The Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)

Balancing Relevance and Diversity in Online Bipartite Matching via Submodularity

John P. Dickerson, Karthik Abinav Sankararaman, Aravind Srinivasan, Pan Xu
{john, kabinav, srin, panxu}@cs.umd.edu
University of Maryland, College Park, MD, USA

"Different roads sometimes lead to the same castle." – George R.R. Martin

Abstract

In bipartite matching problems, vertices on one side of a bipartite graph are paired with those on the other. In its online variant, one side of the graph is available offline, while the vertices on the other side arrive online. When a vertex arrives, an irrevocable and immediate decision should be made by the algorithm; either match it to an available vertex or drop it. Examples of such problems include matching workers to firms, advertisers to keywords, organs to patients, and so on. Much of the literature focuses on maximizing the total relevance—modeled via total weight—of the matching. However, in many real-world problems, it is also important to consider contributions of diversity: hiring a diverse pool of candidates, displaying a relevant but diverse set of ads, and so on. In this paper, we propose the Online Submodular Bipartite Matching (OSBM) problem, where the goal is to maximize a submodular function f over the set of matched edges. This objective is general enough to capture the notion of both diversity (*e.g.,* a weighted coverage function) and relevance (*e.g.,* the traditional linear function)—as well as many other natural objective functions occurring in practice (*e.g.,* limited total budget in advertising settings). We propose novel algorithms that have provable guarantees and are essentially optimal when restricted to various special cases. We also run experiments on real-world and synthetic datasets to validate our algorithms.

Introduction

Online Bipartite Matching (OBM) problems are primarily motivated by Internet advertising. In the basic version of this problem, we are given a bipartite graph $G = (U, V, E)$, where U and V represent the offline vertices (advertisers) and online vertices (keywords or impressions) respectively. An edge $e = (u, v)$ represents a bid by advertiser u for a keyword v. When a keyword v arrives, a central agency must make an *instant and irrevocable* decision to either reject v or assign v to one of its "neighbors" (*i.e.,* a vertex that is connected to v by an edge in G) u and obtain a profit w_e for the match $e = (u, v)$. A matched advertiser u is no longer available for future matches. The goal is to design an efficient online algorithm that maximizes the expected total weight (profit) of the matching. Following the seminal work of Karp, Vazirani, and Vazirani (1990), there has been a large body of research on related variants (Mehta 2012). During the last decade, OBM and its variants have seen wider applications in various matching markets: crowdsourcing Internet marketplaces (*e.g.,* (Assadi, Hsu, and Jabbari 2015; Ho and Vaughan 2012)), online spatial crowdsourcing platforms (*e.g.,* (Tong et al. 2017; 2016b; 2016a)), ride-sharing platforms (*e.g.,* (Dickerson et al. 2018)). In each of these applications we have the following features (1) agents from at least one side appear online (*i.e.,* one-by-one) (2) an online agent on arrival has to either be matched immediately to an offline agent—or be rejected.

Most prior research on online matching focuses on maximizing the total weight of the final matching (Mehta 2012), which captures the quality/relevance of all the matches. In many matching markets, we also care about the *diversity* of the final matching along with relevance. Ahmed, Dickerson, and Fuge (2017) considered a motivating example of matching academic papers to potential reviewers: *just* maximizing the relevance (the quality of each match) could potentially assign a paper to multiple scholars in a single lab due to shared expertise, which is undesirable. Instead, we want to assign each paper to *relevant* experts with diverse backgrounds to obtain comprehensive feedback. Maximizing diversity[1] is of particular importance in various recommendation systems, ranging from recommendations of new books and movies on eBay (Chen et al. 2016) to returning search-engine queries (Agrawal et al. 2009). A common strategy to address diversity is to first formulate a specific objective (typically maximization over a submodular function[2]) capturing the balance of diversity and relevance and then design an efficient algorithm—typically a greedy one—to solve it (*e.g.,* (Ahmed, Dickerson, and Fuge 2017) and references within).

Inspired by the broad applications of OBM and submodular maximization, we propose a variant of the online matching model which we call Online Submodular Bipartite Matching (OSBM). In particular, we answer the main Question 1, defined formally below.

Main model. Suppose we have a bipartite graph $G = (U, V, E)$ where U and V represent the offline and on-

[1]Both individual and aggregate diversity (Adomavicius and Kwon 2012).

[2]See Section for a formal definition and canonical examples.

line agents respectively. We have a finite time horizon T (known beforehand) and for each time (or round) $t \in [T] \doteq \{1, 2, \ldots, T\}$, at most one vertex v is sampled—in which case we say v *arrives*—from a given *known* probability distribution $\{p_v\}$. That is, $\sum_{v \in V} p_v \leq 1$; thus, with probability $1 - \sum_{v \in V} p_v$, none of the vertices from V will arrive at t. The sampling process is independent across different times. Let $r_v \doteq T \cdot p_v$ denote the expected number of arrivals of v in the T online rounds (we interchangeably refer to this as the arrival rate of v). We assume that the value r_v lies in $[0, 1]$.

Once a vertex v arrives, we need to make an *immediate* and *irrevocable* decision: either to reject v or assign v to one of its neighbors in U. Each u has a unit capacity: it will be unavailable in the future upon being matched.[3] We are given a *non-negative monotone submodular* function f over E as an input. Our goal is to design an online matching algorithm such that $\mathbb{E}[f(\mathcal{M})]$ is maximized, where $\mathcal{M}$ is the final (random) matching obtained.

There are two sources contributing to the randomness of $\mathcal{M}$: the stochasticity from the online arrivals of V and the internal randomness used by the algorithm. Note that $\mathcal{M}$ can be a semi-matching, where each u has degree at most 1 while some v may have degree more than 1 (due to multiple online arrivals of v). Following prior work (Mehta 2012), we assume $|V| \gg |U|$ and $T \gg 1$. Throughout this paper, we use edge $e = (u, v)$ and assignment of v to u interchangeably.

Question 1. *Is there a constant-factor competitive ratio*[4] *for online algorithm for the Online Submodular Bipartite Matching problem?*

Related model. One important direction in addressing diversity in online algorithms has been via online convex programming. In particular, Agrawal and Devanur (2014) considered the model of maximizing a concave function under convex constraints. At each time-step a random vector is drawn from an unknown distribution and the goal is to satisfy a convex constraint in expectation. Our work differs from theirs in multiple aspects. First, the offline problem of Agrawal and Devanur (2014) is poly-time solvable, while our problem even in the offline version has unknown hardness (status unknown for both NP- and APX-hardness). Equivalence between discrete and continuous functions exists for submodular minimization via the Lovász extension. However, a similar continuous relaxation for submodular maximization is NP-hard to evaluate[5]. Hence it is unclear how one would use their model to address our problem. Secondly, they consider the large budget regime while all matching type problems differ from allocation problems in that this assumption is not true (in fact, the main challenge is small budgets). The other difference is that our "known i.i.d." gives algorithm design more power as compared to unknown distributions and therefore helps obtain improved ratios rigorously. For example, the online matching problem with linear objectives has been studied both in unknown distribution and known i.i.d. models separately since it presents a natural trade-off—knowing more information about the distribution and the competitive ratio. Based on applications, one would make assumption one-way or the other.

Special cases. Our model generalizes some well-known problems in this literature. Note that if the submodular function is just a linear function of the weights this reduces to online weighted matching. Our model can also capture the Submodular Welfare Maximization (SWM) problem (Kapralov, Post, and Vondrák 2013); this problem and its variants have been widely studied in machine learning and economics (*e.g.,* see (Esfandiari, Korula, and Mirrokni 2016) and references within). Given an instance of SWM we can add polynomially many extra vertices and reduce it to an instance of our problem. All proofs can be found in full version of this paper[6].

Applications. We briefly describe some motivating examples for our problem. The first important application is in *recommender systems*. Consider the problem of recommendation in platforms like Netflix, Amazon, etc. We have a set of users who come online and the system needs to choose a subset of movies, items to recommend to the user which has the most relevance. At the same time, the recommendations to any user needs to be diverse both from an engagement perspective (*e.g.,*, a user doesn't want to see only action movies recommended or just a single brand of items while shopping) and from a fairness perspective (*e.g.,* not showing only stereotypical recommendations based on race, gender, etc.). See Fig. 1 for an example on the MovieLens dataset. This naturally fits our model, where we can capture this trade-off between relevance and diversity via a weighted coverage function (which is monotone and submodular). Another application is in *online advertising and auction design*. Retargeting in personalized advertisements is a major innovation in the past decade where potential advertisers collect background information to provide a better ad experience to their users. One of the major optimization problems for the advertiser is to create various ads to be shown based on the user profile without knowing a-priori the demand for various versions. Hence the advertiser instead specifies a single budget B for a bundle of ads. The goal of the ad-matching agency is to run a matching algorithm with the objective that the revenue they will get for this bundle is $\min\{B, \sum_{e \in \mathcal{M}} w_e\}$, which is a submodular function. Another application is in *matching candidates to jobs in a dynamic job market*. A hiring agency announces openings for various positions and hires the best candidates as and when they come. The most basic version of this problem is called the *secretary problem* (Vanderbei 1980). The goal is usually to hire the best candidates for the open positions. Recently, with increasing awareness of various systemic biases, companies also look to hire diverse candidates (based on various metrics of diversity such as gender, race, and political leanings), which the classical secretary problem and its variants do not consider.[7]

[3] The general case where each u has a given capacity C_u can be reduced to this by creating C_u copies of u.

[4] See Definition 3. Constant refers to value being reasonably bounded away from zero even for large graphs.

[5] *e.g.,* Slide 26 in https://goo.gl/HAhqaZ

[6] http://arxiv.org/abs/1811.05100

[7] These have been explored practically in some recruiting sys-

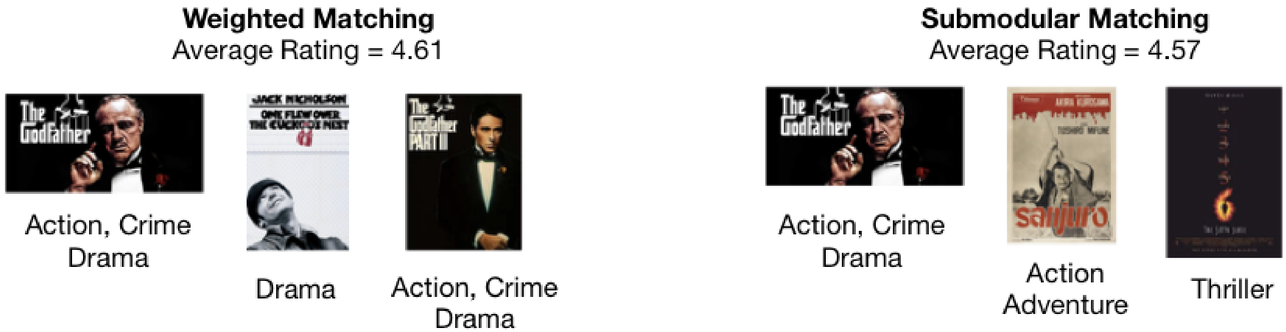

Figure 1: Recommended Movies for User 574. Left - weighted matching (top 3 highest predictions). Right - submodular matching with coverage function (balancing diversity of genres).

However, using our model with a submodular function such as a weighted coverage function over the various metrics as an objective, we can essentially capture hiring the best candidates while maximizing diversity.

Arrival assumption. The literature on Online Matching considers three broad classes of arrival assumptions: Adversarial Order (AO) (*e.g.,* (Assadi, Hsu, and Jabbari 2015)), Random Arrival Order (RAO) (*e.g.,* (Zhao, Li, and Ma 2014; Subramanian et al. 2015)), and Known Independent and Identical Distribution (KIID). In KIID, an online agent's arrival is modeled as a sample (identically with replacement) from a known distribution (*e.g.,* (Singer and Mittal 2013; Singla and Krause 2013)). In this paper, we consider the KIID assumption which captures the fact that the distribution over types can be learnt from historical data; thus can get improved ratios over other models (see (Dickerson et al. 2018) for a discussion).

Our contributions. Our contributions can be summarized as follows. First, we propose the Online Submodular Bipartite Matching (OSBM) model, which abstractly captures the balance between *relevance* and *diversity* in the context of matching markets. Next, we provide two *provably* good algorithms for this model [8]. The first algorithm is based on Contention Resolution schemes used in the offline submodular maximization literature. This algorithm works for the case when the arrival rates are integral. Our second algorithm is based on using a feasible solution to an appropriate mathematical program, where this feasible solution approximates the offline optimal by a factor $1 - 1/e$, to guide the online actions. This algorithm works for the general case of *arbitrary* arrival rates. The ratio achieved by this algorithm is *tight* even when restricted to the special case of linear objectives, however the proof only works when the number of rounds $T \to \infty$. Nonetheless it can be seen as a natural generalization to submodular functions of the LP-based algorithm proposed by Haeupler, Mirrokni, and Zadimoghaddam (2011). Finally, we run experiments on both real-world as well as synthetic datasets on some common submodular functions to validate our algorithms and compare them to natural heuristics.

Related work. The offline version of our problem is the well-studied "maximizing a monotone submodular function subject to a bipartite matching polytope constraint" problem. More generally, the constraint set can be viewed as an intersection of two partition matroids. The general area of submodular maximization is well studied; here, we only survey algorithmic advances related to maximization of a monotone submodular function subject to various constraints. The classical work of Nemhauser, Wolsey, and Fisher (1978) showed that the natural greedy algorithm achieves an $(1 - 1/e)$-approximation under a cardinality constraint, which is optimal in the value oracle model assuming $P \neq NP$ (Nemhauser and Wolsey 1978). Under a general matroid constraint, Calinescu et al. (2011) gave an algorithm achieving the optimal ratio of $1 - 1/e$ (in the value oracle model defined in Section) using the *pipage rounding* technique. Lee, Sviridenko, and Vondrák (2010) considered the constraint case of k matroids with $k \geq 2$ and presented a *local-search* based algorithm. Sarpatwar, Schieber, and Shachnai (2017) studied the case of intersection of k matroids and a single knapsack constraint. Recently a series of works has considered submodular maximization in the online setting. In particular, Buchbinder, Feldman, and Schwartz (2015) and Chan et al. (2017) studied online submodular maximization in the adversarial arrival order with preemption: on arrival of an item, we should decide whether to accept it or not and *possibly rejecting a previously accepted item*. In this paper, we do not allow preemption but consider a more flexible arrival assumption (*i.e.,* KIID). This makes the problem tractable and admits algorithms with non-trivial competitive ratios. Apart from the offline and online models, submodular maximization has received much attention in other models due to its applications in summarization (Tschiatschek et al. 2014), data subset selection and active learning (Wei, Iyer, and Bilmes 2015), and diverse summarization (Mirzasoleiman, Badanidiyuru, and Karbasi 2016), to name a few. It has been studied in the streaming (Badanidiyuru et al. 2014; Mirzasoleiman, Jegelka, and Krause 2018), distributed (Mirza-

tems (Hong et al. 2013).

[8]One of them works under a mild assumption of $|U| = o(\sqrt{T})$

soleiman, Badanidiyuru, and Karbasi 2016; Mirzasoleiman et al. 2016) and stochastic (Karimi et al. 2017; Stan et al. 2017) settings. Online Bipartite matching has been studied with a long line of work, overviewed comprehensively by Mehta (2012). In the KIID arrival model, Feldman et al. (2009) introduced the idea of two suggested matchings and used that to guide the online phase, which was the first to beat $1-1/e$ for the *unweighted* online bipartite matching. A similar idea was used by Haeupler, Mirrokni, and Zadimoghaddam (2011) for *edge-weighted* case. Manshadi, Gharan, and Saberi (2012), Jaillet and Lu (2013), and Brubach et al. (2016) designed ***LP-based*** online algorithms for the unweighted, vertex-weighted and edge-weighted versions of online matching problems to achieve the best known ratios.

Preliminaries

We first describe the notation used throughout this paper. For two vectors $\mathbf{a}$ and $\mathbf{b}$, $\mathbf{a} \leq \mathbf{b}$ denotes the coordinate-wise "$\leq$" operation. For a binary vector $\mathbf{a} = (a_e) \in \{0,1\}^m$, let $Supp(\mathbf{a}) = \{e : a_e = 1\}$ be the support of $\mathbf{a}$; we write $f(\mathbf{a})$ as a short-hand for $f(Supp(\mathbf{a}))$ for any set function over E. In this paper, we use vectors to denote sets (*i.e.,* using the indicator binary vector for a set). e is used both as an edge index as well as Euler's constant; usage will be apparent from the context. We now give a formal definition of the submodular function and describe some canonical examples.

Definition 1 (Submodular function). *A function* $f : 2^{[n]} \rightarrow \mathbb{R}^+$ *on a ground-set of elements* $[n] := \{1, 2, \ldots, n\}$ *is called submodular if for every* $A, B \subseteq [n]$*, we have that* $f(A \cup B) + f(A \cap B) \leq f(A) + f(B)$ *and* $f(\phi) = 0$*. Additionally,* f *is said to be monotone if for every* $A \subseteq B \subseteq [n]$*, we have that* $f(A) \leq f(B)$*.*

For our algorithms, we assume a *value-oracle* access to a submodular function. This means that, there is an oracle which on querying a subset $T \subseteq [n]$, returns the value $f(T)$. The algorithm does not have access to f *explicitly*.

Examples. Some common examples of submodular functions include the coverage function, piece-wise linear functions, budget-additive functions among others. In our experiments section, we use the following two examples.

1. *Coverage function.* Given a universe $\mathcal{U}$ and g subsets $A_1, A_2, \ldots, A_g \subseteq \mathcal{U}$, the function $f(S) = |\cup_{i \in S} A_i|$ is called the coverage function for any $S \subseteq [g]$. This can naturally be extended to the weighted case. Given a non-negative weight function $w : \mathcal{U} \rightarrow \mathbb{R}^+$, then the weighted coverage function is defined as $f(S) = w(\cup_{i \in S} A_i)$.
2. *Budget-additive function.* For a given total budget B and a set of weights $w_i \geq 0$ on the elements $[g]$ of universe $\mathcal{U}$, for any subset $S \subseteq \mathcal{U}$ the budget-additive function is defined as $f(S) = \min\{\sum_{i \in S} w_i, B\}$.

Definition 2 (Multilinear extension). *The multilinear extension of a submodular function* f *is the continuous function* $F : [0,1]^n \rightarrow \mathbb{R}^+$ *defined as* $F(x) := \sum_{T \subseteq [n]} (\prod_{k \in T} x_k \prod_{k \notin T} (1 - x_k)) f(T)$*.*

Note that $F(\mathbf{x}) = f(\mathbf{x})$ for every $\mathbf{x} \in \{0,1\}^n$. The multilinear extension is a useful tool in maximization of submodular objectives. In particular, the above has the following probabilistic interpretation. Let $\mathcal{R}_{\mathbf{x}} \subseteq [n]$ be a random subset of items where each item $i \in [n]$ is added into $\mathcal{R}_{\mathbf{x}}$ independently with probability x_i. We then have $F(\mathbf{x}) = \mathbb{E}[f(\mathcal{R}_{\mathbf{x}})]$.

Offline optimal. Throughout this paper we use the terms offline optimal (or interchangeably offline optimal value) which refers to the following. Given a specific sequence S of (random) arrivals, the "offline problem" is to find the best hindsight matching that maximizes the objective on this sequence, denoted by $\mathrm{OPT}(S)$. Note that $\mathrm{OPT}(S)$ is a random variable since S is random. The offline optimal value, denoted by $\mathbb{E}[\mathrm{OPT}]$, is the expectation of $\mathrm{OPT}(S)$, where the expectation is taken over all possible random sequences S.

Definition 3 (Competitive ratio). *Let* $\mathbb{E}[\mathrm{ALG}(\mathcal{I}, \mathcal{D})]$ *denote the expected value obtained by an algorithm* ALG *on an instance* $\mathcal{I}$ *and arrival distribution* $\mathcal{D}$*. Let* $\mathbb{E}[\mathrm{OPT}(\mathcal{I})]$ *denote the expected* offline optimal*. Then the competitive ratio is defined as* $\min_{\mathcal{I}, \mathcal{D}} \mathbb{E}[\mathrm{ALG}(\mathcal{I}, \mathcal{D})] / \mathbb{E}[\mathrm{OPT}(\mathcal{I})]$.

Challenges and Main Techniques

Our algorithm, like prior work on Online Matching, follows a two-phase approach divided into an Offline phase and an Online phase. **Offline phase.** The first key challenge is to obtain a good handle on the optimal offline solution. For the edge-weighted OBM, the offline version reduces to a maximum weighted matching problem on a bipartite graph, which can be solved efficiently. For OSBM, the offline version which is to maximize a general non-negative monotone submodular function within a bipartite polytope, is non-trivial. Neither polynomial time- nor APX-hardness of this problem is well-understood. We tackle this challenge by first proposing a (offline) Multilinear Maximization Program (MMP) where we maximize the multilinear extension F of the given submodular function f subject to bipartite matching constraints, and then use the continuous greedy algorithm (Calinescu et al. 2011) to solve it. This gives us a marginal distribution $\mathbf{x}^* = \{x_e^*\}$ for each edge being added in the offline optimal satisfying $F(\mathbf{x}^*) \geq (1 - 1/e)\mathbb{E}[\mathrm{OPT}]$.

Online phase. The next challenge is to use the approximate offline marginal distribution $\mathbf{x}^*$ to guide the online phase. We propose two online algorithms which take $\mathbf{x}^*$ as input and make the online decisions by using a modified version of this. The first is inspired by Theorem 4.3 due to Bansal et al. (2012) and its extension. We call this the *CR-based* algorithm which works for the special case of *integral arrival rates*. If f is a linear function, then showing each edge e is added by an online algorithm ALG with probability at least αx_e^* implies that ALG achieves a final ratio of $\alpha(1 - 1/e)$ (the second factor $1 - 1/e$ accounts for the loss in the offline phase). This is because $\mathbb{E}[\mathrm{ALG}] \geq \alpha F(\mathbf{x}^*)$ by *linearity of expectation*. However, this approach fails when F is a multilinear extension of the submodular function f. We overcome this by using Theorem 4.3 of Bansal et al. (2012) (see supplementary materials for the theorem statement), which gives a sufficient condition to ensure that $\mathbb{E}[\mathrm{ALG}] \geq$

$\alpha F(\mathbf{x}^*)$ for any multilinear extension F, provided that each edge e is added in ALG with a marginal distribution at least αx_e^*. This framework is generalized to Contention Resolution (CR) schemes (Vondrák, Chekuri, and Zenklusen 2011), which is used as a tool for the following general problem. Consider a fractional $\mathbf{x} \in \mathcal{P}_{\mathcal{I}}$ where $\mathcal{P}_{\mathcal{I}}$ is a convex relaxation of an integral polytope $\mathcal{I}$ and let $\mathbf{X} = (X_e)$ (not necessarily within $\mathcal{I}$) be a random indicator vector where every X_e is a Bernoulli random variable with mean x_e. Our goal is to round $\mathbf{X}$ to another integral vector $\mathbf{Y}$ such that (1) $\mathbf{Y} \in \mathcal{I}$ and (2) $\mathbb{E}[f(\mathbf{Y})] \geq \alpha \mathbb{E}[f(\mathbf{X})] = \alpha F(\mathbf{x})$ with as large α as possible.

Our second proposed algorithm, which works for arbitrary arrival rates, is an MMP-*based* algorithm (MMP-ALG) described as follows. When a vertex v arrives, sample a neighboring edge $e = (u, v)$ with probability x_e^* and include it iff u is still available. When f is linear, Haeupler, Mirrokni, and Zadimoghaddam (2011) gave a simple and tight analysis[9] showing that MMP-ALG loses a factor of $1 - 1/e$ in the online phase. The tight example is as follows.

Example 1. *Consider an unweighted bipartite graph $G = (U, V, E)$ consisting of a perfect matching with $|U| = |V| = T$ with $x_e^* = 1$ for each e and $f(\mathbf{x}^*) = \sum_e x_e^*$. Notice that each $e = (u, v)$ is added by* MMP-ALG *iff v comes at least once during the T rounds, which occurs with probability equal to $1 - 1/e$. Thus in this example, we have that $\mathbb{E}[\text{MMP-ALG}] = (1 - 1/e) f(\mathbf{x}^*)$.*

In this paper, we give a tight analysis showing that MMP-ALG losses a factor at most $1 - 1/e$ in the online phase even for an arbitrary non-negative monotone submodular function f. The downside is that the bounds hold only in the limit when $T \to \infty$. We believe a careful modification of our proof will lead to a finite time analysis, but do not do so in this paper. In particular when $T \to \infty$, we prove that $\mathbb{E}[\text{MMP-ALG}] \geq (1 - 1/e) F(\mathbf{x}^*)$ and thus this yields a final ratio of $(1 - 1/e)^2$ (after incorporating another factor of $1 - 1/e$ in the offline phase). The main proof idea is through a virtual algorithm ALG which has the same performance as MMP-ALG and by applying pipage rounding (Ageev and Sviridenko 2004) to ALG we show that $\mathbb{E}[\text{ALG}] \geq F((1 - 1/e)\mathbf{x}^*) \geq (1 - 1/e) F(\mathbf{x}^*)$.

Offline Phase

For an edge e, let x_e be the probability that e is chosen in any fixed offline optimal algorithm. For each u (likewise for v), let $E(u)$ ($E(v)$) be the set of neighboring edges incident to u (v) in G. Let $F : [0, 1]^m \to \mathbb{R}_+$ be the multilinear extension of f. Consider the following mathematical program.

$$\begin{aligned} &\text{maximize} \quad F(\mathbf{x}) && && (1) \\ &\text{subject to} \textstyle\sum_{e \in E(v)} x_e \leq r_v && \forall v \in V && (2) \\ &\textstyle\sum_{e \in E(u)} x_e \leq 1 && \forall u \in U && (3) \\ &0 \leq x_e \leq 1 && \forall e \in E && (4) \end{aligned}$$

[9]Here, "tight" refers to the analysis and not the problem formulation itself.

The constraints can be interpreted informally as follows. Constraint (2) states that the expected number of matches for any v is no more than the expected number of arrivals of v (*i.e.,* r_v). Constraint (3) states that the expected number of matches for every u is no more than 1, since u has an unit capacity. Constraint (4) is valid since every x_e is a probability value.

Lemma 1. *There is an efficient algorithm (running in polynomial time) which returns a feasible solution $\mathbf{x}^*$ to the program* (1) *such that $F(\mathbf{x}^*) \geq (1 - 1/e)\mathbb{E}[\text{OPT}]$, where $\mathbb{E}[\text{OPT}]$ is the offline optimal value.*

Online Algorithms

In this section, we present several online algorithms that take $\mathbf{x}^*$ as an input, which is a feasible solution to the program (1) with $F(\mathbf{x}^*) \geq (1 - 1/e)\mathbb{E}[\text{OPT}]$, where $\mathbb{E}[\text{OPT}]$ refers to the offline optimal.

A CR-based online algorithm. In this section, we present a CR-based online algorithm (CR-ALG) for OSBM with integral arrival rates. In this case, we assume that $p_v = 1/T$ and $r_v = 1$ for all v.

The main idea is as follows. We start with $\mathcal{R}_{\mathbf{x}^*}$, which is obtained by independently sampling each edge e with probability x_e^*. Let $\mathbf{X} \in \{0, 1\}^m$ be the integral vector corresponding to $\mathcal{R}_{\mathbf{x}^*}$ such that $X_e = 1$ iff $e \in \mathcal{R}_{\mathbf{x}^*}$. Let $E_{\mathbf{X}}(v) = \{e : e \in E(v), X_e = 1\}$ and $E_{\mathbf{X}}(u) = \{e : e \in E(u), X_e = 1\}$ be the set of sampled edges incident to u and v respectively. Now we obtain another random vector $\mathbf{Y} \in \{0, 1\}^m$ from $\mathbf{X}$ by uniformly sampling an edge from $E_{\mathbf{X}}(u)$ for each u (the sampling process is independent across different u). We then use both $\mathbf{X}$ and $\mathbf{Y}$ to guide the online phase. Algorithm 2 describes this algorithm formally.

Algorithm 1 A CR-based algorithm (CR-ALG)

Offline Phase:

Solve Program (1) using continuous greedy and let $\mathbf{x}^* = (x_e^*)$ be an approximate solution with $F(\mathbf{x}^*) \geq (1 - 1/e)\mathbb{E}[\text{OPT}]$.

Independently sample each edge with probability x_e^*. Let $\mathbf{X} = (X_e) \in \{0, 1\}^m$ be the resultant indicator vector such that $X_e = 1$ iff e is sampled.

For each $w \in U \cup V$, let $E_{\mathbf{X}}(w) = \{e : e \in E(w), X_e = 1\}$ be the set of sampled edges incident to w. Sample one edge uniformly at random from $E_{\mathbf{X}}(u)$ for each u if $E_{\mathbf{X}}(u) \neq \emptyset$. Let $\mathbf{Y} \leq \mathbf{X}$ be the indicator vector of the final edges sampled.

Online Phase:

When v arrives at time t, sample an edge e uniformly from $E_{\mathbf{X}}(v)$. Match it if $Y_e = 1$ and $e = (u, v)$ is safe at t (*i.e.,* u is available); skip it otherwise.

Theorem 1. *There exists an online algorithm* CR-ALG*, which achieves an online competitive ratio of at least $\frac{1}{2}(1 - e^{-1/2})(1 - 1/e)$ for* OSBM *with integral arrival rates.*

An MMP**-based online algorithm.** In this section, we present a MMP-based online algorithm (MMP-ALG) for OSBM. Compared to CR-ALG, it also extends to the regime of arbitrary arrival rates $r_v \in [0, 1]$ for each $v \in V$. Algorithm 2 describes it formally.

Algorithm 2 An MMP-based online algorithm (MMP-ALG)

Offline Phase:
Solve Program (1) using continuous greedy and let $\mathbf{x}^* = (x_e^*)$ be an approximate solution with $F(\mathbf{x}^*) \geq (1 - 1/e)\mathbb{E}[\text{OPT}]$.
Online Phase:
When v arrives at time t, sample an edge e from $E(v)$ with probability $\frac{x_e^*}{r_v}$ (at most one such edge gets sampled). Match it if $e = (u, v)$ is safe at t (*i.e.,* u is available) and skip it otherwise.

Theorem 2. *There exists a* MMP*-based online algorithm* MMP-ALG *which achieves a competitive ratio of at least* $(1 - 1/e)^2$ *for* OSBM *when* $|U| = o(\sqrt{T})$ *and* $T \to \infty$.

Experiments

In this section, we describe the experimental results for the movie recommendation application. Additional experiments using synthetic data on other submodular functions are relegated to the supplemental material. We use the MovieLens dataset (Harper and Konstan 2016) for our purposes[10].

Application. We have a list of movies each associated with some genres and we have a set of users who come into the system at various times (*e.g.,* a user logging into the website for a session). We have information about the ratings of every user for some (different) movies. Our goal is to recommend a small set of movies which the user hasn't rated thus far such that the list is relevant as well as diverse. We quantify the *diversity* by using a weighted coverage function over the set of genres for each user. Hence the goal is to maximize the sum of these weighted coverage functions for all users.[11] This naturally fits within the framework proposed in this paper, where the set of movies form the side U and the set of users form the side V.[12]

Dataset and pre-processing. In this dataset, we have 3952 movies, 6040 users and a total of 100209 ratings of the movies by the users. We choose 200 users who have given the most ratings and sub-sample 100 movies at random; this reduced dataset is used for experiments. Similar to Ahmed, Dickerson, and Fuge (2017), we use a standard collaborative filtering approach (Bradley 2016) to complete the matrix of ratings. To create the graph we do the following. For a given pair of user u and movie m, if u hasn't rated m we add an edge between u and m. For a given user u, we compute the average predicted rating for every genre and use this average as the "weight" in the weighted coverage function. This gives a bias towards genres which the user has highly rated over ones they haven't. For every user we choose a random arrival probability (ensuring that the sum of arrival probabilities equals 1).

Algorithms. We test both our CR-based (Algorithm 1) and the MMP-based (Algorithm 2) experimentally. Additionally we consider the following two heuristics for our study. (1) *Greedy:* At time step t, let S_t be the set of edges chosen so far. When a vertex v arrives, choose an available neighbor u that maximizes $f(S_t \cup \{(u, v)\}) - f(S_t)$. If no neighbor is available, we drop v. (2) *Negative CR-based algorithm (*NEG-CR*):* We tweak CR-ALG by replacing the initial independent sampling with the following procedure. At every u, we use the dependent rounding routine due to Gandhi et al. (2006) to obtain a semi-matching $\mathcal{M}_1$. In the online phase when a v arrives, we sample one of its available neighbors in $\mathcal{M}_1$ uniformly and match it. If not, we drop v.

Results and discussion. We run two kinds of experiments for our purposes. First, we compare our algorithms against the baselines by varying two parameters B and η. B represents the number of times we can match a movie to an user and η represents the number of movies matched to any user on arrival (in the theory $B = 1, \eta = 1$, but we experiment with different values). Second, we want to *measure* diversity of recommendations of the various algorithms. To this end, we compare the various algorithms on the number of users who have various levels of *coverage* (*i.e.,* how many users are shown recommendations greater than x % of the total weight). The plots in Figure 2 and the leftmost plot of Figure 3 show the results for the first kind of experiments, while the right two plots in Figure 3 show the results for the second kind.

In almost all cases, these plots show that MMP-ALG is the clear winner and has the best performance. At times the Greedy algorithm does well, but as B increases the performance drops quickly. Additionally, Greedy makes many calls to the submodular oracle at each online step, as compared to the other algorithms, which can be limiting in if the oracle evaluation is time-consuming. The surprising aspect on these experiments is that the other proposed algorithm CR-ALG does not perform even as well as Greedy. The explanation is that we assign non-integral arrival rates to each user. We show in the supplementary materials that when they are assigned integral rates, CR-ALG's performance is comparable to MMP-ALG and much better than Greedy. As we further show in the supplementary materials, for the budget-additive submodular function, however, even for fractional rates, CR-ALG performs as well as MMP-ALG (and much higher than the theoretical bounds). The diversity histograms show that MMP-ALG is performs well and a good fraction of the users have a coverage greater than 50 % and all the way up to 90% (in the pragmatic case of $b = 15, \eta = 5$). However, the other algorithms have a coverage of at most 20 - 30% for all users (even for the Greedy algorithm when $B = 1, \eta = 1$ where the competitive ratio is higher than

---

[10]Full code can be found at https://bitbucket.org/karthikabinav/submodularmatching/src/master/

[11]The sum of submodular functions is also submodular.

[12]See supplementary materials for details on a Linear Programming formulation of the offline problem.

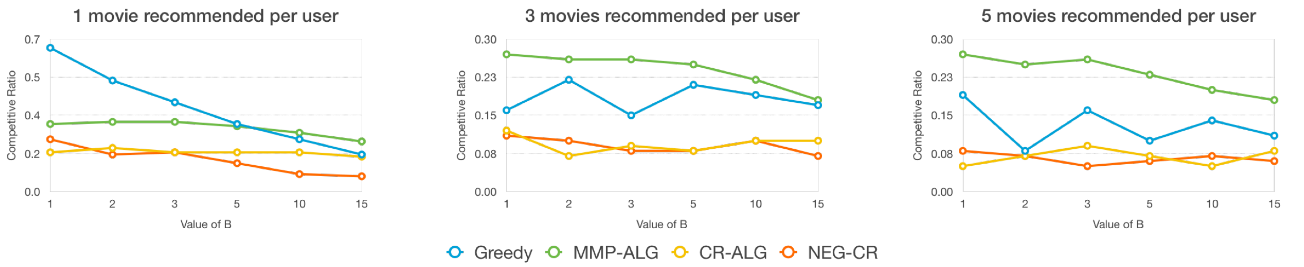

Figure 2: Results when the genre weights are the average of predicted ratings for users. The x-axis varies B and the y-axis represents the ratio. (Left): $\eta = 1$, (Center): $\eta = 3$, (Right): $\eta = 5$.

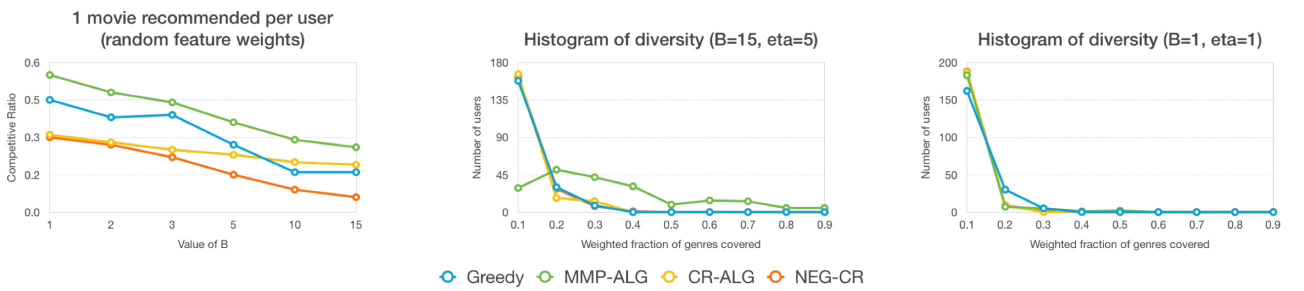

Figure 3: (Left): Same as the left plot in Figure 2 with genre weight chosen $U[0, 1]$. (Center and Right): x-axis percentage of coverage of genres and y-axis number of users who fall in that range.

MMP-ALG).

Conclusion

In this paper, we proposed a new model, Online Submodular bipartite matching (OSBM), which effectively captures notions such as relevance and diversity in matching markets. Many applications such as advertising, hiring diverse candidates, recommending movies or songs naturally fit within this framework. We propose two algorithms, one based on contention-resolution schemes and the other based on using the solution of the mathematical program directly; we give theoretical guarantees on their performance. The algorithm based on using the mathematical program directly is essentially tight even for the special case of linear objectives. Finally, via experiments we show that our algorithms do well in practice. We also proposed heuristics, some of which perform well on specialized submodular functions, and showed that our general algorithm is competitive with such algorithms as well.

Acknowledgements

Aravind Srinivasan's research was supported in part by NSF Awards CNS-1010789, CCF-1422569 and CCF-1749864, and by research awards from Adobe, Inc. The research of Karthik Sankararaman and Pan Xu was supported in part by NSF Awards CNS 1010789 and CCF 1422569.

The authors would like to thank the anonymous reviewers for their helpful feedback.

References

Adomavicius, G., and Kwon, Y. 2012. Improving aggregate recommendation diversity using ranking-based techniques. *IEEE TKDE*.

Ageev, A. A., and Sviridenko, M. I. 2004. Pipage rounding: A new method of constructing algorithms with proven performance guarantee. *Journal of Combinatorial Optimization*.

Agrawal, S., and Devanur, N. R. 2014. Fast algorithms for online stochastic convex programming. In *SODA*.

Agrawal, R.; Gollapudi, S.; Halverson, A.; and Ieong, S. 2009. Diversifying search results. In *WSDM*.

Ahmed, F.; Dickerson, J. P.; and Fuge, M. 2017. Diverse weighted bipartite b-matching. In *IJCAI*.

Assadi, S.; Hsu, J.; and Jabbari, S. 2015. Online assignment of heterogeneous tasks in crowdsourcing markets. In *AAAI-HComp*.

Badanidiyuru, A.; Mirzasoleiman, B.; Karbasi, A.; and Krause, A. 2014. Streaming submodular maximization: Massive data summarization on the fly. In *KDD*.

Bansal, N.; Korula, N.; Nagarajan, V.; and Srinivasan, A. 2012. Solving packing integer programs via randomized rounding with alterations. *Theory of Computing*.

Bradley, A. P. 2016. Movielens collaborative filtering. In *https://github.com/bradleypallen/ keras-movielens-cf, 2016*.

Brubach, B.; Sankararaman, K. A.; Srinivasan, A.; and Xu, P. 2016. New algorithms, better bounds, and a novel model for online stochastic matching. *ESA*.

Buchbinder, N.; Feldman, M.; and Schwartz, R. 2015. Online submodular maximization with preemption. In *SODA*.

Calinescu, G.; Chekuri, C.; Pál, M.; and Vondrák, J. 2011. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*.

Chan, T.; Huang, Z.; Jiang, S. H.-C.; Kang, N.; and Tang, Z. G. 2017. Online submodular maximization with free disposal: Randomization beats 1/4 for partition matroids. In *SODA*.

Chen, C.; Zheng, L.; Srinivasan, V.; Thomo, A.; Wu, K.; and Sukow, A. 2016. Conflict-aware weighted bipartite b-matching and its application to e-commerce. *IEEE TKDE*.

Dickerson, J. P.; Sankararaman, K. A.; Srinivasan, A.; and Xu, P. 2018. Allocation problems in ride-sharing platforms: Online matching with offline reusable resources. *AAAI*.

Esfandiari, H.; Korula, N.; and Mirrokni, V. 2016. Bi-objective online matching and submodular allocations. In *NIPS*.

Feldman, J.; Mehta, A.; Mirrokni, V.; and Muthukrishnan, S. 2009. Online stochastic matching: Beating 1-1/e. In *FOCS*.

Gandhi, R.; Khuller, S.; Parthasarathy, S.; and Srinivasan, A. 2006. Dependent rounding and its applications to approximation algorithms. *Journal of the ACM (JACM)*.

Haeupler, B.; Mirrokni, V. S.; and Zadimoghaddam, M. 2011. Online stochastic weighted matching: Improved approximation algorithms. In *WINE*.

Harper, F. M., and Konstan, J. A. 2016. The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 5(4):19.

Ho, C.-J., and Vaughan, J. W. 2012. Online task assignment in crowdsourcing markets. In *AAAI*.

Hong, W.; Li, L.; Li, T.; and Pan, W. 2013. iHR: an online recruiting system for Xiamen Talent Service Center. In *KDD*.

Jaillet, P., and Lu, X. 2013. Online stochastic matching: New algorithms with better bounds. *Mathematics of Operations Research (MoR)*.

Kapralov, M.; Post, I.; and Vondrák, J. 2013. Online submodular welfare maximization: Greedy is optimal. In *SODA*.

Karimi, M.; Lucic, M.; Hassani, H.; and Krause, A. 2017. Stochastic submodular maximization: The case of coverage functions. In *NIPS*.

Karp, R. M.; Vazirani, U. V.; and Vazirani, V. V. 1990. An optimal algorithm for on-line bipartite matching. In *STOC*.

Lee, J.; Sviridenko, M.; and Vondrák, J. 2010. Submodular maximization over multiple matroids via generalized exchange properties. *Mathematics of Operations Research (MoR)*.

Manshadi, V. H.; Gharan, S. O.; and Saberi, A. 2012. Online stochastic matching: Online actions based on offline statistics. *Mathematics of Operations Research (MoR)*.

Mehta, A. 2012. Online matching and ad allocation. *Theoretical Computer Science*.

Mirzasoleiman, B.; Badanidiyuru, A.; and Karbasi, A. 2016. Fast constrained submodular maximization: Personalized data summarization. In *ICML*.

Mirzasoleiman, B.; Karbasi, A.; Sarkar, R.; and Krause, A. 2016. Distributed submodular maximization. *JMLR*.

Mirzasoleiman, B.; Jegelka, S.; and Krause, A. 2018. Streaming non-monotone submodular maximization: Personalized video summarization on the fly. *AAAI*.

Nemhauser, G. L., and Wolsey, L. A. 1978. Best algorithms for approximating the maximum of a submodular set function. *Mathematics of Operations Research (MOR)*.

Nemhauser, G. L.; Wolsey, L. A.; and Fisher, M. L. 1978. An analysis of approximations for maximizing submodular set functions?i. *Mathematical Programming*.

Sarpatwar, K. K.; Schieber, B.; and Shachnai, H. 2017. Interleaved algorithms for constrained submodular function maximization. *arXiv preprint arXiv:1705.06319*.

Singer, Y., and Mittal, M. 2013. Pricing mechanisms for crowdsourcing markets. In *WWW*.

Singla, A., and Krause, A. 2013. Truthful incentives in crowdsourcing tasks using regret minimization mechanisms. In *WWW*.

Stan, S.; Zadimoghaddam, M.; Krause, A.; and Karbasi, A. 2017. Probabilistic submodular maximization in sub-linear time. In *ICML*.

Subramanian, A.; Kanth, G. S.; Moharir, S.; and Vaze, R. 2015. Online incentive mechanism design for smartphone crowd-sourcing. In *WiOPT*.

Tong, Y.; She, J.; Ding, B.; Chen, L.; Wo, T.; and Xu, K. 2016a. Online minimum matching in real-time spatial data: experiments and analysis. *Proceedings of the VLDB Endowment*.

Tong, Y.; She, J.; Ding, B.; Wang, L.; and Chen, L. 2016b. Online mobile micro-task allocation in spatial crowdsourcing. In *ICDE*.

Tong, Y.; Wang, L.; Zhou, Z.; Ding, B.; Chen, L.; Ye, J.; and Xu, K. 2017. Flexible online task assignment in real-time spatial data. *Proc. VLDB Endow.*

Tschiatschek, S.; Iyer, R. K.; Wei, H.; and Bilmes, J. A. 2014. Learning mixtures of submodular functions for image collection summarization. In *NIPS*.

Vanderbei, R. J. 1980. The optimal choice of a subset of a population. *Mathematics of Operations Research (MoR)*.

Vondrák, J.; Chekuri, C.; and Zenklusen, R. 2011. Submodular function maximization via the multilinear relaxation and contention resolution schemes. In *STOC*.

Wei, K.; Iyer, R.; and Bilmes, J. 2015. Submodularity in data subset selection and active learning. In *ICML*.

Zhao, D.; Li, X.-Y.; and Ma, H. 2014. How to crowdsource tasks truthfully without sacrificing utility: Online incentive mechanisms with budget constraint. In *INFOCOM*.