Blind, Greedy, and Random: Algorithms for Matching and Clustering using only Ordinal Information

Elliot Anshelevich
Department of Computer Science
Rensselaer Polytechnic Institute
eanshel@cs.rpi.edu

Shreyas Sekar
Department of Computer Science
Rensselaer Polytechnic Institute
sekars@rpi.edu

Abstract

We study the Maximum Weighted Matching problem in a partial information setting where the agents' utilities for being matched to other agents are hidden and the mechanism only has access to ordinal preference information. Our model is motivated by the fact that in many settings, agents cannot express the numerical values of their utility for different outcomes, but are still able to rank the outcomes in their order of preference. Specifically, we study problems where the ground truth exists in the form of a weighted graph, and look to design algorithms that approximate the true optimum matching using only the preference orderings for each agent (induced by the hidden weights) as input. If no restrictions are placed on the weights, then one cannot hope to do better than the simple greedy algorithm, which yields a half optimal matching. Perhaps surprisingly, we show that by imposing a little structure on the weights, we can improve upon the trivial algorithm significantly: we design a 1.6-approximation algorithm for instances where the hidden weights obey the metric inequality. Our algorithm is obtained using a simple but powerful framework that allows us to combine greedy and random techniques in unconventional ways. These results are the first non-trivial ordinal approximation algorithms for such problems, and indicate that we can design robust matchings even when we are agnostic to the precise agent utilities.

1 Introduction

Consider the *Maximum Weighted Matching* (MWM) problem, where the input is an undirected complete graph $G = (\mathcal{N}, E)$ and the weight of an edge $w(i,j)$ represents the utility of matching agent i with agent j. The objective is to form a matching (collection of disjoint edges) that maximizes the total utility of the agents. The problem of matching agents and/or items is at the heart of a variety of diverse applications and it is no surprise that this problem and its variants have received extensive consideration in the algorithmic literature (Lovász and Plummer 2009). Perhaps, more importantly, maximum weighted matching is one of the few non-trivial combinatorial optimization problems that can be solved optimally in poly-time (Edmonds 1965). In comparison, we study the MWM problem in a partial information setting where the lack of precise knowledge regarding agents' utilities acts as a barrier against computing optimal matchings, efficiently or otherwise. Furthermore, for the majority of this work, we assume that the edge weights obey the triangle inequality, since in many important applications it is natural to expect that the weights have some geometric structure. Such structure occurs, for instance, when the agents are points in a metric space and the weight of an edge is the distance between the two endpoints.

Partial Information - Ordinal Preferences A crucial question in algorithm and mechanism design is: "How much information about the agent utilities does the algorithm designer possess?". The starting point for the rest of our paper is the observation that in many natural settings, it is unreasonable to expect the mechanism to know the exact weights of the edges in G (Boutilier et al. 2015; Chakrabarty and Swamy 2014). For example, when pairing up students for a class project, it may be difficult to precisely quantify the synergy level for every pair of students; ordinal questions such as 'who is better suited to partner with student x: y or z?' may be easier to answer. Such a situation would also arise when the graph represents a social network of agents, as the agents themselves may not be able to express 'exactly how much each friendship is worth', but would likely be able to form an ordering of their friends from best to worst.

Motivated by this, we consider a model where for every agent $i \in \mathcal{N}$, we *only have access to a preference ordering* among the agents in $\mathcal{N} - \{i\}$ so that if $w(i,j) > w(i,k)$, then $i : j > k$, i.e. i prefers j to k. The common approach in Learning Theory while dealing with such ordinal settings is to estimate the 'true ground state' based on some probabilistic assumptions on the underlying utilities (Oh and Shah 2014; Soufiani, Parkes, and Xia 2012). In this paper we take a different approach, and instead focus on the more demanding objective of designing *robust algorithms*, i.e., algorithms that provide good performance guarantees no matter what the underlying weights are.

Despite the large body of literature on computing matchings in settings with preference orderings, there has been much less work on quantifying the quality of these matchings. As is common in much of *social choice theory*, the implicit assumption in this literature is that the underlying utilities cannot be measured or do not even exist, and hence there is no clear way to define the quality of a matching (Abraham et al. 2007; Bhalgat, Chakrabarty, and Khanna 2011;

Gusfield and Irving 1989). In such papers, the focus therefore is on computing matchings that satisfy normative properties such as stability or optimize a measure of efficiency that depends only on the preference orders, e.g., average rank. On the other hand, the literature on approximation algorithms usually follows the *utilitarian* approach (Harsanyi 1976) of assigning a numerical quality to every solution; the presence of input weights is taken for granted. Our work combines the best of both worlds: we do *not* assume the availability of numerical information (only its latent existence), and yet our approximation algorithms must compete with algorithms that know the true input weights.

Model For the rest of this paper, we assume that the input is a set $\mathcal{N}$ of points or agents with $|\mathcal{N}| = N$, and a strict preference ordering P_i for each $i \in \mathcal{N}$ over the agents in $\mathcal{N} - \{i\}$. We assume that the input preference orderings are derived from a set of underlying **hidden** edge weights ($w(x, y)$ for $x, y \in \mathcal{N}$). Unless mentioned otherwise, we assume that the edge weights satisfy the triangle inequality, i.e., for $x, y, z \in \mathcal{N}$, $w(x, y) \le w(x, z) + w(y, z)$. These weights are considered to represent the *ground truth*, which is not known to the algorithm. We say that the preferences P are induced by weights w if $\forall x, y, z \in \mathcal{N}$, if x prefers y to z, then $w(x, y) \ge w(x, z)$. Our framework captures a number of well-motivated settings; we highlight two of them below.

1. **Forming Diverse Teams** Our setting and objectives align with the research on *diversity maximization algorithms*, a topic that has gained significant traction, particularly with respect to forming diverse teams that capture distinct perspectives (Indyk et al. 2014; Marcolino, Jiang, and Tambe 2013). In these problems, each agent corresponds to a point in a metric space: this point represents the agents's beliefs, skills, or opinions. Given this background, our problem essentially reduces to selecting diverse teams (of size two) based on different diversity goals, since points that are far apart ($w(x, y)$ is large) contribute more to the objective. For instance, one can imagine a teacher pairing up her students who possess differing skill sets or opinions for a class project, which is captured by the maximum weighted matching problem. In section 4, we tackle the problem of forming diverse teams of arbitrary sizes by extending our model to encompass clustering.

2. **Friendship Networks** In structural balance theory (Davis 1977), the statement that *a friend of a friend is my friend* is folklore; this phenomenon is also exhibited by many real-life social networks (Goodreau, Kitts, and Morris 2009). More generally, we can say that a graph with continuous weights has this property if $w(x, y) \ge \alpha[w(x, z) + w(y, z)]$ $\forall x, y, z$, for some suitably large $\alpha \le \frac{1}{2}$ (Anshelevich and Sekar 2015). Friendship networks bear a close relationship to our model; in particular every graph that satisfies the friendship property for $\alpha \ge \frac{1}{3}$ must have metric weights, and thus falls within our framework.

In this paper our main goal is to form **ordinal approximation algorithms** for matching problems. Later, in section 4, we discuss the problem of designing ordinal algorithms for the problem of clustering agents into equal-sized partitions. Generally speaking, an algorithm A is said to be ordinal if it only takes preference orderings $(P)_{i \in \mathcal{N}}$ as input (and not the hidden numerical weights w). It is an α-approximation algorithm if for all possible weights w, and the corresponding induced preferences P, we have that $\frac{OPT(w)}{A(P)} \le \alpha$. Here $OPT(w)$ is the total value of the maximum weight solution with respect to w, and $A(P)$ is the value of the solution returned by the algorithm for preferences $(P)_{i \in \mathcal{N}}$. In other words, such algorithms produce solutions which are always a factor α away from optimum, *without actually knowing* what the weights w are. For randomized algorithms, $A(P)$ denotes the expected performance guarantee.

In the rest of the paper, we focus primarily on the **Maximum Weighted Matching** problem where the goal is to compute a matching to maximize the total (unknown) weight of the edges inside. In addition, we also consider the *Max k-matching* (Mk-M) problem, where the goal is to compute a maximum weight matching consisting of at most k edges.

Challenges and Techniques We describe the challenges involved in designing ordinal algorithms for general problems through the lens of the Maximum Weighted Matching problem. First, different sets of edge weights may give rise to the same preference ordering and moreover, for each of these weights, the optimum matching can be different. Therefore, unlike for the full information setting, no algorithm (deterministic or randomized) can compute the optimum matching using only ordinal information. More generally, the restriction that only ordinal information is available precludes almost all of the well-known algorithms for computing a matching. So, what kind of algorithms use only preference orderings? One algorithm which can still be implemented is a version of the extremely popular greedy matching algorithm, in which we successively select pairs of agents who choose each other as their top choice. Another trivial algorithm is to choose a matching at random: this certainly does not require any numerical information! It is not difficult to show that both these algorithms actually provide an ordinal 2-approximation for the maximum weight matching. The main result of this paper, however, is that by interleaving these basic greedy and random techniques in non-trivial ways, it is actually possible to do much better, and obtain a 1.6-approximation algorithm.

Our Contributions

Our main results are summarized in Table 1. As seen in the table, our ordinal approximation factors are close to the best possible given the information-theoretic constraints imposed by our setting. Moreover, although we cannot solve the problem optimally, our approximation factors are still quite close to 1 indicating that it is possible to find good solutions to matching problems even without knowing any of the true weights, using only preference information instead.

Our central result in this paper is an ordinal 1.6-approximation algorithm for max-weight matching; this is obtained by a careful interleaving of greedy and random matchings. We also present a deterministic 2-approximation algorithm for Max k-Matching. Note that Max k-Matching for $k = \frac{N}{2}$ is the same as the MWM problem.

	Problem	Lower Bound		Our Results	
		Det.	Rand.	Det.	Rand.
w/o Metric	MWM	2	$\frac{5}{3}$	2	2
	Mk-M	∞	∞	-	-
Metric	MWM	1.5	1.25	2	1.6
	Mk-M	2	2	2	2

Table 1: The approximation factors obtained by our deterministic (Det.) and randomized (Rand.) ordinal approximation algorithms along with the information-theoretic lower bounds for the Maximum Weighted Matching (MWM) and the Max k-Matching (Mk-M) problems. The lower bounds are intrinsic to the ordinal information setting that we consider and indicate that no ordinal algorithm can obtain a better approximation factor.

Although our main results are for instances where the hidden weights obey the metric inequality, we also briefly consider ordinal approximation algorithms for general weights. We show that the simple deterministic algorithm that greedily picks edges yields a 2-approximation for the MWM problem even for general weights and is close to the best possible algorithm, deterministic or randomized. Finally, we also consider a strict generalization of the Maximum Weighted Matching problem called *Max k-sum*, where the goal is to partition the agents into k equal sized clusters to maximize the total weight inside the clusters. We present a general black-box approach that uses matching algorithms for forming clusterings, and use this result to obtain an ordinal 3.2-approximation algorithm for Max k-sum.

Techniques: More generally, one of our main contributions is a framework that allows the design of algorithms for problems where the (metric) weights are hidden. Our framework builds on two simple techniques, *greedy* and *random*, and establishes an interesting connection between graph density, matchings, and greedy edges. We believe that this framework may be useful for designing ordinal approximation algorithms in the future.

Related Work

Broadly speaking, the cornucopia of algorithms proposed in the matching literature belong to one of two classes: (i) Ordinal algorithms that ignore agent utilities, and focus on (unquantifiable) axiomatic properties such as *stability*, and (ii) Optimization algorithms where the numerical utilities are fully specified. From our perspective, algorithms belonging to the former class, with the exception of *Greedy*, do not result in good approximations for the hidden optimum, whereas the techniques used in the latter (e.g., Drake and Hougardy; Duan and Pettie) depend heavily on *improving cycles* and thus, are unsuitable for ordinal settings. A notable exception to the above dichotomy is the class of optimization problems studying *ordinal measures of efficiency* (Abraham et al. 2007; Chakrabarty and Swamy 2014), for example, the average rank of an agent's partner in the matching. Such settings often involve the definition of 'new utility functions' based on given preferences, and thus are fundamentally different from our model where preexisting cardinal utilities give rise to ordinal preferences.

The idea of preference orders induced by metric weights (or a more general utility space) was first considered in the work of Irving et al. (1987). Subsequent work has focused mostly on analyzing the greedy algorithm or on settings where the agent utilities are explicitly known (Arkin et al. 2009; Emek, Langner, and Wattenhofer 2015). Most similar to our work is the recent paper by Filos-Ratsikas et al. (2014), who prove that for one-sided matchings, no ordinal algorithm can provide an approximation factor better than $\Theta(\sqrt{N})$. In contrast, for two-sided matchings, there is a simple (greedy) 2-approximation algorithm even when the hidden weights do not obey the metric inequality.

Distortion in Social Choice Our work is partly inspired by the growing body of research in social choice theory studying settings where the voter preferences are induced by a set of hidden utilities (Anshelevich, Bhardwaj, and Postl 2015; Boutilier et al. 2015; Caragiannis and Procaccia 2011; Marcolino et al. 2014; Procaccia and Rosenschein 2006). The voting protocols in these papers are essentially ordinal approximation algorithms, albeit for the very specific problem of selecting the utility-maximizing candidate from a set of alternatives.

Finally, other models of incomplete information have been considered in the Matching literature, most notably Online Algorithms (Kalyanasundaram and Pruhs 1993) and truthful mechanism design (without money) for strategic agents (Dughmi and Ghosh 2010; Procaccia and Tennenholtz 2013). Given the ubiquity of greedy and random algorithms, it would be interesting to see whether such algorithms developed for other partial information models can be extended to our setting.

2 Framework for Ordinal Algorithms

In this section, we present our framework for developing ordinal approximation algorithms and establish tight upper and lower bounds on the performance of algorithms that select matching edges either greedily or uniformly at random. As a simple consequence of this framework, we show that the algorithms that sequentially pick all of the edges greedily or uniformly at random both provide 2-approximations to the maximum weight matching. In the following section, we show how to improve this performance by picking some edges greedily, and some randomly. Finally, we remark that for the sake of convenience and brevity, we will often assume that N is even, and sometimes that it is also divisible by 3. As we discuss in the full version of this paper (Anshelevich and Sekar 2015), our results still hold if this is not the case, with only minor modifications.

Fundamental Subroutine: Greedy

We begin with Algorithm 1 that describes a simple greedy procedure for outputting a matching: at each stage, the algorithm picks one edge (x, y) such that the both x and y prefer this edge to all of the other available edges. We now develop some notation required to analyze this procedure.

Definition (Undominated Edges) Given a set E of edges, $(x, y) \in E$ is said to be an undominated edge if for all (x, a)

and (y, b) in E, $w(x, y) \geq w(x, a)$ and $w(x, y) \geq w(y, b)$.

```
input  : Edge set $E$, preferences $P(\mathcal{N})$, $k$
output: Matching $M_G$ with $k$ edges
while $E$ is not empty (AND) $|M_G| < k$ do
    pick an undominated edge $e = (x, y)$ from $E$ and
      add it to $M_G$;
    remove all edges containing $x$ or $y$ from $E$;
end
```

Algorithm 1: Greedy k-Matching Algorithm

Given a set E, let us use the notation $E_\top$ to denote the set of undominated edges in E. Finally, we say that an edge set E is complete if $\exists$ some $S \subseteq \mathcal{N}$ such that E is the complete graph on the nodes in S (minus the self-loops). We make the following two observations regarding undominated edges

1. Every edge set E has at least one undominated edge. In particular, any maximum weight edge in E is obviously an undominated edge.
2. Given an edge set E, one can efficiently find at least one edge in $E_\top$ *using only the ordinal preference information*. This is done by either finding two agents x and y which are each others' first choices, or if such a pair does not exist, one can find a cycle in the "most-preferred" relationship (i.e., x_1's first choice is x_2, x_2's first choice is x_3, ..., x_i's first choice is x_1), in which case all the edges in this cycle are undominated.

In general, an edge set E may have multiple undominated edges that are not part of a cycle. Our first lemma shows that these different edges are comparable in weight.

Lemma 2.1. *Given a complete edge set E, the weight of any undominated edge is at least half as much as the weight of any other edge in E, i.e., if $e = (x, y) \in E_\top$, then for any $(a, b) \in E$, we have $w(x, y) \geq \frac{1}{2}w(a, b)$. This is true even if (a, b) is another undominated edge.*

Proof. Since (x, y) is an undominated edge, and since E is a complete edge set this means that $w(x, y) \geq w(x, a)$, and $w(x, y) \geq w(x, b)$. Now, from the triangle inequality, we get $w(a, b) \leq w(a, x) + w(b, x) \leq 2w(x, y)$. □

It is not difficult to see that when $k = \frac{N}{2}$, the output of Algorithm 1 coincides with that of the extremely popular greedy algorithm that picks the maximum weight edge in each iteration, and therefore, our algorithm yields an ordinal 2-approximation for the MWM problem. Our next result shows that the approximation factor holds even for Max k-Matching, for any k: this is not a trivial result because at any given stage there may be multiple undominated edges and therefore for $k < \frac{N}{2}$, the output of Algorithm 1 no longer coincides with that of the well known greedy algorithm. In fact, we show the following much stronger lemma,

Lemma 2.2. *Given $k = \alpha\frac{N}{2}$, and $k^* = \alpha^*\frac{N}{2}$, the performance of the greedy k-matching with respect to the optimal k^*-matching (i.e., $\frac{OPT(k^*)}{Greedy(k)}$) is given by,*

1. $\max{(2, 2\dfrac{\alpha^*}{\alpha})}$ *if* $\alpha^* + \alpha < 1$

2. $\max{(2, \dfrac{\alpha^*+1}{\alpha} - 1)}$ *if* $\alpha^* + \alpha \geq 1$

Thus, for example, when $\alpha^* = 1$, and $\alpha = \frac{2}{3}$, we get the factor of 0.5, i.e., in order to obtain a half-approximation to the optimum perfect matching, it suffices to greedily choose two-thirds as many edges as in the perfect matching.

Proof Sketch. The proof is based on an iterative charging algorithm that allows us to cover the edges in the optimum matching (M^*) using the edges in M_G. There are two types of edges in M^*: those that intersect with edges in M_G (Type I), and those that are disjoint (Type II). For Type I edges, the standard greedy argument proceeds by showing that for every $(x, y) \in M^*$, $\exists(x, z) \in M_G$ such that $w(x, z) \geq w(x, y)$. Our algorithm is much more efficient; we show how to cover the type I edges using the minimum possible number of edges in M_G. For a Type II edge (x, y), it may so happen that $\forall(a, b) \in M_G$, $w(a, b)$ is only half of $w(x, y)$; this is problematic if there are a large number of type II edges. We tackle this by establishing strict upper bounds showing that the number of Type II edges cannot be too large when $\alpha + \alpha^* \geq 1$. □

Plugging in $k = k^*$ in the above lemma immediately gives us the following corollary.

Corollary 2.3. *Algorithm 1 is a deterministic, ordinal 2-approximation algorithm for the Max k-Matching problem for all k, and therefore a 2-approximation algorithm for the Maximum Weighted Matching problem.*

Fundamental Subroutine: Random

An even simpler matching algorithm is simply to form a matching completely at random; this does not even depend on the input preferences. This is formally described in Algorithm 2. In what follows, we show upper and lower bounds on the performance of Algorithm 1 for different edges sets.

```
input  : Edge set $E$, $k$
output: Matching $M_R$ with $k$ edges
while $E$ is not empty (AND) $|M_R| < k$ do
    pick an edge from $E$ uniformly at random. Add this
      edge $e = (x, y)$ to $M_R$;
    remove all edges containing $x$ or $y$ from $E$;
end
```

Algorithm 2: Random k-Matching Algorithm

Lemma 2.4. *(Lower Bound)*

1. Suppose $G = (T, E)$ is a complete graph on the set of nodes $T \subseteq \mathcal{N}$ with $|T| = n$. Then, the expected weight of the random (perfect) matching returned by Algorithm 2 for the input E is $E[w(M_R)] \geq \frac{1}{n}\sum_{(x,y)\in E} w(x, y)$.

2. Suppose $G = (T_1, T_2, E)$ is a complete bipartite graph on the set of nodes $T_1, T_2 \subseteq \mathcal{N}$ with $|T_1| = |T_2| = n$. Then, the weight of the random (perfect) matching returned by Algorithm 2 for the input E is $E[w(M_R)] = \frac{1}{n}\sum_{(x,y)\in E} w(x, y)$.

Proof. We show both parts of the theorem using simple symmetry arguments. For the complete (non-bipartite) graph, let $\mathcal{M}$ be the set of all perfect matchings in E. Then, we argue that every matching M in $\mathcal{M}$ is equally likely to occur. Therefore, the expected weight of M_R is

$$E[w(M_R)] = \frac{1}{|\mathcal{M}|}\sum_{M\in\mathcal{M}} w(M) = \sum_{e=(x,y)\in E} p_e w(x,y), \tag{1}$$

where p_e is the probability of edge e occurring in the matching. Since the edges are chosen uniformly at random, the probability that a given edge is present in M_R is the same for all edges in E. So $\forall e$, we have the following bound on p_e, which we can substitute in Equation 1 to get the first result.

$$p_e = \frac{|M_R|}{|E|} = \frac{n/2}{n(n-1)/2} = \frac{1}{n-1} \geq \frac{1}{n},$$

For the second case, where E is the set of edges in a complete bipartite graph, it is not hard to see that once again every edge e is present in the final matching with equal probability. Therefore, $p_e = \frac{|M_R|}{|E|} = \frac{n}{n^2} = \frac{1}{n}$. □

Lemma 2.5. *(Upper Bound) Let $G = (T, E)$ be a complete graph on the set of nodes T with $|T| = n$. Suppose that $\mathcal{S}$ is some superset of T and let M be any perfect matching on $\mathcal{S}$. Then, the following is an upper bound on the weight of M,*

$$w(M) \leq \frac{2}{n}\sum_{\substack{x\in T\\ y\in T}} w(x,y) + \frac{1}{n}\sum_{\substack{x\in T\\ y\in \mathcal{S}\setminus T}} w(x,y)$$

Proof. Fix an edge $e = (x, y) \in M$. Then, by the triangle inequality, the following must hold for every node $z \in T$: $w(x,z) + w(y,z) \geq w(x,y)$. Summing this up over all $z \in T$, we get

$$\sum_{z\in T} w(x,z) + w(y,z) \geq nw(x,y) = n(w_e).$$

Once again, repeating the above process over all $e \in M$, and then all $z \in T$ we have

$$nw(M) \leq 2\sum_{\substack{x\in T\\ y\in T}} w(x,y) + \sum_{\substack{x\in T\\ y\in \mathcal{S}\setminus T}} w(x,y)$$

Each $(x, y) \in E$ appears twice in the RHS: once when we consider the edge in M containing x, and once when we consider the edge with y. □

We conclude by proving that picking edges uniformly at random yields a 2-approximation for the MWM problem.

Claim 2.6. *Algorithm 2 is an ordinal 2-approximation algorithm for the Maximum Weighted Matching problem.*

Proof. From Lemma 2.4, we know that in expectation, the matching output by the algorithm when the input is $\mathcal{N}$ has a weight of at least $\frac{1}{N}\sum_{x\in\mathcal{N}, y\in\mathcal{N}} w(x,y)$. Substituting $T = \mathcal{S} = \mathcal{N}$ in Lemma 2.5 and $M = OPT$ (max-weight matching) gives us the following upper bound on the weight of OPT, $w(OPT) \leq \frac{2}{N}\sum_{x\in\mathcal{N}, y\in\mathcal{N}} w(x,y) \leq 2E[w(M_R)]$. □

3 Ordinal Matching Algorithms

Here we present a better ordinal approximation than simply taking the random or greedy matching. The algorithm first performs the greedy subroutine until it matches $\frac{2}{3}$ of the agents. Then it either creates a random matching on the unmatched agents, *or* it creates a random matching between the unmatched agents and a subset of agents which are already matched. We show that one of these matchings is guaranteed to be close to optimum in weight. Unfortunately since we have no access to the weights themselves, we cannot simply choose the best of these two matchings, and thus are forced to randomly select one, giving us good performance in expectation. More formally, the algorithm is:

```
input  : $\mathcal{N}$, $P(\mathcal{N})$
output: Perfect Matching $M$
$M_0 :=$ Output of (Greedy) Algorithm 1 for $k = \frac{2}{3}\frac{N}{2}$;
$B := \mathcal{N} \setminus \{$Nodes matched in $M_0\}$;
First Algorithm;
$M_1 = M_0 \cup$ (Output of Algorithm 2 on $B, k = \frac{N}{6}$);
Second Algorithm ;
$M_2 := \frac{N}{6}$ edges from $M_0$ chosen uniformly at random;
Let $A$ be the set of nodes in $M_0 \setminus M_2$;
$E_{ab} :=$ edges of complete bipartite graph $(A, B)$;
$M_2 = M_2 \cup \{$Output of Alg 2 with input $E_{ab}$, $k = \frac{N}{3}\}$;
Final Output;
Return $M_1$ with probability $\frac{1}{2}$, $M_2$ with probability $\frac{1}{2}$.
```

Algorithm 3: 1.6-Approximation Algorithm for Maximum Weight Matching

Theorem 3.1. *For every input ranking, Algorithm 3 returns a $\frac{8}{5} = 1.6$-approximation to the maximum-weight matching.*

Proof. First, we provide some high-level intuition on why this algorithm results in a significant improvement over the standard approaches. Observe that in order to obtain a half-approximation to OPT, it is sufficient to greedily select $\frac{2}{3}(N/2)$ edges. Now, let us denote by Top, the set of $\frac{2}{3}N$ nodes that are matched greedily. The main idea behind the second Algorithm is that if the first one performs poorly (not that much better than half), then, all the 'good edges' must be going from Top to Bottom (B). In other words, $\sum_{(x,y)\in Top\times B} w(x,y)$ must be large, and therefore, the randomized algorithm for bipartite graphs should perform well.

We now prove the theorem formally. By linearity of expectation, $E[w(M)] = 0.5(E[w(M_1)] + E[w(M_2)])$. Now, look at the first algorithm: from Lemma 2.2 ($\alpha = \frac{2}{3}, \alpha^* = 1$) we know that $w(M_0) \geq \frac{1}{2}w(OPT)$, and using Lemma 2.4 ($n = \frac{N}{3}$), the expected weight of the random matching on the remaining nodes is $\frac{3}{N}\sum_{(x,y)\in B} w(x,y)$. Therefore, $E[w(M_1)] \geq \frac{OPT}{2} + \frac{3}{N}\sum_{(x,y)\in B} w(x,y)$.

Next, look at the second algorithm: half the edges from M_0 are added to M_2 which in expectation has a weight of at least $0.5w(M_0) \geq \frac{1}{4}w(OPT)$. Let M_{AB} denote the random matching going 'across the cut' from Top to B.

Since half the nodes of Top are chosen uniformly at random, we can use linearity of expectation and the second part of Lemma 2.4 applied to $(Top, B, Top \times B)$, and get $E[w(M_{AB})] \geq \frac{1}{2}\sum_{(x,y)\in Top\times B}\frac{3}{N}w(x,y)$.

Since $Top \cup B = \mathcal{N}$, we can apply our upper bound result of Lemma 2.5 with $T = B$ and $\mathcal{S} = \mathcal{N}$, and get the following inequality after rearranging:

$$\frac{3}{2N}\sum_{(x,y)\in Top\times B} w(x,y) \geq \frac{w(OPT)}{2} - \frac{3}{N}\sum_{\substack{x\in B\\ y\in B}} w(x,y).$$

In summary, our lower bound on $E[M_2]$ can be expressed as $\frac{1}{2}E[w(M_0)]+E[w(M_{AB})] \geq \frac{1}{4}w(OPT)+\frac{1}{2}w(OPT)-\frac{3}{N}\sum_{x\in B, y\in B} w(x,y)$. The final bound comes from adding this quantity to the bound on $E[M_1]$ and multiplying by half. □

Lower Bound Example for Ordinal Matchings

Complementing our main result, we provide examples that highlight the limitations of settings with ordinal information. As mentioned in the Introduction, different sets of weights can give rise to the same preferences, and therefore, we cannot suitably approximate the optimum solution for every possible weight. We now show that even for very simple instances, there can be no deterministic 1.5-approximation algorithm, and no randomized 1.25-approximation algorithm.

Example (3) Consider an instance with 4 nodes having the following preferences: (i) $a : b > c > d$, (ii) $b : a > d > c$, (iii) $c : a > b > d$, (iv) $d : b > a > c$. Since the matching $\{(a,d),(b,c)\}$ is weakly dominated, it suffices to consider algorithms that consider only $M_1 = \{(a,b),(c,d)\}$, and/or $M_2 = \{(a,c),(b,d)\}$.

Now, consider the following two sets of weights, both of which induce the above preferences but whose optima are M_2 and M_1 respectively: W_1 : all weights are 1 except $w(c,d) = \epsilon$, and $W_2 := w(a,b) = 2$, and all other weights are 1. The best deterministic algorithm always chooses the matching M_2, but for the weights W_2, this is only a $\frac{3}{2}$-approximation to OPT. Consider any randomized algorithm that chooses M_1 with probability x, and M_2 with probability $(1-x)$. With a little algebra, we can verify that just for W_1, and W_2, the optimum randomized algorithm has $x = \frac{2}{5}$, yielding an approximation factor of 1.25. ■

For the Max k-Matching problem, our results are tight. For small values of k, it is impossible for any ordinal algorithm to provide a better than 2-approximation factor. To see why, consider an instance with $2N$ nodes $\{a_1, b_1, a_2, b_2, \ldots, a_N, b_N\}$. Every a_i's first choice is b_i and vice-versa, the other preferences can be arbitrary. Pick some i uniformly at random and set $w(a_i, b_i) = 2$, and all the other weights are equal to 1. For $k = 1$, it is easy to see that no randomized algorithm can always pick the max-weight edge and therefore, as $N \to \infty$, we get a lower bound of 2.

4 Extensions

Ordinal Algorithms without Metric Weights

We very briefly discuss the general case where the hidden weights do not obey the triangle inequality. From our discussion in Section 2, we infer that Algorithm 1 still yields a 2-approximation to the MWM problem as its output coincides with that of the classic greedy algorithm. No deterministic algorithm can provide a better approximation; consider the same preference orderings as Example 3 and the following two sets of weights: (i) $w(c,d) = \epsilon$, other weights are 1, and (ii) $w(a,b) = 1$, other weights are ϵ. The only good choice for case (ii) is the matching M_1, which yields a 2-approximation for case (i). In the full version (Anshelevich and Sekar 2015), we provide a more sophisticated example that shows that no randomized algorithm can have a performance guarantee better than $\frac{5}{3}$. For Max k-matching, the situation is much more bleak; using a similar example as before, we can show that no algorithm, deterministic or randomized can provide a reasonable approximation factor if k is small.

Applications to Clustering

In this section, we highlight the efficacy of our framework by showing that it can be applied to derive good algorithms for the Max k-sum clustering problem, where the objective is the partition the N nodes into k equal sized clusters in order to maximize the weight of the edges inside the clusters. When $k = \frac{N}{2}$, this reduces to Max Weighted Matching. We discuss how to utilize our previous results to develop good ordinal approximation algorithms for this problem, a formal treatment can be found in the full version of the paper (Anshelevich and Sekar 2015).

Consider the following algorithm that takes as input a matching M and outputs k clusters. "Arbitrarily divide M into k equal sized sets (with $\frac{N}{2k}$ edges in each) and form clusters using the nodes in each of the sets". Extending our framework, we can show that if M is an α-approximation to the optimum matching, then the weight of above clustering is at most a factor 2α smaller than the optimum clustering. Using our ordinal algorithms to obtain M, we immediately get a deterministic 4-approximation and a randomized 3.2-approximation algorithm for Max k-sum. Furthermore, our framework for matching can be leveraged to obtain good ordinal approximation algorithms for other distance maximization problems on graphs such as *Densest Subgraph* and *Max Traveling Salesman* (Kosaraju, Park, and Stein 1994).

Conclusion

In this paper we study ordinal algorithms, i.e., algorithms which are aware only of preference orderings instead of the hidden weights or utilities which generate such orderings. Perhaps surprisingly, our results imply that for matching, ordinal approximation algorithms are close to optimal indicating that for settings where it is expensive, or impossible, to obtain the true numerical weights or utilities, one can use ordinal matching mechanisms without much loss in welfare. For the MWM problem, it may also be possible to improve the deterministic approximation factor to be better than 2: although this seems to be a difficult problem which would require novel techniques. Finally, a long-term goal is to develop ordinal approximation algorithms for other optimization problems on graphs and more importantly identify what kinds of problems admit good ordinal algorithms.

References

Abraham, D. J.; Irving, R. W.; Kavitha, T.; and Mehlhorn, K. 2007. Popular matchings. *SIAM J. Comput.* 37(4):1030–1045.

Anshelevich, E., and Sekar, S. 2015. Blind, greedy, and random: Ordinal approximation algorithms for graph problems. *(Full Version)*. Available at `http://www.cs.rpi.edu/~eanshel/pubs.html`.

Anshelevich, E.; Bhardwaj, O.; and Postl, J. 2015. Approximating optimal social choice under metric preferences. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 777–783.

Arkin, E. M.; Bae, S. W.; Efrat, A.; Okamoto, K.; Mitchell, J. S. B.; and Polishchuk, V. 2009. Geometric stable roommates. *Inf. Process. Lett.* 109(4):219–224.

Bhalgat, A.; Chakrabarty, D.; and Khanna, S. 2011. Social welfare in one-sided matching markets without money. In *Proceedings of the Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX 2011)*, 87–98.

Boutilier, C.; Caragiannis, I.; Haber, S.; Lu, T.; Procaccia, A. D.; and Sheffet, O. 2015. Optimal social choice functions: A utilitarian view. *Artif. Intell.* 227:190–213.

Caragiannis, I., and Procaccia, A. D. 2011. Voting almost maximizes social welfare despite limited communication. *Artif. Intell.* 175(9-10):1655–1671.

Chakrabarty, D., and Swamy, C. 2014. Welfare maximization and truthfulness in mechanism design with ordinal preferences. In *Proceedings of the 5th Innovations in Theoretical Computer Science conference (ITCS 2014)*, 105–120.

Davis, J. A. 1977. Clustering and structural balance in graphs. *Social networks. A developing paradigm* 27–34.

Drake, D. E., and Hougardy, S. 2003. Improved linear time approximation algorithms for weighted matchings. In *Proceedings of APPROX 2003*, 14–23.

Duan, R., and Pettie, S. 2010. Approximating maximum weight matching in near-linear time. In *Proceedings of the 51st Annual IEEE Symposium on Foundations of Computer Science, (FOCS 2010)*, 673–682.

Dughmi, S., and Ghosh, A. 2010. Truthful assignment without money. In *Proceedings of the 11th ACM Conference on Electronic Commerce (EC 2010),*, 325–334.

Edmonds, J. 1965. Paths, trees, and flowers. *Canadian Journal of mathematics* 17(3):449–467.

Emek, Y.; Langner, T.; and Wattenhofer, R. 2015. The price of matching with metric preferences. In *Proceedings of the 23rd Annual European Symposium of Algorithms (ESA 2015)*, 459–470.

Filos-Ratsikas, A.; Frederiksen, S. K. S.; and Zhang, J. 2014. Social welfare in one-sided matchings: Random priority and beyond. In *Proceedings of the 7th International Symposium on Algorithmic Game Theory (SAGT 2014)*, 1–12.

Goodreau, S. M.; Kitts, J. A.; and Morris, M. 2009. Birds of a feather, or friend of a friend? using exponential random graph models to investigate adolescent social networks. *Demography* 46(1):103–125.

Gusfield, D., and Irving, R. W. 1989. *The stable marriage problem: structure and algorithms*. MIT press.

Harsanyi, J. C. 1976. *Cardinal welfare, individualistic ethics, and interpersonal comparisons of utility*. Springer.

Indyk, P.; Mahabadi, S.; Mahdian, M.; and Mirrokni, V. S. 2014. Composable core-sets for diversity and coverage maximization. In *Proceedings of the 33rd ACM Symposium on Principles of Database Systems, (PODS'14)*, 100–108.

Irving, R. W.; Leather, P.; and Gusfield, D. 1987. An efficient algorithm for the optimal stable marriage. *Journal of the ACM (JACM)* 34(3):532–543.

Kalyanasundaram, B., and Pruhs, K. 1993. Online weighted matching. *J. Algorithms* 14(3):478–488.

Kosaraju, S. R.; Park, J. K.; and Stein, C. 1994. Long tours and short superstrings (preliminary version). In *35th Annual Symposium on Foundations of Computer Science (FOCS 1994)*, 166–177.

Lovász, L., and Plummer, M. D. 2009. *Matching theory*, volume 367. American Mathematical Soc.

Marcolino, L. S.; Xu, H.; Jiang, A. X.; Tambe, M.; and Bowring, E. 2014. Give a hard problem to a diverse team: Exploring large action spaces. *Proceedings of the Twenty-Eigth AAAI Conference on Artificial Intelligence*.

Marcolino, L. S.; Jiang, A. X.; and Tambe, M. 2013. Multi-agent team formation: Diversity beats strength? In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*.

Oh, S., and Shah, D. 2014. Learning mixed multinomial logit model from ordinal data. In *Proceedings of the 28th Annual Conference on Neural Information Processing Systems (NIPS 2014)*, 595–603.

Procaccia, A. D., and Rosenschein, J. S. 2006. The distortion of cardinal preferences in voting. In *Proceedings of 10th Intl. Workshop on Cooperative Information Agents CIA 2006*, 317–331.

Procaccia, A. D., and Tennenholtz, M. 2013. Approximate mechanism design without money. *ACM Trans. Economics and Comput.* 1(4):18.

Soufiani, H. A.; Parkes, D. C.; and Xia, L. 2012. Random utility theory for social choice. In *Proceedings of the 26th Annual Conference on Neural Information Processing Systems (NIPS 2012)*, 126–134.