Assumption-free stability for ranking problems

Anonymous authors
Paper under double-blind review

Abstract

In this work, we consider ranking problems among a finite set of candidates: for instance, selecting the top-k items from a list or obtaining the full ranking of all items in the set. These problems are often unstable: estimating a ranking from noisy data can exhibit high sensitivity to small perturbations. Concretely, if we use data to assign scores to items (say, by aggregating user preference data), then for two items with similar scores, small fluctuations in the data can alter their relative rankings. Many existing theoretical results sidestep this challenge by assuming a separation condition, but real-world data often contains near-ties, limiting the applicability of existing theory. To address this gap, we develop a new algorithmic stability framework for ranking problems, and propose two novel ranking operators for achieving stability: the *inflated top-k* for the top-k selection problem and the *inflated full ranking* for ranking the full list, each of which allows for expressing some uncertainty in the output. Both our proposed methods provide guaranteed stability, with no assumptions on data distributions and no dependence on the total number of candidates to be ranked. Experiments on real-world data confirm that the proposed methods offer stability while retaining informativeness.

1 Introduction

Ranking tasks arise across diverse machine learning settings, powering applications such as search, recommendation, and decision making. Despite their pervasiveness, instability remains a persistent challenge in many of the ranking algorithms employed in practice—the output ranking list is often sensitive to slight perturbations of the training data, which impairs their performance in real-world applications. For example, such sensitivity can sabotage selection fairness (Dwork et al., 2019; Yang et al., 2018), or degrade user experience in the context of recommendation systems (Anelli et al., 2021; Oh et al., 2022).

In this paper, we develop a unified framework that addresses algorithmic stability in ranking problems by adopting a set-valued perspective on the output that allows for ambiguity in the ranking score assignments, such as near-tie scenarios. We propose methods to solve two problems, top-k selection and full ranking, each leveraging recently developed tools in stable classification (Soloff et al., 2024a) to achieve provable ranking stability in an assumption-free way.

1.1 Problem setting

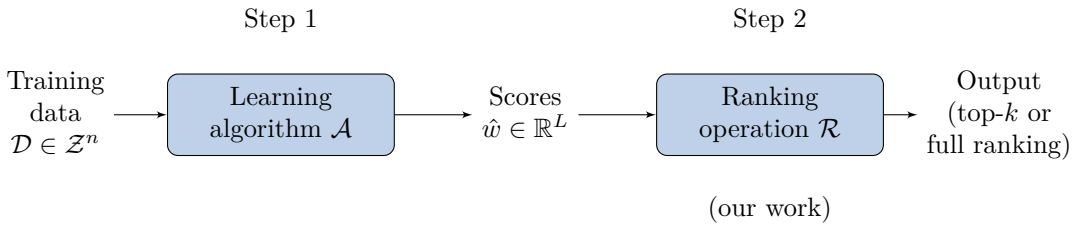
In ranking tasks, we are given a list of L items and a data-based mechanism for estimating the score of each item, and we seek to either select the k items with the top scores (the "top-k problem") or rank the items according to their scores (the "full ranking problem"). The challenge is that small perturbations in the data used to estimate scores can alter the selected set of k items or the order of the ranked items.

Both problems are typically approached via a two-stage process:

Step 1: given a dataset $\mathcal{D}$ containing n samples $Z_1, \ldots, Z_n$, one runs an algorithm $\mathcal{A}$ to learn a vector of *scores* $\hat{w} = (\hat{w}_1, \ldots, \hat{w}_L)$ that assigns scores to the L items;

Step 2: given the scores contained in $\hat{w}$, one either sorts them (ranking) or selects the k items with the largest scores (top-k selection).

For example, for Step 1, in the learning-to-rank (LTR) problem (Liu et al., 2009; Cao et al., 2007), $\hat{w}$ represents the predicted relevance of each item to a query; this may be learned from training data $\mathcal{D}$ using, for instance, a Bradley–Terry model (Bradley & Terry, 1952). As another example, in preference voting problems, $\hat{w}_\ell$ denotes the fraction of votes the ℓ-th item received in the dataset $\mathcal{D}$, where each data value Z_i indicates the choice of the ith voter. In general, we write $\hat{w} = \mathcal{A}(\mathcal{D})$, where $\mathcal{A}$ denotes the learning algorithm mapping a dataset $\mathcal{D} = (Z_1, \ldots, Z_n)$ to the score vector $\hat{w}$. For Step 2, let π denote the permutation on $[L]$ such that $\hat{w}_{\pi(j)}$ is the j-th largest element in $(\hat{w}_1, \cdots, \hat{w}_L)$ for each $j \in [L]$. In top-k selection, typically the k elements with the largest scores are returned: top-$k(\hat{w}) := \{\pi(1), \ldots, \pi(k)\}$. For full ranking, a permutation over $[L]$ based on the sorted scores is produced, i.e., ranking$(\hat{w}) := (\pi(1), \ldots, \pi(L))$. This two-stage process is summarized in the diagram below.



Although the scores $\hat{w}$ obtained in Step 1 are often stable to small perturbations in $\mathcal{D}$ (as in the preference voting setting) or can potentially be stabilized by methods such as bagging (Soloff et al., 2024c), the final output of top-k or ranking can still be highly susceptible to slight changes in the scores. In other words, the stability in the learned scores $\hat{w}$ does not necessarily imply the stability of the induced ranked order, as a ranking operation is intrinsically discontinuous and therefore unstable; the situation is similar for top-k. Indeed, when the scores in $\hat{w}$ are close to each other, even a small perturbation to $\hat{w}$ can lead to drastically different rankings.

In the literature on both full rankings and top-k selection, some degree of separation in the population scores is necessary for identifiability. In this setting, one assumes that $\hat{w}$ is a noisy realization of a "true" set of scores w^*, and theoretical results guaranteeing exact recovery to the true ranking (according to w^*) assume a minimum separation between scores for different items—see, e.g., Chen et al. (2022a;b) and Chen et al. (2019). However, such assumptions are frequently violated in real-world applications. For example, in the Netflix movie rating data, many of the top-rated films have nearly identical average scores, making it difficult to justify any assumption on the minimum separation.

Prior work by Soloff et al. (2024a) studied a special case of top-k selection with $k = 1$ in the context of classification. They proposed the *inflated argmax* operator, which returns a *set-valued* output (i.e., a subset of the candidate items) to account for potential ambiguity in Step 2. In this work, we focus on developing stable versions of top-k and ranking operations for Step 2 that return informative set-valued outputs applicable to general ranking problems. Specifically, in the context of the full ranking problem, our ranking operation would return a set of possible rankings of the L items, where we may return more than one possible ranking if needed to accommodate ambiguities. In the context of the top-k problem, our procedure would return a set of possible top-k items, where the number of items in the set may exceed k if needed. These types of procedures, which allow for ambiguity in the output (i.e., the output essentially corresponds to a *set* of possible answers to the question of full ranking or of top-k), build on works in the literature that propose set-valued or multi-valued estimators for classification problems; see, for example, (Grycko, 1993; Lei, 2014; Chzhen et al., 2021).

1.2 Our contributions

We propose two new notions of algorithmic stability tailored to ranking problems: top-k stability and full ranking stability. As illustrated in the flow chart above, we focus on developing operators to stabilize Step 2 in ranking problems. To this end, we propose the *inflated top-k* and the *inflated full ranking* operations, generalizing the inflated argmax operator of Soloff et al. (2024a). We derive assumption-free stability guarantees for these two operators (Thms. 4 and 9), discuss their connections to the inflated

argmax (Section 5), and show they return minimal sets whenever possible (Props. 5 and 10). Subsequent experiments on both real and synthetic data confirm that our proposed methods exhibit stability while remaining informative.

2 A unified framework for stable ranking

We start by defining notions of stability that are meaningful in the context of the top-k selection problem and the full ranking problem. As discussed above in Section 1.1, in order to enable stability guarantees even when the data might lead to some uncertainty in the ranking, we need to allow for some ambiguity in the output: a procedure for solving the top-k problem might need to return a set of size $> k$, and a procedure for solving the full ranking problem might need to return more than one possible ranking.

We begin with the top-k problem, for any fixed $k \in [L]$. Let $\wp_{\geq k}([L])$ denote the set of subsets of $[L]$ of size $\geq k$ (this is a subset of $\wp([L])$, the power set of $[L]$). We will consider the stability of a function[1]

$$f : \cup_{n \geq 1} \mathcal{Z}^n \to \wp_{\geq k}([L]),$$

which maps a dataset $\mathcal{D} \in \mathcal{Z}^n$ (for any sample size n) to a subset $f(\mathcal{D}) \subseteq [L]$, with $|f(\mathcal{D})| \geq k$. In particular, we expect to have $|f(\mathcal{D})| > k$ if the data exhibits some ambiguity in terms of identifying the top-k items. For any dataset $\mathcal{D} \in \mathcal{Z}^n$, we write $\mathcal{D}^{\setminus i} \in \mathcal{Z}^{n-1}$ to denote this same dataset with the ith point removed—that is, if $\mathcal{D} = (Z_1, \ldots, Z_n)$ then $\mathcal{D}^{\setminus i} = (Z_1, \ldots, Z_{i-1}, Z_{i+1}, \ldots, Z_n)$.

Definition 1 (Stability for top-k selection) *We say that a function $f : \cup_{n \geq 1} \mathcal{Z}^n \to \wp_{\geq k}([L])$ has top-k stability δ at sample size n if*

$$\frac{1}{n} \sum_{i=1}^{n} \mathbf{1} \left\{ \left| f(\mathcal{D}) \cap f(\mathcal{D}^{\setminus i}) \right| \geq k \right\} \geq 1 - \delta \text{ for all } \mathcal{D} \in \mathcal{Z}^n.$$

To explain this definition, we are requiring that the answer returned for a dataset $\mathcal{D}$ and for $\mathcal{D}^{\setminus i}$ should be consistent with each other (for most i) in the sense that their overlap should have size $\geq k$ (since we are seeking to identify the top-k items). In the special case $k = 1$, this definition coincides with the notion of *selection stability* proposed by Soloff et al. (2024a) for the argmax problem (i.e., the problem of identifying the top-ranked item).[2]

Next, we turn to the full ranking problem. For this setting, we will consider functions of the form

$$f : \cup_{n \geq 1} \mathcal{Z}^n \to \wp(\mathcal{S}_L),$$

where $\mathcal{S}_L$ is the set of permutations of $[L]$, and $\wp(\mathcal{S}_L)$ is the power set of $\mathcal{S}_L$. That is, given a dataset $\mathcal{D}$ of any size, $f(\mathcal{D})$ returns a set of permutations, each corresponding to a possible ranking. If $|f(\mathcal{D})| = 1$, this can be interpreted to mean that the data suggests a single ranking, while $|f(\mathcal{D})| > 1$ indicates some ambiguity.

Definition 2 (Stability for full ranking) *We say that a function $f : \cup_{n \geq 1} \mathcal{Z}^n \to \wp(\mathcal{S}_L)$ has full ranking stability δ at sample size n if*

$$\frac{1}{n} \sum_{i=1}^{n} \mathbf{1} \left\{ f(\mathcal{D}) \cap f(\mathcal{D}^{\setminus i}) \neq \varnothing \right\} \geq 1 - \delta \text{ for all } \mathcal{D} \in \mathcal{Z}^n.$$

In other words, the sets of possible rankings $f(\mathcal{D})$ and $f(\mathcal{D}^{\setminus i})$ should be consistent with each other (for most i), in the sense that at least one permutation (i.e., one possible ranking of the L items) should appear in both sets.

[1] In the diagram in Section 1.1, the function f corresponds to $\mathcal{R} \circ \mathcal{A}$.

[2] In many applications, estimation algorithms and ranking algorithms often incorporate some form of randomization—for instance, a random initialization, or, random tie-breaking rules for the ranking procedure. The definitions and results of this paper can be extended naturally to incorporate randomized algorithms, but for conciseness we do not present these extensions here and only consider the deterministic case.

Interpreting these definitions of stability By adopting the set-valued perspective, the current stability guarantees (as in Definitions 1 and 2) are formulated through the notion of compatibility, i.e., by ensuring the existence of at least one consistent ranking output. At first glance, this stability condition may appear quite weak: for instance, we might return two sets each containing many possible rankings, with only one shared ranking. To help interpret this definition, we view it as a minimal baseline: any method that offers a reasonable notion of stability must at least satisfy this condition—and, unfortunately, standard point-valued outputs (such as top-k and full ranking) often fail to satisfy even this minimal condition.

2.1 Stability of the scores, or stability of the ranking?

From this point on, we will focus our attention on procedures that follow the two-stage structure described in Section 1.1, as is common across many applications. Specifically, we will consider functions of the form $f = \mathcal{R} \circ \mathcal{A}$, where $\mathcal{A} : \cup_{n \geq 1} \mathcal{Z}^n \to \mathbb{R}^L$ is a learning algorithm mapping a dataset $\mathcal{D}$ to a vector of scores $\hat{w} = \mathcal{A}(\mathcal{D}) \in \mathbb{R}^L$, and we then apply a ranking procedure $\mathcal{R}$ to produce either an output $\mathcal{R}(\hat{w}) \in \wp_{\geq k}([L])$ (for the top-k problem) or $\mathcal{R}(\hat{w}) \in \wp(\mathcal{S}_L)$ (for the full ranking problem).

As mentioned in Section 1.1, the learning algorithm $\mathcal{A}$ used to learn the scores $\hat{w}$ tends to be stable in many cases, or in other cases, it is easy to modify $\mathcal{A}$ to ensure stability. Formally, we say that $\mathcal{A}$ has (ε, δ)-stability at sample size n if, for all datasets $\mathcal{D} \in \mathcal{Z}^n$,

$$\frac{1}{n} \sum_{i=1}^{n} \mathbf{1} \left\{ \|\hat{w} - \hat{w}^{\setminus i}\| \geq \varepsilon \right\} \leq \delta, \tag{1}$$

where $\hat{w} = \mathcal{A}(\mathcal{D})$ and $\hat{w}^{\setminus i} = \mathcal{A}(\mathcal{D}^{\setminus i})$, and where $\|\cdot\|$ is the usual Euclidean (i.e., ℓ_2) norm on $\mathbb{R}^L$.

This notion of algorithmic stability shares close ties with the formulations appearing in the work of Elisseeff et al. (2005); Soloff et al. (2024b;c); related definitions of stability have previously been considered in the context of generalization bounds and learnability for ranking problems by Lan et al. (2008); Agarwal & Niyogi (2009); Gao & Zhou (2013). We present below two scenarios where the stability condition (1) on $\mathcal{A}$ is achieved in an assumption-free way.

- In the preference voting setting, each participant casts a vote among L candidate items—that is, the data values are $Z_i \in [L]$, denoting the vote from the ith participant. The final ranking scores can then be obtained by simply counting the fraction of votes each item receives, $\hat{w}_\ell = \frac{1}{n} \sum_i \mathbf{1}\{Z_i = \ell\}$, which is (ε, δ)-stable with $\varepsilon = \sqrt{2}/n$ and $\delta = 0$.
- For general learning algorithms $\mathcal{A}$ with bounded outputs, Soloff et al. (2024c) show that one can apply bagging (bootstrapping) with $\mathcal{A}$ as a base algorithm to achieve assumption-free stability, with $\varepsilon^2 \delta \propto 1/n$.

These examples illustrate that, in a two-stage ranking procedure $\mathcal{R} \circ \mathcal{A}$, stability of the learning algorithm $\mathcal{A}$ is often easily achievable—but crucially, this does not necessarily translate to stability in the output ranking list, due to the discontinuity of the ranking operation. For example, in a scenario where participants are voting for their favorite item from two candidate items, suppose that Item 1 receives 51% of the vote while Item 2 receives 49%, but if the ith participant is removed from the vote count then this flips—that is, $\hat{w} = (0.51, 0.49)$ while $\hat{w}^{\setminus i} = (0.49, 0.51)$. Then the perturbation in the output of the learning algorithm $\mathcal{A}$ is equal to $\|\hat{w} - \hat{w}^{\setminus i}\|$, which is small—but the outcome of a ranking procedure $\mathcal{R}$ applied to $\hat{w}$, versus to $\hat{w}^{\setminus i}$, may give completely different answers.

In light of this challenge, our task from this point on is the following:

> We aim to develop a ranking operation $\mathcal{R}$ (for the top-k problem or the full ranking problem) such that, when combined with a learning algorithm $\mathcal{A}$ that is stable as in (1), then the two-stage procedure $\mathcal{R} \circ \mathcal{A}$ is guaranteed to satisfy the appropriate notion of stability (that is, top-k stability as in Defn. 1, or full ranking stability as in Defn. 2).

3 The inflated top-k method

We are now ready to define our proposed ranking procedure for the top-k problem:

Definition 3 (Inflated top-k) *For any $k \in [L]$, any $w \in \mathbb{R}^L$, and any $\varepsilon > 0$, define the ε-inflated top-k ranking as*

$$\text{top-}k^{(\varepsilon)}(w) := \left\{ j \in [L] : \text{dist}(w, C_j^{\varepsilon,k}) < \varepsilon \right\}, \tag{2}$$

where

$$C_j^{\varepsilon,k} = \left\{ v \in \mathbb{R}^L : v_j \geq v_{(k+1)} + \varepsilon/\sqrt{2} \right\}.$$

Here for any set $C \subseteq \mathbb{R}^L$, we define $\text{dist}(w, C) = \inf_{v \in C} \|w - v\|$, and for any vector $v = (v_1, \ldots, v_L) \in \mathbb{R}^L$, we write $v_{(1)} \geq \cdots \geq v_{(L)}$ to denote the order statistics of the values $v_1, \ldots, v_L$.

For the special case $k = 1$, the definition above coincides with the *inflated argmax* proposed by Soloff et al. (2024a)—that is, for $k = 1$, $\text{top-}k^{(\varepsilon)}(w) = \text{argmax}^{(\varepsilon)}(w) := \left\{ j \in [L] : \text{dist}(w, C_j^{\varepsilon,1}) < \varepsilon \right\}$.

To help interpret this definition, we observe that $C_j^{\varepsilon,k}$ is the set of score vectors $v \in \mathbb{R}^L$ for which the jth entry v_j is in the top-k by a positive margin. We can also observe that $\text{top-}k^{(\varepsilon)}(w) \supseteq \text{top-}k(w)$ always holds—that is, the top-k entries are always included in the inflated top-k. Moreover, the inflated top-k is naturally permutation-invariant, satisfying

$$\pi(i) \in \text{top-}k^{(\varepsilon)}\big((w_1, \ldots, w_L)\big) \iff i \in \text{top-}k^{(\varepsilon)}\big((w_{\pi(1)}, \ldots, w_{\pi(L)})\big)$$

for any $w \in \mathbb{R}^L$ and any $\pi \in \mathcal{S}_L$—that is, permuting the entries of w does not alter our assessment of which values w_j may lie in the top-k.

Stability guarantee: Our first main result verifies that, in a two-stage procedure for top-k ranking, we can obtain a stability guarantee by combining the inflated top-k method with any stable learning algorithm $\mathcal{A}$.

Theorem 4 *Fix any $n \geq 2$ and any $k \in [L]$. Let $\mathcal{A}$ be any learning algorithm that has (ε, δ)-stability at sample size n as in* (1)*. Then the two-stage procedure* $\text{top-}k^{(\varepsilon)} \circ \mathcal{A}$ *has top-k stability δ at sample size n, as defined in Defn. 1.*

The key idea behind the proof of this result is the following property of the inflated top-k:

$$\text{For all } w, v \in \mathbb{R}^L, \text{ if } \|w - v\| < \varepsilon \text{ then } \left| \text{top-}k^{(\varepsilon)}(w) \cap \text{top-}k^{(\varepsilon)}(v) \right| \geq k. \tag{3}$$

This property allows us to use the stability property of the learning algorithm $\mathcal{A}$ (1) to derive top-k stability for the two-stage procedure $\text{top-}k^{(\varepsilon)} \circ \mathcal{A}$.

Optimality of the method: Since the inflated top-k permits returning a larger set to handle ambiguity (that is, $\text{top-}k^{(\varepsilon)}(w)$ may contain $> k$ elements), a natural concern is whether this leads to excessive redundancy. Is the method returning overly large sets more often than necessary? The following proposition shows that there is no need for such concern: in fact, the inflated top-k is optimal for the problem of stable top-k selection, in the sense that it returns exactly k elements as often as possible.

Proposition 5 *Consider any function $\mathcal{R} : \mathbb{R}^L \to \wp_{\geq k}([L])$. Suppose that $\mathcal{R}$ is permutation invariant, that* $\text{top-}k(w) \subseteq \mathcal{R}(w)$ *for all $w \in \mathbb{R}^L$, and that $\mathcal{R}$ satisfies*

$$\text{For all } w, v \in \mathbb{R}^L, \text{ if } \|w - v\| < \varepsilon \text{ then } |\mathcal{R}(w) \cap \mathcal{R}(v)| \geq k.$$

Then for any $w \in \mathbb{R}^L$,

$$\text{If } |\mathcal{R}(w)| = k \text{ then } \text{top-}k^{(\varepsilon)}(w) = \mathcal{R}(w)$$

(and consequently $|\text{top-}k^{(\varepsilon)}(w)| = k$*).*

In other words, as long as $\mathcal{R}$ satisfies the conditions needed for ensuring top-k stability when combined with any stable learning algorithm $\mathcal{A}$ (analogous to the property (3) satisfied by $\text{top-}k^{(\varepsilon)}$), then the inflated top-k

method is able to return a set of size k (i.e., no ambiguity in identifying the top-k items) at least as often as $\mathcal{R}$.

As a concrete example illustrating the optimality of the inflated top-k, we show below that it always returns a set no larger than the output produced by a straightforward alternative.

Proposition 6 *Fix any $k \in [L]$. For any $w \in \mathbb{R}^L$ and any $\varepsilon > 0$, define the fixed-margin* top-k *selection rule as,*

$$s^{\varepsilon}_{\text{margin},k}(w) = \left\{ j \in [L] : w_j > w_{(k)} - \varepsilon/\sqrt{2} \right\}.$$

Then $s^{\varepsilon}_{\text{margin},k}$ satisfies

$$\text{For all } w, v \in \mathbb{R}^L, \text{ if } \|w - v\| < \varepsilon \text{ then } \left| s^{\varepsilon}_{\text{margin},k}(w) \cap s^{\varepsilon}_{\text{margin},k}(v) \right| \geq k.$$

Moreover, it holds that

$$\text{top-}k^{(\varepsilon)}(w) \subseteq s^{\varepsilon}_{\text{margin},k}(w).$$

Furthermore, for $L \geq 3$, there exist examples where this set inclusion is strict, i.e., $\text{top-}k^{(\varepsilon)}(w) \subsetneq s^{\varepsilon}_{\text{margin},k}(w)$.

In other words, the simple fixed-margin rule $s^{\varepsilon}_{\text{margin},k}$ satisfies the same property as inflated top-k (i.e., analogous to (3)), which allows us to use this rule to achieve top-k stability; however, the sets returned by $s^{\varepsilon}_{\text{margin},k}$ are always at least as large as those returned by our inflated top-k method.

Efficient computation: Although computing the set returned by the inflated top-k (2) appears complicated at first glance, involving calculating distances to the set $C^{\varepsilon,k}_j$, its computation can be drastically simplified. To present this result, for simplicity we will assume that we are computing $\text{top-}k^{(\varepsilon)}(w)$ for a score vector satisfying $w_1 \geq \cdots \geq w_L$, without loss of generality.

Proposition 7 *Fix any $k \in [L]$, any $\varepsilon > 0$ and any $w \in \mathbb{R}^L$ with $w_1 \geq \cdots \geq w_L$. Then it holds that*

$$\text{top-}k^{(\varepsilon)}(w) = \{1, \ldots, k-1\} \cup \left\{ k - 1 + j : j \in \text{argmax}^{(\varepsilon)} \left((w_k, \ldots, w_L) \right) \right\}.$$

In other words, computing the inflated top-k for a vector of scores $w = (w_1, \ldots, w_L)$ (with $w_1 \geq \cdots \geq w_L$) is only as hard as computing the inflated argmax for the subvector $(w_k, \ldots, w_L)$—and this latter problem has a computationally simple solution (Soloff et al., 2024a, Prop. 11). Moreover, this result offers an interesting perspective on how the inflated top-k method operates: if we think of the inflated argmax as handling any ambiguity at the argmax boundary (i.e., identifying which item is in the top position), then the inflated top-k can be viewed as analogously handling any ambiguity at the boundary between the k-th and $(k+1)$-st position.

4 The inflated full ranking method

Next, we turn to the full ranking problem. In a canonical sorting algorithm, the full ranking list can be constructed sequentially: at each step, the algorithm selects the argmax over the current set of elements, assigns it to the next position in the list, and then repeats this process on the reduced set until all positions are filled. Our inflated full ranking method follows a similar iterative strategy.

Definition 8 (Inflated full ranking) *For any $w \in \mathbb{R}^L$ and any $\varepsilon > 0$, define the ε-inflated full ranking as*

$$\text{ranking}^{(\varepsilon)}(w) := \left\{ \pi \in \mathcal{S}_L : 1 \in \text{argmax}^{(\varepsilon)} \left((w_{\pi(k)}, \ldots, w_{\pi(L)}) \right) \text{ for each } k \in [L] \right\}. \tag{4}$$

In other words, for a permutation π to be included into the set $\text{ranking}^{(\varepsilon)}(w)$, the following must hold: at each step $k \in [L]$, when we examine the subvector $(w_{\pi(k)}, \ldots, w_{\pi(L)})$, the inflated argmax must include its first entry (i.e., the entry corresponding to item $\pi(k)$ in the original vector).

We can immediately verify that the inflated full ranking satisfies several natural properties, by construction. First, it must hold that $\text{ranking}^{(\varepsilon)}(w) \ni \text{ranking}(w)$—for instance, if $w_1 \geq \cdots \geq w_L$, then the identity permutation, $\pi = \text{Id}$, must be included in the set of possible rankings $\text{ranking}^{(\varepsilon)}(w)$. Next, the procedure satisfies permutation invariance: for any $\pi \in \mathcal{S}_L$,

$$\pi \in \text{ranking}^{(\varepsilon)}\left((w_1, \ldots, w_L)\right) \impliedby \text{Id} \in \text{ranking}^{(\varepsilon)}\left((w_{\pi(1)}, \ldots, w_{\pi(L)})\right)$$

Stability guarantee: As for the top-k setting, our next result verifies that, when running a two-stage procedure for full ranking, we obtain a stability guarantee by combining the inflated full ranking method with any stable learning algorithm $\mathcal{A}$.

Theorem 9 *Fix any $n \geq 2$. Let $\mathcal{A}$ be any learning algorithm that has (ε, δ)-stability at sample size n as in* (1)*. Then the two-stage procedure* $\text{ranking}^{(\varepsilon)} \circ \mathcal{A}$ *has full ranking stability δ at sample size n, as defined in Defn. 2.*

As for the stability guarantee for the top-k problem, here the key step will again be to verify that inflated full ranking satisfies the following property:

$$\text{For all } w, v \in \mathbb{R}^L, \text{ if } \|w - v\| < \varepsilon \text{ then } \text{ranking}^{(\varepsilon)}(w) \cap \text{ranking}^{(\varepsilon)}(v) \neq \varnothing. \tag{5}$$

Optimality of the method: Next, we will examine the question of optimality: essentially, does the inflated full ranking return the smallest possible set of candidate permutations? The following result states that the method is optimal for the problem of stable full ranking in this sense.

Proposition 10 *Consider any function $\mathcal{R} : \mathbb{R}^L \to \wp(\mathcal{S}_L)$. Suppose that $\mathcal{R}$ is permutation invariant, that* $\text{ranking}(w) \in \mathcal{R}(w)$ *for all $w \in \mathbb{R}^L$, and that $\mathcal{R}$ satisfies*

$$\text{For all } w, v \in \mathbb{R}^L, \text{ if } \|w - v\| < \varepsilon \text{ then } \mathcal{R}(w) \cap \mathcal{R}(v) \neq \varnothing.$$

Then for any $w \in \mathbb{R}^L$, and any $i \neq j \in [L]$,

$$\text{If } \pi^{-1}(i) < \pi^{-1}(j) \text{ for all } \pi \in \mathcal{R}(w), \text{ then } \pi^{-1}(i) < \pi^{-1}(j) \text{ for all } \pi \in \text{ranking}^{(\varepsilon)}(w).$$

In other words, if $\mathcal{R}(w)$ is able to claim that item i is definitely ranked higher than item j (based on score vector w), then the same is true for $\text{ranking}^{(\varepsilon)}(w)$. In particular, this implies that if $\mathcal{R}(w) = \{\pi\}$ (i.e., $\mathcal{R}(w)$ returns a single permutation), then $\text{ranking}^{(\varepsilon)}(w) = \{\pi\}$ as well. But this result has additional implications as well. For example, suppose there are many near-ties within w, but there is a clear separation between the top-k items and the remaining $L - k$ items—without loss of generality, suppose that $w_1, \ldots, w_k$ are each ranked higher than any of $w_{k+1}, \ldots, w_L$. If this separation is unambiguous according to the ranking rule $\mathcal{R}$ (i.e., for every $\pi \in \mathcal{R}(w)$, it holds that $\{\pi(1), \ldots, \pi(k)\} = \{1, \ldots, k\}$), then the same is true for $\text{ranking}^{(\varepsilon)}(w)$.

As for the top-k setting, we present below a natural alternative for the full ranking problem based on a fixed margin rule, and compare to the inflated ranking rule, as a further illustration of the optimality of our method.

Proposition 11 *For any $w \in \mathbb{R}^L$ and any $\varepsilon > 0$, define*

$$\text{PairRank}^{(\varepsilon)}(w) = \left\{\pi \in \mathcal{S}_L : \ w_{\pi(j)} > w_{\pi(k)} - \varepsilon/\sqrt{2}, \ \forall j < k\right\}.$$

Then $\text{PairRank}^{(\varepsilon)}$ *satisfies*

$$\text{For all } w, v \in \mathbb{R}^L, \text{ if } \|w - v\| < \varepsilon \text{ then } \text{PairRank}^{(\varepsilon)}(w) \cap \text{PairRank}^{(\varepsilon)}(v) \neq \varnothing.$$

Moreover, it holds that

$$\text{ranking}^{(\varepsilon)}(w) \subseteq \text{PairRank}^{(\varepsilon)}(w).$$

Furthermore, there exist examples where this set inclusion is strict, i.e., $\text{ranking}^{(\varepsilon)}(w) \subsetneq \text{PairRank}^{(\varepsilon)}(w)$.

Efficient computation: Since calculating the inflated argmax is computationally efficient (as discussed above), checking whether a given permutation $\pi \in \mathcal{S}_L$ lies in $\text{ranking}^{(\varepsilon)}(w)$ is simple—but, if L is large, enumerating all such permutations π may be computationally challenging. To help with this task, the following result shows that we can restrict our attention to a much smaller set of possible permutations π.

Proposition 12 *For any $w \in \mathbb{R}^L$ and any $\varepsilon > 0$, it holds that*

$$\min\left\{k : j \in \text{top-}k^{(\varepsilon)}(w)\right\} \leq \pi^{-1}(j) \leq \sum_{\ell=1}^{L} \mathbf{1}\{w_\ell > w_j - \varepsilon/\sqrt{2}\},$$

for all $j \in [L]$ and all $\pi \in \text{ranking}^{(\varepsilon)}(w)$.

That is, these upper and lower bounds restrict the possible positions assigned to the jth item by any permutation π in the inflated full ranking.

5 Connecting the full ranking, top-k, and argmax problems

In the setting of standard ranking rules (without inflation), the argmax, top-k, and full ranking questions are all closely connected: for example, if $\pi = \text{ranking}(w)$ determines the ranking of w, then $\pi(1)$ determines the argmax of w, and the top elements of π determine the answer to the top-k problem, i.e., $\text{top-}k(w) = \{\pi(1), \ldots, \pi(k)\}$. Conversely, the ranking of w can be uniquely determined by considering $\text{top-}k(w)$ across all values of k.

In this section, we extend these connections to the setting of the inflated top-k and inflated full ranking methods. Indeed, as we have seen previously, the inflated argmax is a key ingredient in computing the inflated top-k (Prop. 7), and in defining the inflated full ranking (Defn. 8). Here, we examine some additional connections.

Our first result shows that the inflated full ranking of w directly reveals the inflated top-k set.

Proposition 13 *For any $\varepsilon > 0$, any $w \in \mathbb{R}^L$, and any $k \in [L]$,*

$$\text{top-}k^{(\varepsilon)}(w) = \cup_{\pi \in \text{ranking}^{(\varepsilon)}(w)} \{\pi(1), \ldots, \pi(k)\}.$$

In other words, the inflated top-k set consists of all items j that appear in the top k entries of any permutation $\pi \in \text{ranking}^{(\varepsilon)}(w)$.

In the reverse direction, the picture is less straightforward. It turns out that computing the inflated top-k sets (across all k) is not sufficient to reveal the inflated full ranking; the inflated full ranking contains information beyond what is revealed by the inflated top-k sets.

Proposition 14 *For any $\varepsilon > 0$ and any $w \in \mathbb{R}^L$, define*

$$\mathcal{R}(w) = \left\{\pi \in \mathcal{S}_L : \pi(k) \in \text{top-}k^{(\varepsilon)}(w) \ \forall\ k \in [L]\right\}.$$

Then $\text{ranking}^{(\varepsilon)}(w) \subseteq \mathcal{R}(w)$, and moreover, there exist examples where this set inclusion is strict, i.e., $\text{ranking}^{(\varepsilon)}(w) \subsetneq \mathcal{R}(w)$.

As discussed above as well as observed from our constructions for the inflated top-k and inflated ranking, the top-k selection and the full ranking problem are closely related to the argmax problem. This close connection is also reflected in the optimality of our methods: as we show in Appendix A, the optimality of the inflated argmax propagates to both the top-k and full ranking settings, thereby yielding the optimality of the inflated top-k and inflated full ranking methods. See Appendix A for details.

6 Experiments

In this section, we evaluate our proposed methods on real and simulated data.[3]

6.1 Experiments for top-k selection

Data and scores calculation. We use the Netflix Prize data (Bennett & Lanning, 2007),[4] which consists of $480,189$ anonymous Netflix customers' ratings over a total of $L = 17,770$ movies. The ratings are on a scale from 1 to 5 stars. For each trial, we first generate a subsample of $n = 1000$ users, sampled uniformly without replacement; let Z_i denote the ratings data for each user $i \in [n]$. The score vector $\hat{w} = \mathcal{A}(Z_1, \ldots, Z_n)$ is defined by taking $\hat{w}_\ell$ to be the average score for the ℓth movie, averaged over all users who provided a rating for that movie, and modified with a "+1" term in the denominator for shrinkage, $\hat{w}_\ell = \frac{\text{sum of all ratings for the } \ell\text{th movie}}{1+\text{number of ratings for the } \ell\text{th movie}}$, which allows $\hat{w}_\ell$ to be well-defined even if a movie received no ratings among the n sampled users. We then repeat this procedure for $N = 100$ independent trials, i.e., each trial uses a new subsample of n individuals.

Methods and evaluation. We compare the inflated top-k against the usual (uninflated) top-k selection method, with $k = 20$ and with $\varepsilon = 0.01$ for the inflated method, under several metrics.[5]

For each trial $j = 1, \ldots, N$, let $\hat{w}^{(j)} \in \mathbb{R}^L$ denote the score vector obtained by computing the average rating (modified with the "+1" term as described above) over the users included in the jth trial, as described above, and let $\hat{w}^{(j),\setminus i}$ denote the same score vector when computed without the ith user in the subsample, for each $i = 1, \ldots, n$. To assess the top-k stability of each method, we compute, for each trial $j = 1, \ldots, N$,

$$\delta_j = \frac{1}{n} \sum_{i=1}^{n} \mathbf{1} \left\{ \left| \mathcal{R}(\hat{w}^{(j)}) \cap \mathcal{R}(\hat{w}^{(j),\setminus i}) \right| < k \right\},$$

where $\mathcal{R}(\cdot)$ denotes either top-$k(\cdot)$ or top-$k^{(\varepsilon)}(\cdot)$. (Recalling Defn. 1, we should see $\delta_j \leq \delta$ for any method that satisfies top-k stability at level δ.) We also consider an alternative measure to assess stability, the Jaccard similarity (Jaccard, 1912), which measures overlap between sets; a value closer to 1 suggests stronger agreement (i.e., higher stability). For each trial j, and each of the two methods, we compute

$$\text{Jaccard}_j = \frac{1}{n} \sum_{i=1}^{n} \frac{\left| \mathcal{R}(\hat{w}^{(j)}) \cap \mathcal{R}(\hat{w}^{(j),\setminus i}) \right|}{\left| \mathcal{R}(\hat{w}^{(j)}) \cup \mathcal{R}(\hat{w}^{(j),\setminus i}) \right|}.$$

Finally, we also compute the size of the returned set, $\text{Size}_j = \left| \mathcal{R}(\hat{w}^{(j)}) \right|$, which indicates the informativeness of the set; ideally we would want to return exactly k items, but a set size larger than k indicates that there may be ambiguity in the scores, with near-ties between multiple movies.

Results. Table 1 reports the results of the experiment in terms of the evaluation metrics defined above. First, we observe that the inflated top-k shows much better stability than top-k, with substantially lower values of δ_j (and with slightly higher Jaccard similarity), on average across the N trials. Note that even $\max_{j \in [N]} \delta_j$ is quite small for inflated top-k, agreeing with our theoretical finding that inflated top-k offers distribution-free stability, i.e., uniformly over any data set. The high instability of top-k suggests the presence of ambiguity within the data, which makes the returned set of top-k highly sensitive to the removal of a single data point. The inflated top-k accommodates this ambiguity by including slightly more possible candidates in the list. There is relatively little cost to this, since the size of the set returned by the inflated top-k is only

[3] Running the experiments took approximately 30 minutes on a MacBook Pro laptop using a single CPU core. Code to reproduce the experiments is available in the supplementary material.

[4] Data was obtained from `https://www.kaggle.com/datasets/netflix-inc/netflix-prize-data/data`.

[5] For top-k, if movies are tied for the kth position, we break ties by choosing the movie with the smallest index $\ell \in \{1, \ldots, L\}$. Notably, another reason for the "+1" in the denominator is that, without this shrinkage term, we often see many ties between movies near the top of the ranked list (namely, movies that received only a few ratings, and all those ratings are equal to the highest value 5, which often occurs due to the small subsample size n)—and since characterizing the instability of top-k selection in the presence of frequent ties is heavily dependent on our choice of tie-breaking rule, to avoid this issue we instead use the shrinkage term.

Methods	$\max_{j\in[N]} \delta_j$	$\frac{1}{N}\sum_{j\in[N]} \delta_j$	$\frac{1}{N}\sum_{j\in[N]} \text{Jaccard}_j$	$\frac{1}{N}\sum_{j\in[N]} \text{Size}_j$
top-k	0.8530	0.1205 (0.0136)	0.9876 (0.0015)	20.00 (0.0000)
top-$k^{(\varepsilon)}$	0.0380	0.0094 (0.0009)	0.9906 (0.0006)	21.22 (0.1101)

Table 1: Results on the Netflix Prize dataset (see Section 6.1 for details). Evaluation results under various metrics are reported in the table, with standard errors for the averages shown in parentheses.

Methods	$\max_{j\in[N]} \delta_j$	$\frac{1}{N}\sum_{j\in[N]} \delta_j$	$\frac{1}{N}\sum_{j\in[N]} \text{Size}_j$
ranking	0.78	0.1757 (0.0049)	1.00 (0.00)
ranking$^{(\varepsilon)}$	0.12	0.0162 (0.0007)	1.76 (0.04)

Table 2: Results on full ranking stability for simulated data (see Section 6.2 for details). Evaluation results under various metrics are reported in the table, with standard errors for the averages shown in parentheses.

slightly larger, returning (on average) a set of size ≈ 21.22, as compared to $k = 20$. See Section C for figures illustrating the results, and additional results at different values of k.

6.2 Experiments for full ranking

Data and scores calculation. We generate data from a Gaussian linear model, $Y_i = X_i^\top \beta^* + \zeta_i$, where $\zeta_i \overset{\text{iid}}{\sim} \mathcal{N}(0,1)$ and where the feature vectors $X_i \in \mathbb{R}^L$ are drawn as $X_i \overset{\text{iid}}{\sim} \mathcal{N}(0,\Sigma)$, for $\Sigma_{ij} = \rho^{|i-j|}$. We set $n = 50$, $L = 5$, and $\rho = 0.5$. The regression coefficients are taken to be $\beta_j^* = j/\sqrt{\sum_{k=1}^L k^2}$. The learning algorithm $\mathcal{A}$ is defined as follows. First we estimate the coefficients of the regression via ℓ_2-constrained least squares: writing $Z_i = (X_i, Y_i)$ for the ith data point, define

$$\hat{\beta} = \text{argmin}_{\|\beta\|_2 \le 1} \sum_{i=1}^n (Y_i - X_i^\top \beta)^2.$$

(The stability of algorithms of this type (in the sense of (1)) has been well-studied—see, e.g., Bousquet & Elisseeff (2002).) We then take $\hat{w}_\ell = |\hat{\beta}_\ell|$ to estimate the magnitude of the ℓth coefficient—our goal will now be to rank the coefficients—that is, to rank the features in terms of their (estimated) importance in the model.

Methods and evaluation. We compare inflated full ranking (with $\varepsilon = 0.05$) against the usual full ranking procedure, under several metrics: for each $j \in [N]$, we compute

$$\delta_j = \frac{1}{n}\sum_{i=1}^n \mathbf{1}\left\{\mathcal{R}(\hat{w}^{(j)}) \cap \mathcal{R}(\hat{w}^{(j),\setminus i}) = \varnothing\right\}, \quad \text{Size}_j = \left|\mathcal{R}(\hat{w}^{(j)})\right|,$$

for $\mathcal{R}(\cdot)$ denoting either ranking$(\cdot)$ or ranking$^{(\varepsilon)}(\cdot)$, to assess the stability and the informativeness of each procedure.

Results. We present results for this simulation in Table 2. The inflated full ranking procedure shows substantially better stability, with much smaller values of δ_j. This comes at little cost, because the inflated full ranking procedure returns ≈ 1.75 many permutations on average—we can interpret this as saying that there is ambiguity among only very few of the coefficients.

7 Discussion

In this section, we will briefly discuss the practical implications of our methods, and then provide a review of related work on learning to rank and its connections to our proposed stable ranking, as well as the related topic of confidence sets for rankings. Finally, we summarize our main contributions and conclude with a discussion of limitations and potential extensions that suggest interesting directions for future work.

7.1 Interpretation and practical implications of our methods

As discussed in Section 2, the primary goal of our methods is to faithfully and adaptively quantify ambiguity in the score vector $\hat{w}$, and to propagate this ambiguity to the ranking output in a non-conservative manner.

To illustrate, consider the full ranking problem. When there is no ambiguity in a subset of the scores $\{\hat{w}_j : j \in \mathcal{J} \subseteq [L]\}$, i.e., the $\hat{w}_j$ in the set are well separated, the inflated ranking returns a single ranking (if restricted to these items), as shown in Prop. 10. In contrast, when ambiguity is present among these items, this is reflected in the output as a set of rankings, where the relative positions of those ambiguous items vary while others (without ambiguity) remain fixed. More generally, the size of the returned set, together with post-hoc summaries of each item's rank variability (e.g., as characterized in Prop. 12), provides a direct measure of uncertainty and thus a notion of confidence in the ranking.

Therefore, from a practical standpoint, large output sets or substantial rank variability for specific items can serve as indicators of high uncertainty in the underlying score estimates. This, in turn, may guide downstream decisions, such as prioritizing additional data collection for high rank-variability items to improve their score estimates.

7.2 Related work

Devic et al. (2024) also propose a method that can reflect the uncertainty of the learned score predictors from Step 1. However, their work differs from ours in considering score *distributions* for each item (not point scores) and returning a probabilistic distribution over ranking lists, whereas we aim for deterministic set-valued outputs. Notably, the notion of a stochastic type of output to account for the uncertainty in ranking problems has long been employed, though most works focus on promoting fairness instead of algorithmic stability. See, for example Dwork et al. (2019); Singh et al. (2021); Guo et al. (2023). Oh et al. (2024) apply fine-tuning to empirically stabilize a given recommendation system, focusing on sequential recommendation based on users' previous interactions, which is distinct from our setting. Another related line of work focuses on proposing measures of ranking stability, along with corresponding empirical evaluations (Adomavicius & Zhang, 2016; Asudeh et al., 2018; Oh et al., 2022).

Our definition of ranking stability is rooted in the perspective of using set-valued output to quantify uncertainty, which connects to set-valued classification (Grycko, 1993; Del Coz et al., 2009; Lei, 2014; Lorieul et al., 2021; Chzhen et al., 2021; Cortes et al., 2024) and conformal prediction (Vovk et al., 2005; Sadinle et al., 2019; Angelopoulos et al., 2023b;a). Interestingly, Guo et al. (2023) also use this set-valued idea to implicitly encode information about ranking scores' separation when constructing their probabilistic distribution over ranking lists, though they primarily aim to address the fairness problem.

Another closely related line of work focuses on constructing confidence sets for rankings, with the aim of ensuring that the true ranking is contained in the set with high probability. For example, in the context of top-k selection, Goldwasser et al. (2025) use selective inference-based methods to identify the top-k largest Gaussian means with high probability guarantees. For the full ranking problem, built on asymptotic normality, Xie et al. (2009) propose methods for constructing asymptotically valid confidence intervals for item-wise rankings of a population parameter in the presence of ties and near ties. To obtain simultaneous confidence intervals for full rankings, Neuhof & Benjamini (2024) and Al Mohamad et al. (2022) conduct all pairwise comparison tests and then aggregate the results to construct a set of rankings based on the induced partial order determined by test rejections. Relatedly, another way to obtain simultaneous confidence intervals for full rankings is to first construct simultaneous confidence regions for ranking scores and then form a set of rankings consistent with these regions. See, for example, (Klein et al., 2020; Chatzi et al., 2024), among

others. We note that, although this line of work also produces set-valued outputs for rankings, the underlying objectives are different. Our focus is on using the set to reflect the stability of the ranking output (an inherently relative notion), whereas this line of work is concerned with the accuracy of the ranking estimate, aiming to recover the true ranking (an absolute notion).

7.3 Summary and future directions

This work describes novel notions of stability for two ranking problems: identifying the top-k items among a collection of candidates, and ranking all candidates. Our framework seeks to ensure that the removal of a sample from the training dataset will not yield a wholesale change in the returned collection of items (for top-k) or permutations of the items (for full rankings). More specifically, in the context of top-k selection, our method returns a set of k *or more* items, and is stable in the sense that removing one sample and rerunning our method typically yields a set that shares at least k items with the original output. Similarly, in the context of full ranking, our method returns a set of permutations (rankings) of the L items, and is stable in the sense that removing one sample and rerunning our method typically yields a set of permutations with at least one in common with the original output. Of course, these particular definitions offer one specific notion of what it means to be stable in the context of a ranking problem, and exploring other possible formulations of stability is an important open question. For example, one interesting direction is to develop principled ways to extract a stabilized single ranking from the set of plausible outcomes under perturbations, and to translate the current stability guarantee to apply to such point-valued outputs.

Our approach does not rely upon assumptions on the base algorithm $\mathcal{A}$ used to assign scores to each item based on the training dataset $\mathcal{D}$, and does not place any distributional assumptions on $\mathcal{D}$. Our guarantees hold for any dataset size $n \geq 2$. Furthermore, the returned sets of items or permutations are optimal in that they are as small as possible, maximizing the informativeness of the returned top-k selections or full rankings. Our stability guarantees depend on the algorithm $\mathcal{A}$ mapping a dataset $\mathcal{D}$ to a vector of scores for each of the L items being (ε, δ)-stable. Bagging can be used to ensure this property (Soloff et al., 2024c), but whether alternative methods can provide similar assumption-free guarantees with less computational complexity remains an open question for further work.

References

Gediminas Adomavicius and Jingjing Zhang. Classification, ranking, and top-k stability of recommendation algorithms. *INFORMS Journal on Computing*, 28(1):129–147, 2016.

Shivani Agarwal and Partha Niyogi. Generalization bounds for ranking algorithms via algorithmic stability. *Journal of Machine Learning Research*, 10(2), 2009.

Diaa Al Mohamad, Jelle J Goeman, and Erik W van Zwet. Simultaneous confidence intervals for ranks with application to ranking institutions. *Biometrics*, 78(1):238–247, 2022.

Vito Walter Anelli, Yashar Deldjoo, Tommaso Di Noia, Daniele Malitesta, and Felice Antonio Merra. A study of defensive methods to protect visual recommendation against adversarial manipulation of images. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 1094–1103, 2021.

Anastasios N Angelopoulos, Stephen Bates, et al. Conformal prediction: A gentle introduction. *Foundations and Trends® in Machine Learning*, 16(4):494–591, 2023a.

Anastasios N Angelopoulos, Karl Krauth, Stephen Bates, Yixin Wang, and Michael I Jordan. Recommendation systems with distribution-free reliability guarantees. In *Conformal and Probabilistic Prediction with Applications*, pp. 175–193. PMLR, 2023b.

Abolfazl Asudeh, HV Jagadish, Gerome Miklau, and Julia Stoyanovich. On obtaining stable rankings. *Proceedings of the VLDB Endowment*, 12(3):237–250, 2018.

J. Bennett and S. Lanning. The Netflix Prize. In *Proceedings of the KDD Cup Workshop 2007*, pp. 3–6. ACM, August 2007. URL `http://www.cs.uic.edu/~liub/KDD-cup-2007/NetflixPrize-description.pdf`.

Olivier Bousquet and André Elisseeff. Stability and generalization. *The Journal of Machine Learning Research*, 2:499–526, 2002.

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

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In *Proceedings of the 24th International Conference on Machine Learning*, pp. 129–136, 2007.

Ivi Chatzi, Eleni Straitouri, Suhas Thejaswi, and Manuel G Rodriguez. Prediction-powered ranking of large language models. *Advances in Neural Information Processing Systems*, 37:113096–113133, 2024.

Pinhan Chen, Chao Gao, and Anderson Y Zhang. Optimal full ranking from pairwise comparisons. *The Annals of Statistics*, 50(3):1775–1805, 2022a.

Pinhan Chen, Chao Gao, and Anderson Y Zhang. Partial recovery for top-k ranking: optimality of MLE and suboptimality of the spectral method. *The Annals of Statistics*, 50(3):1618–1652, 2022b.

Yuxin Chen, Jianqing Fan, Cong Ma, and Kaizheng Wang. Spectral method and regularized MLE are both optimal for top-k ranking. *The Annals of statistics*, 47(4):2204, 2019.

Evgenii Chzhen, Christophe Denis, Mohamed Hebiri, and Titouan Lorieul. Set-valued classification–overview via a unified framework. *arXiv preprint arXiv:2102.12318*, 2021.

Corinna Cortes, Anqi Mao, Christopher Mohri, Mehryar Mohri, and Yutao Zhong. Cardinality-aware set prediction and top-k classification. *Advances in neural information processing systems*, 37:18265–18309, 2024.

Juan José Del Coz, Jorge Díez, and Antonio Bahamonde. Learning nondeterministic classifiers. *Journal of Machine Learning Research*, 10(10), 2009.

Siddartha Devic, Aleksandra Korolova, David Kempe, and Vatsal Sharan. Stability and multigroup fairness in ranking with uncertain predictions. In *International Conference on Machine Learning*, pp. 10661–10686. PMLR, 2024.

Cynthia Dwork, Michael P Kim, Omer Reingold, Guy N Rothblum, and Gal Yona. Learning from outcomes: Evidence-based rankings. In *2019 IEEE 60th Annual Symposium on Foundations of Computer Science (FOCS)*, pp. 106–125. IEEE, 2019.

Andre Elisseeff, Theodoros Evgeniou, Massimiliano Pontil, and Leslie Pack Kaelbing. Stability of randomized learning algorithms. *Journal of Machine Learning Research*, 6(1), 2005.

Wei Gao and Zhi-Hua Zhou. Uniform convergence, stability and learnability for ranking problems. In *IJCAI*, pp. 1337–1343, 2013.

Jeremy Goldwasser, Will Fithian, and Giles Hooker. Gaussian rank verification. *Stat*, 14(3):e70087, 2025.

Eugen Grycko. Classification with set-valued decision functions. In *Information and Classification: Concepts, Methods and Applications Proceedings of the 16th Annual Conference of the "Gesellschaft für Klassifikation eV" University of Dortmund, April 1–3, 1992*, pp. 218–224. Springer, 1993.

Ruocheng Guo, Jean-François Ton, Yang Liu, and Hang Li. Inference-time stochastic ranking with risk control. *arXiv preprint arXiv:2306.07188*, 2023.

Paul Jaccard. The distribution of the flora in the alpine zone. *The New Phytologist*, 11(2):37–50, 1912. ISSN 0028646X, 14698137. URL `http://www.jstor.org/stable/2427226`.

Martin Klein, Tommy Wright, and Jerzy Wieczorek. A joint confidence region for an overall ranking of populations. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 69(3):589–606, 2020.

Yanyan Lan, Tie-Yan Liu, Tao Qin, Zhiming Ma, and Hang Li. Query-level stability and generalization in learning to rank. In *Proceedings of the 25th International Conference on Machine Learning*, pp. 512–519, 2008.

Jing Lei. Classification with confidence. *Biometrika*, 101(4):755–769, 2014. ISSN 0006-3444,1464-3510. doi: 10.1093/biomet/asu038. URL https://doi.org/10.1093/biomet/asu038.

Tie-Yan Liu et al. Learning to rank for information retrieval. *Foundations and Trends® in Information Retrieval*, 3(3):225–331, 2009.

Titouan Lorieul, Alexis Joly, and Dennis Shasha. Classification under ambiguity: when is average-k better than top-k? *arXiv preprint arXiv:2112.08851*, 2021.

Bitya Neuhof and Yuval Benjamini. Confident feature ranking. In *International Conference on Artificial Intelligence and Statistics*, pp. 1468–1476. PMLR, 2024.

Sejoon Oh, Berk Ustun, Julian McAuley, and Srijan Kumar. Rank list sensitivity of recommender systems to interaction perturbations. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, pp. 1584–1594, 2022.

Sejoon Oh, Berk Ustun, Julian McAuley, and Srijan Kumar. Finest: Stabilizing recommendations by rank-preserving fine-tuning. *ACM Transactions on Knowledge Discovery from Data*, 18(9):1–22, 2024.

Mauricio Sadinle, Jing Lei, and Larry Wasserman. Least ambiguous set-valued classifiers with bounded error levels. *J. Amer. Statist. Assoc.*, 114(525):223–234, 2019. ISSN 0162-1459,1537-274X. doi: 10.1080/01621459.2017.1395341. URL https://doi.org/10.1080/01621459.2017.1395341.

Ashudeep Singh, David Kempe, and Thorsten Joachims. Fairness in ranking under uncertainty. *Advances in Neural Information Processing Systems*, 34:11896–11908, 2021.

Jake Soloff, Rina Barber, and Rebecca Willett. Building a stable classifier with the inflated argmax. *Advances in Neural Information Processing Systems*, 37:70349–70380, 2024a.

Jake A. Soloff, Rina Foygel Barber, and Rebecca Willett. Bagging provides assumption-free stability. *Journal of Machine Learning Research*, 25(131):1–35, 2024b. URL http://jmlr.org/papers/v25/23-0536.html.

Jake A. Soloff, Rina Foygel Barber, and Rebecca Willett. Stability via resampling: statistical problems beyond the real line. *arXiv preprint arXiv:2405.09511*, 2024c.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*. Springer, New York, 2005.

Minge Xie, Kesar Singh, and Cun-Hui Zhang. Confidence intervals for population ranks in the presence of ties and near ties. *Journal of the American Statistical Association*, 104(486):775–788, 2009.

Ke Yang, Julia Stoyanovich, Abolfazl Asudeh, Bill Howe, HV Jagadish, and Gerome Miklau. A nutritional label for rankings. In *Proceedings of the 2018 International Conference on Management of Data*, pp. 1773–1776, 2018.

A Revisiting optimality through the connection to the argmax problem

In this section, we offer an alternative perspective on the optimality of our methods, through the lens of the optimality of the inflated argmax. We start by providing an enhanced version of the optimality result for the inflated argmax.

Proposition 15 *Let $s : \Delta_{L-1} \to \wp([L])$ be any argmax (top-1 selection) rule. Suppose s is*

1. *ε-compatible: for any $\|w - v\| < \varepsilon$, $s(w) \cap s(v) \neq \varnothing$;*
2. *permutation invariant: for any $v, w \in \mathbb{R}^L$, if $v = (w_{\sigma(1)}, \ldots, w_{\sigma(L)})$ for some permutation σ on $[L]$, then $j \in s(v) \Leftrightarrow \sigma(j) \in s(w)$;*
3. *contains the argmax: for any $w \in \mathbb{R}^L$, $\operatorname{argmax}(w) \subseteq s(w)$.*

Then for any $j \in [L]$,

$$s^{-1}(\{j\}) \subseteq C_j^{\varepsilon,1} \equiv \left\{ v : v_j \geq v_{(2)} + \varepsilon/\sqrt{2} \right\}.$$

Moreover, if further above inclusion becomes equality, then it holds for any $w \in \mathbb{R}^L$ that

$$\operatorname{argmax}^{(\varepsilon)}(w) \equiv \left\{ j \in [L] : d\left(w, C_j^{\varepsilon,1}\right) < \varepsilon \right\} \subseteq s(w).$$

In other words, among all argmax rules that satisfy properties 1–3 as stated in the proposition and have the same singleton regions as the inflated argmax, the inflated argmax is uniquely optimal in the sense that it always returns a smaller set. An implication of such optimality is that the top-k selection rule (respectively, the full ranking rule) constructed via the inflated argmax, i.e., the inflated top-k (respectively, inflated full ranking), is optimal among all constructions of this type based on argmax rules. In particular, alternative constructions that rely on more conservative argmax variants necessarily return larger sets. Examples include $s^{\varepsilon}_{\text{margin},k}$ and $\text{PairRank}^{(\varepsilon)}$, where both of them are rooted in $s^{\varepsilon}_{\text{margin},1}$, as we establish in the proofs of Props. 6 and 11. In other words, Prop. 15 provides a unified view of Prop. 6 and Prop. 11, which now appear as special cases.

A.1 Proof of Prop. 15

First, we show that the singleton region of s for each $j \in [L]$ is contained in $C_j^{\varepsilon,1}$. Fix any $j \in [L]$ and suppose $s(w) = \{j\}$. For any $k \neq j$, consider $v \in \mathbb{R}^L$ that is defined as

$$v_\ell = \begin{cases} w_\ell, & \ell \notin \{j, k\} \\ w_j, & \ell = k \\ w_k, & \ell = j. \end{cases}$$

That is, v and w swap the coordinates in positions j, k. Then by permutation invariance property of s, it holds that $s(v) = \{k\}$. Since $s(v) \cap s(w) = \varnothing$ and s satisfies ε-compatibility, it must hold that

$$\varepsilon \leq \|v - w\| = \sqrt{2(w_j - w_k)^2} \iff |w_j - w_k| \geq \varepsilon/\sqrt{2}.$$

Because $s(w)$ contains the argmax, we have that $\operatorname{argmax}(w) \subseteq s(w) = \{j\}$, which implies $w_j > w_j$. Hence, $w_j \geq w_k + \varepsilon/\sqrt{2}$. Finally, since this holds for any $k \neq j$, we have

$$w_j \geq \max_{k \neq j} w_k + \varepsilon/\sqrt{2},$$

and thus $w \in C_j^{\varepsilon,1}$.

Next, if $s^{-1}(\{j\}) = C_j^{\varepsilon,1}$ for any $j \in [L]$, then whenever $d\left(w, C_j^{\varepsilon,1}\right) < \varepsilon$, $s(w)$ must contain item j. Otherwise, we can find some $v \in C_j^{\varepsilon,1}$ that $\|v - w\| < \varepsilon$, but $\{j\} \equiv s(v) \cap s(w) = \varnothing$, which contradicts with the ε-compatibility property satisfied by s. Therefore, $d\left(w, C_j^{\varepsilon,1}\right) < \varepsilon$ must imply $j \in s(w)$, and $\operatorname{argmax}^{(\varepsilon)}(w) \subseteq s(w)$ immediately follows from the definition of the inflated argmax.

A.2 Proof of Prop. 6

Since $s^{\varepsilon}_{\text{margin},1}(w) \supseteq \text{argmax}^{(\varepsilon)}(w)$ for any $w \in \mathbb{R}^L$ (Soloff et al., 2024a, Prop. 20), it then follows from Lemma 16 that

$$\begin{aligned} s^{\varepsilon}_{\text{margin},k}(w) &= \left\{ j \in [L] : w_j > w_{(k)} - \varepsilon/\sqrt{2} \right\} \\ &= \bigcup_{\substack{L-k+1 \le \ell \le L, \\ \text{distinct } i_1,\ldots,i_\ell \in [L]}} \left\{ i_j : w_{i_j} > \max_{t \in \{i_1,\cdots,i_\ell\}} w_t - \varepsilon/\sqrt{2} \right\} \\ &= \bigcup_{\substack{L-k+1 \le \ell \le L, \\ \text{distinct } i_1,\ldots,i_\ell \in [L]}} \left\{ i_j : j \in s^{\varepsilon}_{\text{margin},1}\big((w_{i_1},\ldots,w_{i_\ell})\big) \right\} \\ &\supseteq \bigcup_{\substack{L-k+1 \le \ell \le L, \\ \text{distinct } i_1,\ldots,i_\ell \in [L]}} \left\{ i_j : j \in \text{argmax}^{(\varepsilon)}\left((w_{i_1},\ldots,w_{i_\ell})\right) \right\} = \text{top-}k^{(\varepsilon)}(w). \end{aligned}$$

Moreover, for any $w, v \in \mathbb{R}^L$ that $\|w - v\| < \varepsilon$, we have $s^{\varepsilon}_{\text{margin},k}(w) \cap s^{\varepsilon}_{\text{margin},k}(v) \supseteq \text{top-}k^{(\varepsilon)}(w) \cap \text{top-}k^{(\varepsilon)}(v)$, and thus $|s^{\varepsilon}_{\text{margin},k}(w) \cap s^{\varepsilon}_{\text{margin},k}(v)| \ge k$ due to (3).

For examples of strict inclusion, consider, without loss of generality, $w_1 \ge w_2 \ge \cdots \ge w_L$. Then note that, in this case,

$$s^{\varepsilon}_{\text{margin},k}(w) = \{1, \cdots, k\} \cup \left\{ k - 1 + j : j \in s^{\varepsilon}_{\text{margin},1}\big((w_k,\ldots,w_L)\big) \right\},$$

and

$$\text{top-}k^{(\varepsilon)}(w) = \left\{1,\ldots,k\right\} \cup \left\{ k - 1 + j : j \in \text{argmax}^{(\varepsilon)}\left((w_k,\ldots,w_L)\right) \right\}.$$

Hence, it suffices to construct examples of $\text{argmax}^{(\varepsilon)}(w) \subsetneq s^{\varepsilon}_{\text{margin},1}(w)$ for $L = 3$. To this end, let $\varepsilon = 1/\sqrt{2}$, and

$$w = \left(\frac{1 + \sqrt{7/2}}{6}, \frac{1 + \sqrt{7/2}}{6}, \frac{2 - \sqrt{7/2}}{3} \right).$$

Then we can calculate that

$$\max_{i \ne j} |w_i - w_j| = \frac{\sqrt{7/2} - 1}{2} < \frac{1}{2} = \varepsilon/\sqrt{2},$$

which implies that $s^{\varepsilon}_{\text{margin},1}(w) = \{1, 2, 3\}$. However, we can calculate $\text{argmax}^{(\varepsilon)}(w) = \{1, 2\} \subsetneq s^{\varepsilon}_{\text{margin},1}(w)$.

A.3 Proof of Prop. 11

Since $s^{\varepsilon}_{\text{margin},1}(w) \supseteq \text{argmax}^{(\varepsilon)}(w)$ for any $w \in \mathbb{R}^L$ (Soloff et al., 2024a, Prop. 20), it suffices to show that

$$\begin{aligned} \text{PairRank}^{(\varepsilon)}(w) &= \left\{ \pi \in \mathcal{S}_L : \ w_{\pi(j)} > w_{\pi(k)} - \varepsilon/\sqrt{2}, \ \forall j < k \right\} \\ &= \left\{ \pi \in \mathcal{S}_L : 1 \in s^{\varepsilon}_{\text{margin},1}\big((w_{\pi(k)},\ldots,w_{\pi(L)})\big) \ \text{ for each } k \in [L] \right\}. \end{aligned}$$

To establish the above, note that for any $w \in \mathbb{R}^L$, any $\pi \in \mathcal{S}_L$ and any $j \in [L]$, $w_{\pi(j)} > w_{\pi(k)} - \varepsilon/\sqrt{2}$ for any $k > j$ is equivalent to

$$w_{\pi(j)} > \max_{\ell \ge j} w_{\pi(\ell)} - \varepsilon/\sqrt{2},$$

which by definition of $s^{\varepsilon}_{\text{margin},1}$, is equivalent to $1 \in s^{\varepsilon}_{\text{margin},1}\big((w_{\pi(j)},\ldots,w_{\pi(L)})\big)$. Therefore, we have that

$$\text{PairRank}^{(\varepsilon)}(w) = \left\{ \pi \in \mathcal{S}_L : 1 \in s^{\varepsilon}_{\text{margin},1}\big((w_{\pi(k)},\ldots,w_{\pi(L)})\big) \ \text{ for each } k \in [L] \right\} \supseteq \text{ranking}^{(\varepsilon)}(w).$$

Next, since the inflated ranking satisfies the ε-compatibility property (5), it follows that for any $w, v \in \mathbb{R}^L$ that $\|w - v\| < \varepsilon$,

$$\text{PairRank}^{(\varepsilon)}(w) \cap \text{PairRank}^{(\varepsilon)}(v) \supseteq \text{ranking}^{(\varepsilon)}(w) \cap \text{ranking}^{(\varepsilon)}(v) \ne \varnothing.$$

Finally, consider the following example: Let $L = 3$, $\varepsilon = 1/\sqrt{2}$, and

$$w = \left(\frac{2 - \sqrt{7/2}}{3}, \frac{1 + \sqrt{7/2}}{6}, \frac{1 + \sqrt{7/2}}{6} \right).$$

Then we can calculate that

$$\max_{i \neq j} |w_i - w_j| = \frac{\sqrt{7/2} - 1}{2} < \frac{1}{2} = \varepsilon/\sqrt{2},$$

which implies that $\text{PairRank}^{(\varepsilon)}(w) = \mathcal{S}_3$, i.e., the set of all permutations when $L = 3$.

On the other hand, we can calculate

$$\text{top-}k^{(\varepsilon)}(w) = \begin{cases} \{2, 3\}, & k = 1, \\ \{1, 2, 3\}, & k = 2, \\ \{1, 2, 3\}, & k = 3. \end{cases}$$

Thus, we have that

$$\text{ranking}^{(\varepsilon)}(w) = \{(1, 2, 3), (1, 3, 2), (2, 1, 3), (2, 3, 1)\} \subsetneq \mathcal{S}_3 = \text{PairRank}^{(\varepsilon)}(w).$$

B Proofs of other theoretical results

B.1 Proof of Thm. 4

As discussed after the statement of the theorem, it is sufficient to verify

$$\text{For all } w, v \in \mathbb{R}^L, \text{ if } \|w - v\| < \varepsilon \text{ then } \left| \text{top-}k^{(\varepsilon)}(w) \cap \text{top-}k^{(\varepsilon)}(v) \right| \geq k,$$

since assuming this property holds, we then have

$$\frac{1}{n} \sum_{i=1}^{n} \mathbf{1} \left\{ \left| \text{top-}k^{(\varepsilon)}(\mathcal{A}(\mathcal{D})) \cap \text{top-}k^{(\varepsilon)}(\mathcal{A}(\mathcal{D}^{\setminus i})) \right| \geq k \right\} \geq \frac{1}{n} \sum_{i=1}^{n} \mathbf{1} \left\{ \|\mathcal{A}(\mathcal{D}) - \mathcal{A}(\mathcal{D}^{\setminus i})\| < \varepsilon \right\} \geq 1 - \delta.$$

To prove that the above property holds, we will need to use the fact that the same property holds for the inflated argmax, i.e., for the case $k = 1$ (Soloff et al., 2024a, Thm. 9):

$$\text{For all } w, v \in \mathbb{R}^L, \text{ if } \|w - v\| < \varepsilon \text{ then } \text{argmax}^{(\varepsilon)}(w) \cap \text{argmax}^{(\varepsilon)}(v) \neq \varnothing. \tag{6}$$

We will also need the following lemma:

Lemma 16 *For any $\varepsilon > 0$, any $w \in \mathbb{R}^L$, and any $k \in [L]$,*

$$\text{top-}k^{(\varepsilon)}(w) = \bigcup_{\substack{L-k+1 \leq \ell \leq L, \\ \text{distinct } i_1, \ldots, i_\ell \in [L]}} \left\{ i_j : j \in \text{argmax}^{(\varepsilon)} \left((w_{i_1}, \ldots, w_{i_\ell}) \right) \right\}.$$

That is, the inflated top-k set is given by all entries w_j that are selected by the inflated argmax, when we apply the inflated argmax to any subvector of size $\geq L - k + 1$.

Now fix any $w, v \in \mathbb{R}^L$ with $\|w - v\| < \varepsilon$. Let $S = \text{top-}k^{(\varepsilon)}(w) \cap \text{top-}k^{(\varepsilon)}(v)$. Suppose that $|S| < k$. Let $i_1, \ldots, i_{L-|S|} \in [L]$ enumerate the remaining indices, $[L] \setminus S$. Then

$$\left\| \left(w_{i_1}, \ldots, w_{i_{L-|S|}} \right) - \left(v_{i_1}, \ldots, v_{i_{L-|S|}} \right) \right\| \leq \|w - v\| < \varepsilon,$$

and so by (6),

$$\text{argmax}^{(\varepsilon)} \left((w_{i_1}, \ldots, w_{i_{L-|S|}}) \right) \cap \text{argmax}^{(\varepsilon)} \left((v_{i_1}, \ldots, v_{i_{L-|S|}}) \right) \neq \varnothing.$$

We can therefore find some j with

$$j \in \text{argmax}^{(\varepsilon)} \left((w_{i_1}, \ldots, w_{i_{L-|S|}}) \right) \cap \text{argmax}^{(\varepsilon)} \left((v_{i_1}, \ldots, v_{i_{L-|S|}}) \right).$$

But since $L - |S| > L - k$, by Lemma 16 this means that $i_j \in \text{top-}k^{(\varepsilon)}(w)$ and $i_j \in \text{top-}k^{(\varepsilon)}(v)$. Since $\text{top-}k^{(\varepsilon)}(w) \cap \text{top-}k^{(\varepsilon)}(v) = S \not\ni i_j$, we have reached a contradiction, which completes the proof.

B.2 Proof of Prop. 5

Without loss of generality, assume $w_1 \geq \cdots \geq w_L$. Then $\{1, \ldots, k\} = \text{top-}k(w) \subseteq \mathcal{R}(w)$ by assumption, and so $|\mathcal{R}(w)| = k$ implies $\mathcal{R}(w) = \{1, \ldots, k\}$. Now fix any $\ell > k$ and define

$$v = (w_1, \ldots, w_{k-1}, w_\ell, w_{k+1}, \ldots, w_{\ell-1}, w_k, w_{\ell+1}, \ldots, w_L),$$

which is simply the vector w with kth and ℓth entries swapped. By permutation invariance of $\mathcal{R}$, this means that $\mathcal{R}(v) = \{1, \ldots, k-1, \ell\}$, and consequently $|\mathcal{R}(w) \cap \mathcal{R}(v)| = k - 1 < k$. Therefore,

$$\varepsilon \leq \|w - v\| = \sqrt{2}|w_k - w_\ell|.$$

Since $w_k \geq w_\ell$ by assumption, this means that

$$w_k \geq w_\ell + \varepsilon/\sqrt{2}.$$

This holds for every $\ell > k$, i.e., we have shown that

$$w_k \geq \max\{w_{k+1}, \ldots, w_L\} + \varepsilon/\sqrt{2}.$$

By Soloff et al. (2024a, Lemma 15), this implies that

$$\text{argmax}^{(\varepsilon)}((w_k, \ldots, w_L)) = \{1\},$$

i.e., the inflated argmax (applied to the subvector $(w_k, \ldots, w_L)$) returns a singleton set. By Prop. 7, we have

$$\text{top-}k^{(\varepsilon)}(w) = \{1, \ldots, k-1\} \cup \{k - 1 + j : j \in \text{argmax}^{(\varepsilon)}((w_k, \ldots, w_L))\} = \{1, \ldots, k-1\} \cup \{k\},$$

which completes the proof.

B.3 Proof of Prop. 7

First we will show that

$$\text{top-}k^{(\varepsilon)}(w) \subseteq \big\{1, \ldots, k-1\big\} \cup \Big\{k - 1 + j : j \in \text{argmax}^{(\varepsilon)}\big((w_k, \ldots, w_L)\big)\Big\}.$$

Fix any $j \in \text{top-}k^{(\varepsilon)}(w)$. Since the right-hand side above must include all indices $j \leq k$, we can assume $j > k$ to avoid the trivial case. By definition of $\text{top-}k^{(\varepsilon)}(w)$, we can find some $v \in C_j^{\varepsilon,k}$ with $\|w - v\| < \varepsilon$.

Next, let $v_{(1)}^{-j} \geq \cdots \geq v_{(L-1)}^{-j}$ be the order statistics of $(v_i)_{i \neq j}$, and define

$$\tilde{v} = \big(v_{(1)}^{-j}, \ldots, v_{(j-1)}^{-j}, v_j, v_{(j)}^{-j}, \ldots, v_{(L-1)}^{-j}\big) \in \mathbb{R}^L.$$

Since $w_1 \geq \cdots \geq w_L$, by the rearrangement inequality we have

$$\begin{aligned}
\|w - \tilde{v}\|^2 &= (w_j - v_j)^2 + \Big\|(w_1, \ldots, w_{j-1}, w_{j+1}, \ldots, w_L) - \big(v_{(1)}^{-j}, \ldots, v_{(j-1)}^{-j}, v_{(j)}^{-j}, \ldots, v_{(L-1)}^{-j}\big)\Big\|^2 \\
&\leq (w_j - v_j)^2 + \|(w_1, \ldots, w_{j-1}, w_{j+1}, \ldots, w_L) - (v_1, \ldots, v_{j-1}, v_{j+1}, \ldots, v_L)\|^2 \\
&= \|w - v\|^2 < \varepsilon^2.
\end{aligned}$$

Moreover, $\tilde{v}_{(k+1)} = v_{(k+1)}$ (since $\tilde{v}$ is simply a permutation of v), and therefore we have

$$\tilde{v}_j = v_j \geq v_{(k+1)} + \varepsilon/\sqrt{2} = v_{(k)}^{-j} + \varepsilon/\sqrt{2} = \max_{\ell \geq k, \ell \neq j} \tilde{v}_\ell + \varepsilon/\sqrt{2},$$

where the inequality holds since $v \in C_j^{\varepsilon,k}$, and the following step holds since we clearly must have v_j in the top k elements of v, and so $v_{(k+1)} = v_{(k)}^{-j}$. Finally, by Soloff et al. (2024a, Lemma 15), for any vector u, $\text{argmax}^{(\varepsilon)}(u) = \{j\}$ if and only if $u \in C_j^{\varepsilon,1}$. Consequently,

$$\text{argmax}^{(\varepsilon)}\big((\tilde{v}_k, \ldots, \tilde{v}_L)\big) = \{j - k + 1\},$$

i.e., the unique element selected is the one corresponding to the entry $\tilde{v}_j = v_j$.

Next, we calculate

$$\big\|(w_k, \ldots, w_L) - (\tilde{v}_k, \ldots, \tilde{v}_L)\big\| \leq \|w - \tilde{v}\| < \varepsilon.$$

Therefore, $\mathrm{argmax}^{(\varepsilon)}\big((w_k, \ldots, w_L)\big) \cap \mathrm{argmax}^{(\varepsilon)}\big((\tilde{v}_k, \ldots, \tilde{v}_L)\big) \neq \varnothing$ by (6), which now implies $j - k + 1 \in \mathrm{argmax}^{(\varepsilon)}((w_k, \ldots, w_L))$. This completes the first part of the proof.

Next we need to show the converse, i.e.,

$$\text{top-}k^{(\varepsilon)}(w) \supseteq \big\{1, \ldots, k-1\big\} \cup \Big\{k - 1 + j : j \in \mathrm{argmax}^{(\varepsilon)}\big((w_k, \ldots, w_L)\big)\Big\}.$$

Since $\text{top-}k^{(\varepsilon)}(w) \supseteq \text{top-}k(w)$ by construction, we only need to consider any $j \in \mathrm{argmax}^{(\varepsilon)}\big((w_k, \ldots, w_L)\big)$. If this holds for some j, then by definition of the inflated argmax, there must be some vector $v \in \mathbb{R}^{L-k+1}$ with

$$\|(w_k, \ldots, w_L) - v\| < \varepsilon, \;\; v_j \geq \max_{i \neq j} v_i + \varepsilon/\sqrt{2}.$$

Now define

$$\tilde{v} = (w_1, \ldots, w_{k-1}, v_1, \ldots, v_{L-k+1}).$$

Then clearly,

$$\|w - \tilde{v}\| = \|(w_k, \ldots, w_L) - v\| < \varepsilon.$$

Moreover,

$$\tilde{v}_{k-1+j} = v_j \geq \max_{i \neq j} v_i + \varepsilon/\sqrt{2} = \max_{\ell \geq k, \ell \neq k-1+j} \tilde{v}_\ell + \varepsilon/\sqrt{2} \geq \tilde{v}_{(k+1)} + \varepsilon/\sqrt{2},$$

and therefore $\tilde{v} \in C^{\varepsilon,k}_{k-1+j}$. Consequently, $k - 1 + j \in \text{top-}k^{(\varepsilon)}(w)$ by definition, which completes the proof.

B.4 Proof of Thm. 9

As discussed after the statement of the theorem, it is sufficient to verify

$$\text{For all } w, v \in \mathbb{R}^L, \text{ if } \|w - v\| < \varepsilon \text{ then } \mathrm{ranking}^{(\varepsilon)}(w) \cap \mathrm{ranking}^{(\varepsilon)}(v) \neq \varnothing,$$

since assuming this property holds, we then have

$$\frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\Big\{\mathrm{ranking}^{(\varepsilon)}(\mathcal{A}(\mathcal{D})) \cap \mathrm{ranking}^{(\varepsilon)}(\mathcal{A}(\mathcal{D}^{\setminus i})) \neq \varnothing\Big\} \geq \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\Big\{\|\mathcal{A}(\mathcal{D}) - \mathcal{A}(\mathcal{D}^{\setminus i})\| < \varepsilon\Big\} \geq 1 - \delta.$$

Fix any $w, v \in \mathbb{R}^L$ with $\|w - v\| < \varepsilon$. We will now iteratively construct a permutation $\pi \in \mathrm{ranking}^{(\varepsilon)}(w) \cap \mathrm{ranking}^{(\varepsilon)}(v)$.

First, since $\|w - v\| < \varepsilon$, it holds that $\mathrm{argmax}^{(\varepsilon)}(w) \cap \mathrm{argmax}^{(\varepsilon)}(v) \neq \varnothing$, by (6). We can therefore choose $\pi(1)$ to be any index

$$\pi(1) \in \mathrm{argmax}^{(\varepsilon)}(w) \cap \mathrm{argmax}^{(\varepsilon)}(v).$$

Next we proceed by induction. Suppose that we have defined $\pi(1), \ldots, \pi(k-1)$, and are now ready to define $\pi(k)$. Let $S_k = [L] \backslash \{\pi(1), \ldots, \pi(k-1)\}$. Then

$$\|w_{S_k} - v_{S_k}\| \leq \|w - v\| < \varepsilon,$$

and so $\mathrm{argmax}^{(\varepsilon)}(w_{S_k}) \cap \mathrm{argmax}^{(\varepsilon)}(v_{S_k}) \neq \varnothing$, by (6). In particular, we can choose some $j \in \mathrm{argmax}^{(\varepsilon)}(w_{S_k}) \cap \mathrm{argmax}^{(\varepsilon)}(v_{S_k})$. Now let $\pi(k) \in S_k$ be chosen such that $\pi(k)$ corresponds to the jth element of S_k.

Proceeding iteratively as above, we have defined $\pi \in \mathcal{S}_L$. Now we verify that $\pi \in \mathrm{ranking}^{(\varepsilon)}(w) \cap \mathrm{ranking}^{(\varepsilon)}(v)$. For each k, note that S_k is equal to some permutation of $\{\pi(k), \ldots, \pi(L)\}$. By the permutation invariance of the inflated argmax, and by definition of $\pi(k)$, we therefore have

$$1 \in \mathrm{argmax}^{(\varepsilon)}\big((w_{\pi(k)}, \ldots, w_{\pi(L)})\big).$$

Since this holds for every k, we have shown that $\pi \in \mathrm{ranking}^{(\varepsilon)}(w)$. The same argument holds for v as well, which completes the proof.

B.5 Proof of Prop. 10

Without loss of generality, assume $w_1 \geq \cdots \geq w_L$. Then $\mathrm{Id} = \mathrm{ranking}(w) \in \mathcal{R}(w)$ by assumption. Since we assume $\pi^{-1}(i) < \pi^{-1}(j)$ for all $\pi \in \mathcal{R}(w)$, this means that we must have $i < j$. Next define

$$v = (w_1, \ldots, w_{i-1}, w_j, w_{i+1}, \ldots, w_{j-1}, w_i, w_{j+1}, \ldots, w_L),$$

which is simply the vector w with ith and jth entries swapped. By permutation invariance of $\mathcal{R}$, this means that $\pi^{-1}(i) > \pi^{-1}(j)$ for all $\pi \in \mathcal{R}(v)$. Consequently $\mathcal{R}(w) \cap \mathcal{R}(v) = \varnothing$, and therefore

$$\varepsilon \leq \|w - v\| = \sqrt{2}|w_i - w_j|.$$

Since $w_i \geq w_j$ by assumption, this means that

$$w_i \geq w_j + \varepsilon/\sqrt{2}.$$

Next fix any $\pi \in \mathrm{ranking}^{(\varepsilon)}(w)$. Let $\pi^{-1}(i) = k$ and $\pi^{-1}(j) = \ell$. By definition of $\mathrm{ranking}^{(\varepsilon)}(w)$, we have

$$1 \in \mathrm{argmax}^{(\varepsilon)}\left((w_{\pi(\ell)}, \ldots, w_{\pi(L)})\right).$$

By Soloff et al. (2024a, Prop. 20), for any vector u, $\mathrm{argmax}^{(\varepsilon)}(u) \subseteq \{i : u_i > \max_{i'} u_{i'} - \varepsilon/\sqrt{2}\}$, and therefore this means that

$$w_{\pi(\ell)} > \max_{\ell' \geq \ell} w_{\pi(\ell')} - \varepsilon/\sqrt{2}.$$

On the other hand, from our work above we know that

$$w_{\pi(k)} = w_i \geq w_j + \varepsilon/\sqrt{2} = w_{\pi(\ell)} + \varepsilon/\sqrt{2}.$$

This proves that we cannot have $k \geq \ell$—and consequently, $\pi^{-1}(i) < \pi^{-1}(j)$, as desired.

B.6 Proof of Prop. 12

Suppose $\pi^{-1}(j) = k$. First, by Prop. 13, it holds that $j = \pi(k) \in \text{top-}k^{(\varepsilon)}(w)$. This establishes the lower bound.

Next, by definition of the inflated full ranking, for every $\ell \leq k$ we have

$$1 \in \mathrm{argmax}^{(\varepsilon)}\left((w_{\pi(\ell)}, \ldots, w_{\pi(k)}, \ldots, w_{\pi(L)})\right).$$

By Soloff et al. (2024a, Prop. 20), for any vector v, $\mathrm{argmax}^{(\varepsilon)}(v) \subseteq \{i : v_i > \max_{i'} v_{i'} - \varepsilon/\sqrt{2}\}$, and consequently, we have

$$w_{\pi(\ell)} > \max_{\ell' = \ell, \ldots, L} w_{\pi(\ell')} - \varepsilon/\sqrt{2} \geq w_{\pi(k)} - \varepsilon/\sqrt{2} = w_j - \varepsilon/\sqrt{2}.$$

Since this holds for every $\ell \leq k$, we therefore have

$$\sum_{\ell=1}^{L} \mathbf{1}\{w_\ell > w_j - \varepsilon/\sqrt{2}\} = \sum_{\ell=1}^{L} \mathbf{1}\{w_{\pi(\ell)} > w_j - \varepsilon/\sqrt{2}\} \geq k,$$

which completes the proof of the upper bound.

B.7 Proof of Prop. 13

First, fix any $\pi \in \mathrm{ranking}^{(\varepsilon)}(w)$ and any $j \leq k$. Then $1 \in \mathrm{argmax}^{(\varepsilon)}((w_{\pi(j)}, \ldots, w_{\pi(L)}))$, by definition of the inflated full ranking. Defining $i_1 = \pi(j), \ldots, i_{L-j+1} = \pi(L)$, and applying Lemma 16, we see that $\pi(j) \in \text{top-}k^{(\varepsilon)}(w)$. This proves that

$$\text{top-}k^{(\varepsilon)}(w) \supseteq \cup_{\pi \in \mathrm{ranking}^{(\varepsilon)}(w)} \{\pi(1), \ldots, \pi(k)\}.$$

Now we prove the converse. Without loss of generality assume $w_1 \geq \cdots \geq w_L$. Fix any $j \in \text{top-}k^{(\varepsilon)}(w)$. If $j \leq k$, then $j \in \{\pi(1), \ldots, \pi(k)\}$ for $\pi = \text{Id}$, which satisfies $\pi = \text{ranking}(w) \in \text{ranking}^{(\varepsilon)}(w)$. If instead $j > k$ then define

$$\pi = (1, \ldots, k-1, j, k, \ldots, j-1, j+1, \ldots, L),$$

that is, the permutation π places w_j into position k and otherwise sorts the entries of w from largest to smallest, so that $j \in \{\pi(1), \ldots, \pi(k)\}$. For each $\ell \neq k$, we have $1 \in \text{argmax}^{(\varepsilon)}((w_{\pi(\ell)}, \ldots, w_{\pi(L)}))$, since the first entry of this subvector is its maximum. For $\ell = k$, we have $j - k + 1 \in \text{argmax}^{(\varepsilon)}((w_k, \ldots, w_L))$, by Prop. 7. By permutation invariance, then, $1 \in \text{argmax}^{(\varepsilon)}((w_j, w_k, \ldots, w_{j-1}, w_{j+1}, \ldots, w_L))$. This verifies that $\pi \in \text{ranking}^{(\varepsilon)}(w)$. We have therefore proved that

$$\text{top-}k^{(\varepsilon)}(w) \subseteq \cup_{\pi \in \text{ranking}^{(\varepsilon)}(w)} \{\pi(1), \ldots, \pi(k)\}.$$

B.8 Proof of Prop. 14

First fix any $\pi \in \text{ranking}^{(\varepsilon)}(w)$. Then by definition, for each $k \in [L]$ we have

$$1 \in \text{argmax}^{(\varepsilon)}((w_{\pi(k)}, \ldots, w_{\pi(L)})).$$

By Lemma 16, this means that $\pi(k) \in \text{top-}k^{(\varepsilon)}(w)$. Since this holds for all k, we have proved that $\pi \in \mathcal{R}(w)$—and therefore, $\text{ranking}^{(\varepsilon)}(w) \subseteq \mathcal{R}(w)$.

Next, consider the following example: let $L = 3$, $\varepsilon = 1$, and

$$w = (1, 0.5, 0).$$

Then we can calculate

$$\text{top-}k^{(\varepsilon)}(w) = \begin{cases} \{1, 2\}, & k = 1, \\ \{1, 2, 3\}, & k = 2, \\ \{1, 2, 3\}, & k = 3. \end{cases}$$

Choosing $\pi = (2, 3, 1)$, we therefore see that $\pi \in \mathcal{R}(w)$. However,

$$\text{ranking}^{(\varepsilon)}(w) = \big\{(1, 2, 3), (2, 1, 3), (1, 3, 2)\big\} \not\ni \pi.$$

B.9 Proof of Lemma 16

Without loss of generality, assume $w_1 \geq \cdots \geq w_L$. First, by Prop. 7, we have

$$\text{top-}k^{(\varepsilon)}(w) = \big\{1, \ldots, k-1\big\} \cup \Big\{k - 1 + j : j \in \text{argmax}^{(\varepsilon)}\big((w_k, \ldots, w_L)\big)\Big\}.$$

Now fix any $\ell \in \text{top-}k^{(\varepsilon)}(w)$. If $\ell \leq k - 1$, then let $i_1 = \ell, \ldots, i_{L-\ell+1} = L$. Then

$$1 \in \text{argmax}^{(\varepsilon)}((w_{i_1}, \ldots, w_{i_{L-\ell+1}})),$$

since the first entry of this subvector is the largest. If instead $\ell \geq k$, then we must have

$$\ell = k - 1 + j \text{ where } j \in \text{argmax}^{(\varepsilon)}\big((w_k, \ldots, w_L)\big).$$

Now let $i_1 = k, \ldots, i_{L-k+1} = L$. Then

$$j \in \text{argmax}^{(\varepsilon)}((w_{i_1}, \ldots, w_{i_L})).$$

Combining these cases, we have proved that

$$\text{top-}k^{(\varepsilon)}(w) \subseteq \bigcup_{\substack{L-k+1 \leq \ell \leq L, \\ \text{distinct } i_1, \ldots, i_\ell \in [L]}} \Big\{i_j : j \in \text{argmax}^{(\varepsilon)}\big((w_{i_1}, \ldots, w_{i_\ell})\big)\Big\}.$$

Now we prove the converse. Fix any distinct $i_1, \ldots, i_\ell$ for $\ell \geq L - k + 1$, and suppose $j \in \text{argmax}^{(\varepsilon)}\big((w_{i_1}, \ldots, w_{i_\ell})\big)$. We now need to show that $i_j \in \text{top-}k^{(\varepsilon)}(w)$. By definition of the inflated argmax, there is some vector $v \in \mathbb{R}^\ell$, with $v_j \geq \max_{i \neq j} v_i + \varepsilon/\sqrt{2}$, such that

$$\|(w_{i_1}, \ldots, w_{i_\ell}) - v\| < \varepsilon.$$

Now define $\tilde{v} \in \mathbb{R}^L$ with entries

$$\tilde{v}_{i_1} = v_1, \ \ldots, \tilde{v}_{i_\ell} = v_\ell,$$

and $\tilde{v}_i = w_i$ for all $i \notin \{i_1, \ldots, i_\ell\}$. Then

$$\|w - \tilde{v}\| = \|(w_{i_1}, \ldots, w_{i_\ell}) - v\| < \varepsilon.$$

Moreover,

$$\begin{aligned}\tilde{v}_{i_j} = v_j &\geq \max\{v_1, \ldots, v_{j-1}, v_{j+1}, \ldots, v_\ell\} + \varepsilon/\sqrt{2} \\ &= \max\{\tilde{v}_{i_1}, \ldots, \tilde{v}_{i_{j-1}}, \tilde{v}_{i_{j+1}}, \ldots, \tilde{v}_{i_\ell}\} + \varepsilon/\sqrt{2} \geq \tilde{v}_{(L-\ell+2)} + \varepsilon/\sqrt{2} \geq \tilde{v}_{(k+1)} + \varepsilon/\sqrt{2}.\end{aligned}$$

Therefore, $\tilde{v} \in C_{i_j}^{\varepsilon,k}$, and consequently we have $i_j \in \text{top-}k^{(\varepsilon)}(w)$, as desired. This verifies that

$$\text{top-}k^{(\varepsilon)}(w) \supseteq \bigcup_{\substack{L-k+1 \leq \ell \leq L, \\ \text{distinct } i_1, \ldots, i_\ell \in [L]}} \Big\{ i_j : j \in \text{argmax}^{(\varepsilon)}\big((w_{i_1}, \ldots, w_{i_\ell})\big) \Big\},$$

which completes the proof.

C Additional experiments

Under the same settings as in Section 6.1, here we present additional experiment results for different values of k. We also plot the empirical distribution of δ_j, across trials $j = 1, \ldots, N$, for each value of k. Overall, we observe qualitatively similar results across the different values of k, although both methods are more stable for smaller values of k (since there is less ambiguity among the top few movies, than for larger k).

k	Methods	$\max_{j \in [N]} \delta_j$	$\frac{1}{N}\sum_{j \in [N]} \delta_j$	$\frac{1}{N}\sum_{j \in [N]} \text{Jaccard}_j$	$\frac{1}{N}\sum_{j \in [N]} \text{Size}_j$
$k = 5$	top-k	0.5350	0.0222 (0.0057)	0.9926 (0.0019)	5.00 (0.0000)
	top-$k^{(\varepsilon)}$	0.0210	0.0036 (0.0004)	0.9925 (0.0010)	5.33 (0.0567)
$k = 10$	top-k	0.6560	0.0828 (0.0125)	0.9847 (0.0023)	10.00 (0.0000)
	top-$k^{(\varepsilon)}$	0.0300	0.0059 (0.0006)	0.9911 (0.0008)	10.75 (0.0942)
$k = 20$	top-k	0.8530	0.1205 (0.0136)	0.9876 (0.0015)	20.00 (0.0000)
	top-$k^{(\varepsilon)}$	0.0380	0.0094 (0.0009)	0.9906 (0.0006)	21.22 (0.1101)
$k = 50$	top-k	0.8940	0.2666 (0.0230)	0.9873 (0.0012)	50.00 (0.0000)
	top-$k^{(\varepsilon)}$	0.0800	0.0135 (0.0016)	0.9922 (0.0003)	52.30 (0.1712)
$k = 100$	top-k	0.9150	0.3928 (0.0235)	0.9893 (0.0008)	100.00 (0.0000)
	top-$k^{(\varepsilon)}$	0.0630	0.0122 (0.0011)	0.9939 (0.0003)	103.20 (0.1766)

Table 3: Results on the Netflix Prize dataset (see Section 6.1 for details). Evaluation results under various metrics are reported in the table, with standard errors for the averages shown in parentheses. (The data for $k = 20$ is exactly as reported in Section 6.1.)

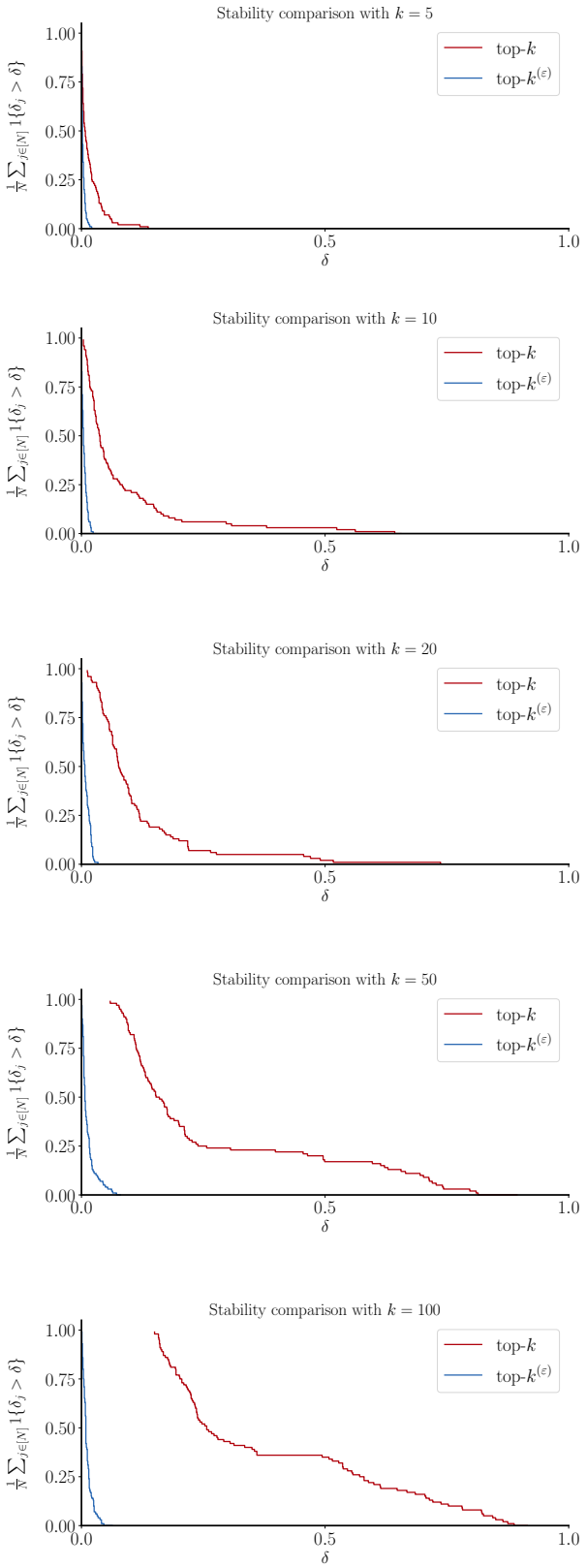


Figure 1: Results on the Netflix Prize dataset (see Section 6.1 for details). The plots show the distribution of δ_j, across trials $j = 1, \ldots, N$, for each choice of k and for each of the two methods.