

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; Tong et al. 2016b; Tong et al. 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 diversity¹ 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 function²) 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-

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

²See 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) $t \in [T] \doteq \{1, 2, \dots, T\}$, 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 ratio⁴ 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 assume small budgets 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. Due to space constraints we defer the proof of this reduction and all other theorems/lemmas in this paper to the supplementary materials.

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.⁶ However, using our model with a submodular function such

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

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

⁵e.g., Slide 26 in <https://goo.gl/HAhqaZ>

⁶These have been explored practically in some recruiting systems (Hong et al. 2013).



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

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 model⁷. 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 $T \rightarrow \infty$. 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 (Tschierschek 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 (Mirzasoleiman, Badanidiyuru, and Karbasi 2016; Mirzasoleiman et al. 2016) and stochastic (Karimi et al. 2017; Stan et al. 2017) settings. Online Bipartite matching has been stud-

⁷One of them works under a mild assumption of $|U| = o(\sqrt{T})$

ied 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 $\text{Supp}(\mathbf{a}) = \{e : a_e = 1\}$ be the support of \mathbf{a} ; we write $f(\mathbf{a})$ as a short-hand for $f(\text{Supp}(\mathbf{a}))$ 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 $[n] := \{1, 2, \dots, n\}$ 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 $f(\emptyset) = 0$. 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 $A_1, A_2, \dots, A_g \subseteq \mathcal{U}$, 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 $F(\mathbf{x}) := \sum_{T \subseteq [n]} (\prod_{k \in T} x_k \prod_{k \notin T} (1 - x_k)) f(T)$.

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 $\text{OPT}(S)$. Note that $\text{OPT}(S)$ is a random variable since S is random. The offline optimal value, denoted by $\mathbb{E}[\text{OPT}]$, is the expectation of $\text{OPT}(S)$, where the expectation is taken over all possible random sequences S .

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

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 $F(\mathbf{x}^*) \geq (1 - 1/e)\mathbb{E}[\text{OPT}]$.

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 $\mathbb{E}[\text{ALG}] \geq \alpha F(\mathbf{x}^*)$ 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 $\mathbb{E}[\text{ALG}] \geq \alpha F(\mathbf{x}^*)$ 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 analysis⁸ 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 loses 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 $T \rightarrow \infty$. 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 $T \rightarrow \infty$, 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 $F : [0, 1]^m \rightarrow \mathbb{R}_+$ be the multilinear extension of f . Consider the following mathematical program.

$$\text{maximize } F(\mathbf{x}) \quad (1)$$

$$\text{subject to } \sum_{e \in E(v)} x_e \leq r_v \quad \forall v \in V \quad (2)$$

$$\sum_{e \in E(u)} x_e \leq 1 \quad \forall u \in U \quad (3)$$

$$0 \leq x_e \leq 1 \quad \forall e \in E \quad (4)$$

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 a 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

⁸Here, “tight” refers to the analysis and not the problem formulation itself.

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 $T \rightarrow \infty$.*

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 purposes⁹.

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.¹⁰ 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 .¹¹

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 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 MMP-ALG).

⁹Full code can be found at <https://bitbucket.org/karthikabinav/submodularmatching/src/master/>

¹⁰The sum of submodular functions is also submodular.

¹¹See supplementary materials for details on a Linear Programming formulation of the offline problem.

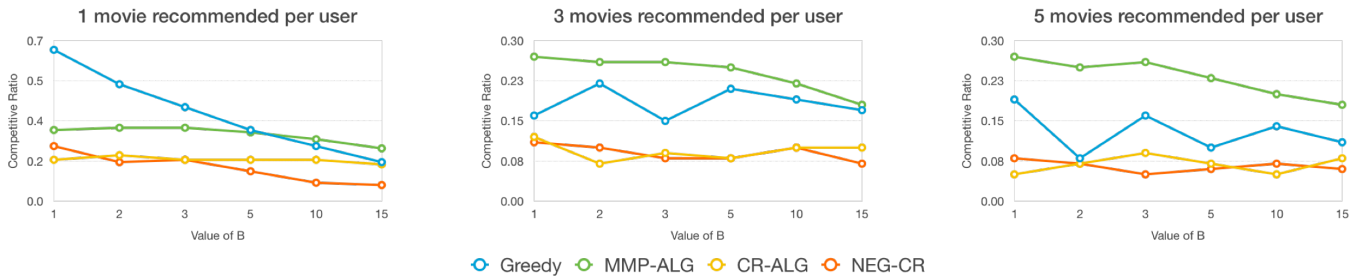


Figure 2: Results when the genre weights are the average of predicted ratings for users. The x -axis varies B and the y -axis represents the ratio. (Left): $\eta = 1$, (Center): $\eta = 3$, (Right): $\eta = 5$.

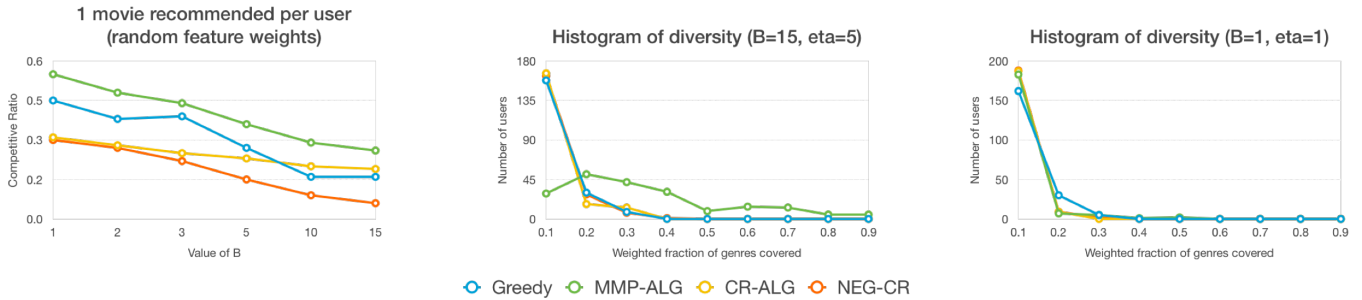


Figure 3: (Left): Same as the left plot in Figure 2 with genre weight chosen $U[0, 1]$. (Center and Right): x -axis percentage of coverage of genres and y -axis number of users who fall in that range.

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

- [2016] Adamczyk, M.; Sviridenko, M.; and Ward, J. 2016. Submodular stochastic probing on matroids. *Mathematics of Operations Research (MoR)*.
- [2012] Adomavicius, G., and Kwon, Y. 2012. Improving aggregate recommendation diversity using ranking-based techniques. *IEEE TKDE*.
- [2004] Ageev, A. A., and Sviridenko, M. I. 2004. Pipage rounding: A new method of constructing algorithms with proven performance guarantee. *Journal of Combinatorial Optimization*.
- [2014] Agrawal, S., and Devanur, N. R. 2014. Fast algorithms for online stochastic convex programming. In *SODA*.
- [2009] Agrawal, R.; Gollapudi, S.; Halverson, A.; and Ieong, S. 2009. Diversifying search results. In *WSDM*.
- [2017] Ahmed, F.; Dickerson, J. P.; and Fuge, M. 2017. Diverse weighted bipartite b-matching. In *IJCAI*.
- [2015] Assadi, S.; Hsu, J.; and Jabbari, S. 2015. Online assignment of heterogeneous tasks in crowdsourcing markets. In *AAAI-HComp*.
- [2014] Badanidiyuru, A.; Mirzasoleiman, B.; Karbasi, A.; and Krause, A. 2014. Streaming submodular maximization: Massive data summarization on the fly. In *KDD*.
- [2012] Bansal, N.; Korula, N.; Nagarajan, V.; and Srinivasan, A. 2012. Solving packing integer programs via randomized rounding with alterations. *Theory of Computing*.
- [2016] Bradley, A. P. 2016. Movielens collaborative filtering. In <https://github.com/bradleypallen/keras-movielens-cf>, 2016.
- [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*.
- [2015] Buchbinder, N.; Feldman, M.; and Schwartz, R. 2015.

- Online submodular maximization with preemption. In *SODA*.
- [2011] Calinescu, G.; Chekuri, C.; Pál, M.; and Vondrák, J. 2011. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*.
- [2017] 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*.
- [2016] 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*.
- [2018] 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*.
- [2016] Esfandiari, H.; Korula, N.; and Mirrokni, V. 2016. Bi-objective online matching and submodular allocations. In *NIPS*.
- [2009] Feldman, J.; Mehta, A.; Mirrokni, V.; and Muthukrishnan, S. 2009. Online stochastic matching: Beating 1-1/e. In *FOCS*.
- [2006] Gandhi, R.; Khuller, S.; Parthasarathy, S.; and Srinivasan, A. 2006. Dependent rounding and its applications to approximation algorithms. *Journal of the ACM (JACM)*.
- [2011] Haeupler, B.; Mirrokni, V. S.; and Zadimoghaddam, M. 2011. Online stochastic weighted matching: Improved approximation algorithms. In *WINE*.
- [2016] Harper, F. M., and Konstan, J. A. 2016. The movie-lens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems (TiiS)* 5(4):19.
- [2012] Ho, C.-J., and Vaughan, J. W. 2012. Online task assignment in crowdsourcing markets. In *AAAI*.
- [2013] Hong, W.; Li, L.; Li, T.; and Pan, W. 2013. iHR: an online recruiting system for Xiamen Talent Service Center. In *KDD*.
- [2013] Jaillet, P., and Lu, X. 2013. Online stochastic matching: New algorithms with better bounds. *Mathematics of Operations Research (MoR)*.
- [2013] Kapralov, M.; Post, I.; and Vondrák, J. 2013. Online submodular welfare maximization: Greedy is optimal. In *SODA*.
- [2017] Karimi, M.; Lucic, M.; Hassani, H.; and Krause, A. 2017. Stochastic submodular maximization: The case of coverage functions. In *NIPS*.
- [1990] Karp, R. M.; Vazirani, U. V.; and Vazirani, V. V. 1990. An optimal algorithm for on-line bipartite matching. In *STOC*.
- [2010] Lee, J.; Sviridenko, M.; and Vondrák, J. 2010. Submodular maximization over multiple matroids via generalized exchange properties. *Mathematics of Operations Research (MoR)*.
- [2012] Manshadi, V. H.; Gharan, S. O.; and Saberi, A. 2012. Online stochastic matching: Online actions based on offline statistics. *Mathematics of Operations Research (MoR)*.
- [2012] Mehta, A. 2012. Online matching and ad allocation. *Theoretical Computer Science*.
- [2016] Mirzasoleiman, B.; Badanidiyuru, A.; and Karbasi, A. 2016. Fast constrained submodular maximization: Personalized data summarization. In *ICML*.
- [2016] Mirzasoleiman, B.; Karbasi, A.; Sarkar, R.; and Krause, A. 2016. Distributed submodular maximization. *JMLR*.
- [2018] Mirzasoleiman, B.; Jegelka, S.; and Krause, A. 2018. Streaming non-monotone submodular maximization: Personalized video summarization on the fly. *AAAI*.
- [1978] Nemhauser, G. L., and Wolsey, L. A. 1978. Best algorithms for approximating the maximum of a submodular set function. *Mathematics of Operations Research (MOR)*.
- [1978] Nemhauser, G. L.; Wolsey, L. A.; and Fisher, M. L. 1978. An analysis of approximations for maximizing submodular set functions?i. *Mathematical Programming*.
- [2017] Sarpatwar, K. K.; Schieber, B.; and Shachnai, H. 2017. Interleaved algorithms for constrained submodular function maximization. *arXiv preprint arXiv:1705.06319*.
- [2013] Singer, Y., and Mittal, M. 2013. Pricing mechanisms for crowdsourcing markets. In *WWW*.
- [2013] Singla, A., and Krause, A. 2013. Truthful incentives in crowdsourcing tasks using regret minimization mechanisms. In *WWW*.
- [2017] Stan, S.; Zadimoghaddam, M.; Krause, A.; and Karbasi, A. 2017. Probabilistic submodular maximization in sub-linear time. In *ICML*.
- [2015] Subramanian, A.; Kanth, G. S.; Moharir, S.; and Vaze, R. 2015. Online incentive mechanism design for smartphone crowd-sourcing. In *WiOPT*.
- [2016a] 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*.
- [2016b] Tong, Y.; She, J.; Ding, B.; Wang, L.; and Chen, L. 2016b. Online mobile micro-task allocation in spatial crowdsourcing. In *ICDE*.
- [2017] 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.*
- [2014] Tschitschek, S.; Iyer, R. K.; Wei, H.; and Bilmes, J. A. 2014. Learning mixtures of submodular functions for image collection summarization. In *NIPS*.
- [1980] Vanderbei, R. J. 1980. The optimal choice of a subset of a population. *Mathematics of Operations Research (MoR)*.
- [2011] Vondrák, J.; Chekuri, C.; and Zenklusen, R. 2011. Submodular function maximization via the multilinear relaxation and contention resolution schemes. In *STOC*.
- [2015] Wei, K.; Iyer, R.; and Bilmes, J. 2015. Submodularity in data subset selection and active learning. In *ICML*.
- [2014] Zhao, D.; Li, X.-Y.; and Ma, H. 2014. How to crowd-source tasks truthfully without sacrificing utility: Online incentive mechanisms with budget constraint. In *INFOCOM*.

Supplementary Materials

Details on prior work on submodularity

As discussed in the preliminaries of the main section, we incorporate ideas from prior work on submodularity. In particular we use the following theorem from (Bansal et al. 2012) and its extension called contention-resolution schemes ((Vondrák, Chekuri, and Zenklusen 2011)).

Theorem 3. ((Bansal et al. 2012)) *Let f be a non-negative monotone submodular function over E with $|E| = m$. Given a fractional vector $\mathbf{x} \in [0, 1]^m$, let $\mathbf{X} = (X_e)_{e \in E}$ be a random binary vector such that each X_e is a Bernoulli random variable with mean x_e . Suppose $\mathbf{Z} \leq \mathbf{X}$ is another random binary vector satisfying the following properties (1) & (2) for some α . Then we have that $\mathbb{E}[f(\mathbf{Z})] \geq \alpha \mathbb{E}[f(\mathbf{X})]$.*

1. **Marginal Property:** $\Pr[Z_e = 1 | X_e = 1] \geq \alpha$.

2. **Monotonicity Property:** $\Pr[Z_e = 1 | \mathbf{X} = \mathbf{a}] \geq \Pr[Z_e = 1 | \mathbf{X} = \mathbf{b}]$, $\forall \mathbf{a} \leq \mathbf{b} \in \{0, 1\}^m$, $\forall e \in \text{Supp}(\mathbf{a}) = \{e' : a_{e'} = 1\}$.

By definition of the multilinear extension we have that $\mathbb{E}[f(\mathbf{X})] = F(\mathbf{x})$. Thus the theorem can be viewed as conditions when “linearity of expectation” can be applied to F .

Submodular Welfare Maximization (SWM) as a special case of our problem

We will now show how we can reduce the SWM problem to our model. Recall that the SWM problem is defined as follows. We have n vertices in U which are available offline. With each vertex $u \in U$ we have a monotone submodular function $g_u : 2^V \rightarrow \mathbb{R}$ associated with it. When a vertex v arrives we need to match v to one of its neighbors in U . At the end of the online phase, let S_1, S_2, \dots, S_n be the set of vertices assigned to vertices $1, 2, \dots, n$ respectively. The goal is to maximize the sum $\sum_{i=1}^n w_i(S_i)$.

Note that sum of submodular functions is a submodular function. Hence to make the objective function of SWM fit our framework, we do the following. Let E_u denote the set of edges incident to u . Define a function $\tilde{g}_u : E_u \rightarrow \mathbb{R}$. For any subset $S \subset E_u$, we let $\tilde{g}(S) := g(S_v)$ where S_v is the set of endpoints of S in V . The non-trivial part to handle is that in SWM any vertex in U can be matched multiple times, while in our setting we allow any U to be matched exactly once. We can overcome this by creating additional vertices as follows. For any $u \in U$, create $|\delta(u)| * T$ copies. A copy, indexed by $v \in \delta(u)$ and $t \in [T]$ has an edge only to v . Additionally, wlog we can enforce that at time t , an algorithm can be matched to vertices whose copy is indexed by t (this doesn’t change the optimal value). Therefore, we now have an instance with a submodular objective and matching constraints.

Consider an arrival sequence τ . Let the optimal allocation in SWM under this arrival be $S_{\tau,1}, S_{\tau,2}, \dots, S_{\tau,n}$ at times T_1, T_2, \dots, T_n . Then note that by considering the edges $(u_{S_{\tau,i}, T_{\tau,i}}, S_{\tau,i})$ yields the same value to the objective in the OSBM model. Additionally, note that the above argument also holds in the other direction since there is a one-to-one mapping between the two optimal solutions.

Proof of Lemma 1

Proof. Observe that the polytope \mathcal{P} defined by Constraints (2)-(4) is downward closed. By Lemma 4 in (Adamczyk, Sviridenko, and Ward 2016), we can apply continuous greedy algorithm to get an approximation solution \mathbf{x}^* such that \mathbf{x}^* is feasible to \mathcal{P} and $F(\mathbf{x}^*) \geq (1 - 1/e) \max_{\mathbf{y} \in \mathcal{P}} f^+(\mathbf{y})$, where f^+ is the concave closure of f . For each given $\mathbf{y} \in \mathcal{P}$, by definition $f^+(\mathbf{y})$ is the maximum value of $\mathbb{E}[f(\mathcal{R})]$ over all possible random sets \mathcal{R} satisfying the marginal distribution that $\Pr[e \in \mathcal{R}] \leq y_e$ for each e . Thus we conclude that $\max_{\mathbf{y} \in \mathcal{P}} f^+(\mathbf{y})$ is an upper bound of the offline optimal and hence proving our claim.

Proof of Theorem 1

Proof. Let $\mathbf{Z} = (Z_e)$ be the indicator vector for e being matched in the CR-ALG and let \mathbf{X} be as defined in Theorem 3 with $\mathbf{x} = \mathbf{x}^*$ where \mathbf{x}^* is the optimal solution to the offline program. We show that \mathbf{Z} satisfies the two properties stated in Theorem 3 with $\alpha = \frac{1}{2}(1 - e^{-1/2})$. Hence we have that $\mathbb{E}[f(\mathbf{Z})] \geq \alpha \mathbb{E}[f(\mathbf{X})]$. Moreover observe that $\mathbb{E}[f(\mathbf{X})] = F(\mathbf{x}^*) \geq (1 - 1/e) \mathbb{E}[\text{OPT}]$. Combining these two facts proves Theorem 1.

Consider an edge $e = (u, v)$ with $X_e = 1$. Let $|E_{\mathbf{X}}(u)| = k_1$ and $|E_{\mathbf{X}}(v)| = k_2$, where k_1 and k_2 are the number of edges incident to u and v in \mathbf{X} ¹² including e , respectively. A sufficient condition for $Z_e = 1$ given $X_e = 1$, $|E_{\mathbf{X}}(u)| = k_1$, $|E_{\mathbf{X}}(v)| = k_2$ is that $Y_e = 1$ in this conditional space and that $Z_e = 1$ given that $Y_e = 1$. Thus we have,

$$\begin{aligned} & \Pr[Z_e = 1 \mid X_e = 1, |E_{\mathbf{X}}(u)| = k_1, |E_{\mathbf{X}}(v)| = k_2] \\ & \geq \Pr[Y_e = 1 \mid X_e = 1, |E_{\mathbf{X}}(u)| = k_1] * \Pr[Z_e = 1 \mid Y_e = 1, |E_{\mathbf{X}}(v)| = k_2, |E_{\mathbf{X}}(u)| = k_1] \\ & = \frac{1}{k_1} \sum_{t=1}^T \frac{1}{T} \frac{1}{k_2} \left(1 - \frac{1}{T \cdot k_2}\right)^{t-1} = \frac{1}{k_1} \left(1 - \exp\left(-\frac{1}{k_2}\right)\right). \end{aligned}$$

¹²More precisely, we consider the sub-graph induced by the edges e with $X_e = 1$.

In the second-last equality we used the fact that $Y_e = 1$ is set uniformly for every edge $e \in E(u)$ and thus $\Pr[Y_e = 1 \mid X_e = 1, |E_{\mathbf{X}}(u)| = k_1] = \frac{1}{k_1}$. Moreover we claimed that $\Pr[Z_e = 1 \mid Y_e = 1, |E_{\mathbf{X}}(v)| = k_2, |E_{\mathbf{X}}(u)| = k_1] = \sum_{t=1}^T \frac{1}{T} \frac{1}{k_2} \left(1 - \frac{1}{T \cdot k_2}\right)^{t-1}$. The argument is as follows. Consider a given $e = (u, v)$ with $Y_e = 1$. Note that from the algorithm, the only vertex u can be matched to is a previous arrival of v . At a given time-step $t \in [T]$, this happens with probability $\frac{1}{k_2 |V|}$. Moreover from the integral arrival rates assumption wlog we have that $|V| = T$. Thus the probability at u is safe at $t \in [T]$ is $\left(1 - \frac{1}{T \cdot k_2}\right)^{t-1}$. The probability that v arrives at $t \in [T]$ and is assigned to u is $\frac{1}{k_2 |V|}$. The event that (u, v) is matched is mutually exclusive across the T time-steps. Putting these arguments together we get that $\Pr[Z_e = 1 \mid Y_e = 1, |E_{\mathbf{X}}(v)| = k_2, |E_{\mathbf{X}}(u)| = k_1] = \sum_{t=1}^T \frac{1}{T} \frac{1}{k_2} \left(1 - \frac{1}{T \cdot k_2}\right)^{t-1}$.

Given the above derivation, we are now ready to prove the two properties. First we show that the **Marginal Property** holds with $\alpha = (1 - e^{-1/2})$. Notice that $\mathbb{E}[k_1] = 1 + \sum_{e' \in E(u), e' \neq e} x_{e'}^* \leq 2$ and $\mathbb{E}[k_2] = 1 + \sum_{e' \in E(v), e' \neq e} x_{e'}^* \leq r_v = 2$, since $\mathbf{x}^* = (x_e^*)$ is feasible to MMP (1). Also the two functions, $\frac{1}{k_1}$ and $\left(1 - \exp\left(-\frac{1}{k_2}\right)\right)$, are convex in k_1 and k_2 respectively. Thus by Jensen's inequality, we have the following.

$$\Pr[Z_e = 1 \mid X_e = 1] \geq \mathbb{E}_{k_1, k_2} \left[\Pr[Z_e = 1 \mid X_e = 1, |E_{\mathbf{X}}(u)| = k_1, |E_{\mathbf{X}}(v)| = k_2] \right] \geq \frac{1}{2} (1 - e^{-1/2}) \doteq \alpha$$

The **Monotonicity Property** can be derived from the fact that $\frac{1}{k_1}$ and $\left(1 - \exp\left(-\frac{1}{k_2}\right)\right)$ are decreasing in k_1 and k_2 respectively. Thus we are done.

Full Proof of Theorem 2

The bounds in Theorem 2 comes from two parts. The first $(1 - 1/e)$ factor is obtained from approximately solving MMP (1) offline while the second $(1 - 1/e)$ factor is the ratio obtained because of the online nature of the problem. *Note that our online analysis is tight: MMP-ALG losses at least one factor of $(1 - 1/e)$ during the online stage even when f is linear.* We first provide a sketch here and then prove it in detail. Recall that \mathbf{X} denotes the indicator for whether an edge is matched. The notations for this proof are overloaded and hence different from that in Theorem 1.

Proof sketch. For each u , let \mathbf{I}_u indicate if u is finally matched ($\mathbf{I}_u = 1$) after the T online rounds. Set $x_u^* = \sum_{e \in E(u)} x_e^*$. We then prove the following.

Lemma 2. $\{\mathbf{I}_u : u \in U\}$ are asymptotically independent with each $\Pr[\mathbf{I}_u = 1] = 1 - e^{-x_u^*}$ when $T \rightarrow \infty$ and $|U| = o(\sqrt{T})$.

Let $\mathbf{X} \in \{0, 1\}^m$ be the indicator vector for e being matched in MMP-ALG. Let $\mathbf{X}_u \in \{0, 1\}^m$ be the restriction of \mathbf{X} onto the block $E(u)$ with remaining entries all zeros such that $\mathbf{X} = \sum_{u \in U} \mathbf{X}_u$. W.l.o.g. assume that the edges are grouped by blocks of $E(u)$ for all $u \in U$: the first few entries of \mathbf{X} corresponds to some $E(u)$ followed by another $E(u')$ and so on. Let $\mathbf{1}_e$ be an m -dimensional unit vector with a 1 only at coordinate e . Lemma 3 characterizes the distribution of \mathbf{X}_u .

Lemma 3. $\Pr[\mathbf{X}_u = \mathbf{0}] = e^{-x_u^*}$, $\Pr[\mathbf{X}_u = \mathbf{1}_e] = (1 - e^{-x_u^*})^{\frac{x_e^*}{x_u^*}}$, $\forall e \in E(u)$

Combining Lemmas 2 and 3, we can view \mathbf{X} alternatively as follows. First, start with an arbitrary vector \mathbf{A} of dimension m . Then update the first block of \mathbf{A} corresponding to some $E(u)$ by setting $\mathbf{A}_u = \mathbf{0}$ with probability $e^{-x_u^*}$ and $\mathbf{A}_u = \mathbf{1}_e$ with probability $\left(1 - e^{-x_u^*}\right)^{\frac{x_e^*}{x_u^*}}$, for every $e \in E(u)$ (denote this procedure as $(*)$). Now apply $(*)$ to the remaining blocks one-by-one independently. Set $\mathbf{X} = \mathbf{A}$ at the end of this process.

In order to get the desired lower bound on $\mathbb{E}[f(\mathbf{X})]$ we consider the following random vector \mathbf{Y} . Set $Y_e = 1$ with probability $(1 - 1/e)x_e^*$ and with the remaining probability set it to be 0. From Lemma 4.2 in (Bansal et al. 2012), this implies that $\mathbb{E}[f(\mathbf{Y})] = F(\mathbf{Y}) \geq (1 - 1/e)F(\mathbf{x}^*)$. The main idea of the proof is to show that after every update of $(*)$ the invariant that the value of $\mathbb{E}[f(\mathbf{X})]$ is at least as large as $F(\mathbf{Y})$ continues to hold. Thus the final random vector \mathbf{X} satisfies that $\mathbb{E}[f(\mathbf{X})] \geq F(\mathbf{Y})$. Thus we get $\mathbb{E}[f(\mathbf{X})] \geq F(\mathbf{Y}) \geq (1 - 1/e)F(\mathbf{x}^*)$. This proves Theorem 2 since $F(\mathbf{x}^*) \geq (1 - 1/e)\mathbb{E}[\text{OPT}]$.

Proof of Lemma 2

Proof. Consider a given u . Notice that in each round t , u is assigned to a neighboring edge in $E(u)$ with probability x_u^*/T . Thus after T online rounds it gets assigned at least once with probability $1 - \left(1 - x_u^*/T\right)^T \sim 1 - e^{-x_u^*}$. Now we prove the statement about the independence for different u . Consider two given disjoint subsets U_1 and U_2 and a given vertex $u \notin U_1 \cup U_2$, we

show that

$$\Pr \left[\mathbf{I}_u = x \mid \bigwedge_{w \in U_1} (\mathbf{I}_w = 1) \bigwedge_{w \in U_2} (\mathbf{I}_w = 0) \right] = \Pr[\mathbf{I}_u = x], \forall x \in \{0, 1\}$$

For event $\bigwedge_{w \in U_2} (\mathbf{I}_w = 0)$ to occur, a sufficient condition is that in each round, no edge from $\bigcup_{w \in U_2} E(w)$ arrives. This occurs with probability $1 - \sum_{w \in U_2} x_w^*/T$. For event $\bigwedge_{w \in U_1} (\mathbf{I}_w = 1)$ to occur a sufficient condition is that in some rounds only the vertices in U_1 are assigned. For each $w \in U_1$, let N_w be the number of assignments received by w during the T online rounds, *i.e.*, the number of times w gets assigned an edge by MMP-ALG (irrespective of whether it is eventually matched or not). Observe that $N_w \sim \text{Pois}(x_w^*)$ when T is large. We can verify that ¹³ $\Pr[N_w > \ln T] = O(T^{-2})$.

Thus with probability $1 - O(|U| * T^{-2}) = 1 - o(T^{-1})$, N_w is no larger than $\ln T$ for all $w \in U_1$. Hence with a probability of at least $1 - o(T^{-1})$, the total number of rounds reserved for the shots of vertices in U_1 is at most $|U| * \ln T$. Thus, with probability a probability at least $1 - o(T^{-1})$, we have the following.

$$\begin{aligned} \left(1 - \frac{x_u^*/T}{1 - \sum_{w \in U_1 \cup U_2} x_w^*/T} \right)^T &\leq \Pr \left[\mathbf{I}_u = 0 \mid \bigwedge_{w \in U_1} (\mathbf{I}_w = 1) \bigwedge_{w \in U_2} (\mathbf{I}_w = 0) \right] \\ &\leq \left(1 - \frac{x_u^*/T}{1 - \sum_{w \in U_1 \cup U_2} x_w^*/T} \right)^{T - |U| * \ln T} \end{aligned}$$

Note that both sides of the inequality approaches $e^{-x_u^*}$ when $|U| = o(\sqrt{T})$ and $T \rightarrow \infty$. Thus we are done.

Proof of Lemma 3

Proof. The first part of the Lemma follows directly from Lemma 2 with $\mathbf{I}_u = 0$. We will now prove the second part. Let $\mathbf{I}_{u,t}$ indicate that u is matched (for the first time) at time t . Consider a given $e \in E(u)$.

$$\begin{aligned} \Pr[\mathbf{X}_u = \mathbf{1}_e | \mathbf{I}_u = 1] &= \sum_{t=1}^T \Pr[\mathbf{X}_u = \mathbf{1}_e | \mathbf{I}_{u,t} = 1] \cdot \Pr[\mathbf{I}_{u,t} = 1 | \mathbf{I}_u = 1] \\ &= \sum_{t=1}^T \frac{x_e^*/T}{\sum_{e' \in E(u)} x_{e'}^*/T} \cdot \Pr[\mathbf{I}_{u,t} = 1 | \mathbf{I}_u = 1] \\ &= \frac{x_e^*}{x_u^*} \end{aligned}$$

The second equality follows from the following argument. In each time t , u will be assigned an edge $e' \in E(u)$ with probability $x_{e'}^*/T$. Thus conditioning on the event that u gets matched at t , the conditional probability that u gets matched by edge e is exactly $\frac{x_e^*/T}{\sum_{e' \in E(u)} x_{e'}^*/T}$. Therefore we have

$$\Pr[\mathbf{X}_u = \mathbf{1}_e] = \Pr[\mathbf{X}_u = \mathbf{1}_e | \mathbf{I}_u = 1] \cdot \Pr[\mathbf{I}_u = 1] = \left(1 - e^{-x_u^*} \right) \frac{x_e^*}{x_u^*}.$$

Theorem 4. $\mathbb{E}[f(\mathbf{X})] \geq (1 - 1/e)F(\mathbf{x}^*)$

Proof. Let \mathbf{Y} , \mathbf{A} be as defined in the proof sketch above. Let \mathbf{Z} be the random vector obtained after updating the first block of \mathbf{A} , say $E(u)$. We show that $\mathbb{E}[f(\mathbf{Z})] \geq F(\mathbf{A})$. Let $\mathbf{R} = (1 - 1/e)\mathbf{x}^*$.

Let $\mathbf{R}_{-u} \in [0, 1]^m$ be the restriction of \mathbf{R} onto all blocks other than $E(u)$ (hence the entries in the block $E(u)$ are all zeroes). Note that both \mathbf{R} and \mathbf{R}_{-u} have the same length m . Let \mathcal{R}_{-u} be the random indicator vector to denote whether an edge was sampled (independently with probability according to \mathbf{R}_{-u}).

¹³<https://eventuallyalmosteverywhere.wordpress.com/2013/02/26/poisson-tails/>

From the above analysis, we have the following.

$$\mathbb{E}[f(\mathbf{Z})] \tag{5}$$

$$= \sum_{e \in E(u)} \left(1 - e^{-x_u^*}\right) \frac{x_e^*}{x_u^*} F(\mathbf{R}_{-u} + \mathbf{1}_e) + e^{-x_u^*} F(\mathbf{R}_{-u}) \tag{6}$$

$$\geq \sum_{e \in E(u)} (1 - 1/e) x_e^* F(\mathbf{R}_{-u} + \mathbf{1}_e) + \left(1 - (1 - 1/e) x_u^*\right) F(\mathbf{R}_{-u}) \tag{7}$$

$$= \sum_{e \in E(u)} (1 - 1/e) x_e^* \left(\sum_{\mathbf{a}} \Pr[\mathcal{R}_{-u} = \mathbf{a}] F(\mathbf{a} + \mathbf{1}_e) \right) + \left(1 - (1 - 1/e) x_u^*\right) \left(\sum_{\mathbf{a}} \Pr[\mathcal{R}_{-u} = \mathbf{a}] F(\mathbf{a}) \right) \tag{8}$$

$$= \sum_{\mathbf{a}} \Pr[\mathcal{R}_{-u} = \mathbf{a}] \left(\sum_{e \in E(u)} (1 - 1/e) x_e^* F(\mathbf{a} + \mathbf{1}_e) + (1 - (1 - 1/e) x_u^*) F(\mathbf{a}) \right) \tag{9}$$

Inequality (7) is due to the monotonicity of f : we have that F is non-decreasing over each entry and thus $F(\mathbf{R}_{-u} + \mathbf{1}_e) \geq F(\mathbf{R}_{-u})$ for all e ; note that the probability mass distributed over each $F(\mathbf{R}_{-u} + \mathbf{1}_e)$ is $\left(1 - e^{-x_u^*}\right) \frac{x_e^*}{x_u^*} \geq (1 - 1/e) x_e^*$ and thus by squeezing the surplus mass from each lower bound to the last item $F(\mathbf{R}_{-u})$, we get our claim. Inequality (8) follows directly from the definition of the multilinear extension F .

For a given realization \mathbf{a} of \mathcal{R}_{-u} , define a set function $g_{\mathbf{a}}$ over $E(u)$ as follows. Consider a given binary vector $\mathbf{b}' \in \{0, 1\}^{|E(u)|}$. For notational convenience, we extend \mathbf{b}' to a vector of length of m by filling the remaining entries as 0; let \mathbf{b} be the resultant vector. Define $g_{\mathbf{a}}(\mathbf{b}') = f(\mathbf{a} + \mathbf{b})$ for each $\mathbf{b}' \in \{0, 1\}^{|E(u)|}$. From Lemma 4, we have that each given \mathbf{a} , $g_{\mathbf{a}}$ is a monotone submodular set function over $E(u)$. Let $\mathbf{R}_u \in [0, 1]^m$ be the restriction of \mathbf{R} onto the block $E(u)$ with the remaining entries being 0 and $\mathbf{R}'_u \in [0, 1]^{|E(u)|}$ be the truncated version of \mathbf{R}_u onto $E(u)$. Let $G_{\mathbf{a}}$ be the multilinear extension of $g_{\mathbf{a}}$. Focus on the second part of the expression in (9).

$$\Lambda \doteq \sum_{e \in E(u)} (1 - 1/e) x_e^* F(\mathbf{a} + \mathbf{1}_e) + (1 - (1 - 1/e) x_u^*) F(\mathbf{a}) \tag{10}$$

$$= \sum_{e \in E(u)} (1 - 1/e) x_e^* g_{\mathbf{a}}(\mathbf{1}'_e) + (1 - (1 - 1/e) x_u^*) g_{\mathbf{a}}(\mathbf{0}) \tag{11}$$

$$\geq G_{\mathbf{a}}(\mathbf{Y}'_u) \tag{12}$$

$$= \sum_{\mathbf{b}'} \Pr[\mathcal{R}'_u = \mathbf{b}'] g_{\mathbf{a}}(\mathbf{b}') \tag{13}$$

$$= \sum_{\mathbf{b}} \Pr[\mathcal{R}_u = \mathbf{b}] f(\mathbf{a} + \mathbf{b}) \tag{14}$$

In (11), both $\mathbf{1}'_e$ (elementary vector of e) and $\mathbf{0}$ have the same length $|E(u)|$. Recall that $\mathbf{R} = (1 - 1/e) \mathbf{x}^*$ and thus $\mathbf{R}'_u = \left((1 - 1/e) x_e^* : u \in E(u)\right)$. Inequality (12) follows from applying Lemma 5, as shown later, to the function $g_{\mathbf{a}}$ and \mathbf{R}'_u . In equalities (13) and (14), $\mathcal{R}'_u \in \{0, 1\}^{|E(u)|}$ and $\mathcal{R}_u \in \{0, 1\}^m$ are the respective random indicator vectors denoting if an edge was sampled (independently according to \mathbf{R}'_u and \mathbf{R}_u). The two equalities (13) and (14) follow directly from the definitions of $g_{\mathbf{a}}$ and $G_{\mathbf{a}}$.

Substituting the result in (14) back to (9), we have that

$$\mathbb{E}[f(\mathbf{Z})] \geq \sum_{\mathbf{a}} \Pr[\mathcal{R}_{-u} = \mathbf{a}] \sum_{\mathbf{b}} \Pr[\mathcal{R}_u = \mathbf{b}] f(\mathbf{a} + \mathbf{b}) = \mathbb{E}[f(\mathbf{Y})] = F(\mathbf{Y}).$$

By applying the above analysis repeatedly to each block $E(u)$, we have that the final random integral vector \mathbf{Z} is same as \mathbf{X} and that it satisfies $\mathbb{E}[f(\mathbf{X})] \geq F(\mathbf{Y}) \geq (1 - 1/e) F(\mathbf{x}^*)$.

Technical Lemmas

Throughout this section, we assume f is a general non-negative monotone submodular function over the ground set $[m] = \{1, 2, \dots, m\}$. Consider a given subset $S \subseteq [m]$ and define g_S as a set function over $[m] - S$ as follows: for each $S' \subseteq [m] - S$, $g_S(S') = f(S \cup S')$.

Lemma 4. *For each given $S \subseteq [m]$, g_S is a monotone submodular function over $[m] - S$.*

Proof. Consider any two subsets $S'_1, S'_2 \subseteq [m] - S$. Observe that

$$\begin{aligned} g_S(S'_1) + g_S(S'_2) &= f(S \cup S'_1) + f(S \cup S'_2) \\ &\geq f\left(S \cup \left(S'_1 \cup S'_2\right)\right) + f\left(S \cup \left(S'_1 \cap S'_2\right)\right) \\ &= g_S(S'_1 \cup S'_2) + g_S(S'_1 \cap S'_2) \end{aligned}$$

Thus by definition, we have that g_S is submodular. The monotonicity of g_S follows directly from that of f .

Let F be the the multilinear extension of f . Consider a given vector $\mathbf{x} \in [0, 1]^m$ such that $\sum_{i \in [m]} x_i \leq 1$. Let $\mathbf{1}_i \in \{0, 1\}^m$ be the standard unit vector such that only the entry at position i is 1 and all rest are 0.

Lemma 5. $F(\mathbf{x}) \leq \sum_{i=1}^m x_i f(\mathbf{1}_i) + (1 - \sum_{i=1}^m x_i) f(\mathbf{0})$

Proof. We prove by induction on m . For the base case $m = 1$, we can verify that the inequality becomes tight from the definition of F . Now assume our claim is valid for all $m \leq k - 1$. Consider the case $m = k$.

Let $\mathbf{x}_{-k} \in [0, 1]^k$ be the restriction of \mathbf{x} onto the first $k - 1$ coordinates with the last entry being 0. Similarly let $\mathbf{x}_k \in [0, 1]^k$ be the restriction of \mathbf{x} onto the last coordinate. Note that both \mathbf{x}_{-k} and \mathbf{x}_k have the same length \mathbf{x} such that $\mathbf{x} = \mathbf{x}_{-k} + \mathbf{x}_k$. We then have the following, which completes the proof.

$$F(\mathbf{x}) = F(\mathbf{x}_{-k} + \mathbf{x}_k) \tag{15}$$

$$= x_k F(\mathbf{x}_{-k} + \mathbf{1}_k) + (1 - x_k) F(\mathbf{x}_{-k}) \tag{16}$$

$$\leq x_k \left(F(\mathbf{x}_{-k}) + f(\mathbf{1}_k) - f(\mathbf{0}) \right) + (1 - x_k) F(\mathbf{x}_{-k}) \tag{17}$$

$$= F(\mathbf{x}_{-k}) + x_k \left(f(\mathbf{1}_k) - f(\mathbf{0}) \right) \tag{18}$$

$$\leq \sum_{i=1}^{k-1} x_i f(\mathbf{1}_i) + \left(1 - \sum_{i=1}^{k-1} x_i \right) f(\mathbf{0}) + x_k \left(f(\mathbf{1}_k) - f(\mathbf{0}) \right) \tag{19}$$

$$= \sum_{i=1}^k x_i f(\mathbf{1}_i) + \left(1 - \sum_{i=1}^k x_i \right) f(\mathbf{0}) \tag{20}$$

Equality (16) follows from the definition of F ; Inequality(17) can be proved as follows. Let \mathcal{X}_{-k} be the random indicate vector to denote if an element was sampled (independently according to \mathbf{x}_{-k}). For every realization $\mathcal{X}_{-k} = \mathbf{a} \in \{0, 1\}^k$, we have that $f(\mathbf{a} + \mathbf{1}_k) \leq f(\mathbf{a}) + f(\mathbf{1}_k) - f(\mathbf{0})$, since \mathbf{a} and $\mathbf{1}_k$ represents two disjoint sets over $[k]$. Thus by taking expectation over \mathcal{X}_{-k} , we get the Inequality (17). Inequality (19) follows from the inductive hypothesis.

Supplementary section for experiments

Mathematical Program for the experiments in main section

Recall that with every vertex v , there was a weighted coverage function $f_v : 2^E \rightarrow \mathbb{R}$ associated. The objective was $\sum_{v \in V} f_v$. Our offline program can be reformulated using the epigraph form of the mathematical program and reduces to the following Linear Program. Let $[g]$ denote the set of genres and we use z to index into each genre.

$$\text{maximize } \sum_{v \in V} \sum_{z \in [g]} w_{z,v} \gamma_{z,v} \tag{21}$$

$$\text{subject to } \sum_{e \in E(v)} x_e \leq \eta * r_v \quad \forall v \in V \tag{22}$$

$$\sum_{e \in E(u)} x_e \leq B \quad \forall u \in U \tag{23}$$

$$0 \leq x_e \leq 1 \quad \forall e \in E \tag{24}$$

$$\gamma_{z,v} \leq \sum_{e \in \delta(v): q_e[z]=1} x_e \quad \forall z \in [g], v \in V \tag{25}$$

$$\gamma_{z,v} \leq 1 \quad \forall z \in [g], v \in V \tag{26}$$

Since (21) is a linear program, we can use standard tools to solve this linear program exactly.

Additional experiment for MovieLens dataset with integral arrival rates

Here we describe an additional experiment on the MovieLens dataset. We consider the case of $\eta = 1$ and compare the performance of various algorithms. The experimental setting is the same as that described in the main section. The key difference is that here we let every user come with probability $1/T$ at each time-step. Hence $T = |V| = 200$ in these experiments. Figure 4 plots the performance for varying b . As seen in the plot, both MMP-ALG and CR-ALG comprehensively beats the baselines under this setting.

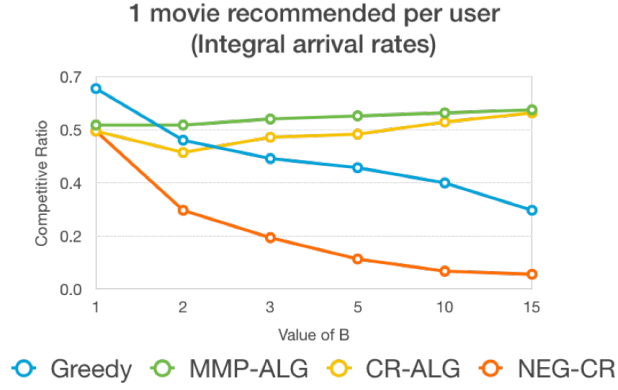


Figure 4: Comparing algorithms with $\eta = 1$ on MovieLens dataset with integral arrival rates (i.e., $r_v = 1$)

Experiments on Synthetic Data

We now describe our experimental results on synthetic data. The landscape of this problem is vast and we do not aim to be exhaustive. Instead, we focus on two important submodular functions, namely the budget-additive function and the coverage function (both defined in the preliminaries). We consider the same setting as the experiments in the main section. For both these special cases, the offline program can be reduced to a linear program (LP) which can be solved efficiently.

Offline programs. We will now describe the offline programs for these two objectives. In our experiments, we use the optimal solution of these respective programs for both benchmark of the offline as well as a guide to our online algorithms.

1. *Budget-additive function.* We want to solve the following mathematical program as our offline benchmark.

$$\text{maximize } \min\{B, \sum_{e \in E} w_e x_e\} \quad (27)$$

$$\text{subject to } \sum_{e \in E(v)} x_e \leq r_v \quad \forall v \in V \quad (28)$$

$$\sum_{e \in E(u)} x_e \leq 1 \quad \forall u \in U \quad (29)$$

$$0 \leq x_e \leq 1 \quad \forall e \in E \quad (30)$$

This can be converted to a LP using a standard transformation (similar to the epigraph form of convex programs) as follows.

$$\text{maximize } \gamma \quad (31)$$

$$\text{subject to } \sum_{e \in E(v)} x_e \leq r_v \quad \forall v \in V \quad (32)$$

$$\sum_{e \in E(u)} x_e \leq 1 \quad \forall u \in U \quad (33)$$

$$0 \leq x_e \leq 1 \quad \forall e \in E \quad (34)$$

$$\gamma \leq \sum_{e \in E} w_e x_e \quad (35)$$

$$\gamma \leq B \quad (36)$$

2. *Coverage function.* Similar to above we can convert it to a standard LP form and here we directly describe the final form. Every edge is associated with a binary feature-vector \mathbf{q}_e of dimension g . Weight of the z^{th} dimension is represented as w_z .

$$\text{maximize } \sum_{z \in [g]} w_z \gamma_z \quad (37)$$

$$\text{subject to } \sum_{e \in E(v)} x_e \leq r_v \quad \forall v \in V \quad (38)$$

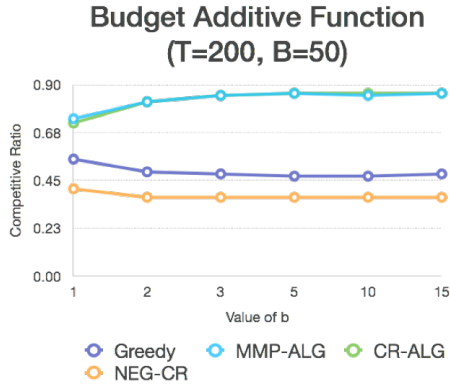
$$\sum_{e \in E(u)} x_e \leq 1 \quad \forall u \in U \quad (39)$$

$$0 \leq x_e \leq 1 \quad \forall e \in E \quad (40)$$

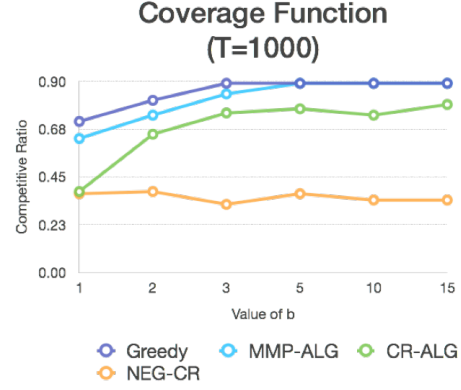
$$\gamma_z \leq \sum_{e \in E: q_e[z]=1} x_e \quad \forall z \in [g] \quad (41)$$

$$\gamma_z \leq 1 \quad \forall z \in [g] \quad (42)$$

Dataset. We use a simulated dataset for our experimental purposes as follows. First simulate an instance of the graph and then simulate the arrival sequence. For both objectives, we associate the online type with an arrival rate chosen uniformly at random



(a) Comparing algorithms on the budget-additive function. x and y -axes represent the b value in b -matching and competitive ratio respectively.



(b) Comparing algorithms on the coverage function.

in the range $[0, 1]$. Then simulate the arrival process based on these rates. The edges are chosen as follows. For every online type, we set a random set of at most 10 vertices in the offline side as its neighbor and edges between this online type and those offline vertices. We consider the b -matching scenario and vary b in the set $\{1, 2, 3, 5, 10, 15\}$. The cardinalities of offline vertices and online vertices vary based on the experiment which we will describe below.

1. *Coverage function.* For this experiment, the number of offline vertices to be 40 and the number of online types to be 200. The time-horizon is set to 1000 (hence a lot of time-steps has no arrival of a online vertex). Each offline and online type is associated with a random subset, of size at most 10, of $\{1, 2, \dots, 1000\}$ as the set of features. For every edge $e = (u, v)$, the feature vector $\mathbb{1}_e$ associated with it is given by the union of the features of its end-points u and v . Every feature in $[1000]$ is associated with a weight chosen uniformly at random in the interval $[0, 1]$.
2. *Budget-additive function.* In this experiment, the number of offline vertices to be 100 and the number of online types to be 200. The time-horizon is 200. For every edge e we choose a random number uniformly in $[0, 1]$ and set the weight w_e as this random number. We set the budget $B = 50$.

Results. Figures 5a and 5b plot the results of the experiments on the budget-additive function and coverage function, respectively. In budget-additive case, our algorithms MMP-ALG and CR-ALG significantly outperform the heuristic baselines, namely, greedy and NEG-CR. Additionally the competitive ratios of our algorithm are much higher than the theoretical benchmark (0.63 and 0.20 respectively). Note that as b increases for our experimental parameters the competitive ratios only slightly increase. In the coverage function scenario, we see that greedy out-performs both our algorithms when b is small. But as soon as b is slightly increased, our algorithm matches the performance of Greedy. This suggests that for very special coverage functions one might be able to construct algorithms that do better. However, even on such cases our general algorithm is *very competitive*. In this scenario, increasing the value of b has a larger effect than before, on the competitive ratio, yet not drastically larger. These experiments are a further evidence to support that analysis on $b = 1$ gives a good proxy for understanding the performance on larger b .