq 1 + 2x \), thus, \( \Pr[\mathcal{E}_2 \mid y_1 \leq y_2] = -(x + 1)x \),
- \( \mathcal{E}_3 \): when \( y_1 < x \) and \( x < y_2 < 2x - x \), thus, \( \Pr[\mathcal{E}_3 \mid y_1 \leq y_2] = (x + 1)(x - x) \),
- \( \mathcal{E}_4 \): when \( y_1 < x \) and \( 2x - x \leq y_2 \leq 1 + 2x \), thus, \( \Pr[\mathcal{E}_4 \mid y_1 \leq y_2] = \frac{(x + 1)^2}{2} \).

Figure 3(a) in Appendix A visualizes the events to complement the analysis. We now interpret the meaning of these events.

**Event \( \mathcal{E}_1 \).** Event \( \mathcal{E}_1 \) represents the case in which \( y_1 \) and \( y_2 \) are on the same side of \( x \). In this case, any movement of \( x \) without passing \( y_1, y_2 \) will not change the probability that \( y_1 > y_2 \) occurs (i.e., \( \Pr[y_1 > y_2 \mid y_1, y_2, x, \mathcal{E}_1] = \Pr[y_1 > y_2 \mid y_1, y_2, x, x_i, \mathcal{E}_1] \) for any sufficiently small \( \delta \)). Therefore, \( \mathcal{E}_1 \) does not impact \( \mathbb{E}[\Phi] \) and \( \mathbb{E}[\Phi \mid x, x_i, \mathcal{E}_1] = 0 \).

**Event \( \mathcal{E}_2 \).** Under this event, one can check that \( |x_i - y_1| < |x_i - y_2| \) (i.e., \( x_i \) is closer to \( y_1 \) than to \( y_2 \)). In this case, when we increase the value of \( x \) (recall that \( y_1 < x < x_i \)), NK(R_i y_1 y_2, R_x y_1 y_2) will increase. This is equivalent to \( \mathbb{E}[\Phi \mid \mathcal{E}_2, x, x_i] > 0 \).

**Event \( \mathcal{E}_3 \).** Under this event, one can check that \( |x_i - y_1| > |x_i - y_2| \). In this case, when we increase the value of \( x \), NK(R_i y_1 y_2, R_x y_1 y_2) will decrease. This is equivalent to \( \mathbb{E}[\Phi \mid x, x_i, \mathcal{E}_3] < 0 \).

**Event \( \mathcal{E}_4 \).** Under this event \( |x_i - y_1| - |x_i - y_2| \) is positive (we call this a positive event) with probability 0.5 and is negative.
(we call this a negative event) with probability 0.5. The first order term conditioned under the positive event cancels out that conditioned under the negative event. Therefore, $E[\Phi | x, x_i, E_4]$ will be a "small" term.

With the above intuition, we have

$$E[\Phi | x, x_i] \approx E[\Phi | x, x_i, E_4] \Pr[E_2 | x, x_i] + E[\Phi | x, x_i, E_3] \Pr[E_3 | x, x_i].$$

We now relate $E_2$ and $E_3$ to Example 1. Event $E_2$ corresponds to the setting where $x_i$ is closer to $y_i$ than to $y_2$. As explained in Example 1, when we move $x$ to the right, we increase the expected KT distance, whereas in $E_3$ when we move $x$ to the right, we decrease the expected KT distance. The crucial observation is that $E_2$ is much more likely to happen than $E_3$. The observation becomes clear with visualization in Fig. 3b (i.e., the area for $E_2$ is much larger than that for $E_3$). Using these intuitions, we have (see Appendix B.2 for the full analysis):

**Fact 1. Using notations above, we have**

$$E[\Phi | x, x_i] = \sum_{t=2,3,4} E[\Phi | x, x_i, E_t] \Pr[E_t | x, x_i] > 0.$$  

This completes the proof of Lemma 2.

## 4 Anchor-based nearest neighbor algorithms

This section develops a new (high-dimensional) feature $F_i$ for each agent $i$ so that $|F_i - F_j|$ is small if and only if $x_i$ and $x_j$ are close. We present the positive results in the most general form in 1-dim latent space (i.e., we allow $D_x$ and $D_Y$ to be any near-uniform distribution, $p = o(1)$, and the latent space is $[-c, c]$ for any constant $c$).

**Index convention.** This section uses $i$, $j$, and $t$ to index agents and $\ell$ (include $\ell_1, \ell_2$, etc.) to index alternatives.

**Intuition of the design of $F_i$.** Our key idea is to leverage a third agent, namely an anchor agent, to determine the closeness of two agents $x_i$ and $x_j$. Let $x_1$ be a third agent. Compute $NK(R_{i1}, R_{j1})$ and $NK(R_{i1}, R_{j1})$. If $x_i$ were chosen appropriately, then $NK(R_{i1}, R_{j1}) \approx NK(R_{i1}, R_{j1})$ if and only if $x_i$ and $x_j$ are close. For example, if $x_i = -c$, then $NK(R_{i1}, R_{j1}) \approx G_r(x_1)$ and the function $G_r(x)$ is a monotone function (see the blue curve in Fig. 1a). We have $G_r(x_1) \approx G_r(x_1)$ if and only if $x_i \approx x_j$.

We face two key technical challenges: $C1$. Not all agents can be served as effective anchor agents. For example, when $x_0 = 0, G_r(x) = G_r(-x)$, which cannot separate $x_i$ and $x_j$ when $x_i = -x_j$. C2. The efficacy of the anchor agent is sensitive to $x_i$ and $x_j$. For example, when $x_i = -c$, the function $G_r(x)$ grows in a sub-linear manner so it is less effective in detecting the closeness of $x_i$ and $x_j$ when they are both close to $-c$.

To address C1, we use all possible agents as anchor agents. To address C2, we need to develop new probabilistic techniques to analyze $G_r(x)$.

**Our features.** Let $F_{i1} = (F_{i1,1}, F_{i1,2}, \cdots, F_{i1,n})$, where

$$F_{i,t} \equiv \mathbb{E}_{R_t, R_t, y}[NK(R_{i1}, R_{j1}) | X] = G_r(x_i).$$

Then we use the $L_1$-distance between $F_{i1}$ and $\hat{F}_{i1}$ to determine whether $x_i$ and $x_j$ are close. Define

$$D(x_i, x_j) = \frac{1}{n-2} \sum_{t \notin \{i,j\}} |F_{i,t} - \hat{F}_{j,t}|. \quad (5)$$

The summation excludes $i$ and $j$ because $x_i$ and $x_j$ themselves cannot be used as anchor agents.

**Replacing the features $F$ by estimates.** $F_{i,t}$ is not directly given so we use the empirical estimate as the plug-in estimator. Define $\hat{F}_{i,t} = NK(R_{i1}, R_{j1})$ and our estimator is $\hat{D}(x_i, x_j) = \frac{1}{n-2} \sum_{t \notin \{i,j\}} |\hat{F}_{i,t} - \hat{F}_{j,t}|$ (see Algorithm 2).

**Algorithm 2: Anchor-kNN**

1. **Input:** $\{R_{j1}\}_{j \in [n]}, k$, and agent $x_i$.
2. **Output:** $k$ neighbors near agent $i$.
3. Compute $\hat{F}_{i,j} = NK(R_{i1}, R_{j1})$ for all $i, j \in [n]$.
4. Compute $\hat{D}(x_i, x_j) = \frac{1}{n-2} \sum_{t \notin \{i,j\}} |\hat{F}_{i,t} - \hat{F}_{j,t}|$.
5. Find $j_1, j_2, \cdots, j_{n-1}$ ($\in [n] \setminus \{i\}$) such that $D(x_i, x_{j_1}) \leq \hat{D}(x_i, x_{j_2}) \cdots \leq \hat{D}(x_i, x_{j_{n-1}})$.
6. Return $\mathcal{A}_{\text{Anchor-kNN}} \leftarrow \{j_1, \cdots, j_k\}$.

**Theorem 2.** Consider Algorithm Anchor-kNN under distance-based random preference model. Let $D_X$ and $D_Y$ be any near-uniform distribution on $[-c, c]$. Let $\tau(n)$ be an arbitrary quality parameter so that $\tau(n) = o(1)$ and $\tau(n) = \omega\left(n^{-\frac{1}{2}} \sqrt{n} \right)$. For any $x_i \in [-c, c]$, any $m = \omega\left(n^{\frac{3}{2}} \sqrt{n} \cdot \ln^2 \left(\frac{n^{\frac{3}{2}} \ln n}{\tau(n)}\right)\right)$ and any $k = o(n \cdot \tau^2(n))$.

---

**Figure 1:** (a) Functions $G_r(x)$ for $c = 1$ and $x_i = -1, 0.4$, and 1. Observations: (i) when $x_i = \pm 1$, the function $G_r(x)$ is a monotone function; (ii) for all $x_i$, $G_r(x)$ grows in sublinear manner when $x_i$ is close to $\pm 1$. (b) Definition of $I_1, I_2$ and $I_3$ used in the lower bound proof for Lemma 3 (Section 4). Agents in $I_1$ and $I_3$ are effective anchor agents. (c) Intuition of $E_2$ and $E_3$ in the proof of Lemma 4 (Section 4).
\[ \max_{x \in \text{Anchor} \cap n(N((R^\delta_i), k, x_i))} ||x - x_i||_2 \leq \tau(n) \]

with high probability. The probability comes from random \( X \\{x_i\} \), random \( Y \), and random preferences.

**Remark.** \( \tau(n) \) cannot be too small because our function \( D(x_i, x_j) \) cannot measure the distance among two agents well if they are too close. \( m \) needs to grow when \( p \) (fewer samples) or \( \tau(n) \) decreases (higher quality requirement), which is intuitive. \( k \) measures the number of numbers an algorithm can find so that their distance is within \( \tau(n) \); larger \( k \) means the algorithm is more powerful.

Let \( D(x_i, x_j) = \mathbb{E}[D(x_i, x_j)] \). In the remainder of this section, we analyze the behavior of \( D \). The function \( D \) concentrates at \( D \) and can be shown by using simple Chernoff bounds (see Appendix C.2 for a complete analysis).

**Lemma 3.** For any near-uniform \( D_X, D_Y \) on \([-c, c]\) and any two agents \( x_i, x_j \), we have

\[
c_3(c) \cdot (\ln^{-1} n) \cdot |x_i - x_j|^2 \leq D(x_i, x_j) \leq |x_i - x_j|,
\]

where \( c_3 \) is a constant that depends only on \( c \).

**Upper bound proof for Lemma 3.** The upper bound requires only a straightforward calculation. Recall that \( p_i = p_{1i} + p_{2i} \) is \( \mathbb{E}[y_i \mid y_1, y_2, x_i] \). We have

\[
D(x_i, x_j) = \mathbb{E}_{x_i, x_j} \left[ \mathbb{E}_{y_1, y_2} \left[ |p_i - p_j| \cdot \mathbb{I}[|x_i - x_j| < 1] \right] \right] \leq \mathbb{E}_{y_1, y_2} \left[ |p_i - p_j| \cdot \mathbb{I}[|x_i - x_j| < 1] \right].
\]

The last inequality is shown in Fact 6 in Appendix C.1.

**Lower bound proof for Lemma 3.** Here we analyze only the case \( |x_i - x_j| < 2c - 2\ln n \). When \( |x_i - x_j| > 2c - 2\ln n \) (e.g., \( x_i \) and \( x_j \) are around the boundaries \(-c \) and \( c \), respectively), the result is trivial.

Wlog, assume that \( x_i < x_j \). We partition \([-c, c]\) into three intervals and consider anchor agents in each of these intervals. Specifically, define (see also Figure 1(b))

\[
I_1 = \left[-c, \frac{x_i - c}{2}\right], \quad I_2 = \left[\frac{x_i - c}{2}, \frac{x_i + c}{2}\right], \quad I_3 = \left[\frac{x_i + c}{2}, c\right].
\]

The agents in \( I_2 \) are “less effective anchors” (C2). We use trivial bound for terms in \( I_2 \) (\( \mathbb{E}_{I_d} - F_{I_d, I} \geq 0 \)). Focusing on \( I_1 \) and \( I_3 \), we have

\[
D(x_i, x_j) \geq \frac{x_i + c}{4c_e} \cdot \mathbb{E}_{x_i} \left[ |G_t(x_i) - G_t(x_j)| \mid x_t \in I_1, x_i, x_j \right] + \frac{c - x_j}{4c_e} \cdot \mathbb{E}_{x_j} \left[ |G_t(x_i) - G_t(x_j)| \mid x_t \in I_3, x_i, x_j \right].
\]

Note that \( \frac{x_i + c}{4c_e} + \frac{c - x_j}{4c_e} \) is at least \( \tilde{O}(1) \). Now we show that \( \mathbb{E}_{x_i} \left[ |G_t(x_i) - G_t(x_j)| \mid x_t \in I_1, x_i, x_j \right] \) is at least in the order of \( |x_i - x_j|^2 \). The analysis for the other term is similar. Below is the lemma we need (related to C2):

**Lemma 4.** For any near-uniform \( D_X, D_Y \) on \([-c, c]\) such that \( |x_i - x_j| \geq 2c - 2\ln n \) and \( x_1 \in I_1 \), we have

\[
|G_t(x_i) - G_t(x_j)| \geq c_4(c) \cdot |x_i - x_j|^2,
\]

where \( c_4(c) = \frac{1 - e^{-c/4}}{1 + e^{-c/4}} \cdot \frac{1}{90c^2 \cdot \cosh^2(c/2)} \in (0, \frac{1}{2c}) \).

**Proof of Lemma 4.** Note that \( |x_i - x_j| \geq 2c - 2\ln n \) and \( x_1 \in I_1 \) imply \( x_i \geq x_1 \geq 2x_i + c \). Because \( |G_t(x_i) - G_t(x_j)| = \int_{x_i}^{x_j} \frac{\partial G_t(x)}{\partial x} \, dx \), we aim to give a bound for \( \frac{\partial G_t(x)}{\partial x} \) specifically.

\[
\frac{\partial G_t(x)}{\partial x} \geq 3c_4(c) \cdot (c^2 - x^2).
\]

when \( x \geq 2x_i + c \) (or equivalently, \( x_i \leq \frac{x - x_i}{2} \)). Re-cycle the definition of \( \Phi, \Delta_{2i}^{(i)}, \) and \( \Delta_{1i}^{(i)} \) used in the analysis of Theorem 1 (see also Appendix A for the notation summary) so that \( \frac{\partial G_t(x)}{\partial x} = \mathbb{E}_{y_1, y_2} \left[ \Phi(y_1, y_2, x_i, x_1) \right] \mid x_i \). We partition the positions of \( \{y_1, y_2\} \) into events (see also Figure 1c for a visualization):

- \( \mathcal{E}_1 \): when \( x_i \leq y_1 \leq y_2 \) or \( y_1 \leq y_2 \leq x \).
- \( \mathcal{E}_2 \): when \( y_1 \geq \frac{|x - 7c|}{8} \) and \( y_2 \geq \frac{x + c}{2} \). We have \( \mathbb{P}[\mathcal{E}_2 \mid y_1 < y_2] \leq \frac{c^2 - x^2}{16c^2} \).
- \( \mathcal{E}_3 \): when \( (y_1, y_2) \notin \mathcal{E}_1 \cup \mathcal{E}_2 \). As explained below, we do not need to explicitly calculate \( \mathbb{P}[\mathcal{E}_3 \mid y_1 \geq y_2] \).

We now explain the intuition associated with these events.

**Event \( \mathcal{E}_1 \).** Because \( y_1 \) and \( y_2 \) are on the same side of \( x \), we have \( \mathbb{E}[\Phi \mid \mathcal{E}_1, x, x_i] = 0 \) (see also Lemma 2).

**Event \( \mathcal{E}_2 \) and event \( \mathcal{E}_3 \).** When \( (y_1, y_2) \in \mathcal{E}_2 \cup \mathcal{E}_3 \), we have \( |y_1 - x| \leq |y_2 - x| \). This is because \( (i) \) \( y_2 - x_i \geq x_i - x \) when \( y_2 \geq x_2 \) and \( x_1 \in I_1 \), and \( (ii) \) \( y_1 - x \leq \max(x_i - c, x - x_i) \leq x_i - x \) since \( y_1 \leq x \) and \( x \geq 2x_i + c \). This conclusion is trivial when the positions of the agents are visualized (Figure 1c).

In these events, an increment in \( x \) will result in an increment in \( G_t(x) \). Therefore \( \mathbb{E}[\Phi \mid \mathcal{E}_2 \cup \mathcal{E}_3, x, x_i] \geq 0 \).

**Event \( \mathcal{E}_2 \).** Knowing \( \Phi \) can be arbitrarily close to 0 when \( \mathcal{E}_2 \cup \mathcal{E}_3 \) happens. Here, we need also identify an event so that \( \Phi \) is at least a positive constant. Event \( \mathcal{E}_2 \) serves for this purpose because it ensures \( x \) to be far from the mid-point of \( y_1 \) and \( y_2 \), so \( G(x) \) behaves like a linear function in this region (and thus its derivative behaves like a constant).

**Intuition on dependencies on \( |x_i - x_j|^2 \).** We now explain why \( |G_t(x_i) - G_t(x_j)| \) depends on \( |x_i - x_j|^2 \) instead of \( |x_i - x_j| \). Only if \( (x_i, x_j) \notin \mathcal{E}_1 \), \( \Phi \) will be non-zero. This requires \( y_1 \leq x \leq y_2 \). Recall we only interests to \( x_i \leq x \leq x_j \). When \( x_i \) and \( x_j \) get too close to \(-c \), \( \mathbb{P}[y_1 < x < y_2 \mid x] = \Theta(x_i + c) \approx \Theta(|x_i - x_j|) \). When we carry out integration over \( \partial G_t(x)/\partial x \), this term will lead to an additional factor of \( |x_i - x_j|^2 \) (see also Figure 1a for simulated \( G_t(x) \)).
Our experiments aim to consider the kNN. Ground-Truth-kNN assumes an oracle access to an 1-dim synthetic data. In this experiment we compare the theoretical world data are an promising direction for future work.

(iii) settings, and understand the behavior of and . Figure 2(a) plots the performance of Anchor KT outperforms KT when . Figure 2(b) plots the completeness performance for different . Figure 2(c) shows the average latent distance between the return set and . The experiments confirm that (i) Anchor-kNN is consistently better, (ii) KT-kNN does not return near neighbors when , and (iii) when KT-kNN is the poorest at completing preferences. Appendix E.1 provides additional experiments for different settings on .

High-dim synthetic data. We repeat the experiments for . Figure 2(d) is the prediction error and Figure 2(e) is the average distance. Anchor-kNN continues to outperform KT-kNN, suggesting that our algorithm works for because of the curse-of-dimensionality problem in neighbor-based algorithms.

Real dataset. We examine the performance of Anchor-kNN using the standard Netflix dataset. The baselines are KT-kNN and a standard collaborative filtering algorithm using cosine-similarity. Table 2 in Supplementary materials presents the results. Anchor-kNN consistently outperforms KT-kNN and collaborative filtering using cosine-similarity. Our experiments suggest that our distance-based random preference model and Anchor-kNN seem to be quite practical.

5 Experiments

Our experiments aim to (i) confirm the behaviors of KT-kNN and Anchor-kNN for finite sample size when , (ii) understand the behavior of Anchor-kNN in high-dim settings, and (iii) validate the practicality of our algorithm over real-world datasets.

Details of all experiments are in Appendix E.1. Note that this is a theoretical paper. Extensive evaluations on real-world data are an promising direction for future work.

1-dim synthetic data. In this experiment we compare the efficacy of KT-kNN, Anchor-kNN, and Ground-Truth-kNN. Ground-Truth-kNN assumes an oracle access to an input’s (ground-truth) k-nearest neighbors so this is an optimal kNN. Figure 2(a) plots the performance of Anchor-kNN, KT-kNN and Ground-truth-kNN in completing the preferences for different . Figure 2(b) plots the completion/prediction error for different using an optimal chosen by using cross-validations for Ground-Truth-kNN). Figure 2(c) shows the average latent distance between the return set and . The experiments confirm that (i) Anchor-kNN is consistently better, (ii) KT-kNN does not return near neighbors when , and (iii) when KT-kNN is the poorest at completing preferences. Appendix E.1 provides additional experiments for different settings on .

6 Conclusion

This paper introduced a natural learning-to-rank model, and showed that under this model a widely used KT-distance based kNN algorithm failed to find similar agents (users), challenging the assumptions made in many prior preference completion algorithms. To fix the problem, we introduced a new anchor-based algorithm Anchor-kNN that uses all the agents’ ranking data to determine the closeness of two agents. Our approach is in sharp contrast to most existing feature engineering methods. We provided a rigorous analysis for Anchor-kNN for 1-dim latent space, and performed experiments on both synthetic and real datasets. Our experiments showed that Anchor-kNN works in high dim space and promises to outperform other widely used techniques.

7 Acknowledgments

We thank all anonymous reviewers for helpful comments and suggestions. AL and LX are supported by NSF #1453542 and ONR #N00014-17-1-2621. QW and ZL are supported by NSF CRII:III #1755769.