Distinguishing Feature Model for Learning From Pairwise Comparisons

Elisha Parhi[1] **Arun Rajkumar**[1]

Abstract

We consider the problem of learning to predict outcomes of unseen pairwise comparisons over a set of items when a small set of pairwise comparisons are available. When the underlying preferences are intransitive in nature, which is common occurrence in real world data sets, this becomes a challenging problem both in terms of modeling and learning. Towards this, we introduce a flexible and natural parametric model for pairwise comparisons that we call the *Distinguishing Feature Model* (DF). Under this model, the items have an unknown but fixed embeddings and the pairwise comparison between a pair of items depends probabilistically on the feature in the embedding that can best distinguish the items. The proposed DF model generalizes the popular transitive Bradley-Terry-Luce model and with embedding dimension as low as $d = 3$, 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 style neural network architecture which can be used 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 is either comparable or outperforms standard baselines in both synthetic and real world data-sets.

1. Introduction

We consider the problem of learning to predict outcomes of unseen pairwise comparisons over a set of items given a small set of pairwise comparisons as training data. This problem has applications in predicting outcomes of sports tournaments, e-commerce, meta-ranking, etc. Typically, one assumes a statistical model for comparisons where item i is preferred to item j with some underlying fixed but unknown probability P_{ij}. The performance of an algorithm is then measured by how *closely* the algorithm is able to predict the underlying probability matrix $\mathbf{P}$ either in terms of pairwise agreement or in terms of mean squared error (RMSE). Imposing parametric assumptions on $\mathbf{P}$ lead to various popular preference models. 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. Under the BTL model, 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 13. While the sample complexity of learning is attractive, the downside of the BTL assumption 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 due to the fact that items typically have multiple features or scores associated with them and one is preferred over the other based on some function of all these features. In the simplistic BTL model, items have 1 dimensional features (their associated score) and are hence restrictive for real world applications.

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

[1] Department of CSE, IIT Madras, India. Correspondence to: Elisha Parhi <elisha.parhi@gmail.com>, Arun Rajkumar <arunr@cse.iitm.ac.in>.

The Many Facets of Preference Learning Workshop at the International Conference on Machine Learning (ICML), Honolulu, Hawaii, USA, 2023. Copyright 2023 by the author(s).

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.

Main Contributions: 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.

On the learning side, we develop a 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 well in terms of prediction accuracy and RMSE when compared to the state of the art algorithms including those that are proposed for the Salient Feature model (Bower & Balzano, 2020), Blade-Chest Model (Chen & Joachims, 2016), GNNRank Model (He et al., 2022), Low Rank Pairwise Ranking Model (Rajkumar & Agarwal, 2016), Majority Vote Model (Makhijani, 2018), and BTL model (Negahban et al., 2015).

2. Related Work

The problem of learning to predict outcomes of pairwise comparisons has been studied extensively in several areas including theoretical computer science, AI/ML, social choice, operations research, etc.

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(Bradley & Terry, 1952)(Luce, 2012). 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.

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. (Makhijani, 2018; Makhijani & Ugander, 2019)] propose the Majority vote model which is a random utility model (RUM) with a d dimensional feature embedding for each item. The Blade-Chest inner (BC) model (Chen & Joachims, 2016)] 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 $< i_{\text{chest}}, j_{\text{blade}} > - < i_{\text{blade}}, j_{\text{chest}} > + s_i - s_j$. Previous work (Gleich & Lim, 2011),(Rajkumar & Agarwal, 2016) 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(Bower & Balzano, 2020) 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 along with the items. However, if features are available, our algorithms will still learn an embedding from the feature space to an embedding space automatically. The GNNRank (He et al., 2022) (and it's several variants) produces a ranking by learning embeddings from pairwise comparisons. As rankings are inherently transitive, this algorithm may not be suited in cases where intransitivity is expected in the underlying preferences.

3. Problem Setting and Preliminaries

Let $[n] = \{1, 2, \dots, n\}$ 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 $k = 1, \dots, m$, $i_k, j_k \in [n]$ 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 ties are not allowed i.e., $P_{ij} \neq 0.5 \, \forall i \neq j$.

Performance Measure: We will use *Prediction Accuracy* as one of the performance of an algorithm. This is the fraction of pairwise preferences predicted correctly with respect to the underlying true probability preference matrix. 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 prediction accuracy is defined as below:

$$predAcc(\hat{\mathbf{P}}, \mathbf{P}) =$$

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

As one can observe, the prediction accuracy captures the fraction of correct predictions but does not capture the absolute values of $\hat{\mathbf{P}}$ with respect to $\mathbf{P}$. To capture this, we also use RMSE as a performance measure.

$$RMSE(\hat{\mathbf{P}}, \mathbf{P}) = \sqrt{\frac{1}{\binom{n}{2}} \sum_{i<j} (\hat{P}_{ij} - P_{ij})^2} \quad (1)$$

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:

$$k^* = \arg\max_k |e_{ik} - e_{jk}| = \|\mathbf{e}_i - \mathbf{e}_j\|_\infty \quad (2)$$

where, e_{ik} represents the k^{th} feature of the embedding $\mathbf{e}_i$. In the above equation, ties are broken arbitrarily. In the second step, the preference probability is calculated as $P_{ij} = \phi(e_{ik^*} - e_{jk^*})$, where ϕ 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.

Remark: It is straightforward to see that the DF model with embeddings of items in d dimensions is equivalent to the following generalization of the BTL model: For any set of d linearly independent vectors $\{\mathbf{w}_1, \ldots, \mathbf{w}_d\}$, define $P_{ij} = \phi({\mathbf{w}_{\mathbf{k}*}}^T(\mathbf{e}_i - \mathbf{e}_j))$ where $k^* = \arg\max_k |{\mathbf{w}_\mathbf{k}}^T(\mathbf{e}_i - \mathbf{e}_j)|$. When $\{\mathbf{w}_1, \ldots, \mathbf{w}_d\}$ is the set of standard basis vectors, we obtain the distinguishing feature model described earlier. For any general set of linearly independent vectors, the embeddings can be linearly transformed to achieve the same effect as the DF model. Moreover, when $d = 1$, we obtain the BTL model.

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.

Proposition 5.1. *For $d = 1$, the Distinguishing Feature model with the logit link function is exactly same as the Bradley-Terry-Luce model.*

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 5.2. *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 $d \geq 3$.*

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

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 5.4. *Let $\mathbf{T}^{forb}$ be the tournament on 8 nodes described in Figure 1 (right). 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 $\mathbf{T}(\mathbf{P}) = \mathbf{T}^{forb}$, i.e., $\mathbf{T}^{forb}$ is a forbidden tournament under the DF model for $d = 3$.*

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.

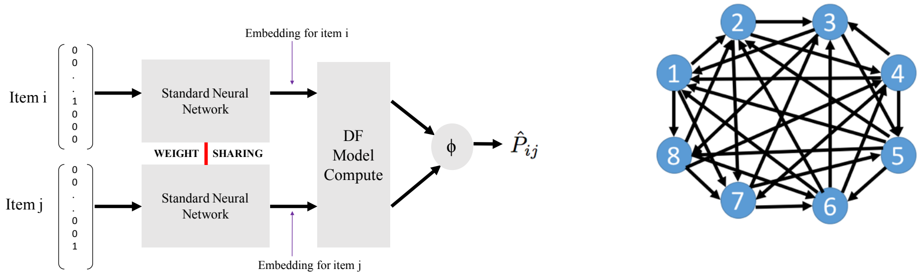

Figure 1. (left) **DFLearn** - A Siamese style architecture for learning under the Distinguishing Feature Model (right) $\mathbf{T}^{forb}$ - A 8 node tournament that is forbinnden i.e., cannot be realized by the Distinguishing Feature Model with $d = 3$;

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 1 (left). 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 $\mathbb{R}^{10}$ 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):** 3 dimensional embeddings are generated for 100 nodes such that they realize the tournament in Figure 2 (left). In particular, there are 3 clusters with 66, 22 and 12 items each. Each of these clusters has 3 sub-clusters. Each circle in the figure indicates the corresponding number of nodes in that subgroup. For the nodes in each subgroup, the embeddings are generated according to a 3 dimensional Gaussian distribution with mean embeddings respecting the pairwise relation with other subgroups and with a co-variance of $0.0001\mathbf{I}$.

The BTL model was chosen as it the most commonly used transitive model, the SF model was chosen as we wanted to test the DF learn algorithm's performance under model mis-specification. We run the following algorithms for data generated according to each of the above models.

- **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 (Negahban et al., 2015).
- **LRPR-2** - This algorithm that computes the embeddings using a matrix completion procedure as proposed in (Rajkumar & Agarwal, 2016). The rank is chosen as 2 after trying out different ranks and finding that 2 gives the best results in general.
- **SF-MLE** - This algorithm computes the maximum likelihood estimator for the weights under the context dependent Salient Features model (Bower & Balzano, 2020)
- **Blade-Chest** - This algorithm that computes embeddings assuming that data is generated according to the Blade Chest (Inner) model of (Chen & Joachims, 2016)
- **Majority Vote** - This algorithm that computes embeddings assuming the data is generated according to the 3D majority vote model of (Makhijani & Ugander, 2019)

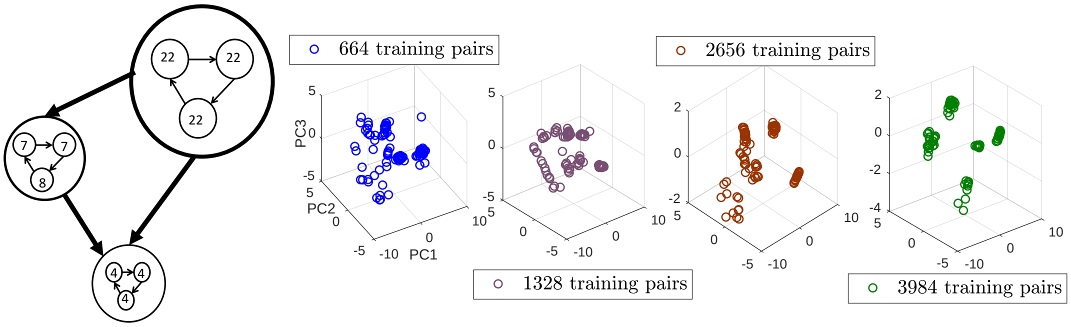

Figure 2. (left) The Tournament used to generated the synthetic data from the DF model. (Right) Visualization of the top 3 principal components of the learnt embeddings by the DFlearn algorithm with increasing number of training pairs. As can be seen, the embeddings become better with increasing data and visually correspond to the ground truth tournament shown on the left.

c	RC	LRPR-2	SF-MLE	Blade-Chest	3D Majority Vote	DFLearn	GNNRank-Best
1	0.741 (0.004)	0.802 (0.002)	0.784 (0.002)	0.815 (0.004)	0.782 (0.007)	0.89 (0.004)	0.745 (0.005)
2	0.729 (0.003)	0.803 (0.002)	0.8 (0.002)	0.898 (0.003)	0.857 (0.008)	0.903 (0.003)	0.728 (0.007)
3	0.728 (0.003)	0.8 (0.003)	0.806 (0.002)	0.91 (0.002)	0.747 (0.012)	0.912 (0.003)	0.733 (0.005)
4	0.727 (0.002)	0.807 (0.002)	0.808 (0.002)	0.918 (0.001)	0.768 (0.017)	0.901 (0.004)	0.74 (0.006)
5	0.718 (0.004)	0.805 (0.005)	0.807 (0.003)	0.917 (0.002)	0.81 (0.005)	0.905 (0.004)	0.738 (0.005)
6	0.71 (0.005)	0.81 (0.005)	0.805 (0.003)	0.922 (0.003)	0.86 (0.012)	0.906 (0.004)	0.732 (0.006)

Figure 3. Prediction Accuracy (higher is better) of various algorithms when the data follows the DF model and the number of training pairs are varied as $cn \log n$ for various choices of c, red implies the best and blue implies the second best

- **GNNRank-Best** - There are 5 algorithmic variants proposed in their work(He et al., 2022). We run all 5 variants and report the results for the best variant in each case.
- **DF-Learn** - This is the algorithm proposed in this work. The architecture is as shown in Figure 1 (right) where the weight shared neural network is a fully connected network with 2 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 i.e., the i-th item is represented using a n dimensional vector (where n is the number of items) where the i-th entry is 1 and the rest are 0.

7.2. Synthetic Data - Experimental Results

For each of the 3 models generating the data and for each of the 7 algorithms considered above, we test the performance using various measures discussed earlier namely the Prediction accuracy and the root mean squared error (We also measure the Kendall-Tau correlation for all the experiments where a ranking is obtained suitably from both ground-truth and the predicted preference matrix. We present the root mean squared error and Kendall-Tau correlation results in the appendix due to lack of space).

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. We report performance measures along with their standard errors averaged over 10 runs. As there are 100 items, there are $\binom{100}{2} = 4950$ unique pairs in total. We vary the number of pairs during training as $cn \log n$ for $c = 1, 2, ..., 6$. The number of times each pair is compared is fixed to be 6 which is equal to $\log(n)$ as $n = 100$ for these experiments. In each case, the accuracy is measured with respect to the pairs not seen in the training data. We make the following observations.

Transitive Model - BTL Data: The results of the algorithms when the data is generated according to the BTL model is given in Table 7. Here, we observe that the DF-learn model performs exceedingly well, sometimes even better than the Rank Centrality (RC) algorithm which is designed to work for the BTL model.

Model Mis-specification - SF Data: The results of the algorithms when the data is generated according to the SF model is given in Table 8. This experiment is to study the robustness of the DF learn algorithm to model mis-specification. As expected, in this case the SF-MLE algorithm works the best. However, the RC algorithm performs reasonably well in this case whereas DF-learn performs slightly worse than RC.

DF Model Data: The results of the algorithms when the data is generated according to the DF model described earlier is given in Table 3. DFLearn outperforms all other

Accuracy	**RC**	**LRPR-2**	**SF-MLE**	**Blade-Chest**	**Majority Vote**	**DFLearn**	**GNNRank-Best**
Jester	0.87(0.001)	0.87(0.001)	0.726(0.003)	0.84(0.005)	0.849(0.01)	0.876(0.004)	0.875(0.003)
MovieLens	0.649(0.001)	0.676(0.001)	0.600(0.003)	0.599(0.003)	0.661(0.003)	0.676(0.003)	0.554(0.003)
DoTA	0.603(0.007)	0.556(0.02)	0.533(0.013)	0.72(0.009)	0.578(0.007)	0.64(0.013)	0.512(0.01)
StarCraft II : WoL	0.605(0.002)	0.63(0.006)	0.546(0.005)	0.816(0.003)	0.63(0.005)	0.663(0.005)	0.518(0.006)
StarCraft II : HoTs	0.627(0.002)	0.69(0.003)	0.574(0.005)	0.84(0.002)	0.646(0.005)	0.699(0.006)	0.52(0.005)

Figure 4. Prediction Accuracy (higher is better) of various algorithms for real world datasets.

algorithms in terms of accuracy except Blade-Chest which outperforms DFLearn in some cases. Both Blade-Chest and DFLearn perform much better in terms of RMSE 2 than every other algorithm. We have also run an experiment[10], with the data following the DF model, where the number of comparisons is varied for a fixed number of training pairs ($n \log n = 664$) as $c \log n$ for various choices of c.

7.3. Further Experiments on DFLearn

Visulaization of Learnt Embeddings: We visualized the 10 dimensional embeddings learnt by the DF learn algorithm when the ground truth embedding corresponds to the tournanment in Figure 2 (left). We plot the top 3 principal components (obtained using PCA) to see if the embeddings visually correspond to the tournament. As can be seen, with increasing number of pairs, the embeddings becomes visually similar to the tournament structure. In particular, the clusters with $66(3 * 22)$ node corresponds to 3 cluster regions and the cluster with $22(2 * 7 + 8)$ and $12(2 * 4)$ nodes correspond to 2 other cluster regions indicating that the algorithm is able to learn the tournament embeddings very well.

Effect of Dimension on Cycles Captured: We 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 3. Note that the RC algorithm cannot capture any cycles and the fraction will always be 0. The result is shown in Table 5. As the dimension increases, the fraction of cycles captured increases indicating that the algorithm is able to predict intransitive preferences well on unseen data.

Effect of Dimension on Accuracy: We test the effect of learning dimension and prediction accuracy. Again the ground-truth dimension is fixed to be 3 while we vary the learning dimension and the number of training pairs. The result is shown in Table 6.

7.4. 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 (Goldberg et al., 2001) 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 $m = 891404$.
- **MovieLens:** This dataset (Herlocker et al., 1999) contains real-world ratings of movies. We consider $n = 1682$ movies and $m = 139982$ comaprisons among them derived from ratings.
- **DoTA:** This dataset (dt) 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 (sc).
- **StartCraft II: HoTs** Another pairwise matches dataset with $n = 2287$ users and $m = 28582$ matches (sc).

7.5. RealWord Data - Results

All the algorithms were run on real world datasets where the learning rates were tuned using cross validation. For DF learn algorithm, a SGD optimizer was used. 200 epochs were used for LRPR-2, Majority Vote, DF-Learn algorithm. For Blade-Chest, SF-MLE and GNN-Rank, we used the epochs specified as default in the publicly available implementations of these algorithms. The results of running our experiments on real world data are shown in Table 4,9. We note that the datasets Jester and Movielens are inherently ranking/rating based datasets where the pairwise comparisons are obtained from underlying scores given by users to movies/jokes. In these datasets, we observe that the DF learn algorithm performs the best both in terms of accuracy and RMSE. For the other datasets, DF learn algorithm performs the second best while the Blade-Chest algorithm is the best. The performance of the remaining algorithms are below par when compared to DF-learn.

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 includes understanding the exact class of tournaments that can be modelled under DF model.

References

Dota data : http://www.datdota.com/.

Starcraft ii : http://aligulac.com/.

Bower, A. and Balzano, L. Preference modeling with context-dependent salient features. In *International Conference on Machine Learning*, pp. 1067–1077. PMLR, 2020.

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

Chen, S. and Joachims, T. 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.

Gleich, D. F. and Lim, L.-h. 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.

Goldberg, K., Roeder, T., Gupta, D., and Perkins, C. Eigentaste: A constant time collaborative filtering algorithm. *information retrieval*, 4(2):133–151, 2001.

He, Y., Gan, Q., Wipf, D., Reinert, G. D., Yan, J., and Cucuringu, M. Gnnrank: Learning global rankings from pairwise comparisons via directed graph neural networks. In *International Conference on Machine Learning*, pp. 8581–8612. PMLR, 2022.

Herlocker, J. L., Konstan, J. A., Borchers, A., and Riedl, J. An algorithmic framework for performing collaborative filtering. In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*, pp. 230–237, 1999.

Luce, R. D. *Individual choice behavior: A theoretical analysis*. Courier Corporation, 2012.

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

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

Negahban, S., Oh, S., and Shah, D. Rank centrality: Ranking from pair-wise comparisons, 2015.

Rajkumar, A. and Agarwal, S. 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.

A. Appendix

Code for Reproducing results All datasets and code are available here

Proof of Proposition 1

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. □

In all the proofs given below, if $i \geq j$, the tie is broken in favor of i.

Proof of Theorem 1

Proof. We already know 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 x and y. Let three items a, b, c be represented using the following embeddings in $\mathbb{R}^2$: $[x_a, y_a], [x_b, y_b], [x_c, y_c]$.

Assume wlog, $a \succ b(1), b \succ c(1)$ where $i \succ j(k)$ means that node i beats node j and the distinguishing feature dimension is k. Thus, $|x_a - x_b| \geq |y_a - y_b|, x_a > x_b$, and $|x_b - x_c| \geq |y_b - y_c|, x_b > x_c$. For a 3-cycle to exist among these three items with only two dimensions, it should be the case that $c \succ a(2)$, because $i \succ j(l), j \succ k(l) \implies i \succ k(l)$. So, if we can show that $|x_a - x_c| < |y_a - y_c|, y_c > y_a$, it would imply that two dimensions are sufficient for realizing a 3-cycle.

As, $x_c < x_b < x_a$, we have the following cases depending on the relation of y_b w.r.t y_a and y_c.

Case 1 : $y_a < y_b < y_c$: Let $y_c - y_a = (y_c - y_b) + (y_b - y_a) > (x_a - x_c) = (x_a - x_b) + (x_b - x_c)$. But, $(y_b - y_a) < (x_a - x_b) \implies (y_c - y_b) > (x_b - x_c)$ which is a contradiction.

Case 2 : $y_b < y_a < y_c$ **or** $y_a < y_c < y_b$
$y_b < y_a < y_c \implies (y_c - y_b) > (y_c - y_a) > (x_a - x_c) = (x_a - x_b) + (x_b - x_c)$
$(x_a - x_b) > 0 \implies (y_c - y_b) > (x_b - x_c)$, which contradicts our previous assumption.
A similar argument shows the case $y_a < y_c < y_b$ also leads to a contradiction.

□

Proof of Theorem 2

Proof. From theorem 1, it is clear that we need at least three dimensions to realize a 3-cycle in a tournament based on the DF model.

Suppose that a tournament consists of only 3 items a, b, c and each item has a 3-dimensional feature embedding given by $[x_a, y_a, z_a]$, $[x_b, y_b, z_b]$, and $[x_c, y_c, z_c]$ respectively and together they form a 3-cycle.
Wlog $a \succ b(1), b \succ c(2), c \succ a(3)$. Hence,

1) $|x_a - x_b| \geq |y_a - y_b|$, $|x_a - x_b| \geq |z_a - z_b|$ and $x_a > x_b$,

2) $|y_b - y_c| \geq |x_b - x_c|$, $|y_b - y_c| \geq |z_b - z_c|$ and $y_b > y_c$,

3) $|z_c - z_a| \geq |x_c - x_a|$, $|z_c - z_a| \geq |y_c - y_a|$ and $z_c > z_a$.

In order to show that the above claim holds, we will give a constructive proof below.
Let's fix an ordering for the feature embeddings of the items based on the above relations,

$$x_a > x_b > x_c, y_a > y_b > y_c, z_c > z_b > z_a \tag{3}$$

The following set of inequalities follow:

$$(x_a - x_b) \geq (y_a - y_b), (x_a - x_b) \geq (z_b - z_a) \tag{4}$$
$$(y_b - y_c) \geq (x_b - x_c), (y_b - y_c) \geq (z_c - z_b) \tag{5}$$
$$(z_c - z_a) \geq (x_a - x_c), (z_c - z_a) \geq (y_a - y_c) \tag{6}$$

Let, $x_a = y_a$ and $x_c = y_c \implies x_a - x_c = y_a - y_c$.

Now, if we add a node d such that, $z_d = \frac{(z_a+z_c)}{2}, x_d = \frac{(x_a+x_c)}{2}, y_d = \frac{(y_a+y_c)}{2}$, then $(z_c - z_d) \geq (x_d - x_c)$ and $(z_c - z_d) \geq (y_d - y_c)$.

Similarly, $(z_d - z_a) \geq (x_a - x_d)$ and $(z_d - z_a) \geq (y_a - y_d)$.

In the same way, we can add further nodes to the tournament and bisect the intervals (z_a, z_d), (x_d, x_a) and (y_d, y_a) to accommodate each such node in the cycle thereby increasing its length arbitrarily.

□

Proof of Theorem 3

Proof. Consider the tournament in Figure 1. We begin by fixing a part of the dimension assignment wlog as follows, $1 \succ 2(1), 2 \succ 3(2), 3 \succ 1(3)$. Note that we have $i \succ j(l), j \succ k(l) \implies i \succ k(l)$. Also from theorem 1, 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. In all the below possible combinations, an invalid assignment occurs whenever a dimension has already been assigned to another pair of items for comparison in the same three cycle. As, one directed edge might be a part of more than one three cycles in the tournament, it might be possible that the dimension capturing the respective edge/pairwise comparison is not available for further assignment. We can consider

- $2 \succ 4(2), 4 \succ 1(3)$
- $2 \succ 4(3), 4 \succ 1(2)$
- $7 \succ 3\ (1\ or\ 2)$

If $2 \succ 4(3), 4 \succ 1(2)$, then $5 \succ 2(1)$ because $2 \succ 3(2), 2 \succ 4(3)$ and both $5 \succ 2 \succ 3 \succ 5$ and $5 \succ 2 \succ 4 \succ 5$ are three cycles. If $2 \succ 4(2), 4 \succ 1(3)$, then $5 \succ 2(1\ or\ 3)$. We have six possible cases to consider.

Case 1) Let $2 \succ 4(3), 4 \succ 1(2), 7 \succ 3(1) => 1 \succ 7(2), 5 \succ 2(1) => 7 \succ 5(3)$ (because it is the common edge of $5 \succ 2 \succ 7 \succ 5, 5 \succ 1 \succ 7 \succ 5), 5 \succ 1(1)$. Now, $4 \succ 5(2), 7 \succ 5(3) => 5 \succ 8(1) => 8 \succ 4(3)$. So, $3 \succ 8$ has no dimension left to be assigned to.

Case 2) Let $2 \succ 4(3), 4 \succ 1(2), 7 \succ 3(2) => 1 \succ 7(1)$. (a) If $7 \succ 5(3)$, then $5 \succ 8(1)$. Now, $8 \succ 7$ has no dimension left. (b) If $7 \succ 5(2) => 5 \succ 8(3\ or\ 1)$.

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

Case 3) Let $2 \succ 4(2), 4 \succ 1(3), 7 \succ 3(2), 5 \succ 2(1)$. Then, $3 \succ 5(3), 4 \succ 5(3), 1 \succ 7(1)$.

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

If $7 \succ 5(2)$, then $5 \succ 8(1), 8 \succ 7(3), 8 \succ 4(2), 1 \succ 8(1), 3 \succ 8(1), 6 \succ 3(2), 8 \succ 6(3), 6 \succ 1(2), 7 \succ 6(3)$, but $2 \succ 7(3)$ (invalid)

Case 4) Let $2 \succ 4(2), 4 \succ 1(3), 7 \succ 3(2), 5 \succ 2(3)$. Then, $1 \succ 7(1), 7 \succ 5(2), 3 \succ 5(1), 4 \succ 5(1), 5 \succ 8(3), 8 \succ 7(1), 3 \succ 8(3), 8 \succ 4(2), 1 \succ 8(1), 6 \succ 3(2), 8 \succ 6(1)$ (invalid)

Case 5) Let $2 \succ 4(2), 4 \succ 1(3), 7 \succ 3(1), 5 \succ 2(1)$. Then, $1 \succ 7(2), 7 \succ 5(3), 2 \succ 7(2), 3 \succ 5(3), 4 \succ 5(3)$.
If $5 \succ 8(2)$, then $8 \succ 7(1)$ (invalid).
If $5 \succ 8(1)$, then $8 \succ 7(2), 3 \succ 8(3), 8 \succ 4(2), 1 \succ 8(1), 8 \succ 6(2), 6 \succ 1(3), 7 \succ 6(1), 6 \succ 3(1), 5 \succ 6(2), 6 \succ 4(1)$ (invalid)

Case 6) Let $2 \succ 4(2), 4 \succ 1(3), 7 \succ 3(1), 5 \succ 2(3)$. Then, $3 \succ 5(1), 4 \succ 5(1), 1 \succ 7(2), 7 \succ 5(1)$,
If $5 \succ 8(2), 8 \succ 7(3), 3 \succ 8(2)$ (invalid).
If $5 \succ 8(3), 8 \succ 7(2), 3 \succ 8(3), 6 \succ 3(2), 8 \succ 6(1), 8 \succ 4(2), 1 \succ 8(1)$(invalid).

□

B. Experimental Results

For each of the models generating the data and for each of the algorithms considered above, we test the performance using various measures discussed below. All the hyper-parameters used in the DFLearn and 3D majority vote models are tuned using the cross validation method for both synthetic and real-world datasets.

For synthetic experiments, ADAM optimizer with binary cross-entropy loss is used in DFLearn with 100 epochs and batch size 32. For both Blade-Chest model and DFLearn, the number of embedding dimensions is a hyperparameter, which is fixed to be 10 for the synthetic experiments for data generated from the DF model. For data generated from other two models and real-world data, the number of dimensions for items is taken as 50 for both Blade-Chest model and DFLearn after doing the hyperparameter tuning. The other hyper-parameters in DFLearn are learning rate, kernel regularization parameter and number of hidden nodes in both the hidden layers.

In 3D majority vote, the parameters are generated from a uniform distribution for real-world data as well as synthetic data generated from DF model. But for data generated from the Salient Features model and BTL model, the parameters are generated from a Gaussian distribution with mean 0 and standard deviation 1. A standard deviation of $\frac{1}{\sqrt{k}}$ is used in 3D majority vote in order to generate the probabilities using the normal cumulative distribution function, where k is taken as a hyper-parameter and tuned using cross-validation. For GNNRank algorithm, $K = \log n$, where $n =$ number of items and K = number of top eigen vectors used, when the input features are unavailable.

B.1. 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.

Ground Truth Ranking: For the BTL model, the ground truth ranking is obtained by sorting the true scores in descending order. For both SF model and DF model, it is obtained by the Copeland procedure i.e., associating the score of an item as the number of 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 all other algorithms, 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 can observe that in general, DFLearn gives better rank correlation compared to the other baselines for the data generated from DF model.

c	RC	LRPR-2	SF-MLE	Blade-Chest	3D Majority Vote	DFLearn	GNNRank-Best
1	0.535 (0.013)	0.517 (0.005)	0.486 (0.007)	0.545 (0.01)	0.554 (0.036)	0.545 (0.012)	0.54 (0.01)
2	0.491 (0.015)	0.512 (0.003)	0.507 (0.003)	0.513 (0.012)	0.541 (0.02)	0.551 (0.01)	0.51 (0.012)
3	0.486 (0.009)	0.503 (0.004)	0.509 (0.003)	0.52 (0.01)	0.543 (0.01)	0.555 (0.016)	0.515 (0.009)
4	0.508 (0.01)	0.509 (0.003)	0.509 (0.004)	0.54 (0.009)	0.546 (0.01)	0.546 (0.009)	0.52 (0.009)
5	0.492 (0.016)	0.507 (0.004)	0.526 (0.003)	0.536 (0.009)	0.539 (0.014)	0.564 (0.009)	0.53 (0.01)
6	0.52 (0.009)	0.513 (0.004)	0.51 (0.003)	0.541 (0.012)	0.553 (0.02)	0.559 (0.02)	0.53 (0.01)

Table 1. Kendall-Tau Correlation of various algorithms when the data follows the DF model and the number of training pairs are varied as $cn \log n$ for various choices of c red implies the best, blue implies the second best

c	RC	LRPR-2	SF-MLE	Blade-Chest	3D Majority Vote	DFLearn	GNNRank - Best
1	0.427 (0.0005)	0.366 (0.002)	0.387 (0.0013)	0.319 (0.005)	0.39 (0.0089)	0.204 (0.009)	0.419 (0.0006)
2	0.397 (0.0004)	0.353 (0.001)	0.387 (0.0007)	0.159 (0.003)	0.225 (0.0153)	0.141 (0.008)	0.39 (0.0006)
3	0.372 (0.0004)	0.346 (0.001)	0.387 (0.0007)	0.117 (0.005)	0.304 (0.007)	0.117 (0.005)	0.37 (0.0005)
4	0.35 (0.0007)	0.34 (0.002)	0.389 (0.0011)	0.097 (0.002)	0.226 (0.016)	0.118 (0.006)	0.32 (0.0048)
5	0.333 (0.001)	0.339 (0.003)	0.39 (0.0008)	0.087 (0.003)	0.269 (0.012)	0.115 (0.006)	0.32 (0.0007)
6	0.321 (0.0011)	0.34 (0.004)	0.392 (0.0013)	0.077 (0.001)	0.136 (0.013)	0.106 (0.007)	0.32 (0.001)

Table 2. RMSE values of various algorithms when the data follows the DF model and the number of training pairs are varied as $cn \log n$ for various choices of c, red implies the best, blue implies the second best

c	d = 10	d = 20	d = 30	d = 40	d = 50
1	0.864 (0.023)	0.915 (0.018)	0.919 (0.01)	0.925 (0.007)	**0.93** (0.007)
2	0.946 (0.008)	0.949 (0.006)	0.951 (0.01)	0.966 (0.011)	**0.974** (0.006)
3	0.96 (0.004)	0.955 (0.005)	0.962 (0.011)	0.969 (0.012)	**0.978** (0.004)
4	0.956 (0.004)	0.956 (0.004)	0.96 (0.022)	0.966 (0.023)	**0.971** (0.01)
5	0.959 (0.005)	0.96 (0.005)	0.975 (0.012)	0.977 (0.011)	**0.985** (0.003)
6	0.959 (0.005)	0.961 (0.005)	0.969 (0.02)	0.97 (0.013)	**0.977** (0.003)

Figure 5. Effect of dimension on the fraction of cycles captured by the DFlearn algorithm under the DF model when trained with $cn \log(n)$ training pairs, with the average number of cycles $= 33,544$

c	d = 10	d = 20	d = 30	d = 40	d = 50
1	0.89 (0.004)	0.892(0.005)	0.895(0.005)	**0.909**(0.005)	0.897 (0.006)
2	0.903 (0.003)	0.911 (0.005)	0.903(0.003)	0.914(0.003)	**0.915**(0.004)
3	0.912 (0.003)	0.901(0.003)	0.906(0.005)	0.912(0.004)	**0.918**(0.004)
4	0.901 (0.004)	0.909(0.004)	0.**911**(0.004)	0.91(0.004)	0.907(0.004)
5	0.905 (0.004)	0.907(0.006)	0.906(0.005)	0.912(0.004)	**0.918**(0.004)
6	0.906 (0.004)	0.911(0.002)	0.907(0.003)	0.91(0.003)	**0.915**(0.003)

Figure 6. Effect of dimension on accuracy by the DFlearn algorithm under the DF model when trained with $cn \log(n)$ training pairs

c	RC	LRPR-2	SF-MLE	Blade-Chest	3D Majority Vote	DFLearn	GNNRank-Best
1	0.849(0.003)	0.705(0.012)	0.708(0.005)	0.787(0.007)	0.761(0.02)	0.867(0.004)	0.503(0.008)
2	0.892(0.003)	0.817(0.007)	0.723(0.004)	0.8(0.005)	0.856(0.003)	0.899(0.002)	0.506(0.007)
3	0.912(0.002)	0.863(0.007)	0.736(0.004)	0.795(0.005)	0.89 (0.004)	0.914(0.003)	0.514(0.011)
4	0.928(0.002)	0.876(0.004)	0.732(0.005)	0.788(0.008)	0.901(0.003)	0.928(0.002)	0.517(0.012)
5	0.939(0.003)	0.877(0.004)	0.74(0.005)	0.782(0.005)	0.913(0.003)	0.935(0.002)	0.512(0.01)
6	0.945(0.003)	0.889(0.014)	0.737(0.01)	0.77(0.003)	0.923(0.004)	0.94(0.003)	0.532(0.011)

	RC	LRPR-2	SF-MLE	Blade-Chest	3D Majority Vote	DFLearn	GNNRank - Best
1	0.2228(0.0006)	0.3008(0.0058)	0.2035(0.0049)	0.1914(0.0025)	0.2131(0.0301)	0.1029(0.0063)	0.2036(0.0004)
2	0.1916(0.0011)	0.2198(0.0036)	0.2056(0.0051)	0.1868(0.0009)	0.1094(0.0022)	0.0722(0.0036)	0.2035(0.0004)
3	0.1588(0.0011)	0.1892(0.0027)	0.2054(0.0052)	0.1909(0.0015)	0.0694(0.0007)	0.0598(0.0034)	0.2032(0.0004)
4	0.1309(0.0013)	0.181(0.0024)	0.2049(0.0053)	0.1983(0.002)	0.0604(0.0007)	0.0455(0.0018)	0.2031(0.0005)
5	0.0971(0.0006)	0.1736(0.003)	0.2059(0.0057)	0.2089(0.0018)	0.053(0.0008)	0.0404(0.0008)	0.2027(0.0005)
6	0.0681(0.0015)	0.1704(0.003)	0.2031(0.0055)	0.2188(0.001)	0.0477(0.0011)	0.0379(0.0014)	0.2025(0.0003)

c	RC	LRPR-2	SF-MLE	Blade-Chest	3D Majority Vote	DFLearn	GNNRank-Best
1	0.704 (0.007)	0.505 (0.018)	0.437 (0.009)	0.06 (0.08)	0.648(0.051)	0.735 (0.008)	0.004 (0.028)
2	0.786 (0.006)	0.645 (0.014)	0.464 (0.009)	0.069 (0.084)	0.815 (0.006)	0.799 (0.004)	0.012 (0.009)
3	0.83 (0.004)	0.732 (0.013)	0.485 (0.008)	0.074 (0.085)	0.843 (0.006)	0.83 (0.007)	0.036 (0.032)
4	0.856 (0.005)	0.756 (0.009)	0.477 (0.008)	0.084 (0.086)	0.856 (0.006)	0.856 (0.005)	0.036 (0.035)
5	0.88 (0.004)	0.761 (0.006)	0.491 (0.009)	0.085 (0.088)	0.88 (0.004)	0.874 (0.002)	0.024 (0.026)
6	0.892 (0.004)	0.779 (0.007)	0.487 (0.009)	0.08 (0.09)	0.896 (0.006)	0.884 (0.004)	0.046 (0.021)

Figure 7. Prediction Accuracy (top, higher is better), RMSE (middle, lower is better) and Kendall-Tau Correlation (bottom, higher is better) of various algorithms when the data follow and s the BTL model and the number of training pairs are varied as $cn \log n$ for various choices of c, red implies the best, blue implies the second best

c	RC	LRPR-2	SF-MLE	Blade-Chest	3D Majority Vote	DFLearn	GNN Rank- Best
1	0.608(0.01)	0.498(0.003)	0.899(0.034)	0.549(0.003)	0.559(0.021)	0.584(0.009)	0.516(0.013)
2	0.631(0.007)	0.503(0.002)	0.901(0.029)	0.548(0.004)	0.58(0.005)	0.614(0.01)	0.509(0.013)
3	0.651(0.008)	0.506(0.003)	0.938(0.029)	0.553(0.005)	0.589(0.005)	0.638(0.01)	0.512(0.017)
4	0.657(0.009)	0.521(0.006)	0.931(0.018)	0.551(0.006)	0.599(0.008)	0.643(0.008)	0.512(0.014)
5	0.681(0.005)	0.529(0.009)	0.938(0.023)	0.549(0.003)	0.622(0.01)	0.67(0.007)	0.528(0.013)
6	0.668(0.008)	0.546(0.011)	0.972(0.014)	0.544(0.004)	0.565(0.026)	0.657(0.01)	0.509(0.016)

c	RC	LRPR-2	SF-MLE	Blade-Chest	3D Majority Vote	DFLearn	GNN Rank - Best
1	0.0653(0.0009)	0.2823(0.0058)	0.0476(0.0061)	0.2205(0.0018)	0.147(0.0024)	0.117(0.0044)	0.203(0.0003)
2	0.0629(0.0006)	0.2237(0.0045)	0.0443(0.0058)	0.2138(0.0011)	0.13(0.0013)	0.0851(0.0019)	0.2032(0.0004)
3	0.0601(0.0005)	0.1963(0.0048)	0.0465(0.0053)	0.216(0.0008)	0.111(0.0022)	0.0735(0.0012)	0.203(0.0004)
4	0.0593(0.0007)	0.187(0.0035)	0.042(0.0039)	0.2247(0.0011)	0.1(0.001)	0.0706(0.0017)	0.2028(0.0006)
5	0.0578(0.0005)	0.174(0.003)	0.0396(0.0035)	0.2313(0.0012)	0.09(0.0016)	0.0623(0.0006)	0.2022(0.0005)
6	0.0583(0.0005)	0.164(0.0035)	0.0389(0.0045)	0.2399(0.0014)	0.132(0.0039)	0.0632(0.0011)	0.2032(0.0008)

c	RC	LRPR-2	SF-MLE	Blade-Chest	3D Majority Vote	DFLearn	GNNRank-Best
1	0.283 (0.025)	0.04 (0.025)	0.713 (0.09)	0.216 (0.017)	0.219 (0.034)	0.222 (0.022)	0.04 (0.031)
2	0.347 (0.018)	0.083 (0.033)	0.775 (0.076)	0.298 (0.018)	0.291 (0.026)	0.306 (0.023)	0.035 (0.032)
3	0.412 (0.017)	0.074 (0.024)	0.799 (0.094)	0.349 (0.018)	0.378 (0.024)	0.363 (0.024)	0.03 (0.04)
4	0.43 (0.023)	0.14 (0.023)	0.818 (0.061)	0.36 (0.021)	0.396 (0.014)	0.38 (0.019)	0.035 (0.037)
5	0.489 (0.015)	0.164 (0.037)	0.877 (0.064)	0.343 (0.01)	0.458 (0.025)	0.461 (0.017)	0.068 (0.031)
6	0.474 (0.016)	0.218 (0.036)	0.929 (0.047)	0.363 (0.017)	0.36 (0.05)	0.429 (0.017)	0.039 (0.04)

Figure 8. Prediction Accuracy (top, higher is better), RMSE (middle, lower is better) and Kendall-Tau Correlation (bottom, higher is better) of various algorithms when the data follows the Salient Features model and the number of training pairs are varied as $cn \log n$ for various choices of c, red implies the best, blue implies the second best

RMSE	RC	LRPR-2	SF-MLE	Blade-Chest	Majority Vote	DFLearn	GNNRank-Best
Jester	0.066(0.0002)	0.084(0.0003)	0.12(0.0002)	0.078(0.0003)	0.082(0.005)	0.065(0.0003)	0.157(0.0003)
MovieLens	0.471(0.0002)	0.472(0.0005)	0.449(0.0002)	0.532(0.0002)	0.441(0.0002)	0.416(0.0002)	0.477(0.0002)
DoTA	0.378(0.0013)	0.418(0.0043)	0.368(0.0013)	0.291(0.0013)	0.398(0.005)	0.332(0.0013)	0.379(0.0011)
StarCraft II : WoL	0.466(0.0003)	0.467(0.0006)	0.453(0.0009)	0.307(0.0005)	0.456(0.0006)	0.413(0.0005)	0.467(0.0004)
StarCraft II : HoTs	0.482(0.0003)	0.461(0.0031)	0.466(0.0005)	0.309(0.0004)	0.484(0.01)	0.419(0.0006)	0.482(0.0006)

Figure 9. RMSE values (lower is better) of various algorithms for real data, red implies the best, blue implies the second best

c	RC	LRPR-2	SF-MLE	Blade-Chest	3D Majority Vote	DFLearn	GNNRank-Best
1	0.741 (0.004)	0.802 (0.002)	0.784 (0.002)	0.815 (0.004)	0.768 (0.007)	0.89 (0.004)	0.745 (0.005)
2	0.732 (0.002)	0.797 (0.002)	0.782 (0.005)	0.801 (0.011)	0.797 (0.009)	0.898 (0.007)	0.75 (0.007)
3	0.73(0.003)	0.796 (0.002)	0.782 (0.004)	0.806(0.005)	0.751(0.009)	0.892 (0.01)	0.737(0.005)
4	0.728 (0.002)	0.797 (0.002)	0.784 (0.001)	0.802 (0.004)	0.755 (0.007)	0.892 (0.004)	0.744 (0.006)
5	0.736 (0.002)	0.798 (0.002)	0.785 (0.01)	0.821 (0.003)	0.761 (0.007)	0.892 (0.01)	0.74 (0.005)
6	0.738 (0.003)	0.801 (0.002)	0.784(0.003)	0.82 (0.002)	0.768 (0.009)	0.898 (0.009)	0.75 (0.006)

c	RC	LRPR-2	SF-MLE	Blade-Chest	3D Majority Vote	DFLearn	GNNRank-Best
1	0.424 (0.0005)	0.366 (0.002)	0.387 (0.0013)	0.319 (0.005)	0.39 (0.0089)	0.204 (0.009)	0.419 (0.0006)
2	0.427 (0.0003)	0.368 (0.003)	0.389 (0.001)	0.328 (0.018)	0.331(0.008)	0.165 (0.021)	0.404 (0.0006)
3	0.428 (0.0004)	0.364 (0.001)	0.391 (0.002)	0.323 (0.012)	0.33 (0.009)	0.193 (0.031)	0.405 (0.0005)
4	0.427 (0.0003)	0.362 (0.002)	0.392 (0.002)	0.324 (0.005)	0.331 (0.008)	0.193 (0.017)	0.403 (0.0048)
5	0.427 (0.0003)	0.364 (0.001)	0.392 (0.002)	0.301 (0.007)	0.33 (0.006)	0.175 (0.027)	0.404 (0.0007)
6	0.427 (0.0005)	0.363 (0.001)	0.393 (0.001)	0.31 (0.005)	0.33 (0.006)	0.16 (0.022)	0.4 (0.001)

c	RC	LRPR-2	SF-MLE	Blade-Chest	3D Majority Vote	DFLearn	GNNRank-Best
1	0.533 (0.013)	0.517 (0.005)	0.486 (0.007)	0.545 (0.01)	0.554 (0.036)	0.545 (0.012)	0.54 (0.01)
2	0.483 (0.012)	0.505 (0.005)	0.483 (0.007)	0.504(0.017)	0.541 (0.017)	0.523 (0.02)	0.51 (0.012)
3	0.471 (0.011)	0.503 (0.005)	0.482 (0.006)	0.493(0.009)	0.553 (0.032)	0.542 (0.027)	0.5 (0.009)
4	0.456 (0.009)	0.506 (0.004)	0.484 (0.004)	0.506 (0.022)	0.558 (0.029)	0.544 (0.013)	0.506 (0.009)
5	0.511 (0.009)	0.506 (0.003)	0.486 (0.02)	0.508 (0.009)	0.553 (0.019)	0.518 (0.014)	0.511 (0.01)
6	0.518 (0.016)	0.516 (0.007)	0.489 (0.01)	0.531 (0.016)	0.554 (0.025)	0.524(0.027)	0.52 (0.01)

Figure 10. Prediction Accuracy (top, higher is better), RMSE (middle, lower is better), Kendall-Tau Correlation (bottom, higher is better) of various algorithms when the data follows the DF model and the number of comparisons is varied for a fixed number of training pairs ($n \log n = 664$) as $c \log n$ for various choices of c, red implies the best, blue implies the second best