[02/26/2025], Lecture Note: Fair Ranking – Part II: Learned Ranking

[Kodati Shruthi, Anand Meena]

CS 516: Responsible Data Science and Algorithmic Fairness; Spring 2025
Abolfazl Asudeh; www.cs.uic.edu/~asudeh/teaching/archive/cs516spring25/

1 Introduction

Modern applications such as search engines, recommendation systems, college admissions, and job recruitment increasingly rely on ranking algorithms to order items by relevance or quality. Unlike classification, where decisions are made independently for each item, Ranking is inherently competitive; each candidate's or document's position depends on the relative scores compared to others. Consequently, fairness in the Ranking is a multidimensional challenge that considers individual scores and the exposure and opportunity allocated between groups. These notes provide a comprehensive overview of the following:

- The conceptual and technical differences between Ranking and classification.
- Various fairness definitions for ranking, including top-k parity, moving top-k, exposure fairness, and utility parity.
- Learning-to-rank (LTR) techniques that transform ranking into supervised learning problems.
- Recent fairness-aware approaches in supervised learning-to-rank and recommendation systems, including pre-processing, in-processing, and postprocessing methods.
- Illustrative class examples showing practical scenarios and challenges.

This document synthesizes lecture content, class notes, and insights from the survey paper by Zehlike *et al.* (2022) on fairness in Ranking.

2 Ranking vs. Classification

2.1 Core Differences

Ranking involves ordering items according to their scores, with the position of each item influenced by the scores of all other items in the dataset. In contrast, **classification** assigns a label to each item independently of the others. This difference has important implications:

- In the Ranking, even if multiple items are of high quality, only a limited number of positions (e.g., top-10 results) can be highlighted, making relative performance crucial.
- In classification, each item is judged on its own merits, without competition for a fixed number of "slots" or positions.

2.2 Examples

- **Hotel Ratings:** Although hotels are often categorized as five-star, four-star, etc., this task resembles classification since each hotel is assigned an independent label based on predetermined criteria. The evaluation does not require comparing hotels relative to one another to determine their order; each hotel's label stands on its own, which is characteristic of classification.

- **Search Engine Results:** In contrast, search engines rank documents by relevance. Even if many documents are relevant to a query, only the top few positions are presented prominently. The Ranking is competitive and depends on the relative scores of documents, which makes user attention and exposure critical factors.

3 Fairness in Ranking

Fairness in Ranking ensures that the ordering process does not systematically disadvantage certain groups. This is a complex issue because fairness must consider the numerical scores and the relative positions determining exposure.

3.1 Top-k and Moving Top-k Fairness

Top-k Fairness requires that the proportion of items from each group in the top-k positions reflects their overall presence. For example, if 30% of candidates belong to a protected group, then roughly 30% of the top-k should be from that group.

Moving Top-k Fairness extends this idea by evaluating fairness over a range of values for k. This approach incorporates discounting factors (e.g., $\frac{1}{\log(k+1)}$) to capture the diminishing importance of lower-ranked positions. This ensures that fairness is maintained consistently as the Ranking extends rather than at a single arbitrary cutoff.

3.2 Exposure Fairness

Exposure fairness is based on the observation that items higher in a ranking receive disproportionately more attention. A typical formulation assigns an exposure value to each rank:

$$v(i) = \frac{1}{\log(i+1)},$$

This reflects that the first result is far more visible than the tenth. Fairness definitions based on exposure typically fall into two categories:

- **Probability-Based Fairness:** This approach employs statistical tests to assess whether a given ranking could have emerged from a fair random process (e.g., coin tosses).
- **Exposure-Based Fairness:** This method directly computes the expected attention each item or group receives and enforces that exposure is allocated proportionately based on either merit or group size.

3.3 Utility Parity

Each candidate has a relevance score $r(e, q)$ for a given query q in information retrieval. The overall utility of a ranking is defined as:

$$U = \sum_{i=1}^{n} v(i) \cdot r(e_i, q).$$

Utility parity ensures that the cumulative utility for each group is proportional to its size so that highly relevant candidates are not systematically underexposed.

4 Learning to Rank (LTR)

Learning-to-rank (LTR) is a supervised learning framework that aims to learn a ranking function from data. Unlike traditional regression or classification tasks, LTR must capture the relative ordering of items.

4.1 Pairwise Transformation

One popular strategy is to convert the ranking task into a set of pairwise comparisons:

- Given a ranking (e.g., $e_1 > e_2 > \cdots > e_n$), form pairs of items.
- For each pair (e_i, e_j), assign a label $+1$ if e_i should be ranked higher than e_j, and -1 otherwise.
- The feature vector for each pair is constructed by concatenating the features of both items, resulting in a $2d$-dimensional vector if each item has d features.
- A classifier is then trained on these pairs to predict the correct ordering, thereby allowing the model to generalize and rank new items by comparing their concatenated feature representations.

4.2 Geometric Interpretation and Minimal Adjustments

Consider a linear ranking function of the form:

$$f(x) = \theta_1 x_1 + \theta_2 x_2.$$

This function projects candidate feature vectors onto the weight vector θ. Key insights include:

- **Partitioning Function Space:** The space of all possible weight vectors can be divided by hyperplanes representing boundaries where the relative ordering of a candidate pair changes.
- **Minimal Adjustments:** If an initial ranking is determined to be unfair, minor adjustments to the weights (for instance, modifying θ_1 or θ_2 by a slight margin) can shift the Ranking into a region where fairness criteria (such as balanced top-k representation) are satisfied. This approach seeks to preserve the original merit ordering as much as possible while achieving fairness.

5 Illustrative Examples from Class

The lecture included several examples that illustrate these concepts in practical scenarios. The following expanded descriptions provide further detail:

Example 1: Hotel Ratings

In this example, hotels are rated with star classifications (e.g., five-star, four-star). The task is a classification problem because each hotel receives an independent label based on predetermined quality criteria. There is no inherent competition between hotels in Ranking; each hotel is evaluated on a fixed scale. Consequently, this task does not involve ordering hotels by relative merit; thus, exposure considerations typical in ranking problems are absent.

Example 2: University Admissions

This example examines the Ranking of university applicants. Each candidate possesses features such as high school GPA and GRE scores and a sensitive attribute such as gender. An initial ranking function might be defined as:

$$f(x) = \theta_1\,\mathrm{GPA} + \theta_2\,\mathrm{GRE},$$

with $\theta_1 = \theta_2 = 1$. In the class, it was discussed that even minor adjustments (for example, setting $\theta_1 = 0.9$ and $\theta_2 = 1.1$) can alter the ordering significantly. Such adjustments may impact the fairness of the Ranking—if, for instance, one gender systematically benefits from certain test score distributions. The objective is to determine the minimal change required to achieve a fair ranking where one group does not dominate the top positions, thus ensuring equitable access to opportunities.

Example 3: Pairwise Comparison for LTR

This example demonstrates converting a ranking list into a pairwise dataset, a common strategy in LTR. Consider ranking applicants A, B, and C, where A is ranked above B and B above C. The pairwise comparisons generated from this Ranking are:

- (A, B): Label +1 indicating A should be ranked higher than B.
- (A, C): Label +1 indicating A should be ranked higher than C.
- (B, C): Label +1 indicating B should be ranked higher than C.

The features of each candidate (A, B, and C) are concatenated for each pair to form a training instance. This approach allows the learning algorithm to focus on relative differences between candidates rather than absolute scores. When new applicants are introduced, the model can apply these learned pairwise relationships to predict the appropriate ranking order, even if the exact pair combinations were not encountered during training.

6 Fairness in Supervised Learning-to-Rank and Recommender Systems

Recent research extends LTR methods to incorporate fairness constraints directly within the learning process. The survey by Zehlike *et al.* (2022) categorizes these approaches into pre-processing, in-processing, and postprocessing methods.

6.1 Pre-Processing Methods

Pre-processing methods mitigate bias by transforming the training data before model training. A prominent example is:

- $iFair$**:** This method learns a fair representation $\tilde{X}$ of the original data X. It maps the original feature vector to a new space where similar individuals (based on non-sensitive attributes) remain close while any correlations with sensitive attributes are minimized. The mapping is achieved by minimizing a combined loss function:

$$L = \lambda \cdot L_{util}(X, \tilde{X}) + \mu \cdot L_{fair}(X, \tilde{X}),$$

  where L_{util} ensures retention of useful information and L_{fair} enforces fairness by preserving pairwise distances on non-sensitive features.

6.2 In-Processing Methods

In-processing methods incorporate fairness constraints directly into the training objective. Two notable approaches are:

- $DELTR$**:** This approach extends the ListNet algorithm by including a fairness penalty term that targets exposure disparities between protected and non-protected groups:

$$L_{DELTR}(Y, \hat{Y}) = L(Y, \hat{Y}) + \gamma\, D(\hat{Y}),$$

  where $D(\hat{Y})$ quantifies the difference in exposure. The penalty encourages the model to learn representations that reduce bias in exposure.

- $Fair$**-***PG***-***Rank***:** This method enforces fairness by ensuring that candidates receive exposure proportional to their utility. It introduces fairness constraints at an individual level (by comparing pairwise exposure discrepancies) and at a group level, aligning the distribution of attention with candidate merit.

6.3 Post-Processing Methods

Postprocessing methods modify the final ranking output to improve fairness. Examples include:

- FA^*IR**:** This algorithm reorders the Ranking so that the number of protected candidates in every prefix meets a predefined minimum proportion p. A pre-computed table based on the binomial cumulative distribution enforces these constraints.
- **LinkedIn's Fairness-Aware Ranking:** Developed for practical deployment, this method sets the minimum and maximum thresholds for each group at every rank position and selects the next candidate based on whether the group's quota is met.
- **Continuous Fairness with Optimal Transport (**CFA_θ**):** This framework interpolates between two fairness paradigms—*WYSIWYG* (meritocratic) and *WAE* (correcting for bias)—using a fairness parameter $\theta \in [0, 1]$. Adjustments to score distributions are made to bring them closer together, thereby achieving statistical parity.

6.4 Evaluation Metrics

Accurate evaluation is critical in LTR. Common metrics include:

- **Normalized Discounted Cumulative Gain (NDCG):** This metric measures how closely the predicted Ranking approximates an ideal ranking by applying a logarithmic discount to lower positions:

$$\mathrm{NDCG}_k = \frac{1}{\mathrm{IDCG}_k} \sum_{i=1}^{k} \frac{\hat{Y}_{\tau(i)}}{\log(i+1)}.$$

- **Mean Average Precision (MAP):** This metric averages the precision at each relevant cutoff, providing a comprehensive summary statistic across multiple queries.

Additionally, fairness metrics are employed to quantify disparities in exposure or utility between groups. For instance, group exposure disparity is expressed as:

$$D(G_1, G_2) = \left| \frac{1}{|G_1|} \sum_{a \in G_1} \mathrm{Exposure}(a) - \frac{1}{|G_2|} \sum_{b \in G_2} \mathrm{Exposure}(b) \right|.$$

7 Conclusion

Fairness in Ranking extends beyond merely ordering items by merit; it also involves ensuring equitable exposure and opportunity across all groups. By transforming ranking tasks into pairwise comparisons, applying minimal adjustments to model parameters, and integrating fairness constraints directly into the learning objective (or reordering predictions post hoc), modern systems can mitigate bias while maintaining high relevance. The methods surveyed—including $iFair$, $DELTR$, *Fair-PG-Rank*, FA^*IR, and CFA_θ—illustrate the diverse approaches available for achieving fairness in real-world systems. Although these techniques often involve trade-offs between accuracy and fairness, ongoing research aims to optimize these trade-offs to develop systems that are both effective and equitable.

Reference:

M. Zehlike, K. Yang, and J. Stoyanovich, *Fairness in Ranking, Part II: Learning-to-Rank and Recommender Systems*, ACM Comput. Surv. 55, 6, Article 117 (December 2022), 41 pages.
https://doi.org/10.1145/3533380