How Many Pairwise Preferences Do We Need to Rank a Graph Consistently?

Aadirupa Saha
Department of CSA
Indian Institute of Science, Bangalore
aadirupa@iisc.ac.in

Rakesh Shivanna
Google Brain, Mountain View
rakeshshivanna@google.com

Chiranjib Bhattacharyya
Department of CSA and Robert Bosch
Center for Cyberphysical Systems
Indian Institute of Science, Bangalore
chiru@iisc.ac.in

Abstract

We consider the problem of optimal recovery of true ranking of n items from a randomly chosen subset of their pairwise preferences. It is well known that without any further assumption, one requires a sample size of $\Omega(n^2)$ for the purpose. We analyze the problem with an additional structure of relational graph $G([n], E)$ over the n items added with an assumption of *locality*: Neighboring items are similar in their rankings. Noting the preferential nature of the data, we choose to embed not the graph, but, its *strong product* to capture the pairwise node relationships. Furthermore, unlike existing literature that uses Laplacian embedding for graph based learning problems, we use a richer class of graph embeddings—*orthonormal representations*—that includes (normalized) Laplacian as its special case. Our proposed algorithm, *Pref-Rank*, predicts the underlying ranking using an SVM based approach using the chosen embedding of the product graph, and is the first to provide *statistical consistency* on two ranking losses: *Kendall's tau* and *Spearman's footrule*, with a required *sample complexity* of $O(n^2\chi(\bar{G}))^{\frac{2}{3}}$ pairs, $\chi(\bar{G})$ being the *chromatic number* of the complement graph $\bar{G}$. Clearly, our sample complexity is smaller for dense graphs, with $\chi(\bar{G})$ characterizing the degree of node connectivity, which is also intuitive due to the *locality* assumption e.g. $O(n^{\frac{4}{3}})$ for union of k-cliques, or $O(n^{\frac{5}{3}})$ for random and power law graphs etc.—a quantity much smaller than the fundamental limit of $\Omega(n^2)$ for large n. This, for the first time, relates ranking complexity to structural properties of the graph. We also report experimental evaluations on different synthetic and real-world datasets, where our algorithm is shown to outperform the state of the art methods.

1 Introduction

The problem of ranking from pairwise preferences has widespread applications in various real-world scenarios e.g. web search (Page et al. 1998; Kleinberg 1999), gene classification, recommender systems (Theodoridis, Kotropoulos, and Panagakis 2013), image search (Geng, Yang, and Hua 2009) and more. Its of no surprise why the problem is so well studied in various disciplines of research, be that computer science, statistics, operational research or computational biology. In particular, we study the problem of ranking (or ordering) of set of n items on a graph, given some partial information of the relative ordering of the item pairs.

It is well known from the standard results of classical sorting algorithms, for any set of n items associated to an unknown deterministic ordering, say $\boldsymbol{\sigma}_n^*$, and given the learner has access to only preferences of the item pairs, in general one requires to observe $\Omega(n \log n)$ *actively* selected pairs (where the learner can choose which pair to observe next) to obtain the true underlying ranking $\boldsymbol{\sigma}_n^*$; whereas, with *random* selection of pairs, it could be as bad as $\Omega(n^2)$.

Related Work. Over the years, numerous attempts have been made to improve the above sample complexities by imposing different structural assumptions on the set of items or the underlying ranking model. In active sampling setting, (Jamieson and Nowak 2011) gives a sample complexity of $O(d \log^2 n)$, provided the true ranking is realizable in a d-dimensional embedding; (Braverman and Mossel 2008) and (Ailon 2012) proposed a near optimal recovery with sample complexity of $O(n \log n)$ and $O(n\text{poly}(\log n))$ respectively, under noisy permutation and tournament ranking model. For the non-active (random) sampling setting, (Wauthier, Jordan, and Jojic 2013) and (Negahban, Oh, and Shah 2012) gave a sample complexity bound of $O(n \log n)$ under noisy permutation (with $O(\log n)$ repeated sampling) and Bradley-Terry-Luce (BTL) ranking model. Recently, (Rajkumar and Agarwal 2016) showed a recovery guarantee of $O(nr \log n)$, given the preference matrix is rank r under suitable transformation.

However, existing literature on sample complexity for *graph based ranking problems* is sparse, where it goes without saying that the underlying structural representation of the data is extremely relevant in various real-world applications where the edge connections model item similarities e.g. In social network, connection among friends can be modelled as a graph, or in recommender systems, movies under same the genre should lie in close neighbourhood. It is important to note that a relational graph is different from imposing item dependencies through feature representations and much more practical, since side information of exact features may not even be available to the learner as required in the latter case.

Furthermore, the only few algorithmic contributions made on the problem of ranking on graphs – (Page et al. 1998; He et al. 2017; Del Corso and Romani 2016; Hsu et al. 2017) have not explored their theoretical performance. (Agarwal 2010; 2008) proposed an SVM-rank based algorithm, with

A full version of this paper is available at http://arxiv.org/abs/1811.02161

Table 1: Summary of sample complexities for ranking from pairwise preferences.

Reference	Assumption on the Ranking Model	Sampling Technique	Sample Complexity
(Braverman and Mossel 2008)	Noisy permutation	Active	$O(n \log n)$
(Jamieson and Nowak 2011)	Low d-dimensional embedding	Active	$O(d \log^2 n)$
(Ailon 2012)	Deterministic tournament	Active	$O(n\text{poly}(\log n))$
(Gleich and Lim 2011)	Rank-r pairwise preference with ν incoherence	Random	$O(n\nu r \log^2 n)$
(Negahban, Oh, and Shah 2012)	Bradley Terry Luce (BTL)	Random	$O(n \log n)$
(Wauthier, Jordan, and Jojic 2013)	Noisy permutation	Random	$O(n \log n)$
(Rajkumar and Agarwal 2016)	Low r-rank pairwise preference	Random	$O(nr \log n)$
(Niranjan and Rajkumar 2017)	Low d-rank feature with BTL	Random	$O(d^2 \log n)$
(Agarwal 2010)	Graph + Laplacian based ranking	Random	✗
Pref-Rank (This paper)	Graph + Edge similarity based ranking	Random	$O(n^2\chi(\bar{G}))^{\frac{2}{3}}$

generalization error bounds for the inductive and transductive graph ranking problems. (Agarwal and Chakrabarti 2007) derived generalization guarantees for PageRank algorithm. To the best of our knowledge, we are not aware of any literature which provide *statistical consistency* guarantees to recover the true ranking and analyze the required sample complexity, which remains the primary focus of this work.

Problem Setting. We precisely address the question: Given the additional knowledge of a relational graph on the set of n items, say $G([n], E)$, can we find the underlying ranking $\boldsymbol{\sigma}_n^*$ efficiently (i.e. with a sample complexity less than $\Omega(n^2)$)? Of course, in order to hope for achieving a better sample complexity, there must be a connection between the graph and the underlying ranking – question is how to model this?

A natural modelling could be to assume that similar items connected by an edge are close in terms of their rankings or similar node pairs have similar pairwise preferences e.g. In movie recommendations, if two movies A and B belongs to thriller genre and C belongs to comedy, and it is known that A is preferred over C (i.e. the true ranking over latent topics prefers thriller over comedy), then it is likely that B would be preferred over C; and the learner might not require an explicit (B, C) labelled pair – thus one could hope to reduce the sample complexity by inferring preference information of the neighbouring similar nodes. However, how to impose such a smoothness constraint remains an open problem.

One way out could be to assume the true ranking to be a smooth function over the graph Laplacian as also assumed in (Agarwal 2010). However, why should we confine ourself to the notion of Laplacian embedding based similarity when several other graph embeddings could be explored for the purpose? In particular, we use a broader class of *orthonormal representation* of graphs for the purpose, which subsumes (normalized) Laplacian embedding as a special case, and assume the ranking to be a *smooth function* with respect to the underlying embedding (see Sec. 2.1 for details).

Our Contributions. Under the smoothness assumptions, we show a sample complexity guarantee of $O(n^2\chi(\bar{G}))^{\frac{2}{3}}$ to achieve *ranking consistency* – the result is intuitive as it indicates smaller sample complexity for densely connected graph, as one can expect to gather more information about the neighboring nodes compared to a sparse graph. Our proposed *Pref-Rank* algorithm, to the best of our knowledge, is the first attempt in proving *consistency* on a large class of graph families with $\vartheta(G) = o(n)$, in terms of *Kendall's tau* and *Spearman's footrule* losses – It is developed on the novel idea of embedding nodes of the strong product graph $G \boxtimes G$, drawing inference from the preferential nature of the data and finally uses a kernelized-SVM approach to learn the underlying ranking. We summarize our contributions:

- *The choice of graph embedding*: Unlike the existing literature, which is restricted to Laplacian graph embedding (Ando and Zhang 2007), we choose to embed the strong product $G \boxtimes G$ instead of G, as our ranking performance measures penalizes every pairwise misprediction; and use a general class of orthonormal representations, which subsumes (normalized) Laplacian as a special case.
- *Our proposed preference based ranking algorithm: Pref-Rank* is a kernelized-SVM based method that inputs an embedding of pairwise graph $G \boxtimes G$. The generalization error of *Pref-Rank* involves computing the transductive *Rademacher complexity* of the function class associated with the underlying embedding used (see Thm. 3, Sec. 3).
- For the above, we propose to embed the nodes of $G \boxtimes G$ with 3 different orthonormal representations: (a) Kron-Lab($G \boxtimes G$) (b) PD-Lab(G) and (c) *LS*-labelling; and derive generalization error bounds for the same (Sec. 4).
- *Consistency:* We prove the existence of an optimal embedding in Kron-Lab($G \boxtimes G$) for which *Pref-Rank* is statistically consistent (Thm. 10, Sec. 5) over a large class of graphs, including power law and random graphs. To the best of our knowledge, this is the first attempt at establishing algorithmic consistency for graph ranking problems.
- *Graph Ranking Sample Complexity:* Furthermore, we show that observing $O(n^2\chi(\bar{G}))^{\frac{2}{3}}$ pairwise preferences is sufficient for *Pref-Rank* to be consistent (Thm. 12, Sec. 5.1), which implies that a *densely connected graph requires much smaller training data compared to a sparse graph* for learning the optimal ranking – as also intuitive. Our result is the first to connect the complexity of graph ranking problem to its structural properties. Our proposed bound is a significant improvement in sample complexity (for *random* selection of pairs) for dense graphs e.g. $O(n^{\frac{4}{3}})$ for union of k-cliques; and $O(n^{\frac{5}{3}})$ for random and power law graphs – a quantity much smaller than $\Omega(n^2)$.

Our experimental results demonstrate the superiority of *Pref-Rank* algorithm compared to *Graph Rank* (Agarwal 2010), Rank Centrality (Negahban, Oh, and Shah 2012) and

Inductive Pairwise Ranking (Niranjan and Rajkumar 2017) on various synthetic and real-world datasets; validating our theoretical claims. Table 1 summarizes our contributions.

2 Preliminaries and Problem Statement

Notations. Let $[n] := \{1, 2, \ldots, n\}$, for $n \in \mathbb{Z}_+$. Let x_i denote the i^{th} component of a vector $\mathbf{x} \in \mathbb{R}^n$. Let $\mathbf{1}\{\varphi\}$ denote an indicator function that takes the value 1, if the predicate φ is true and 0 otherwise. Let $\mathbf{1}_n$ denote an n-dimensional vector of all 1's. Let $S^{n-1} = \{\mathbf{u} \in \mathbb{R}^n \big| \|\mathbf{u}\|_2 = 1\}$ denote a $(n-1)$ dimensional sphere. For any matrix $\mathbf{M} \in \mathbb{R}^{m \times n}$, we denote the i^{th} column by $\mathbf{M}_i$, $\forall i \in [n]$ and $\lambda_1(\mathbf{M}) \geq \ldots \geq \lambda_n(\mathbf{M})$ to denote its sorted eigenvalues, $tr(\mathbf{M})$ to be its trace. Let $\mathbf{S}_n^+ \in \mathbb{R}^{n \times n}$ denote a set of $n \times n$ square symmetric positive semi-definite matrices. Let $G(V, E)$ denote a simple undirected graph, with vertex set $V = [n]$ and edge set $E \subseteq V \times V$. We denote its adjacency matrix by $\mathbf{A}_G$. Let $\bar{G}$ denote the complement graph of G, with the adjacency matrix $\mathbf{A}_{\bar{G}} = \mathbf{1}_n^\top \mathbf{1}_n - \mathbf{I} - \mathbf{A}_G$, $\mathbf{I}$ being the identity matrix.

Orthonormal Representation of Graphs. (Lovász 1979) An orthonormal representation of $G(V, E)$, $V = [n]$ is $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_n] \in \mathbb{R}^{d \times n}$ such that $\mathbf{u}_i^\top \mathbf{u}_j = 0$ whenever $(i, j) \notin E$ and $\mathbf{u}_i \in S^{d-1}$, $\forall i \in [n]$. Let $Lab(G)$ denote the set of all possible orthonormal representations of G given by $Lab(G) := \{\mathbf{U} \mid \mathbf{U} \text{ is an Orthonormal Representation}\}$. Consider the set of graph kernels $\mathcal{K}(G) := \{\mathbf{K} \in \mathbf{S}_n^+ \mid K_{ii} = 1, \forall i \in [n];\ K_{ij} = 0, \forall (i, j) \notin E\}$. (Jethava et al. 2013) showed the two sets to be equivalent i.e. for every $\mathbf{U} \in Lab(G)$, one can construct $\mathbf{K} \in \mathcal{K}(G)$ and vice-versa.

Definition 1. Lovász Number. *(Lovász 1979) Orthonormal representations $Lab(G)$ of a graph G is associated with an interesting quantity – Lovász number of G, defined as*

$$\vartheta(G) := \min_{\mathbf{U}\in Lab(G)} \min_{\mathbf{c}\in S^{d-1}} \max_{i\in V} \frac{1}{(\mathbf{c}^\top \mathbf{u}_i)^2}$$

Lovász Sandwich Theorem: If $I(G)$ and $\chi(G)$ denote the independence number and chromatic number of the graph G, then $I(G) \leq \vartheta(G) \leq \chi(\bar{G})$ (Lovász 1979).

Strong Product of Graphs. Given a graph $G(V, E)$, strong product of G with itself, denoted by $G \boxtimes G$, is defined over the vertex set $V(G \boxtimes G) = V \times V$, such that two nodes $(i, j), (i', j') \in V(G \boxtimes G)$ is adjacent in $G \boxtimes G$ iff $i = i'$ and $(j, j') \in E$, or $(i, i') \in E$ and $j = j'$, or $(i, i') \in E$ and $(j, j') \in E$. Also, it is well known from the classical work of (Lovász 1979) that $\vartheta(G \boxtimes G) = \vartheta^2(G)$.

2.1 Problem Statement

We study the problem of graph ranking on a simple, undirected graph $G(V, E)$, $V = [n]$. Suppose there exists a true underlying ranking $\boldsymbol{\sigma}_n^* \in \Sigma_n$ of the nodes V, where Σ_n is the set of all permutations of $[n]$, such that for any two distinct nodes $i, j \in V$, i is said to be preferred over j iff $\sigma_n^*(i) < \sigma_n^*(j)$. Clearly, without any structural assumption on how $\boldsymbol{\sigma}_n^*$ relates to the underlying graph $G(V, E)$, the knowledge of $G(V, E)$ is not very helpful in predicting $\boldsymbol{\sigma}_n^*$:

Ranking on Graphs: Locality Property. A ranking $\boldsymbol{\sigma}_n$ is said to have *locality property* if $\exists$ at least one ranking function $\mathbf{f} \in \mathbb{R}^n$ such that $f(i) > f(j)$ iff $\sigma(i) < \sigma(j)$ and

$$|f(i) - f(j)| \leq c, \text{ whenever } (i, j) \in E, \tag{1}$$

where $c > 0$ is a small constant that quantifies the "locality smoothness" of $\mathbf{f}$. One way is to model $\mathbf{f}$ as a smooth function over the Laplacian embedding $\mathbf{L}$ (Agarwal 2010) such that $\mathbf{f}^\top \mathbf{L} \mathbf{f} = \sum_{(i,j)\in E} A_G(i, j)\big(f_i - f_j\big)^2$ is small. However, we generalize this notion to a broader class of embeddings:

Locality with Orthonormal Representations: Formally, we try to solve for $\mathbf{f} \in \text{RKHS}(\mathbf{K})$[1] i.e. $\mathbf{f} = \mathbf{K}\boldsymbol{\alpha}$, for some $\boldsymbol{\alpha} \in \mathbb{R}^n$, where the *locality* here implies $\mathbf{f}$ to be a smooth function over the embedding $\mathbf{K} \in \mathcal{K}(G)$, or alternatively $\mathbf{f}^\top \mathbf{K}^\dagger \mathbf{f} \leq B$, where $\mathbf{K}^\dagger$ is the pseudo inverse of $\mathbf{K}$ and $B > 0$ is a small constant (details given in Appendix of the full arXiv version). Note that if G is a completely disconnected graph, then $\mathcal{K}(G) = \{\mathbf{I}_n\}$ is the only choice for $\mathbf{K}$ and f_i's are independent of each other, and the problem is as hard as the classical sorting of n independent items. But as the density of G increases, or equivalently $\vartheta(G) \leq \chi(\bar{G}) \ll n$, then $\mathcal{K}(G)$ becomes more expressive and the problem enters into an interesting regime, as the node dependencies come to play, aiding to faster learning rate. Recall that, however, we only have access to G, our task is to find a suitable kernel $\mathbf{K}$ that fits $\mathbf{f}$ on G and estimate $\boldsymbol{\sigma}_n^*$ accurately.

Problem Setup. Consider the set of all node pairs $\mathcal{P}_n = \big\{(i, j) \in V \times V \big| i < j\big\}$. Clearly $|\mathcal{P}_n| = \binom{n}{2}$. We will use $N = \binom{n}{2}$ and denote the pairwise preference label of the k^{th} pair (i_k, j_k) as $y_k \in \{\pm 1\}$, such that $y_k := \text{sign}(\sigma_n^*(i_k) - \sigma_n^*(j_k))$, $\forall k \in [N]$. The learning algorithm is given access to a set of randomly chosen node-pairs $S_m \subseteq \mathcal{P}_n$, such that $|S_m| = m \in [N]$. Without loss of generality, by renumbering the pairs we will assume the first m pairs to be labelled $S_m = \{(i_k, j_k)\}_{k=1}^m$, with the corresponding pairwise preference labels $\mathbf{y}_{S_m} = \{y_k\}_{k=1}^m$, and set of unlabelled pairs $\bar{S}_m = \mathcal{P}_n \backslash S_m = \{(i_k, j_k)\}_{k=m+1}^N$. Given G, S_m and $\mathbf{y}_{S_m}$, the goal of the learner is to predict a ranking $\hat{\boldsymbol{\sigma}}_n \in \Sigma_n$ over the nodes V, that gives an accurate estimate of the underlying true ranking $\boldsymbol{\sigma}_n^*$. We use the following ranking losses to measure performance (Monjardet 1998): **Kendall's Tau loss:** $d_k(\boldsymbol{\sigma}^*, \hat{\boldsymbol{\sigma}}) = \frac{1}{N}\sum_{k=1}^N \mathbf{1}\big((\sigma^*(i_k) - \sigma^*(j_k))(\hat{\sigma}(i_k) - \hat{\sigma}(j_k)) < 0\big)$ and **Spearman's Footrule loss:** $d_s(\boldsymbol{\sigma}^*, \hat{\boldsymbol{\sigma}}) = \frac{1}{n}\sum_{i=1}^n \Big|\sigma^*(i) - \hat{\sigma}(i)\Big|$. d_k measures the average number of mispredicted pairs, whereas d_s measures the average displacement of the ranking order. By Diaconi-Graham inequality (Kumar and Vassilvitskii 2010), we know for any $\boldsymbol{\sigma}, \boldsymbol{\sigma'} \in \Sigma_n$, $d_k(\boldsymbol{\sigma}, \boldsymbol{\sigma'}) \leq d_s(\boldsymbol{\sigma}, \boldsymbol{\sigma'}) \leq 2d_k(\boldsymbol{\sigma}, \boldsymbol{\sigma'})$.

Now instead of predicting $\hat{\boldsymbol{\sigma}}_n \in \Sigma_n$, suppose the learner is allowed to predict a pairwise score function $\mathbf{f} : \mathcal{P}_n \mapsto \mathbb{R} \setminus \{0\}$. Note, $\mathbf{f} = [f_k]_{k=1}^N \in (\mathbb{R} \setminus \{0\})^N$ can also be realized as a vector, where f_k denotes the score for every k^{th} pair (i_k, j_k), $k \in [N]$. We measure the prediction accuracy as **pairwise** (0-1) **loss:** $\ell^{0-1}(y_k, f_k) = \mathbf{1}\left(f_k y_k < 0\right)$, or using the convex surrogate loss functions – **hinge loss:** $\ell^{\text{hinge}}(y_k, f_k) = (1 - f_k y_k)_+$ or **ramp loss:** $\ell^{\text{ramp}}(y_k, f_k) = \min\{1, (1 - f_k y_k)_+\}$, where $(a)_+ = \max(a, 0)$.

[1] RKHS: Reproducing Kernel Hilbert Space.

In general, given a transductive learning framework, following the notations from (Ando and Zhang 2007; El-Yaniv and Pechyony 2007), for any pairwise preference loss ℓ, we denote the empirical (training) ℓ-error of $\mathbf{f}$ as $er^{\ell}_{S_m}(\mathbf{f}) = \frac{1}{m}\sum_{k=1}^{m} \ell(y_k, f_k)$, the generalization (test set) error as $er^{\ell}_{\bar{S}_m}(\mathbf{f}) = \frac{1}{N-m}\sum_{k=m+1}^{N} \ell(y_k, f_k)$ and the average pairwise misprediction error as $er^{\ell}_{n}(\mathbf{f}) = \frac{1}{N}\sum_{k=1}^{N} \ell(y_k, f_k)$.

2.2 Learners' Objective - Statistical Consistency for Graph Ranking from Pairwise Preferences

Let $\mathcal{G}$ be a graph family with infinite sequence of nodes $\mathcal{V} = \{v_n\}_{n=1}^{\infty}$. Let V_n denote the first n nodes of $\mathcal{V}$ and $G_n \in \mathcal{G}$ be a graph instance defined over $(V_n, E_1 \cup \ldots \cup E_n)$, where E_n is the edge information of node v_n with previously observed nodes $V_{n-1},\ n \geq 2$. Let $\boldsymbol{\sigma}^*_n \in \Sigma_n$ be the true ranking of the nodes V_n. Now, given G_n and $f \in (0,1)$ a fixed fraction, let Π_f be a uniform distribution on the random draw of $m(f) = \lceil Nf \rceil$ pairs of nodes from N possible pairs $\mathcal{P}_n$. Let $S_{m(f)} = \{(i_k, j_k) \in \mathcal{P}_n\}_{k=1}^{m(f)}$ be an instance of the draw, with corresponding pairwise preferences $\mathbf{y}_{S_{m(f)}} = \{y_k\}_{k=1}^{m(f)}$. Given $(G_n, S_{m(f)}, \mathbf{y}_{S_{m(f)}})$, a learning algorithm $\mathcal{A}$ that returns a ranking $\hat{\sigma}_n$ on the node set V_n is said to be statistically d-rank consistent *w.r.t.* $\mathcal{G}$ if

$$Pr_{S_{m(f)} \sim \Pi_f}\left(d(\boldsymbol{\sigma}^*_n, \hat{\boldsymbol{\sigma}}_n) \geq \epsilon\right) \to 0 \quad as \quad n \to \infty,$$

for any $\epsilon > 0$ and d being the Kendall's tau (d_k) or Spearman's footrule (d_s) ranking losses. In the next section we propose *Pref-Rank*, an SVM based graph ranking algorithm and prove it to be statistically d-rank consistent (Sec. 5) with 'optimal embedding' in Kron-Lab($G \boxtimes G$) (Sec. 4.1).

3 *Pref-Rank* - Preference Ranking Algorithm

Given a graph $G(V, E)$ and training set of pairwise preferences $(S_m, \mathbf{y}_{S_m})$, we design an SVM based ranking algorithm that treats each observed pair in S_m as a binary labelled training instance and outputs a pairwise score function $\mathbf{f} \in \mathbb{R}^N$, which is used to estimate the final rank $\hat{\boldsymbol{\sigma}}_n$.

Step 1. Select an embedding ($\tilde{\mathbf{U}}$): Choose a pairwise node embedding $\tilde{\mathbf{U}} = [\tilde{\mathbf{u}}_1, \cdots \tilde{\mathbf{u}}_N] \in \mathbb{R}^{d \times N}$, where any node pair $(i_k, j_k) \in \mathcal{P}_n$ is represented by $\tilde{\mathbf{u}}_k,\ \forall k \in [N]$. We discuss the suitable embedding schemes in Sec. 4.

Step 2. Predict pairwise scores ($\mathbf{f}^* \in \mathbb{R}^N$): We solve the binary classification problem given the embeddings $\tilde{\mathbf{U}}$ and pairwise node preferences $\{(\tilde{\mathbf{u}}_k, \mathbf{y}_k)\}_{k=1}^{m}$ using an SVM:

$$\min_{\mathbf{w} \in \mathbb{R}^d} \frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{k=1}^{m} \ell^{\text{hinge}}(y_k, \mathbf{w}^\top \tilde{\mathbf{u}}_k) \tag{2}$$

where $C > 0$ is a regularization hyperparameter. Note that the dual of the above formulation is given by:

$$\max_{\boldsymbol{\alpha} \in \mathbb{R}^m_+,\ \|\boldsymbol{\alpha}\|_\infty \leq C} \sum_{k=1}^{m} \alpha_k - \frac{1}{2}\sum_{k,k'=1}^{m} \alpha_k \alpha_{k'} y_k y_{k'} \tilde{\mathbf{K}}_{k,k'}$$

where $\tilde{\mathbf{K}} = \tilde{\mathbf{U}}^\top \tilde{\mathbf{U}}$ denotes the embedding kernel of the pairwise node instances. From standard results of SVM, we know that optimal solution of (2) gives $\mathbf{w}^* = \sum_{k=1}^{m} y_k \tilde{\mathbf{u}}_k \alpha_k = \tilde{\mathbf{U}}\boldsymbol{\beta}$, where $\boldsymbol{\beta} \in \mathbb{R}^N$ is such that $\beta_k = y_k \alpha_k,\ \forall k \in [m]$ and 0 otherwise. Since $y_k \in \{\pm 1\}$, $\|\boldsymbol{\alpha}\|_\infty = \|\boldsymbol{\beta}\|_\infty \leq C$. Thus for any $k \in [N]$, the score of the pair (i_k, j_k) is given by $f^*_k = \mathbf{w}^{*\top}\tilde{\mathbf{u}}_k = \sum_{l=1}^{m} y_l \alpha_l \tilde{\mathbf{u}}_l^\top \tilde{\mathbf{u}}_k$ or equivalently $\mathbf{f}^* = \tilde{\mathbf{U}}^\top \mathbf{w}^* = \tilde{\mathbf{U}}^\top \tilde{\mathbf{U}}\boldsymbol{\beta} = \tilde{\mathbf{K}}\boldsymbol{\beta}$, which suggests the following alternative formulation of SVM:

$$\max_{\mathbf{f} \in \mathbb{R}^N} \frac{1}{2}\mathbf{f}^\top \tilde{\mathbf{K}}^\dagger \mathbf{f} + C\sum_{k=1}^{m} \ell^{\text{hinge}}(y_k, f_k) \tag{3}$$

Clearly, if $\mathbf{f}^*$ denotes the optimal solution of (3), then we have $\mathbf{f}^* \in \{\mathbf{f} \mid \mathbf{f} = \tilde{\mathbf{K}}\boldsymbol{\beta},\ \boldsymbol{\beta} \in \mathbb{R}^N,\ \|\boldsymbol{\beta}\|_\infty \leq C\}$.

Remark 1. The regularization $\mathbf{f}^\top \tilde{\mathbf{K}}^\dagger \mathbf{f}$, precisely enforces the *locality* assumption of Sec. 2.1 *(see full version on arXiv).*

Step 3. Predict $\hat{\boldsymbol{\sigma}}_n \in \Sigma_n$ from pairwise scores $\mathbf{f}^*$: Given the score vector $\mathbf{f}^* \in \mathbb{R}^N$ as computed above, predict a ranking $\hat{\boldsymbol{\sigma}}_n \in \Sigma_n$ over the nodes V of G as follows:

1. Let $c(i)$ denote the number of wins of node $i \in V$ given by $\sum_{\{k=(i_k,j_k) \mid i_k = i\}} \mathbf{1}(f^*_k > 0) + \sum_{\{k=(i_k,j_k) \mid j_k = i\}} \mathbf{1}(f^*_k < 0)$.
2. Predict the ranking of nodes by sorting *w.r.t.* $c(i)$, i.e. choose any $\hat{\boldsymbol{\sigma}}_n \in$ argsort$(\mathbf{c})$, where argsort$(\mathbf{c}) = \{\boldsymbol{\sigma} \in \Sigma_n \mid \sigma(i) < \sigma(j),\ \text{if } c(i) > c(j),\ \forall i, j \in V\}$.

A brief outline of *Pref-Rank* is given below:

Algorithm 1 *Pref-Rank*

```
Input: $G(V, E)$ and subset of preferences $(S_m, \mathbf{y}_{S_m})$.
Init: Pairwise graph embedding $\tilde{\mathbf{U}} \in \mathbb{R}^{d \times N},\ d \in \mathbb{N}_+$.
Get $\mathbf{w}^*$ by solving the SVM objective (2).
Compute preference scores $\mathbf{f}^* = \tilde{\mathbf{U}}^\top \mathbf{w}^*$.
Count number of wins $c(i)$ for each node $i \in V$:
$c(i) := \sum_{\{k=(i_k,j_k) \mid i_k = i\}} \mathbf{1}(f^*_k > 0) + \sum_{\{k=(i_k,j_k) \mid j_k = i\}} \mathbf{1}(f^*_k < 0)$
Return: Ranking of nodes $\hat{\boldsymbol{\sigma}}_n \in$ argsort$(\mathbf{c})$.
```

3.1 Generalization Error of *Pref-Rank*

We now derive generalization guarantees of *Pref-Rank* on its test set error $er^{\ell^\rho}_{\bar{S}_m}(\mathbf{f}^*) = \frac{1}{N-m}\sum_{k=m+1}^{N} \ell^\rho(y_k, f^*_k)$, *w.r.t.* some loss function $\ell^\rho : \{\pm 1\} \times \mathbb{R} \mapsto \mathbb{R}_+$, where ℓ^ρ is assumed to be ρ-lipschitz ($\rho > 0$) with respect to its second argument i.e. $|\ell^\rho(y_k, f_k) - \ell^\rho(y_k, f'_k)| \leq \frac{1}{\rho}|f_k - f'_k|$, where $\mathbf{f}, \mathbf{f}' : \mathcal{P}_n \mapsto \mathbb{R}^N$ be any two pairwise score functions. We find it convenient to define the following function class complexity measure associated with orthonormal embeddings of pairwise preference strong product of graphs (as motivated in (Pelckmans, Suykens, and Moor 2007)):

Definition 2 **(Transductive Rademacher Complexity).** *Given a graph $G(V, E)$, let $\tilde{\mathbf{U}} \in \mathbb{R}^{d \times N}$ be any pairwise embedding of G and let col$(\tilde{\mathbf{U}})$ denote the column space spanned by $\tilde{\mathbf{U}}$. Then, for any function class $\mathcal{H}_{\tilde{\mathbf{U}}} = \{\mathbf{h} \mid$*

$\mathbf{h} : col(\tilde{\mathbf{U}}) \mapsto \mathbb{R}\}$ *associated with* $\tilde{\mathbf{U}}$*, its transductive Rademacher complexity is defined as*

$$R(\mathcal{H}_{\tilde{\mathbf{U}}}, \tilde{\mathbf{U}}, p) = \frac{1}{N}\mathbb{E}_{\boldsymbol{\gamma}}\left[\sup_{\mathbf{h}\in\mathcal{H}_{\tilde{\mathbf{U}}}} \sum_{k=1}^{N} \gamma_k \mathbf{h}(\tilde{\mathbf{u}}_k)\right],$$

where for any fixed $p \in (0, 1/2]$*,* $\boldsymbol{\gamma} = (\gamma_1, \ldots, \gamma_N)$ *is a vector of* i.i.d. *random variables such that* $\gamma_i \sim \{+1, -1, 0\}$ *with probability* p*,* p *and* $1 - 2p$ *respectively.*

We bound the generalization error of *Pref-Rank* in terms of the Rademacher complexity. Note the result below crucially depends on the fact that any score vector $\mathbf{f}^*$ returned by *Pref-Rank*, is of the form $\mathbf{f}^* = \tilde{\mathbf{U}}^\top\mathbf{w}^*$, for some $\mathbf{w}^* \in \{\mathbf{h} \mid \mathbf{h} = \tilde{\mathbf{U}}\boldsymbol{\beta}, \boldsymbol{\beta} \in \mathbb{R}^N, \|\boldsymbol{\beta}\|_\infty \leq C\}$, where $\tilde{\mathbf{U}} \in \mathbb{R}^{d\times N}$ be the embedding used in *Pref-Rank* (refer (2), (3) for details).

Theorem 3 (Generalization Error of *Pref-Rank*). *Given a graph* $G(V, E)$*, let* $\tilde{\mathbf{U}} \in \mathbb{R}^{d\times N}$ *be any pairwise embedding of G. For any* $f \in (0, 1/2]$*, let* Π_f *be a uniform distribution on the random draw of* $m(f) = \lceil Nf \rceil$ *pairs of nodes from* $\mathcal{P}_n$*, such that* $S_{m(f)} = \{(i_k, j_k) \in \mathcal{P}_n\}_{k=1}^{m(f)} \sim \Pi_f$*, with corresponding pairwise preference* $\mathbf{y}_{S_{m(f)}}$*. Let* $\bar{S}_{m(f)} = \mathcal{P}_n \backslash S_{m(f)}$*. Let* $\mathcal{H}_{\tilde{\mathbf{U}}} = \{\mathbf{w} \mid \mathbf{w} = \tilde{\mathbf{U}}\boldsymbol{\beta},\ \boldsymbol{\beta} \in \mathbb{R}^N,\ \|\boldsymbol{\beta}\|_\infty \leq C,\ C > 0\}$ *and* $\ell^\rho : \{\pm1\} \times \mathbb{R} \mapsto [0, B]$ *be a bounded,* ρ*-Lipschitz loss function. Then for any* $\delta > 0$*, with probability* $\geq 1 - \delta$ *over* $S_{m(f)} \sim \Pi_f$*,*

$$er^{\ell^\rho}_{\bar{S}_{m(f)}}(\mathbf{f}^*) \leq er^{\ell^\rho}_{S_{m(f)}}(\mathbf{f}^*) + \frac{R(\mathcal{H}_{\tilde{\mathbf{U}}}, \tilde{\mathbf{U}}, p)}{\rho f(1-f)} + \frac{C_1 B\sqrt{\ln\left(\frac{1}{\delta}\right)}}{(1-f)\sqrt{Nf}},$$

where $p = f(1 - f)$ *and* $\mathbf{f}^* = \tilde{\mathbf{U}}^\top\mathbf{w}^* \in \mathbb{R}^N$ *is pairwise score vector output by Pref-Rank and* $C_1 > 0$ *is a constant.*

Remark 2. It might appear from above that a higher value of $R(\mathcal{H}_{\tilde{\mathbf{U}}}, \tilde{\mathbf{U}}, p)$ leads to increased generalization error. However, note that there is a *tradeoff* between the first and second term, since a higher Rademacher complexity implies a richer function class $\mathcal{H}_{\tilde{\mathbf{U}}}$, which in turn is capable of producing a better prediction estimate $\mathbf{f}^* = \tilde{\mathbf{U}}^\top\mathbf{w}$, resulting in a much lower training set error $er^{\ell^\rho}_{S_{m(f)}}(\mathbf{f}^*)$. Thus, a *higher value of* $R(\mathcal{H}_{\tilde{\mathbf{U}}}, \tilde{\mathbf{U}}, p)$ *is desired* for better generalization.

Taking insights from Thm. 3, it follows that the performance of *Pref-Rank* crucially depends on the *Rademacher complexity* $R(\mathcal{H}_{\tilde{\mathbf{U}}}, \tilde{\mathbf{U}}, p)$ of the underlying function class $\mathcal{H}_{\tilde{\mathbf{U}}}$, which boils down to the problem of finding a "good" embedding $\tilde{\mathbf{U}}$. We address this issue in the next section.

4 Choice of Embeddings

We discuss different classes of pairwise graph embeddings and their generalization guarantees. Recalling the results of (Ando and Zhang 2007) (see Thm. 1), which provides a crucial characterization of the class of optimal embeddings for any graph based regularization algorithms, we choose to work with embeddings with normalized kernels, i.e. $\tilde{\mathbf{K}} = \tilde{\mathbf{U}}^\top\tilde{\mathbf{U}}$ such that $\tilde{K}_{kk} = 1, \forall k \in [N]$. The following theorem analyses the Rademacher complexity of 'normalized' embeddings:

Theorem 4 (Rademacher Complexity of Orthonormal Embeddings). *Given* $G(V, E)$*, let* $\tilde{\mathbf{U}} \in \mathbb{R}^{d\times N}$ *be any 'normalized' node-pair embedding of* $G \boxtimes G$*, let* $\tilde{\mathbf{K}} = \tilde{\mathbf{U}}^\top\tilde{\mathbf{U}}$ *be the corresponding graph-kernel, then* $R(\mathcal{H}_{\tilde{\mathbf{U}}}, \tilde{\mathbf{U}}, p) \leq C\sqrt{2p\lambda_1(\tilde{\mathbf{K}})}$*, where* $\lambda_1(\tilde{\mathbf{K}})$ *is the largest eigenvalue of* $\tilde{\mathbf{K}}$*.*

Note that the above result does not educate us on the choice of $\tilde{\mathbf{U}}$ – we impose more structural constraints and narrow down the search space of optimal 'normalized' graph embeddings and propose the following special classes:

4.1 Kron-Lab($G \boxtimes G$): Kronecker Product Orthogonal Embedding

Given any graph $G(V, E)$, with $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n] \in \mathbb{R}^{d\times n}$ being an orthogonal embedding of G, i.e. $\mathbf{U} \in \text{Lab}(G)$, its Kronecker Product Orthogonal Embedding is given by:

$$\begin{aligned}\text{Kron-Lab}(G \boxtimes G) := \{&\tilde{\mathbf{U}} \in \mathbb{R}^{d^2\times n^2} \mid \tilde{\mathbf{U}} = \mathbf{U} \otimes \mathbf{U},\\ &\mathbf{U} \in \mathbb{R}^{d\times n} \text{ such that } \mathbf{U} \in \text{Lab}(G)\},\end{aligned}$$

where $\otimes$ is the kronecker (or outer) product of two matrix. The *'niceness'* of the above embedding lies in the fact that one can construct $\tilde{\mathbf{U}} \in$ Kron-Lab($G \boxtimes G$) from any orthogonal embedding of the original graph $\mathbf{U} \in \text{Lab}(G)$ – let $\mathbf{K} := \mathbf{U}^\top\mathbf{U}$ and $\tilde{\mathbf{K}} := \tilde{\mathbf{U}}^\top\tilde{\mathbf{U}}$, we see that for any two $k, k' \in [n^2]$, $\tilde{\mathbf{K}}_{kk'} = \tilde{\mathbf{u}}_k^\top\tilde{\mathbf{u}}'_k = (\mathbf{u}_{i_k} \otimes \mathbf{u}_{j_k})^\top(\mathbf{u}_{i_{k'}} \otimes \mathbf{u}_{j_{k'}}) = (\mathbf{u}_{i_k}^\top\mathbf{u}_{i_{k'}})(\mathbf{u}_{j_k}^\top\mathbf{u}_{j_{k'}}) = \mathbf{K}_{i_k i_{k'}}\mathbf{K}_{j_k j_{k'}}$, where $(i_{(\cdot)}, j_{(\cdot)}) \in [n] \times [n]$ are the node pairs corresponding to k, k'. Hence, $\tilde{\mathbf{K}} = \mathbf{K} \otimes \mathbf{K}$. Note that when $k = k'$, we have $\tilde{\mathbf{K}}_{kk} = 1$, as $\mathbf{U} \in \text{Lab}(G)$ and $K_{ii} = 1, \forall i \in [n]$. This ensures that the kronecker product graph kernel $\tilde{\mathbf{K}}$ satisfies the optimality criterion of 'normalized' embedding as previously discussed.

Lemma 5 (Rademacher Complexity of Kron-Lab($G \boxtimes G$)). *Consider any* $\mathbf{U} \in$ *Lab(G),* $\mathbf{K} = \mathbf{U}^\top\mathbf{U}$ *and the corresponding* $\tilde{\mathbf{U}} \in$ *Kron-Lab(G* $\boxtimes$ *G). Then for any* $p \in (0, 1/2]$ *and* $\mathcal{H}_{\tilde{\mathbf{U}}} = \{\mathbf{w} \mid \mathbf{w} = \tilde{\mathbf{U}}\boldsymbol{\beta},\ \boldsymbol{\beta} \in R^N,\ \|\boldsymbol{\beta}\|_\infty \leq C, C > 0\}$ *we have,* $R(\mathcal{H}_{\tilde{\mathbf{U}}}, \tilde{\mathbf{U}}, p) \leq C\lambda_1(\mathbf{K})\sqrt{2p}$*.*

Above leads to the following generalization guarantee:

Theorem 6 (Generalization Error of *Pref-Rank* with Kron-Lab($G \boxtimes G$)). *For the setting as in Thm. 3 and Lem. 5, for any* $\tilde{\mathbf{U}} \in$ *Kron-Lab(G* $\boxtimes$ *G), we have*

$$er^{\ell^\rho}_{\bar{S}}(\mathbf{f}^*) \leq er^{\ell^\rho}_{S}(\mathbf{f}^*) + \frac{C\lambda_1(\mathbf{K})\sqrt{2}}{\rho\sqrt{f(1-f)}} + \frac{C_1 B}{1-f}\sqrt{\frac{\log(\frac{1}{\delta})}{Nf}}$$

4.2 Pairwise Difference Orthogonal Embedding

Given graph $G(V, E)$, let $\mathbf{U} = [\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n] \in \mathbb{R}^{d\times n}$ be such that $\mathbf{U} \in \text{Lab}(G)$. We define the class of *Pairwise Difference Orthogonal Embeddings* of G as:

$$\begin{aligned}\text{PD-Lab}(G) := \{&\tilde{\mathbf{U}} \in \mathbb{R}^{d\times N} \mid \tilde{\mathbf{u}}_{ij} = \mathbf{u}_i - \mathbf{u}_j\ \forall (i, j) \in \mathcal{P}_n,\\ &\mathbf{U} \in \mathbb{R}^{d\times n} \text{ such that } \mathbf{U} \in \text{Lab}(G)\}\end{aligned}$$

Let $\mathbb{E} = [\mathbf{e}_i - \mathbf{e}_j]_{(i,j)\in\mathcal{P}_n} \in \{0, \pm1\}^{n\times N}$, where $\mathbf{e}_i$ denotes the i^{th} standard basis of $\mathbb{R}^n$, $\forall i \in [n]$; then it is easy to note that $\tilde{\mathbf{U}} = \mathbf{U}\mathbb{E} \in$ PD-Lab(G) and the corresponding graph kernel is given by $\tilde{\mathbf{K}} = \mathbb{E}^\top \mathbf{K}\mathbb{E}$. For PD embedding, we get:

Lemma 7 (Rademacher Complexity of PD-Lab(G)). *Consider any $\mathbf{U} \in$ Lab(G), $\mathbf{K} = \mathbf{U}^\top \mathbf{U}$ and the corresponding $\tilde{\mathbf{U}} \in$ PD-Lab(G). Then for any $p \in (0, 1/2]$ and $\mathcal{H}_{\tilde{\mathbf{U}}} = \{\mathbf{w} \mid \mathbf{w} = \tilde{\mathbf{U}}\boldsymbol{\beta},\ \boldsymbol{\beta} \in R^N,\ \|\boldsymbol{\beta}\|_2 \le tC\sqrt{N},\ C > 0\}$, we have $R(\mathcal{H}_{\tilde{\mathbf{U}}}, \tilde{\mathbf{U}}, p) \le 2C\sqrt{pn\lambda_1(\mathbf{K})}$.*

Similarly as before, using above result we can show that:

Theorem 8 (Generalization Error of *Pref-Rank* with PD-Lab(G)). *For the setting as in Thm. 3 and Lem. 7, for any $\tilde{\mathbf{U}} \in$ PD-Lab(G), we have*

$$er_{\bar{S}}^{\ell^\rho}(\mathbf{f}^*) \le er_{S}^{\ell^\rho}(\mathbf{f}^*) + \frac{2C\sqrt{n\lambda_1(\mathbf{K})}}{\rho\sqrt{f(1-f)}} + \frac{C_1 B}{1-f}\sqrt{\frac{\log(\frac{1}{\delta})}{Nf}}$$

Recall from Thm. 3 that $\mathbf{f}^* = \tilde{\mathbf{U}}^\top \mathbf{w}$. Thus the *'niceness'* of PD-Lab(G) lies in the fact that it comes with the free transitivity property – for any two node pairs $k_1 := (i, j)$ and $k_2 := (j, l)$, if $\mathbf{f}^*$ scores node i higher than j i.e. $f^*_{k_1} > 0$, and node j higher than node l i.e. $f^*_{k_2} > 0$; then for any three nodes $i, j, l \in [n]$, this automatically implies $f^*_{k_3} > 0$, where $k_3 := (i, l)$ i.e. node i gets a score higher than node l.

Remark 3. Although Lem. 5 and 7 shows that both Kron-Lab($G \boxtimes G$) and PD-Lab(G) are associated to rich expressive function classes with high Rademacher complexity, *the superiority of Kron-Lab($G \boxtimes G$) comes with an additional consistency guarantee*, as we will derive in Sec. 5.

4.3 *LS*-labelling based Embedding

The embedding (graph kernel) corresponding to *LS*-labelling (Luz and Schrijver 2005) of graph G is given by:

$$\mathbf{K}_{LS}(G) = \frac{\mathbf{A}_G}{\tau} + \mathbf{I}_n, \text{ where } \tau \ge |\lambda_n(\mathbf{A}_G)|, \tag{4}$$

where $\mathbf{A}_G$ is the adjacency matrix of graph G. It is known that $\mathbf{K}_{LS} \in \mathbb{R}^{n \times n}$ is symmetric and positive semi-definite, and hence defines a valid graph kernel; also $\exists \mathbf{U}_{LS} \in$ Lab(G) such that $\mathbf{U}_{LS}^\top \mathbf{U}_{LS} = \mathbf{K}_{LS}$. We denote $\mathbf{U}_{LS}$ to be the corresponding embedding matrix for *LS*-labelling. We define *LS*-labelling of the strong product of graphs as:

$$\tilde{\mathbf{K}}_{LS}(G \boxtimes G) = \mathbf{K}_{LS}(G) \otimes \mathbf{K}_{LS}(G) \tag{5}$$

and equivalently, the embedding matrix $\tilde{\mathbf{U}}_{LS}(G \boxtimes G) = \mathbf{U}_{LS}(G) \otimes \mathbf{U}_{LS}(G)$. Similar to Kron-Lab($G \boxtimes G$), we have $\tilde{\mathbf{K}}_{LS}(k, k) = 1,\ \forall k \in [n^2]$, since $\mathbf{K}_{LS}(i, i) = 1,\ \forall i \in [n]$. Following result shows that $\tilde{\mathbf{K}}_{LS}(G \boxtimes G)$ has high Rademacher complexity on random $G(n, q)$ graphs.

Lemma 9. *Let $G(n, q)$ be a Erdós-Réyni random graph, where each edge is present independently with probability $q \in [0, 1]$, $q = O(1)$. Then, the Rademacher complexity of the function class associated with $\tilde{\mathbf{K}}_{LS}(G \boxtimes G)$ is $O(\sqrt{n})$.*

Laplacian based Embedding. This is the most popular choice of graph embedding that uses the inverse of the Laplacian matrix for the purpose. Formally, let d_i denotes the degree of vertex $i \in [n]$ in graph G, i.e. $d_i = (\mathbf{A}_G)_i^\top \mathbf{1}_n$, and $\mathbf{D}$ denote a diagonal matrix such that $D_{ii} = d_i, \forall i \in [n]$. Then, the Laplacian and normalized Laplacian kernel of G is defined as follows[2]: $\mathbf{K}_{Lap}(G) = (\mathbf{D} - \mathbf{A}_G)^\dagger$ and $\mathbf{K}_{nLap}(G) = (\mathbf{I}_n - \mathbf{D}^{-1/2}\mathbf{A}_G\mathbf{D}^{-1/2})^\dagger$. Though widely used (Agarwal 2010; Ando and Zhang 2007), it is not very expressive on dense graphs with high $\chi(G)$ – we observe that the Rademacher complexity of function class associated with Laplacian is an order magnitude smaller than that of *LS*-labelling. See full version on arXiv for details.

5 Consistency with Kron-Lab($G \boxtimes G$)

In this section, we show that *Pref-Rank* is provably statistically consistent while working with *kronecker product orthogonal embedding* Kron-Lab($G \boxtimes G$)(see Sec. 4.1).

Theorem 10 (Rank-Consistency). *For the setting as in Sec. 2.2, there exists an embedding $\tilde{\mathbf{U}}_n \in$ Kron-Lab($G_n \boxtimes G_n$) such that, if $\boldsymbol{\sigma}_n \in \mathbb{R}^N$ denotes the pairwise scores returned by Pref-Rank on input $(\tilde{\mathbf{U}}_n, S_m(f), \mathbf{y}_{S_{m(f)}})$, then $\forall G_n \in \mathcal{G}$, with probability $\ge \left(1 - \frac{1}{N}\right)$ over $S_{m(f)} \sim \Pi_f$*

$$d(\boldsymbol{\sigma}_n^*, \hat{\boldsymbol{\sigma}}_n) = O\left(\left(\frac{\vartheta(G_n)}{nf}\sqrt{\frac{1-f}{f}}\right)^{\frac{1}{2}} + \sqrt{\frac{\ln n}{Nf}}\right),$$

where d denotes Kendall's tau (d_k) or Spearman's footrule (d_s) ranking loss functions.

Consistency follows from the fact that for large families of graphs including random graphs (Coja-Oghlan 2005) and power law graphs (Jethava et al. 2013), $\vartheta(G_n) = o(n)$.

5.1 Sample Complexity for Ranking Consistency

We analyze the minimum fraction of pairwise node preferences f^* to be observed for *Pref-Rank* algorithm to be statistically ranking consistent. We refer the required sample size $m(f^*) = \lceil Nf^* \rceil$ as *ranking sample complexity*.

Lemma 11. *If $\mathcal{G}$ in Thm. 10 is such that $\vartheta(G_n) = n^c$, $0 \le c < 1$, then observing only $f^* = O\left(\frac{\sqrt{\vartheta(G_n)}}{n^{\frac{1}{2} - \varepsilon}}\right)^{\frac{4}{3}}$ fraction of pairwise node preferences is sufficient for Pref-Rank to be statistically ranking consistent, for any $0 < \varepsilon < \frac{(1-c)}{2}$.*

Note that one could potentially choose any $\varepsilon \in \left(0, \frac{1-c}{2}\right)$ for the purpose – the *tradeoff* lies in the fact that a higher ε leads to *faster convergence rate of $d(\boldsymbol{\sigma}_n^*, \hat{\boldsymbol{\sigma}}_n) = O(\frac{1}{n^\varepsilon})$*, although at the *cost of increased sample complexity*; on the contrary setting $\varepsilon \to 0$ gives a smaller sample complexity, with significantly slower convergence rate (see proof of Lem. 11 in the full version). We further extend Lem. 11 and relate ranking sample complexity to structural properties of the graph – *coloring number* of the complement graph $\chi(\bar{G})$.

Theorem 12. *Consider a graph family $\mathcal{G}$ such that $\chi(\bar{G}_n) = o(n)$, $\forall G_n \in \mathcal{G}$. Then observing $O(n^2\chi(\bar{G}))^{\frac{2}{3}}$ pairwise preferences is sufficient for Pref-Rank to be ranking consistent.*

[2] † denotes the pseudo inverse.

Above conveys that for *dense graphs we need fewer pairwise samples compared to sparse graphs* as $\chi(\bar{G})$ reduces with increasing graph density. We discuss the sample complexities for some special graphs below where $\vartheta(G) = o(n)$.

Corollary 13 (**Ranking Consistency on Special Graphs**). *Pref-Rank algorithm achieves consistency on the following graph families, with the required sample complexities –* (a) Complete graphs*:* $O(n^{\frac{4}{3}})$ (b) Union of k disjoint cliques*:* $O(n^{\frac{4}{3}}k^{\frac{2}{3}})$ (c) Complement of power-law graphs*:* $O(n^{\frac{5}{3}})$ (d) Complement of k-colorable graphs*:* $O(n^{\frac{4}{3}}k^{\frac{2}{3}})$ (e) Erdős Réyni random $G(n, q)$ graphs *with* $q = O(1)$*:* $O(n^{\frac{5}{3}})$.

Remark 4. Thm. 10 along with Lem. 11 suggest that if the graph satisfies a crucial structural property: $\vartheta(G) = o(n)$ and given sufficient sample of $\Omega(n^2\vartheta(G))^{\frac{2}{3}}$ pairwise preferences, *Pref-Rank* yields consistency. Note that $\vartheta(G) \le \chi(\bar{G}) \le n$, where the last inequality is tight for completely disconnected graph – which implies one need to observe $\Omega(n^2)$ pairs for consistency, as a disconnected graph does not impose any structure on the rank. Smaller the $\vartheta(G)$, denser the graph and we attain consistency observing a smaller number of node pairs, the best is of course is when G is a clique, as $\vartheta(G) = 1$! Thus, for sparse graphs with $\vartheta(G) = \Theta(n)$, consistency and learnability is far fetched without observing $\Omega(n^2)$ pairs.

Note that proof of Thm. 10 relies on the fact that the maximum SVM margin attained for the formulation (2) is $\vartheta(G \boxtimes G)$, which is achieved by *LS*-labelling on Erdős Réyni random graphs (Shivanna and Bhattacharyya 2014); and thus guarantee consistency, with $O(n^{\frac{5}{3}})$ sample complexity.

6 Experiments

We conducted experiments on both real-world and synthetic graphs, comparing *Pref-Rank* with the following algorithms:

Algorithms. (a) **PR-Kron**: *Pref-Rank* with $\tilde{\mathbf{K}}_{LS}(G \boxtimes G)$ (see Eqn. (5)) (b) **PR-PD**: *Pref-Rank* with PD-Lab(G) with *LS*-labelling i.e. $\mathbf{U} = \mathbf{U}_{LS}$, (c) **GR**: Graph Rank (Agarwal 2010), (d) **RC**: Rank Centrality (Negahban, Oh, and Shah 2012) and (e) **IPR**: Inductive Pairwise Ranking, with Laplacian as feature embedding (Niranjan and Rajkumar 2017).

Recall from the list of algorithms in Table 1. Except (Agarwal 2010), none of the others applies to ranking on graphs. Moreover, they work under specific models, e.g. *noisy permutations* (Wauthier, Jordan, and Jojic 2013), (Rajkumar and Agarwal 2016) requires knowledge of the preference matrix rank. Nevertheless, we compare with Rank Centrality (works under BTL model) and Inductive Pairwise Ranking (requires item features), but as expected both perform poorly.

Performance Measure. Note the generalization guarantee of Thm. 3 not only holds for full ranking but for any *general preference learning problem*, where the nodes of G are assigned to an underlying preference vector $\boldsymbol{\sigma}_n^* \in \mathbb{R}^n$. Similar as before, the goal is to predict a pairwise score vector $\mathbf{f} \in \mathbb{R}^N$ to optimize the average pairwise mispredictions *w.r.t.* some loss function $\ell : \{\pm 1\} \times \mathbb{R} \setminus \{0\} \mapsto \mathbb{R}_+$ defined as:

$$er_D^{\ell}(\mathbf{f}) = \frac{1}{|D|}\sum_{k \in D} \ell(y_k, f_k), \qquad (6)$$

where $D = \{(i_k, j_k) \in \mathcal{P}_n \mid \sigma_n^*(i_k) \neq \sigma_n^*(j_k),\ k \in [N]\} \subseteq \mathcal{P}_n$ denotes the subset of node pairs with distinct preferences and $y_k = \text{sign}(\sigma_n^*(j_k) - \sigma_n^*(i_k)),\ \forall k \in D$. In particular, *Pref-Rank* applies to *bipartite ranking* **(BR)**, where $\boldsymbol{\sigma}_n^* \in \{\pm 1\}^n$, *categorical or d-class ordinal ranking* **(OR)**, where $\boldsymbol{\sigma}_n^* \in [d]^n,\ d < n$, and the original *full ranking* **(FR)** problem as motivated in Sec. 2.1. We consider all three tasks with **pairwise** 0-1 **loss**, i.e. $\ell(y_k, f_k) = \mathbf{1}(y_k f_k < 0)$.

6.1 Synthetic Experiments

Graphs. We use 3 *types of graphs*, each with $n = 30$ nodes: (a) *Union of k-disconnected cliques* with $k = 2$ and 10, (b) *r-Regular graphs* with $r = 5$ and 15; and (c) $G(n, q)$ *Erdős Réyni random graphs* with edge probability $q = 0.2$ and 0.6.

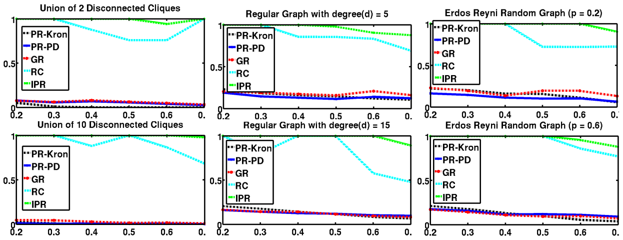

Figure 1: Synthetic Data: Average number of mispredictions ($er_D^{\ell^{0\text{-}1}}(\mathbf{f})$, Eqn. (6)) vs fraction of sampled pairs (f).

Generating $\boldsymbol{\sigma}_n^*$. For each of the above graphs, we compute $\mathbf{f}^* = \mathbf{A}_G\boldsymbol{\alpha}$, where $\boldsymbol{\alpha} \in [0, 1]^n$ is generated randomly, and set $\boldsymbol{\sigma}_n^* = \text{argsort}(\mathbf{f}^*)$ (see *Pref-Rank*, Step 3 for definition).

We report the average performance across 10 repeated runs in Fig. 6.1. In all the cases, our proposed algorithms **PR-Kron** and **PR-PD** outperforms the rest, with **GR** performing competitively. As expected, **RC** and **IPR** perform poorly as they could not exploit the underlying graph *locality* property.

6.2 Real-World Experiments

Datasets. We use 6 standard real-world datasets[3] for three graph learning tasks – (a) *Heart* and *Fourclass* for **BR**, (b) *Vehicle* and *Vowel* for **OR**, and (c) *House* and *Mg* for **FR**.

Graph Generation. For each dataset, we select 10 random subsets of 40 items each and construct a similarity matrix using RBF kernel, where $(i, j)^{\text{th}}$ entry is given by $\exp\left(\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\mu^2}\right)$, $\mathbf{x}_i$ being the feature vector and μ the average distance. For each of the 10 subsets, we constructed a graph by thresholding the similarity matrices about the mean.

Generating $\boldsymbol{\sigma}_n^*$. For each dataset, the provided item labels are used as the score vector $\mathbf{f}^*$ and we set $\boldsymbol{\sigma}_n^* = \text{argsort}(\mathbf{f}^*)$.

For each of the task, we report the average error across 10 randomly drawn subsets in Fig. 6.2. As before, our proposed methods **PR-Kron** and **PR-PD** perform the best, followed by **GR**. Once again **RC** and **IPR** perform poorly[4]. Note that, the performance error increases from bipartite ranking **(BR)** to full ranking **(FR)**, former being a relatively simpler task.

[3] https://www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/

[4] We omit them for **BR** and **OR** for better comparisons.

Results on more datasets are provided in the supplementary of the full version on arXiv.

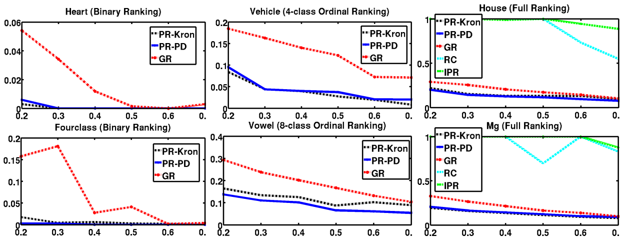

Figure 2: Real-World Data: Average number of mispredictions ($er_D^{\ell^{0\text{-}1}}(\mathbf{f})$, Eqn. (6)) vs fraction of sampled pairs (f).

7 Conclusion and Future Work

In this paper we address the problem of ranking nodes of a graph $G([n], E)$ given a random subsample of their pairwise preferences. Our proposed *Pref-Rank* algorithm, guarantees consistency with a required *sample complexity* of $O\big(n^2\chi(\bar{G})\big)^{\frac{2}{3}}$ – also gives novel insights by relating the ranking sample complexity with graph structural properties through the chromatic number of $\bar{G}$, i.e. $\chi(\bar{G})$, for the first time. One possible future direction is to extend the setting to noisy preferences e.g. using BTL model (Negahban, Oh, and Shah 2012), or analyse the problem with other measures of ranking losses e.g. NDCG, MAP (Agarwal 2008). Furthermore, proving consistency of *Pref-Rank* algorithm using PD-Lab(G) also remains an interesting direction to explore.

Acknowledgements

Thanks to the anonymous reviewers for their insightful comments. This work is partially supported by an Amazon grant to the Department of CSA, IISc and Qualcomm travel grant.

References

Agarwal, A., and Chakrabarti, S. 2007. Learning Random Walks to Rank Nodes in Graphs. In *Proceedings of the 24th international conference on Machine learning*, 9–16. ACM.

Agarwal, S. 2008. Transductive Ranking on Graphs. *Tech Report*.

Agarwal, S. 2010. Learning to Rank on Graphs. *Machine learning* 81(3):333–357.

Ailon, N. 2012. An Active Learning Algorithm for Ranking from Pairwise Preferences with an Almost Optimal Query Complexity. *Journal of Machine Learning Research* 13(Jan):137–164.

Ando, R. K., and Zhang, T. 2007. Learning on Graph with Laplacian Regularization. In *Advances in Neural Information Processing Systems*, 25–32.

Braverman, M., and Mossel, E. 2008. Noisy Sorting without Resampling. In *Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, 268–276. Society for Industrial and Applied Mathematics.

Coja-Oghlan, A. 2005. The Lovász Number of Random Graphs. *Combinatorics, Probability and Computing* 14(04):439–465.

Del Corso, G. M., and Romani, F. 2016. A Multi-class Approach for Ranking Graph Nodes: Models and Experiments with Incomplete Data. *Information Sciences* 329:619–637.

El-Yaniv, R., and Pechyony, D. 2007. Transductive Rademacher Complexity and its Applications. In *Learning Theory*. Springer.

Geng, B.; Yang, L.; and Hua, X.-S. 2009. Learning to Rank with Graph Consistency.

Gleich, D. F., and Lim, L.-h. 2011. Rank Aggregation via Nuclear Norm Minimization. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

He, X.; Gao, M.; Kan, M.-Y.; and Wang, D. 2017. BiRank: Towards Ranking on Bipartite Graphs. *IEEE Transactions on Knowledge and Data Engineering* 29(1):57–71.

Hsu, C.-C.; Lai, Y.-A.; Chen, W.-H.; Feng, M.-H.; and Lin, S.-D. 2017. Unsupervised Ranking using Graph Structures and Node Attributes. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, 771–779. ACM.

Jamieson, K. G., and Nowak, R. 2011. Active Ranking using Pairwise Comparisons. In *Advances in Neural Information Processing Systems*, 2240–2248.

Jethava, V.; Martinsson, A.; Bhattacharyya, C.; and Dubhashi, D. P. 2013. Lovász ϑ Function, SVMs and Finding Dense Subgraphs. *Journal of Machine Learning Research* 14(1):3495–3536.

Kleinberg, J. M. 1999. Authoritative Sources in a Hyperlinked Environment. *Journal of the ACM (JACM)* 46(5):604–632.

Kumar, R., and Vassilvitskii, S. 2010. Generalized Distances between Rankings. In *Proceedings of the 19th international conference on World wide web*, 571–580. ACM.

Lovász, L. 1979. On the Shannon Capacity of a Graph. *Information Theory, IEEE Transactions on* 25(1):1–7.

Luz, C. J., and Schrijver, A. 2005. A Convex Quadratic Characterization of the Lovász Theta Number. *SIAM Journal on Discrete Mathematics* 19(2):382–387.

Monjardet, B. 1998. On the Comparison of the Spearman and Kendall Metrics between Linear Orders. *Discrete mathematics*.

Negahban, S.; Oh, S.; and Shah, D. 2012. Iterative Ranking from Pair-wise Comparisons. In *Advances in Neural Information Processing Systems*, 2474–2482.

Niranjan, U., and Rajkumar, A. 2017. Inductive Pairwise Ranking: Going Beyond the $nlog(n)$ Barrier. In *AAAI*, 2436–2442.

Page, L.; Brin, S.; Motwani, R.; and Winograd, T. 1998. The PageRank Citation Ranking: Bringing Order to the Web. In *Proceedings of the 7th International World Wide Web Conference*, 161–172.

Pelckmans, K.; Suykens, J. A.; and Moor, B. 2007. Transductive Rademacher Complexities for Learning over a Graph. In *MLG*.

Rajkumar, A., and Agarwal, S. 2016. When Can We Rank Well from Comparisons of $O(n \log n)$ Non-Actively Chosen Pairs? In *Conference on Learning Theory*, 1376–1401.

Shivanna, R., and Bhattacharyya, C. 2014. Learning on Graphs Using Orthonormal Representation is Statistically Consistent. In *Advances in Neural Information Processing Systems*, 3635–3643.

Theodoridis, A.; Kotropoulos, C.; and Panagakis, Y. 2013. Music Recommendation Using Hypergraphs and Group Sparsity. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, 56–60. IEEE.

Wauthier, F.; Jordan, M.; and Jojic, N. 2013. Efficient Ranking from Pairwise Comparisons. In *International Conference on Machine Learning*, 109–117.