# DISTINGUISHING FEATURE MODEL FOR RANKING FROM PAIRWISE COMPARISONS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

We consider the problem of ranking a set of items from pairwise comparisons among them when the underlying preferences are intransitive in nature. Intransitivity is a common occurrence in real world data sets and we introduce a flexible and natural parametric model for pairwise comparisons that we call the *Distinguishing Feature Model* (DF) to capture this. Under this model, the items have an unknown but fixed embedding and the pairwise comparison between a pair of items depends probabilisitically on the feature in the embedding that can best distinguish the items. We study several theoretical properties including how it generalizes the popular transitive Bradley-Terry-Luce model. With just an embedding dimension $d = 3$, we show that the proposed model can capture arbitrarily long cyclic dependencies. Furthermore, we explicitly show the type of preference relations that cannot be modelled under the DF model for $d = 3$. On the algorithmic side, we propose a Siamese type neural network based algorithm which can learn to predict well under the DF model while at the same time being interpretable in the sense that the embeddings learnt can be extracted directly from the learnt model. Our experimental results show that the model outperforms standard baselines in both synthetic and real world ranking datasets.

## 1 INTRODUCTION

We consider the problem of learning to rank from pairwise comparisons where one is given a set of pairwise comparisons over a bunch of items and the goal is to obtain a *good* global ranking over these items. Typically, one assumes a statistical model for comparisons where when a pair of items $i$ and $j$ are compared, item $i$ is preferred to item $j$ with some underlying fixed but unknown probability $P_{ij}$. The goodness of a ranking over the items is then given by how closely the ranking aligns with the underlying probability matrix $\mathbf{P}$ typically in terms of pairwise agreement. Different pairwise comparison models make different parametric assumptions about $\mathbf{P}$ and attempt to learn the parameters of the model. A typical score based assumption leads to the popular Bradley-Terry-Luce Model (BTL) where $P_{ij} = \frac{w_i}{w_i+w_j}$ where $w_i \in \mathbb{R}_+$ is the score of item $i$. In such a model, it is known that one can learn effectively from $\mathcal{O}(n \log n)$ pairs uniformly chosen from the set of all $\binom{n}{2}$ pairs and where each chosen pair is compared only $\mathcal{O}(\log n)$ times. While the sample complexity of learning is attractive, the downside of the BTL model is that it can model only transitive preferences i.e., for any three items $i, j, k$, $P_{ij} \geq 0.5 \ \& \ P_{jk} \geq 0.5 \implies P_{ik} \geq 0.5$. Real world preferences are almost always intransitive and hence the transitive models are often not good enough for modelling them. There are several hypotheses as to why human reasoning oftentimes lead to intransitivity. While some argue using the inherent irrationality in human judgement, other it by saying that humans often associate items with multiple features or scores and decide to prefer one over the other based on some function of all these features. The latter explanation is what we will lean towards in this work.

In this work, we propose a flexible model for pairwise comparisons called the Distinguishing Feature (DF) model. The DF model is based on the simple hypothesis that when two items are being compared by a human, there are several implicit features of these items that are considered and the preference is decided based on the *most distinguishing feature* among all features. For instance, if the items being compared are mobile phones, the scores/features to be considered may include price, battery life, weight and/or some weighted combination of these (all normalized to have the

same scale) and the comparison between two mobile phones will depend only on the distinguishing feature i.e, the feature whose absolute difference of scores is the largest. Once the distinguishing feature is identified, the preference is a probabilistic choice that depends on the exact values of the items for the distinguishing feature. We stress that the features that the model uses are implicit i.e., they are not necessarily known to an algorithm that attempts to learn a ranking from pairwise comparisons. As we will see, this is a key property that makes the model much more flexible than recently proposed models such as the salient feature model where the features are assumed to be known.

We investigate the proposed DF model by first theoretically understanding the type of preferences that can be realized under the model. We first show that even with just 3 features per item, the DF model can model intransitive preferences i.e., cyclic preferences of arbitrary lengths. We explicitly identify a preference tournament on $8$ nodes that cannot be realized using only $3$ features. We also explore the representation power of the DF model by mapping it to a edge colouring problem on tournaments and derive several interesting sufficient conditions on the coloring that ensures representability.

On the learning side, we develop an explainable neural network based algorithm (DF-Learn) to learn the implicit features from pairwise comparisons generated according to the DF model. DF-Learn is competitive when compared to a standard black box neural network model in terms of it's prediction accuracy while at the same time is designed such that the features corresponding to each item can be easily extracted from the architecture. Furthermore, we demonstrate that DF-Learn requires very few number of comparisons to learn a good predictive model, thus being sample efficient.

Finally, we demonstrate the power of the DF model on both synthetic and real world datasets. As we will see in the experimental results, the DF learn algorithm performs exceedingly well in terms of ranking quality and prediction accuracy when compared to the state of the art models including the Salient Feature model based algorithm and Rank Centrality.

## 2 Related Work

The problem of ranking from pairwise comparisons has been studied extensively for several decades in several areas including theoretical computer science, AI/ML, social choice, operations research, etc. We focus here on work that is relevant to our approach.

**Prior work on learning from Transitive pairwise Comparisons Models:** A vast number of works have considered learning from transitive pairwise models especially focusing on the Bradley-Terry-Luce (BTL) model. We discuss here only those approaches that are closely related to either our models or our approaches.The classical statistics approach is to pose the problem in the estimation framework and it is known that the maximum likelihood estimator for the BTL scores asymptotically converges to the underlying parameters of the model. Shah et al. (2015) work in a min-max framework and give tight upper and lower bounds for estimation of the underlying parameter vector when data is generated from the BTL or closely related Thurstone model and Hajek et al. (2014) work in the partial ranking setting. Spectral ranking was studied in Negahban et al. (2015) where Rank-centrality, an algorithm that produces good rankings from $O(nlog(n)^2)$ comparisons was introduced. Several improvements to the vanilla spectral ranking have appeared such as the accelerated spectral ranking Agarwal et al. (2018). Rajkumar & Agarwal (2014) study several algorithms including the Borda count algorithm and give statistical performance guarantees for optimal recovery. Matrix completion based approaches for rank aggregation were explored in Gleich & Lim (2011). The work in Rajkumar & Agarwal (2016) is where the low rank pairwise comparison model was introduced and a matrix completion based algorithm was developed. However, the model considered low rank preferences with an additional transitivity constraint and showed sample complexity bound for recovering an almost optimal ranking for the same. In this work, we remove the transitivity restriction for rank 2 models and allow for potential cycles.

**Prior work on Intransitive Preference Modelling:** While the BTL model can be seen as using a 1 dimensional embedding of the item using a score vector, studies have considered higher dimensional embeddings. A 2 dimensional embedding was considered in [2]. While their model can give rise to cyclic tournaments, it is not clear what type of cycles are possible. [6; 7] propose

the Majority vote model which is a random utility model (RUM) with a $d$ dimensional feature embedding for each item. The Majority vote model is powerful enough to produce arbitrarily long cycles and can express any probability sub-matrix over a fixed triplet. The Blade-Chest inner (BCI) model [3] embeds each item into two $d$ dimensional vectors (blade vector and a chest vector) and a score vector $\mathbf{s}$ where the probability of $i$ being preferred over $j$ depends on $< i_{\text{chest}}, j_{\text{blade}} > - < i_{\text{blade}}, j_{\text{chest}} > + s_i - s_j$. [10]. Previous work [4,10] have proposed matrix completion based algorithms to obtain optimal ranking for the LRPR type models assuming transitivity of preferences. In this work, we make no such assumptions. More recently, the context dependent salient feature model was introduced where the preference probabilities for a pair of items depends on a subset of items that are specific to the pair. In this work, we don't require features to be available alongwith the items. However, if features are available, our algorithms will still learn an embedding from the feature space to an embedding space automatically.

## 3 Problem Setting and Preliminaries

Let $[n] = \{1, 2, \ldots, n\}$ be a set of items that need to be ranked. We assume that the learner is given a set of $m$ pairwise comparisons $\{i_k, j_k, y_k\}$ where $k = 1, \ldots, m$, $i_k, j_k \in [n]$ and $y_k \in \{0, 1\}$ for all $k$. For each $k$, $(i_k, j_k)$ refers to the pair of items that were compared and $y_k = 1$ indicates that item $i_k$ was preferred to $j_k$ and $y_k = 0$ indicates otherwise. The goal of the learner is to produce a *good* global ranking over the set of items.

**Probability Preference Matrix:** We assume that whenever two items $i$ and $j$ are compared, item $i$ is preferred to item $j$ with probability $P_{ij}$. Thus for all $k$, $y_k$ is a Bernoulli random variable with proability $P_{i_k j_k}$. We refer to the matrix $\mathbf{P} \in [0, 1]^{n \times n}$ as the probability preference matrix. We have $P_{ij} + P_{ji} = 1 \ \forall i, j$. We assume that $P_{ij} \neq 0.5 \ \forall i \neq j$.

**Performance Measure:** Let $\mathbb{S}_n$ denote the set of all permutations/rankings on $n$ items. Given a ranking $\sigma \in \mathbb{S}_n$, the performance measure of $\sigma$ with respect to the underlying probability preference matrix $\mathbf{P}$ is given by the *pairwise disagreement error* which is defined as follows:

$$dis(\sigma, \mathbf{P}) = \frac{1}{\binom{n}{2}} \sum_{i<j} \mathbb{I}\big((P_{ij} > 0.5 \ \& \ \sigma(i) < \sigma(j)) \ || \ \mathbb{I}((P_{ij} < 0.5 \ \& \ \sigma(i) > \sigma(j))\big) \tag{1}$$

Here $\sigma(i)$ is the rank of the $i$-th item and $\mathbb{I}(\cdot)$ is the indicator function. In words, the pairwise disagreement error captures the fraction of pairs $(i, j)$ for which the permutation $\sigma$ ranks item $i$ above item $j$ whereas the probability of item $i$ being preferred over item $j$ is less than $0.5$ and vice versa.

An algorithm that outputs a ranking can in general not be able to achieve $0$ pairwise disagreement error no matter which ranking it outputs. This is true if the underlying $\mathbf{P}$ has cyclic or intransitive preferences. In general, even if the underlying $\mathbf{P}$ is known, finding the $\sigma$ that minimizes the pairwise disagreement error is a NP-hard problem . Thus, we will not restrict our algorithms to only output a ranking but allow for algorithms to predict pairwise probabilities for any given pair of items. Note that such algorithms may be able to predict cyclic dependencies correctly. Thus, another measure that we will use to measure the performance of an algorithm is the fraction of pairwise preferences predicted incorrectly. Let $\hat{\mathbf{P}}$ be the predicted probability preference matrix where $\hat{P}_{ij}$ is the predicted pairwise preference probability for the pair $(i, j)$. The pairwise probability disagreement is defined as below:

$$disProb(\hat{\mathbf{P}}, \mathbf{P}) = \frac{1}{\binom{n}{2}} \sum_{i<j} \mathbb{I}\big((P_{ij} > 0.5 \ \& \hat{P}_{ij} < 0.5) \ || \ \mathbb{I}((P_{ij} < 0.5 \ \& \hat{P}_{ij} > 0.5)\big) \tag{2}$$

Note that $dis(\sigma, \mathbf{P}) = disProb(\hat{\mathbf{P}}, \mathbf{P})$ for algorithms which only output rankings [1] whereas they may be different for algorithms which learn to predict probabilities given any pair of items.

[1] This is true because for algorithms that output only a ranking $\sigma$, we assume $\hat{P}_{ij} = \mathbb{I}(\sigma(i) < \sigma(j))$.

## 4 DISTINGUISHING FEATURE (DF) MODEL FOR PAIRWISE COMPARISONS

We now introduce the distinguishing feature model for pairwise comparisons. The model assumes that each item $i$ is associated with an embedding $\mathbf{e}_i \in \mathbb{R}^d$ in some dimension $d$. When two items $i$ and $j$ are compared, a two step procedure if followed to decide the preferred item. In the first step, the feature which contributes to the highest absolute difference is calculated as follows:

$$k^* = \arg\max_k |e_{ik} - e_{jk}|$$

In the above equation, ties are broken arbitrarily.

In the second step, the preference probability is calculated as follows:

$$P_{ij} = \phi(e_{ik^*} - e_{jk^*})$$

where $\phi$ is a probability link function that satisfies $\phi(0) = 0.5$, $\lim_{x\to\infty} \phi(x) = 1$ and $\lim_{x\to-\infty} \phi(x) = 0$ and $\phi(x) \in [0, 1] \forall x \in \mathbb{R}$. A simple example of a probability link function is the logit function defined as $\phi(x) = \frac{1}{1+e^{-x}}$.

We say that a probability preference matrix $\mathbf{P}$ satisfies the Distinguishing Feature model with dimension $d$ if there exists a set of $d$ dimensional embeddings for the items such that $P_{ij}$ can be obtained as described above for all $i, j$.

## 5 THEORETICAL ASPECTS OF DF MODEL

In this section, we prove several theoretical properties of the Dinstinguishing Feature model. We begin by setting some notation that will be useful to state our results.

Let $\mathbf{P} \in [0, 1]^{n\times n}$ be a probability preference matrix. We denote by $\mathbf{T}(\mathbf{P})$ the tournament on $n$ nodes associated with $\mathbf{P}$ where $\mathbf{T}(\mathbf{P})$ is a complete directed graph and there is an edge from node $i$ to node $j$ in $\mathbf{T}(\mathbf{P})$ if and only if $P_{ij} > 0.5$. For a general tournament $\mathbf{T}$, we will say $i \succ_{\mathbf{T}} j$ if and only if there is an edge from $i$ to $j$ in $\mathbf{T}$.

We first begin with a couple of results that show that our model can effectively subsume several popular and recent models for pairwise comparisons including the BTL model and the context dependent Salient feature model.

**Theorem 1.** *For $d = 1$, the Distinguishing Feature model with the logit link function is exactly same as the Bradley-Terry-Luce model.*

*Proof.* When $d = 1$, the only feature present is indeed the distinguishing feature as well. Thus, the feature can be thought of as a score for each item and the item with the higher score dominates the one with lower score. The exact probability with which it dominates is given by $logit(s_i - s_j)$ where $\mathbf{s} \in \mathbb{R}^n$ is the score vector. This is indeed equivalent to a BTL model where the score for the $i$-th item is given by $e^{s_i}$ for every item $i$. □

Next, we take a closer look at the preferences that can be modelled using the DF model. One of the important ways of understanding a particular model for pairwise comparisons is to understand the set of tournaments that are achievable under the model. In this section, we will show several results which will illustrate the flexibility of the DF model in representing tournaments. We start with a result on 3 cycles.

**Theorem 2.** *Let $\mathbf{P} \in [0, 1]^{n\times n}$ be a probability preference matrix that satisfies the Distinguishing Feature model with dimension $d$. If $\mathbf{T}(\mathbf{P})$ contains a cycle, then $d \geq 3$.*

*Proof.* We already argued in Theorem 1 that dimension 1 corresponds to the BTL model and it is known that the BTL model cannot model cyclic relations. Thus, assume there are only two dimensions. Let three items be represented using the embedding $(x1, y1), (x2, y2), (x3, y3)$. Given, $|x1 - x2| >= |y1 - y2|, x1 > x2$, and $|x2 - x3| >= |y2 - y3|, x2 > x3$, we will show that

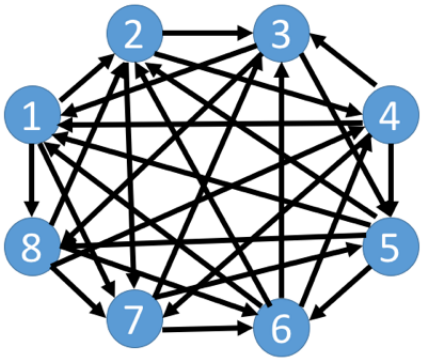


Figure 1: A 8 node tournament that cannot be realized by the Distinguishing Feature Model with $d = 3$

$|x1 - x3| < |y1 - y3|, y3 > y1$. Suppose, $x3 < x2 < x1$, then either $y2$ lies between $y1$ and $y3$ or it doesn't lie between them. We have the following cases.
**Case 1 :** $y1 < y2 < y3$: Let $y3 - y1 = (y3 - y2) + (y2 - y1) > (x1 - x3) = (x1 - x2) + (x2 - x3)$. But, $(y2 - y1) < (x1 - x2) => (y3 - y2) > (x2 - x3)$. which is a contradiction.

**Case 2 :** $y2 < y1 < y3$ **or** $y1 < y3 < y2$
$y2 < y1 < y3 \implies (y3 - y2) > (y3 - y1) > (x1 - x3) = (x1 - x2) + (x2 - x3)$
$(x1 - x2) > 0 \implies (y3 - y2) > (x2 - x3)$, which contradicts our previous assumption.
A similar argument shows the case $y1 < y3 < y2$ also leads to a contradiction.

□

While the above theorem shows that we need at least 3 dimensional embeddings to model cycles, we next show that $d = 3$ is already powerful enough to model arbitrarily long cycles.

**Theorem 3.** *The DF model can capture arbitrarily long cycles with just* 3 *dimensions.*

The proof of the above theorem is in the appendix.

The main result of this section is the following theorem where we explicitly characterize the tournament that cannot be modelled using the DF model with only 3 dimension.

**Theorem 4.** *Let* $\mathbf{T}^{forb}$ *be the tournament on* 8 *nodes described in Figure 5. There does not exist any probability preference matrix* $\mathbf{P} \in [0, 1]^{8\times 8}$ *that satisfies the DF model with* $d = 3$ *such that* $\mathbf{T}(\mathbf{P}) = \mathbf{T}^{forb}$.

**Proof Sketch:** We defer the proof to appendix. The idea is to start with a three cycle which needs 3 different dimensions due to Theorem 2. Given the nature of the tournament, this will immediately impose restrictions on dimensions of the other edges. A careful case by case argument will show that there is no possible way to use just three dimensions to obtain this tournament.

**Remark:** The above theorem shows that tournaments with complicated cyclic dependencies are those that end up being forbidden by the DF-Model even for dimension $d = 3$. In practice, we don't expect tournaments to have complicated cyclic dependencies and hence the above theorem can be seen as a reassurance that the DF model is a good enough model to capture most useful cyclic dependencies in practice.

## 6 Learning under the DF Model

In this section, we turn to the question of learning under the DF model. We propose a simple algorithm DF-Learn which is based on a Siamese type neural network architecture as shown in Figure

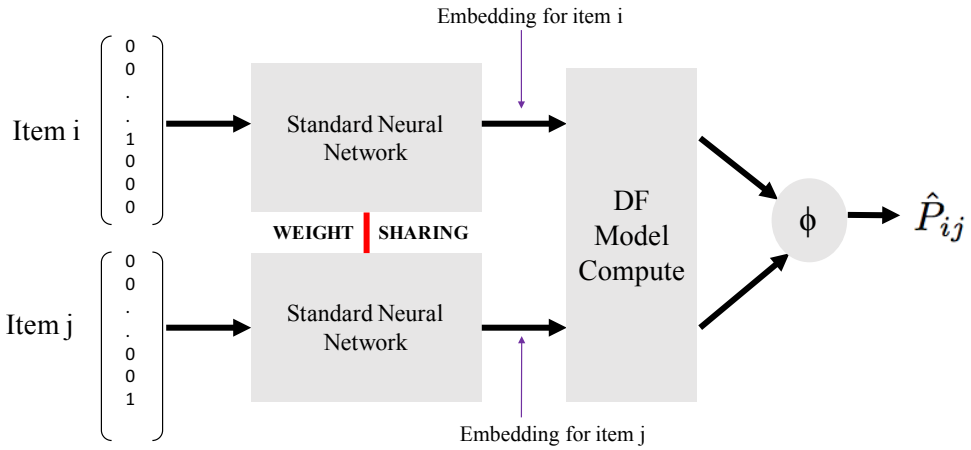


Figure 2: DFLearn - A Siamese style architecture for learning under the Distinguishing Feature Model

2. The network takes as input a pair of items represented using either their features if available or using a one-hot representation and learns embeddings of each item via a shared input to embedding network. The embeddings are then passed into a DF-Model compute module which computes the difference of scores for these embeddings corresponding to the most distinguishing feature, which is then converted into a probability of one item being preferred over another using a link function. Given a pairwise comparison dataset, the network is trained in the usual Siamese network training fashion to obtain the weights. The Siamese nature of the architecture ensures $P_{ij} + P_{ji} = 1$ for all $i, j$.

## 7 Experiments

In this section, we describe our experimental results on synthetic as well as real world datasets. We begin by describing our experimental setup.

### 7.1 Synthetic Data Experimental Setup:

We perform experiments on synthetic data by generating probability preferences over $n = 100$ items using three different models as described below.

- **BTL Model:** A score vector $\mathbf{s} \in \mathbb{R}^{100}$ is generated at random from a uniform distribution in $[0, 1]$.
- **Salient Feature Model (SF):** The embeddings in $\mathbb{R}^d$ for items are generated randomly where each component is drawn from a Gaussian distribution with mean $0$ and standard deviation $\frac{1}{\sqrt{d}}$. Furthermore, each component of the weights are drawn according to a $0$ mean Gaussian with standard deviation $\frac{4}{\sqrt{d}}$.
- **Distinguishing Feature Model (DF):** The embeddings are generated in the same manner as the SF model. There are no weights associated with the DF model.

We run the following three algorithms for data generated according to each of the above models.

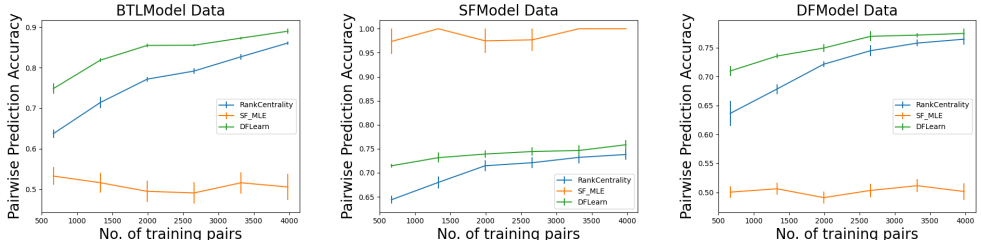


Figure 3: Experimental Results on pairwise prediction accuracy on Synthetic Data for all three algorithms considered. Left - Data generated according to BTL model, Middle - SF Model and Right - DF Model.

- **Rank Centrality** - This algorithm computes a Markov chain transition probability matrix from the training pairwise comparisons and outputs the stationary distribution of the Markov chain as the score vector.
- **SF-MLE** - This algorithm computes the maximum likelihood estimator for the weights under the context dependent Salient Features model.
- **DF-Learn** - This is the algorithm proposed in this work. The architecture is as shown in Figure 2 where the weight shared neural network is a fully connected network with 3 hidden layers and ReLu activation.

**Remark:** While the Rank Centrality algorithm and the DF learn algorithm does not require any features, the SF-MLE algorithm requires features to learn from data. However, the BTL model and the DF model do not have any explicit feature information. Thus, when we run the SF-MLE algorithm, we use the one-hot encoding of the items as the features.

### 7.2 Synthetic Data - Experimental Results

For each of the model generating the data and for each of the algorithms considered above, we test the performance using various measures discussed below.

#### 7.2.1 Pairwise Prediction Accuracy

The basic performance measure we use in the pairwise prediction accuracy as a function of the number of unique pairs considered during training. As there are $100$ items, there are $\binom{100}{2} = 4950$ unique pairs in total. We vary the number of pairs during training from $500$ in steps of $500$ until $4000$ pairs. The number of times each pair is compared is fixed to be $7$ which is equal to $\lceil \log(n) \rceil$ as $n = 100$ for these experiments. In each case, the accuracy is measured with respect to the pairs not seen in the training data. The results are presented in Figure 3. As can be seen, the DF-learn algorithm performs better than both the Rank Centrality and SF-MLE algorithms when the underlying model generating the data is either BTL or the DF model. When the underlying model is the SF-Model, the SF-MLE algorithm has an unfair advantage as it uses the features (embeddings) whereas both RC and DF-learn are not privy to the features. Even under such model mis-specification and the unavailability of features, the DF-learn algorithm still achives around $70-75\%$ accuracy irrespective of the size of the training data. In contrast, when the data is generated according to the DF-model, the SF-MLE algorithm is unable to adapt to the model mis-specification and performs only as well as a coin toss (i.e., accuracy stays around $50\%$). Thus, we conclude from these experiments that the DF-learn algorithm is able to perform well not only when the data adheres to the model but also under model mis-specification.

#### 7.2.2 Kendall-Tau Correlation

We next measure the Kendall-Tau correlation of global rankings on $n$ items obtained from the algorithms and compare it with the global rankings obtained from the underlying ground truth probability preferences.

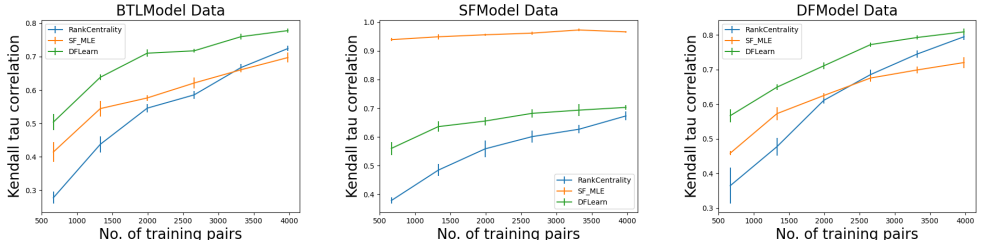


Figure 4: Experimental Results on Kendall Tau correlation on Synthetic Data for all three algorithms considered. Left - Data generated according to BTL model, Middle - SF Model and Right - DF Model.

**Ground Truth Ranking:** For the BTL model, the ground truth ranking is obtained by sorting the true scores in descending order. For the SF model, the ground truth ranking is obtained by sorting the scores where the score of item $i$ is the $w^T f_i$ where $w$ is the weight vector and $f_i$ is the feature vector corresponding to item $i$. For the DF model it is obtained by the Copeland procedure i.e., associating the score of an item as the number items it beats with probability greater than $0.5$ in pairwise contests. The Copeland score is known to be a $5$-approximation of the NP-hard problem of obtaining the ranking that minimizes pairwise disagreement error with the ground truth preference matrix.

**Ranking Output by Algorithm:** For the RC algorithm, the scores output by the algorithm is sorted to obtain the predicted ranking. For SF-MLE, the scores are obtained similar to ground truth but now by using the estimated $\hat{w}$ vector and the features. For DF learn, all pairwise probabilities are computed and a ranking is obtained by sorting the Copeland scores as described earlier.

The Kendall Tau correlation computes how well the output ranking aligns with the ground-truth ranking. We plot this for various models and algorithms in Figure 4. As can be seen again, the correlations improve with number of pairs seen in training data for all three algorithms. The DF-learn model has much better correlation with ground-truth for BTL and DF models and beats the other two algorithms irrespective of the number of pairs compared. However, for SF model, as before the SF-MLE algorithm uses features and so has an unfair advantage which shows up in the Kendall Tau correlation as well.

### 7.2.3 Effect of Dimension on Cycles Captured

We next test what fractions of 3-cycles are captured by the DF-learn model as we vary the learning dimension when the ground-truth dimension is $10$. Note that the RC algorithm cannot capture any cycles and the fraction will always be $0$. The result is shown in Figure 5 (left). As the dimension increases, the fraction of cycles captured increases indicating that the algorithm is able to predict intransitive preferences well on unseen data.

### 7.2.4 Effect of Dimension on Accuracy

We test the effect of learning dimension and prediction accuracy. Again the ground-truth dimension is fixed to be $10$ while we vary the learning dimension and the number of training pairs. The result is shown in Figure 5 (right). As can be seen, increasing the dimension tends to help increase prediction accuracy in general.

## 7.3 RealWord Data - Setup

To test the performance of our algorithm on real world data, we did the experiments using the following datasets.

- **Jester:** This is a dataset of ratings of jokes given by several users. The ratings are converted into pairwise comparisons. Number of jokes $n = 100$ and number of pairs compared $m = 891404$.

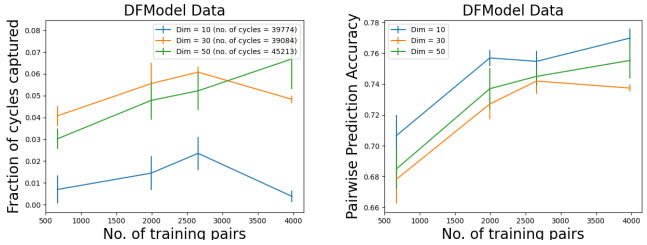


Figure 5: (left) Dimension vs Cycles; (right) Dimension vs accuracy for the DF model

| | RC | SF-MLE | DF-Learn |
|---|---|---|---|
| **Jester** | 0.813(0.009) | 0.726(0.003) | **0.864 (0.004)** |
| **MovieLens** | 0.638(0.003) | 0.600(0.003) | **0.669 (0.003)** |
| **DoTA** | **0.645 (0.015)** | 0.533(0.013) | 0.640(0.013) |
| **StarCraft II: WoL** | 0.635(0.004) | 0.546(0.005) | **0.651 (0.005)** |
| **StartCraft II: HoTs** | 0.657(0.005) | 0.574(0.005) | **0.689 (0.006)** |

Table 1: Table showing accuracy of different algorithms for various real world datasets. RC - Rank Centrality, SF-MLE - Maximum Likelihood estimation for Salient Feature Model, DF-Learn - Distinguishing Feature Model proposed in this paper.

- **MovieLens:** This dataset contains realworld ratings of movies. We consider $n = 1682$ movies and $m = 139982$ comaprisons among them derived from ratings.
- **DoTA:** This dataset contains match-ups of players of the online video game. Number of players considered $n = 757$ and number of matches considered $m = 10,442$.
- **StarCraft II: WoL** Similar to DoTA dataset where pairwise match results of online video games are considered for $n = 4381$ with $m = 61657$ matches.
- **StartCraft II: HoTs** Another pairwise matches dataset with $n = 2287$ users and $m = 28582$ matches.

### 7.4 REALWORD DATA - RESULTS

For each of the above dataset, the pairwise comparisons were split in ratio of $70 : 30$ for train and test respectively. The algorithms were run on the train set and they were tested for accuracy of prediction on the test set. The accuracy for various datasets are shown in Table 7.4. As can be seen, the DF-Learn algorithm beats the baselines in four out of five datasets.

## 8 CONCLUSION

In this work, we proposed the distinguishing feature model for pairwise comparisons. We analysed certain theoretical properties of the class of tournaments that can be obtained via this model, developed an algorithm called DF-Learn to learn from pairwise comparisons generated according to this model and showed superior experimental results on both real world and synthetic data as compared to standard baselines. Future work include understanding the exact class of tournaments that can be modelled under the DF model.

**Ethical Considerations:** There are no major ethical considerations of this work. However, any learning algorithm needs to be aware of potential risks involved due to data/algorithmic bias. The current work proposes a new model for pairwise comparisons and fairness/bias considerations of the proposed model is deferred to future work.

## REFERENCES

Arpit Agarwal, Prathamesh Patil, and Shivani Agarwal. Accelerated spectral ranking. In *International Conference on Machine Learning*, pp. 70–79. PMLR, 2018.

David Causeur and François Husson. A 2-dimensional extension of the bradley–terry model for paired comparisons. *Journal of statistical planning and inference*, 135(2):245–259, 2005.

Shuo Chen and Thorsten Joachims. Modeling intransitivity in matchup and comparison data. In *Proceedings of the ninth acm international conference on web search and data mining*, pp. 227–236, 2016.

David F Gleich and Lek-heng Lim. Rank aggregation via nuclear norm minimization. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 60–68, 2011.

Bruce Hajek, Sewoong Oh, and Jiaming Xu. Minimax-optimal inference from partial rankings. *arXiv preprint arXiv:1406.5638*, 2014.

Rahul Makhijani. Social choice random utility models of intransitive pairwise comparisons. *arXiv preprint arXiv:1810.02518*, 2018.

Rahul Makhijani and Johan Ugander. Parametric models for intransitivity in pairwise rankings. In *The World Wide Web Conference*, pp. 3056–3062, 2019.

Sahand Negahban, Sewoong Oh, and Devavrat Shah. Rank centrality: Ranking from pair-wise comparisons, 2015.

Arun Rajkumar and Shivani Agarwal. A statistical convergence perspective of algorithms for rank aggregation from pairwise data. In *International Conference on Machine Learning*, pp. 118–126. PMLR, 2014.

Arun Rajkumar and Shivani Agarwal. When can we rank well from comparisons of o (n\log (n)) non-actively chosen pairs? In *Conference on Learning Theory*, pp. 1376–1401. PMLR, 2016.

Nihar Shah, Sivaraman Balakrishnan, Joseph Bradley, Abhay Parekh, Kannan Ramchandran, and Martin Wainwright. Estimation from pairwise comparisons: Sharp minimax bounds with topology dependence. In *Artificial Intelligence and Statistics*, pp. 856–865. PMLR, 2015.

**Proof of Theorem 4:**

*Proof.* Consider the tournament in Figure 5. We begin by fixing a part of the dimension assignment wlog as follows,
$1- > 2(1), 2- > 3(2), 3- > 1(3)$ where $i- > j(k)$ means that node $i$ beats node $j$ and the distinguishing feature dimension is $k$. Note that we have $i- > j(l), j- > k(l) \implies i- > k(l)$. Also from theorem 2, we know that for any three cycle, all three dimension must be used. We will now look at all possibilities given that the above assignment was fixed wlog.

We can consider
$1)2- > 4(2), 4- > 1(3)$ or
$2)2- > 4(3), 4- > 1(2)$
$3)7- > 3$ (1 or 2)

If $2- > 4(3), 4- > 1(2)$, then $5- > 2$ is 1 because $2- > 3 = 2, 2- > 4 = 3$. If $2- > 4(2), 4- > 1(3)$, then $5- > 2$ is 1 or 3.
1) Let $2- > 4(3), 4- > 1(2), 7- > 3(1) => 1- > 7 = 2, 5- > 2 = 1 => 7- > 5 = 3$ (because it is the common edge of $5- > 2- > 7- > 5, 5- > 1- > 7- > 5), 5- > 1 = 1$. Now, $4- > 5 = 2, 7- > 5 = 3 => 5- > 8 = 1 => 8- > 4 = 3$. So, $3- > 8$ has no color left to be assigned to.

2) Let $2- > 4(3), 4- > 1(2), 7- > 3(2) => 1- > 7 = 1$. (a) If $7- > 5 = 3$, then $5- > 8 = 1$. Now, $8- > 7$ has no color left.
(b) If $7- > 5 = 2 => 5- > 8 = 3/1$.

Let $5- > 8 = 3, => 8- > 7 = 1, 8- > 4 = 1, 1- > 8 = 3, 3- > 8 = 3, 6- > 1 = 2, 7- > 6 = 3$, but $2- > 7 = 3$(invalid)
Let $5- > 8 = 1, => 8- > 7 = 3, 8- > 4 = 3, 1- > 8 = 1, 3- > 8 = 1, 2- > 7 = 3, 7- > 6 = 2, 6- > 1 = 3, 8- > 6 = 2, 6- > 3 = 3$, but $3- > 5 = 3$ (invalid)

3) Let $2- > 4(2), 4- > 1(3), 7- > 3(2), 5- > 2(1)$. Then, $3- > 5(3), 4- > 5(3), 1- > 7(1)$.

If $7- > 5(3)$ and $5- > 8 = 1 => 8- > 7 = 2$, but $7- > 3(2)$ (invalid)
If $7- > 5(3)$ and $5- > 8 = 2 => 8- > 7 = 1, 3- > 8 = 3, 8- > 4 = 1, 1- > 8 = 2, 6- > 1 = 3, 7- > 6 = 2$, but $2- > 7 = 2$ (invalid)

If $7- > 5 = 2$, then $5- > 8 = 1, 8- > 7 = 3, 8- > 4 = 2, 1- > 8 = 1, 3- > 8 = 1, 6- > 3 = 2, 8- > 6 = 3, 6- > 1 = 2, 7- > 6 = 3$, but $2- > 7 = 3$ (invalid)

4) Let $2- > 4(2), 4- > 1(3), 7- > 3(2), 5- > 2(3)$. Then, $1- > 7 = 1, 7- > 5 = 2, 3- > 5 = 1, 4- > 5 = 1, 5- > 8 = 3, 8- > 7 = 1, 3- > 8 = 3, 8- > 4 = 2, 1- > 8 = 1, 6- > 3 = 2, 8- > 6 = 1$ (invalid)

5) Let $2- > 4(2), 4- > 1(3), 7- > 3(1), 5- > 2(1)$. Then, $1- > 7 = 2, 7- > 5 = 3, 2- > 7 = 2, 3- > 5 = 3, 4- > 5 = 3$.
If $5- > 8 = 2$, then $8- > 7 = 1$ (invalid).
If $5- > 8 = 1$, then $8- > 7 = 2, 3- > 8 = 3, 8- > 4 = 2, 1- > 8 = 1, 8- > 6 = 2, 6- > 1 = 3, 7- > 6 = 1, 6- > 3 = 1, 5- > 6 = 2, 6- > 4 = 1$ (invalid)

6) Let $2- > 4(2), 4- > 1(3), 7- > 3(1), 5- > 2(3)$. Then, $3- > 5 = 1, 4- > 5 = 1, 1- > 7 = 2, 7- > 5 = 1$,
If $5- > 8 = 2, 8- > 7 = 3, 3- > 8 = 2$ (invalid).
If $5- > 8 = 3, 8- > 7 = 2, 3- > 8 = 3, 6- > 3 = 2, 8- > 6 = 1, 8- > 4 = 2, 1- > 8 =$

1(invalid).

□

**Proof of Theorem 3**

*Proof.* We know that we can capture a 3-cycle using 3 dimensions. If $|x1 - x2| >= |y1 - y2|$ and $|x1-x2| >= |z1-z2|$ and $x1 > x2$, then 1 beats 2 in 1. If $|y2-y3| >= |x2-x3|$ and $|y2-y3| >= |z2 - z3|$ and $y2 > y3$ , then 2 beats 3 in 2. Hence, $|z3-z1| >= |x3-x1|, |z3-z1| >= |y3-y1|$ and $z3 > z1$ so that 3 beats 1 in 3. $(1, 2, 3)$ becomes the order of the coordinates for the chosen features wlog.

Let, $x1 > x2 > x3, y1 > y2 > y3$ and $z3 > z2 > z1$.

$$
\begin{aligned}
&=> (x1 - x2) >= (y1 - y2) \text{ and } (x1 - x2) >= (z2 - z1).\\
&(y2 - y3) >= (x2 - x3) \text{ and } (y2 - y3) >= (z3 - z2).\\
&z3 - z1 = (z3 - z2) + (z2 - z1) >= x1 - x3 = (x1 - x2) + (x2 - x3)\\
&(x1 - x2) >= (z2 - z1) => (x2 - x3) <= (z3 - z2)\\
&z3 - z1 = (z3 - z2) + (z2 - z1) >= y1 - y3 = (y1 - y2) + (y2 - y3)\\
&(y2 - y3) >= (z3 - z2) => (y1 - y2) <= (z2 - z1).\\
&=> z3 - z1 = (z3 - z2) + (z2 - z1) >= (x2 - x3) + (y1 - y2).
\end{aligned}
$$

$x1 = y1$ and $x3 = y3 \implies x1 - x3 = y1 - y3$. Now, If we add a node 4 such that $z4 = (z1+z3)/2, x4 = (x1+x3)/2, y4 = (y1+y3)/2$, then $(z3-z4) >= (x4-x3)$ and $(z3-z4) >= (y4 - y3)$. Similarly, $(z4 - z1) >= (x1 - x4)$ and $(z4 - z1) >= (y1 - y4)$ So, 3 beats 4 in 3 and 4 beats 1 in 3.

Similarly, if we add another node 5 to the set of nodes and bisect the intervals $(z1, z4)$, $(x4, x1)$ and $(y4, y1)$ to add the 2 directed edges 4 beats 5, $4- > 5$) and (5 beats 1, $5- > 1$), then both $4- > 5$ and $5- > 1$ can be realized in 3.

It is easy to see that a cycle of length $n >= 3$ can be formed in 3-D by creating the features like this. □