
Regret Guarantees for Item-Item Collaborative Filtering

Guy Bresler Devavrat Shah Luis F. Voloach

Laboratory for Information and Decision Systems
Department of Electrical Engineering and Computer Science
Massachusetts Institute of Technology
{guy, devavrat, voloch}@mit.edu

Abstract

There is much empirical evidence that item-item collaborative filtering works well in practice. Motivated to understand this, we provide a framework to design and analyze various recommendation algorithms. The setup amounts to online binary matrix completion, where at each time a random user requests a recommendation and the algorithm chooses an entry to reveal in the user’s row. The goal is to minimize regret, or equivalently to maximize the number of +1 entries revealed at any time. We analyze an item-item collaborative filtering algorithm that can achieve fundamentally better performance compared to user-user collaborative filtering. The algorithm achieves good “cold-start” performance (appropriately defined) by quickly making good recommendations to new users about whom there is little information.

1 Introduction

A natural approach to automated recommendation systems is to use content specific data: similar words in two books’ titles suggest that they are similar, and similarly for user’s age and geographic location. Recommending based on such content-specific data is called content filtering. In contrast, a technique called collaborative filtering (CF) provides recommendation in a content-agnostic way by exploiting patterns in general purchase or usage data to determine similarity (the term collaborative filtering was coined in [1]). For example, if 90% of users agree on two items, a CF algorithm might recommend the second item to a user who likes first item.

Virtually all industrial recommendation systems use CF. There are two main paradigms in neighborhood-based CF: user-user and item-item. In the user-user paradigm, recommendation to user u is done by finding users similar to u and recommending items liked by these users. In the item-item paradigm, in contrast, items similar to those liked by the user are found and then recommended. Empirical evidence shows that the item-item paradigm performs well (cf. [2] and [3]). In this paper we provide theoretical justification by introducing a model for recommendation and analyze the performance of a simple item-item CF algorithm.

1.1 Model

We consider a system with N users and collection of items \mathcal{I} . For each item $i \in \mathcal{I}$, user u has binary preference $L_{u,i}$ equal to +1 (like) or -1 (dislike). Recommendation systems typically operate in an online setting, meaning that when a user logs into a virtual store (such as Amazon), a

recommendation must be made immediately. At each discrete time step $t = 1, 2, 3, \dots$ a uniformly random user $U_t \in \{1, \dots, N\}$ requests a recommendation. The recommendation algorithm selects an item I_t to recommend from the set of available items \mathcal{I} , after which U_t gives feedback L_{U_t, I_t} . The recommendation must depend only on previous feedback: I_t is required to be measurable with respect to the sigma-field generated by the history $(U_1, I_1, L_{U_1, I_1}), \dots, (U_{t-1}, I_{t-1}, L_{U_{t-1}, I_{t-1}})$.

We impose the constraint that the recommendation algorithm may only recommend each item to a given user at most once. This captures the situation where users do not want to watch a movie or read a book more than once and focuses attention on the ability to recommend new items to users. For each recommendation, the algorithm may therefore either recommend an item that has been previously recommended to *other* users (in which case it has some information about the item) or recommend a new item from \mathcal{I} .

We are interested in the situation where there are many items, and will assume that \mathcal{I} is infinite. For a given item i , the corresponding i^{th} column $L_{\cdot, i} \in \{-1, +1\}^N$ of each user's preferences is called the *type* of item i . It is convenient to represent the population of items by a probability measure μ over $\{-1, +1\}^N$. When the algorithm selects an item that has not yet been recommended, the item's type is drawn from this distribution in an i.i.d. manner. Recommending a new item corresponds to adding a column to the rating matrix, with binary preferences jointly distributed according to μ .

1.2 Performance measure

As is standard in the online decision-making literature, algorithm performance is measured by regret relative to an all-knowing algorithm that makes no bad recommendations. The regret at time T is

$$\mathcal{R}(T) \triangleq \frac{1}{N} \sum_{t=1}^{T \cdot N} \frac{1}{2} (1 - L_{U_t, I_t}).$$

Recall that at time t user $U_t \in \{1, \dots, N\}$ desires a recommendation, I_t is the recommended item, and $L_{u, i}$ is equal to $+1$ (resp. -1) if u likes (resp. dislikes) item i . The regret $\mathcal{R}(T)$ is the number of bad recommendations per user after having made an average of T recommendations per user. Dependence on the algorithm is implicit through I_t .

1.3 Main results

We now describe two high-level objectives in designing a recommendation system and the corresponding guarantees obtained for our proposed algorithm ITEM-ITEM-CF, which is described in Sections 3 and 4. The results are stated in more detail in Section 5.

1.3.1 Cold-start time

With no prior information, the algorithm should give reliable recommendations as quickly as possible. The *cold-start* time of a recommendation algorithm is defined as

$$T_{\text{cold-start}} = \min \left\{ T + \Gamma \text{ s.t. } T, \Gamma \geq 0, \mathbb{E} [\mathcal{R}(T + \Delta) - \mathcal{R}(T)] \leq 0.1\Delta, \forall \Delta > \Gamma \right\}. \quad (1)$$

This is the first time after which the slope of the expected regret is bounded by 0.1: after $T_{\text{cold-start}}$ the algorithm makes a bad recommendation to a randomly chosen user with probability at most 0.1.¹

Our results: As described in Section 2, we assume that each user likes at least $\nu > 0$ fraction of the items. In Theorem 5.1 we show that algorithm ITEM-ITEM-CF achieves $T_{\text{cold-start}} = \tilde{O}(\frac{1}{\nu})^2$ for $N \geq N_0(d)$. Note that one must typically randomly sample $\Omega(\frac{1}{\nu})$ items to find a *single* liked

¹The choice 0.1 is arbitrary. We will assume that users like only a small fraction of the items, so the cold-start time is the minimum time after which the algorithm can recommend significantly better than random.

²Here and throughout, $r = \tilde{O}(x)$ means $r \leq Cx \log^c x$ for some numerical constants c, C .

item. Our results show that this amount of time-investment suffices in order to give consistently good recommendations.

1.3.2 Improving accuracy

The algorithm should give increasingly more reliable recommendations as it gains information about the users and items. This is captured by having *sublinear expected regret* $\mathbb{E}[\mathcal{R}(T)] = o(T)$.

Our results: Proposition 2.1 shows that without assumptions on the item space, it is impossible for any online algorithm to achieve sublinear regret for any positive length of time. In this paper we assume that the item space has doubling dimension (a measure of complexity of the space, defined and motivated later) bounded by d . In Theorem 5.2 we show that after time $T_{\text{cold-start}}$ (until which we incur linear regret), algorithm ITEM-ITEM-CF achieves sublinear expected regret $\tilde{\mathcal{O}}(T^{\frac{d+1}{d+2}})$ up until a certain time T_{max} . After T_{max} the expected regret again grows linearly (but with much smaller slope), and this behavior is shown in Theorem 5.3 to be unavoidable. As will be made explicit, performance improves with increasing number of users: T_{max} (and hence the length of the sublinear time-period) increases with N and the eventual linear slope decreases with N , both of which illustrate the so-called collaborative gain.

Remark 1.1. The mathematical formulation of cold-start time is to the best of our knowledge new. The strong guarantee we obtain on cold-start time (independent of doubling dimension d) is distinct from and does not follow as an implication of the sublinear regret result (which *does* depend on d).

1.4 Related Work

A latent source model of user types is used by [4] to give performance guarantees for user-user collaborative filtering. The assumptions on users and items are closely related since K items types induce at most 2^K user types and vice versa (the K item types liked by a user fully identify the user’s preferences, and there are at most 2^K such choices). Since we study algorithms that cluster similar items together, in this paper we assume a latent structure of items. We note that unlike the standard mixture model with minimum separation between mixture components (as assumed in [4]), our setup does not have any such gap condition. In contrast, we allow an effectively arbitrary model, and we prove performance guarantees based on a notion of dimensionality of the item space.

Our model can be thought of as a certain extended multi-armed bandit problem. The papers [5, 6] use notions of dimensionality similar to the one in this paper. They assume that the set of arms \mathcal{X} has some geometry: [5] assumes that the arm space is endowed with a metric, and [6] assumes that the arms have a dissimilarity function (which is not necessarily a metric). The expected rewards are then related to this geometry of the arms. In the former work, the difference in expected rewards is Lipschitz³ in the distance, and in the latter work the dissimilarity function constrains the slope of the reward around its maxima.

The regret of the algorithms in [5, 6] is $\tilde{\mathcal{O}}(T^{\frac{d'+1}{d'+2}})$, where d' is a weaker notion (than the one we use) of the covering number of \mathcal{X} , and is closely related to the doubling dimension (which we define later) in the case of a metric. The regret bound in Theorem 5.2 for the sublinear regime is of the same form, but two important aspects of our model require a different algorithm and more intricate arguments: (i) in our case no repeat recommendations (i.e. pulling the same arm) can be made to the same user, and (ii) we do not have an oracle for distances between users and items, and instead we must estimate distances by making carefully chosen exploratory recommendations.

Aside from these differences, the nature of the collaborative filtering problem leads to additional novelty relative to existing work on multi-armed bandits. First, we formalize the cold-start problem and prove strong guarantees in this regard. Second, all of our bounds are in terms of system parameters. This allows, for example, to see the role of the number of users N as an important resource allowing for collaboration.

³That is, $|\mathbb{E}[r_i] - \mathbb{E}[r_j]| \leq C \cdot d(i, j)$, where d is the distance between arms, and C is a constant.

The works of [7] and [8] on online learning and matrix completion are also relevant. In their case, however, the matrix entries to be predicted are not chosen by the algorithm and hence there is no explore-exploit trade-off. The paper [9] considers collaborative filtering under a mixture model in the *offline* setting, and they make separation assumptions between the item types (called genres in their paper). The work [10] considers a setting similar to ours (but with finite number of user and item types) and proves certain guarantees on a moving horizon approximation rather than the cumulative anytime regret. The paper [11] proves asymptotic consistency guarantees on estimating the ratings of unrecommended items. The recent paper [12] considers a different model in which repeat recommendations are also not allowed, but they make recommendations by exploiting existing information about users' interests.

It is possible that using the similarities between users, and not just between items as we do, is also useful. This has been studied theoretically in the user-user collaborative filtering framework in [4], via bandits in a wide variety of settings (for instance [13, 14, 15]), with focus on benefits to the cold-start problem [16, 17], and in practice (cf. [18, 19]). In this paper, in order to capture the power of purely item-item collaborative filtering, we intentionally avoid using any user-user similarities.

2 Structure in Data

The main intuition behind all variants of collaborative filtering is that users and items can typically be clustered in a meaningful way even ignoring context specific data. Items, for example, can often be grouped into a few different types that tend to be liked by the same users.

2.1 Need for Structure

As discussed, a good recommendation algorithm suggests items to users that are liked but have not been recommended to them before. In order to motivate the need for assumptions on the item space, we begin by stating the intuitive result that in the worst case when μ has little structure, no online algorithm can do better than recommending random items.

Proposition 2.1 (Lower Bound). *Let μ be the uniform distribution over $\{-1, +1\}^N$. Then for all $T \geq 1$, the expected regret of any online recommendation algorithm is lower bounded as $\mathbb{E}[\mathcal{R}(T)] \geq T/2$. Conversely, recommending a random item at each time step achieves $\mathbb{E}[\mathcal{R}(T)] = T/2$.*

Proposition 2.1 states that no online algorithm can have sublinear regret for any period of time unless some structural assumptions are made. Hence, to have any *collaborative gain* we need to capture the fact that items tend to come in clusters of similar items. We make two assumptions.

- (A1) The distribution μ over the item space has doubling dimension at most d for a given $d \geq 0$.
- (A2) Each user likes a random item drawn from μ with probability between ν and 2ν , and each item is liked by a fraction between ν and 2ν of the users, for a given $\nu \in (0, 1/4)$.

Assumption A1 captures structure in the item space through the notion of doubling dimension, defined and motivated in Section 2.2. Assumption A2 is made to avoid the extreme situations where almost no items are liked (in which case good recommendations are impossible) or most items are liked (in which case the regret benchmark becomes meaningless).

2.2 Item Types and Doubling Dimension

We endow the N -dimensional Hamming cube $\{-1, +1\}^N$ with the following normalized ℓ_1 metric: for any two item types $x, y \in \{-1, +1\}^N$, define their distance

$$\gamma_{x,y} \equiv \gamma(x,y) \triangleq \frac{1}{N} \sum_{k=1}^N \frac{1}{2} |x_k - y_k|.$$

When we write γ_{ij} for items i, j , we mean the distance between their types, which is the fraction of users that disagree on them.

Let $\mathcal{B}(x, r) = \{y \in \{-1, +1\}^N : \gamma_{x,y} \leq r\}$ be the ball of radius r centered at x .

Definition 2.1 (Doubling Dimension). The doubling dimension d of a measure μ on $\{-1, +1\}^N$ is the least d such that for each $x \in \{-1, +1\}^N$ with $\mu(x) > 0$ we have $\sup_{r>0} \frac{\mu(\mathcal{B}(x, 2r))}{\mu(\mathcal{B}(x, r))} \leq 2^d$. A measure with finite doubling dimension is called a doubling measure.

Measures of low doubling dimension capture the observed clustering phenomenon⁴. It follows directly from the definition, for instance, that a small doubling dimension ensures that the balls around any item type must have a significant mass. In particular, any item type $x \in \{-1, +1\}^N$ with $\mu(x) > 0$ has $\mu(\mathcal{B}(x, r)) \geq r^d$. Appendix D contains some examples of measures with low doubling dimension, and the reader is directed there to gain more intuition for the concept. Appendix D also contains experiments indicating that the doubling dimension is often small in practice, and describes in more detail why it is an assumption strictly more general than the ones made in [4].

3 Item-item collaborative filtering algorithm

In this section we describe our algorithm ITEM-ITEM-CF. The algorithm carries out a certain procedure over increasingly longer epochs (blocks of time), where the epoch index is denoted by $\tau \geq 1$. In each epoch the algorithm carefully balances *Explore* and *Exploit* steps.

In the Explore steps of epoch τ , a partition $\{P_k^{(\tau+1)}\}$ of a set of items is created for use in the subsequent epoch. Each epoch has a target precision ε_τ (specified below) such that if two items i and j are in the same block $P_k^{(\tau+1)}$, then usually $\gamma_{ij} \leq \varepsilon_{\tau+1}$.

In the Exploit steps of epoch τ , the partition $\{P_k^{(\tau)}\}$ created in the previous epoch is used for recommendation. Exploit recommendations to a user u are made as follows: u samples a random item i from a random block $P_k^{(\tau)}$, and if u likes i ($L_{u,i} = +1$) then the rest of $P_k^{(\tau)}$ is recommended to u in subsequent Exploit steps. After all items in $P_k^{(\tau)}$ have been recommended to u , the user repeats the process by sampling random items in random blocks until liking some item j in $P_{k'}^{(\tau)}$, upon which the rest of $P_{k'}^{(\tau)}$ is recommended.

In the first epoch there is no possibility of exploiting a partition created in a previous epoch, so the algorithm begins with a purely exploratory “cold-start” period. The pseudo-code of the algorithm is as follows.

⁴The above definition is a natural adaptation to probability measures on metric spaces of the notion of doubling dimension for metric spaces (cf. [20], [21] and [22]). As noted in, for instance [22], this is equivalent to enforcing that $\mu(B_\gamma(x, \alpha r)) \leq \alpha^d \cdot \mu(B_\gamma(x, r))$ for any $r > 0$ and any $x \in \{-1, +1\}^N$ with $\mu(x) > 0$. For Euclidean spaces, the doubling dimension coincides with the ambient dimension, which reinforces the intuition that metric spaces of low doubling dimension have properties of low dimensional Euclidean spaces.

ITEM-ITEM-CF(N)

1 **Algorithm parameters:**

$$\begin{aligned} \varepsilon_N &= \left(\frac{2^{5d+18}}{\nu} \cdot 630(2d+11)(d+2)^4 \frac{1}{N} \right)^{\frac{1}{d+5}}, C = \frac{\nu}{148} \frac{1}{20} \\ \varepsilon_\tau &= \max\left(\frac{1}{2^\tau}, \varepsilon_N\right) \cdot C, \text{ for } \tau \geq 1 \text{ (target accuracy for epoch)} \\ M_\tau &= \frac{2^{\max(3.5d, 8)}}{\nu} \frac{(3d+1)}{\varepsilon_\tau^{d+2}} \ln\left(\frac{2}{\varepsilon_\tau}\right), \text{ for } \tau \geq 1 \text{ (number of items introduced in epoch)} \\ D_\tau &= \frac{\nu}{2} M_\tau, \text{ for } \tau \geq 1 \text{ (duration of epoch)} \end{aligned}$$

2 **Cold-Start:**

$$\{P_k^{(1)}\} = \text{MAKE-PARTITION}(M_1, \varepsilon_1, \varepsilon_1)$$

3 **Subsequent epochs:**

for $\tau \geq 1$

do for $t = 1$ to $N \cdot D_\tau$

do $U_t = \text{random user}$

w.p. $1 - \varepsilon_\tau$: *exploit* $\{P_k^{(\tau)}\}$ to recommend an item to U_t

w.p. ε_τ : U_t *explores* to help construct partition $\{P_k^{(\tau+1)}\}$ (see Section 4)

4 Explore: making a partition

Recall that during epoch τ the goal of the explore recommendations is to create a partition $\{P_k^{(\tau+1)}\}$ of items such that whenever $i, j \in P_k^{(\tau+1)}$ then $\gamma_{ij} \leq \varepsilon_{\tau+1}$. We later prove that this can be done by executing the routine $\text{MAKE-PARTITION}(M_{\tau+1}, \varepsilon_{\tau+1}, \varepsilon_{\tau+1})$ described below, which at any point makes recommendations to a randomly chosen user. Hence, given the random user making the recommendation, ITEM-ITEM-CF provides explore recommendations in whatever order MAKE-PARTITION would have recommended (had it been run sequentially)⁵.

Definition 4.1. (ε -net) For any $\varepsilon > 0$, a collection of items \mathcal{C} is called ε -net of the item space represented by distribution μ on $\{-1, +1\}^N$ if (a) for any pair $i, j \in \mathcal{C}$, we have $\gamma_{ij} > \varepsilon/2$, and (b) for any item of type ℓ with $\mu(\ell) > 0$, there exists $i \in \mathcal{C}$ so that $\gamma_{i\ell} \leq \varepsilon$.

MAKE-PARTITION first finds a *net* \mathcal{C} for the item space (using the subroutine GET-NET described later). To each item in the net there is associated a block in a partition. M randomly sampled items are assigned to the blocks as follows: for each sampled item j , an item $i \in \mathcal{C}$ is found that is similar to j , and j is assigned to the partition block P_i (if there is more than one item i similar to j , the algorithm chooses among the relevant blocks at random). Finally, the algorithm breaks up large blocks into blocks of size on the order of $1/\varepsilon$. This guarantees that there will be many blocks in the partition, which turns out to be important in Theorem 5.1 showing brief cold-start time⁶.

MAKE-PARTITION(M, ε, δ)

- 1 $\mathcal{C} = \text{GET-NET}(\varepsilon/2, \delta/2)$
- 2 $\mathcal{M} = M$ randomly drawn items from item space
- 3 for each $i \in \mathcal{C}$, let $P_i = \emptyset$
- 4 for each $j \in \mathcal{M}$, let $S_j = \{i \in \mathcal{C} \mid \text{SIMILAR}(i, j, 0.6\varepsilon, \frac{\delta}{4M|\mathcal{C}|}) \text{ returns TRUE}\}$
- 5 if $|S_j| > 0$ then $P_i = P_i \cup \{j\}$, for i chosen u.a.r. from S_j
- 6 for each i , if $|P_i| > 1/\varepsilon$, then partition P_i into blocks of size at least $\frac{1}{2\varepsilon}$ and no more than $1/\varepsilon$
- 7 return $\{P_i\}$

⁵For instance, suppose that time t is the first in some epoch τ . We might have that times $t+5$ and $t+30$ are the first two explore recommendations of the epoch, then for those two recommendations the algorithm makes whatever the first two recommendations would have been in MAKE-PARTITION. If the execution of MAKE-PARTITION has finished, the algorithm resorts to an exploit recommendation instead.

⁶It is crucial that blocks in the partition are not too small because we would like the reward for exploration to be large when a user finds a likable item (reward in the sense of many new items to recommend). Although the algorithm does not explicitly ensure that blocks are not too small (as it did in ensuring the blocks are not too large) it comes as a byproduct of a property proven in Proposition D.2, which shows that there are not many items in the net close to any given item j .

The subroutine SIMILAR is used in MAKE-PARTITION; it determines whether most users have the same preference (like or dislike) for two given items i and j . This is accomplished by sampling many random users and counting the number of disagreements on the two items.

```

SIMILAR( $i, j, \varepsilon, \delta$ )
1  $q_{\varepsilon, \delta} = \lceil 630 \frac{d+1}{\varepsilon} \ln(\frac{1}{\delta}) \rceil$ 
2 for  $n = 1$  to  $q_{\varepsilon, \delta}$ 
3   do sample a uniformly random user  $u$ 
4   let  $X_u = \mathbf{1}_{L_{u,i} \neq L_{u,j}}$ 
5 if  $\frac{1}{q_{\varepsilon, \delta}} \sum_{\text{sampled } u} X_u \geq 0.9\varepsilon$  then
6   return FALSE
7 return TRUE

```

The subroutine GET-NET below is a natural greedy procedure for constructing an ε -net. Given parameters ε and δ , it finds a set of items \mathcal{C} that is an ε -net for μ with probability at least $1 - \delta$ (proven in the appendix). It does so by keeping a set of items \mathcal{C} and whenever it samples an item i that currently has no similar item in \mathcal{C} , it adds i to \mathcal{C} .

```

GET-NET( $\varepsilon, \delta$ )
1  $\mathcal{C} = \emptyset, \text{COUNT} = 0$ 
2  $\text{MAX-SIZE} = (4/\varepsilon)^d, \text{MAX-WAIT} = (\frac{5}{\varepsilon})^d \ln(\frac{2 \cdot \text{MAX-SIZE}}{\delta}), \delta' = \delta / (4 \cdot \text{MAX-WAIT} \cdot \text{MAX-SIZE}^2)$ 
3 while  $\text{COUNT} \leq \text{MAX-WAIT}$  and  $|\mathcal{C}| < \text{MAX-SIZE}$ 
4   do draw item  $i$  from  $\mu$ 
5   if SIMILAR( $i, j, \varepsilon, \delta'$ ) for any  $j \in \mathcal{C}$  then
6      $\text{COUNT} = \text{COUNT} + 1$ 
7   else  $\mathcal{C} = \mathcal{C} \cup i, \text{COUNT} = 0$ 
8 return  $\mathcal{C}$ 

```

5 Main Results

5.1 Cold-start time and regret bound

In Section 3 we described how ITEM-ITEM-CF starts recommending items to a user as soon as it finds one item that the user likes. This leads to a short cold-start time.

Theorem 5.1 (Cold-Start Performance). *Suppose assumptions A1 and A2 are satisfied. Then the algorithm ITEM-ITEM-CF has cold-start time $T_{\text{cold-start}} = \frac{f(\nu, d)}{N} + \tilde{\mathcal{O}}(1/\nu)$.*

Hence, the algorithm ITEM-ITEM-CF has cold-start time $\tilde{\mathcal{O}}(1/\nu)$ for N sufficiently large. This differs from that of [4] for user-user paradigm, where the cold-start time increases with user space complexity and the effect is not counteracted with more users present.

The next result shows that after the cold-start period and until a time T_{max} , the expected regret is sublinear.

Theorem 5.2 (Regret Upper Bound of ITEM-ITEM-CF). *Suppose assumptions A1 and A2 are satisfied. Then ITEM-ITEM-CF achieves expected regret*

$$\mathbb{E}[\mathcal{R}(T)] \leq \begin{cases} T_{\min} + \alpha(\nu, d) \cdot (T - T_{\min})^{\frac{d+1}{d+2}} \log_2(T - T_{\min}) & T_{\min} < T \leq T_{\max} \\ \beta + \varepsilon_N (T - T_{\max}) & T > T_{\max} \end{cases}, \quad (2)$$

where $T_{\min} = \tilde{\mathcal{O}}(\frac{1}{\nu}) + \frac{f(d, \nu)}{N}$, $T_{\max} = g(\nu, d) N^{\frac{d+2}{d+5}}$, $\varepsilon_{N, d, \nu} = h(d, \nu) (\frac{1}{N})^{\frac{1}{d+5}}$, $\beta = T_{\min} + \alpha(\nu, d) \cdot (T_{\max} - T_{\min})^{\frac{d+1}{d+2}} \log_2(T_{\max} - T_{\min})$.

The reader is directed to the proof in the appendix for the exact constants. Also note that T_{\max} increases with N and the asymptotic slope ε_N decreases as a function with N , both of which illustrate the so-called collaboration gain. Finally, the regret bound in Theorem 5.2 has an asymptotic linear regime. The next result shows that with a finite number of users such linear regret is unavoidable.

Theorem 5.3 (Asymptotic linear regret is unavoidable). *Consider an item space μ satisfying assumptions A1 and A2. Then any online algorithm must have expected asymptotic regret $\mathbb{E}[\mathcal{R}(T)] \geq C(\nu, N) \cdot T$, where $C(\nu, N) = (1 - 2\nu)/N$.*

5.2 Comparison with user-user CF

In this section we will contrast the cold-start performance of user-user collaborative filtering to that of our item-item algorithm. In particular, we give a heuristic argument showing that the cold-start time grows with the complexity of the user space. This is in contrast to our Theorem 5.1, where for any doubling dimension of the item space, if there are sufficiently many users then the cold-start time is independent of system complexity.

We consider a simple scenario with K user clusters. First, let $\tilde{\gamma}_{uv}$ denote the probability that users u and v agree on an item randomly drawn from the item space. We have K equally sized clusters of users, such that $\tilde{\gamma}_{uv} = 0$ for users u, v in the same cluster, and $\tilde{\gamma}_{uv} \in (0.1\nu, 0.2\nu)$ for users u, v in different clusters.

Consider now a given user u . A user-user algorithm seeks to find another user v who is similar to u , so that the items liked by v can be recommended to u . In order to recommend with at most (say) 0.1 probability of error, the similar user v should have distance $\tilde{\gamma}_{uv}$ at most 0.1ν . The extra factor ν is present because inference can only effectively be made from the ν fraction of liked items.

Concretely, we sample a random user v , and attempt to decide if it is from the same cluster as u . Suppose u and v have rated q items in common. The problem then reduces to a classical hypothesis test: after observing q items in common from two users, determine whether or not they are from the same cluster. The goal is to understand what is the minimal value of q needed so that the above procedure works with at least probability 1/2.

We consider the maximum a posteriori rule for deciding that v is from u 's cluster. If u and v disagree on any single item, then they cannot be from the same cluster. Conversely, if u and v agree on all q sampled items, the MAP rule declares v to be from u 's cluster only if

$$\frac{K-1}{K}(1-0.2\nu)^q \leq \frac{1}{K}.$$

This means that if q is too small, we will never declare v to be from u 's cluster and therefore will be unable to make recommendations. Rearranging gives $q \geq \Omega(\log(K)/\nu)$.

Hence, an algorithm based on user similarity needs at least $T = \Omega(\log(K)/\nu)$ steps simply to determine if two users are similar to each other, a prerequisite to making good recommendations. In contrast, we have shown that ITEM-ITEM-CF achieves cold start time $\mathcal{O}(1/\nu)$, which in particular does not increase with the complexity of the item space.

This contrast between cold-start times highlights the *asymmetry between item-item and user-user collaborative filtering*. It is much faster to compare two items than it is to compare two users: it takes a long time to make many recommendations to two particular users, but comparing two items can be done in *parallel* by sampling different users.

6 Discussion

This paper analyzes a collaborative filtering algorithm based on item similarity, and proves guarantees on its regret. Our algorithm exploits structure only in the item space. It would be desirable to have a matching lower bound, in the spirit of lower bound for multi-armed bandits in metric spaces

shown in [5] and [6]. Furthermore, many practitioners use a hybrid of user-user and item-item paradigms [23] and [24], and formally analyzing such algorithms is an open problem.

Finally, the main challenge of the cold-start problem is that initially we do not have any information about item-item similarities. In practice, however, some similarity can be inferred via content specific information. For instance, two books with similar words in the title can have a prior for having a higher similarity than books with no similar words in the title. In practice such hybrid content/collaborative filtering algorithms have had good performance [25]. Formally analyzing such hybrid algorithms has not been done and can shed light onto how to best combine content information with the collaborative filtering information.

References

- [1] David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 1992.
- [2] Greg Linden, Brent Smith, and Jeremy York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, 2003.
- [3] Yehuda Koren and Robert Bell. Advances in collaborative filtering. In *Recommender Systems Handbook*, pages 145–186. Springer US, 2011.
- [4] Guy Bresler, George Chen, and Devavrat Shah. A latent source model for online collaborative filtering. In *NIPS*, 2014.
- [5] Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Bandits and experts in metric spaces. *arXiv preprint arXiv:1312.1277*, 2013.
- [6] Sébastien Bubeck, Rémi Munos, Gilles Stoltz, and Csaba Szepesvari. X-armed bandits. *JMLR*, 2011.
- [7] Elad Hazan, Satyen Kale, and Shai Shalev-Shwartz. Near-optimal algorithms for online matrix prediction. *COLT*, 2012.
- [8] Nicolò Cesa-Bianchi and Ohad Shamir. Efficient transductive online learning via randomized rounding. In *Empirical Inference*, pages 177–194. Springer, 2013.
- [9] Jon Kleinberg and Mark Sandler. Using mixture models for collaborative filtering. In *STOC*, 2004.
- [10] Onkar Dabeer. Adaptive collaborating filtering: The low noise regime. In *ISIT*, 2013.
- [11] Gerard Biau, Benoit Cadre, and Laurent Rouviere. Statistical analysis of k-nearest neighbor collaborative recommendation. *The Annals of Statistics*, 38(3):1568–1592, 2010.
- [12] Laurent Massoulié, Mesrob Ohannessian, and Alexandre Proutiere. Greedy-bayes approach for targeted news dissemination. *Sigmetrics*, 2015.
- [13] Noga Alon, Nicolò Cesa-Bianchi, Claudio Gentile, Shie Mannor, Yishay Mansour, and Ohad Shamir. Nonstochastic multi-armed bandits with graph-structured feedback. *arXiv Technical Report arXiv:1409.8428*, 2014.
- [14] Aleksandrs Slivkins. Contextual bandits with similarity information. *The Journal of Machine Learning Research*, 2014.
- [15] Nicolo Cesa-Bianchi, Claudio Gentile, and Giovanni Zappella. A gang of bandits. In *NIPS*, 2013.
- [16] Claudio Gentile, Shuai Li, and Giovanni Zappella. Online clustering of bandits. *ICML*, 2014.
- [17] Stéphane Caron and Smriti Bhagat. Mixing bandits: A recipe for improved cold-start recommendations in a social network. In *Workshop on Social Network Mining and Analysis*, 2013.
- [18] Abhinandan S Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram. Google news personalization: scalable online collaborative filtering. In *WWW*, 2007.
- [19] Alejandro Bellogin and Javier Parapar. Using graph partitioning techniques for neighbour selection in user-based collaborative filtering. In *RecSys*, 2012.
- [20] Juha Heinonen. *Lectures on analysis on metric spaces*. Springer, 2001.
- [21] Sarel Har-Peled and Manor Mendel. Fast construction of nets in low-dimensional metrics and their applications. *SIAM Journal on Computing*, 35(5):1148–1184, 2006.
- [22] Sanjoy Dasgupta and Kaushik Sinha. Randomized partition trees for nearest neighbor search. *Algorithmica*, 2014.
- [23] Jun Wang, Arjen De Vries, and Marcel Reinders. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *ACM SIGIR*, 2006.
- [24] Koen Verstrepen and Bart Goethals. Unifying nearest neighbors collaborative filtering. In *Recsys*, 2014.
- [25] Prem Melville, Raymond J Mooney, and Ramadass Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *AAAI/IAAI*, 2002.
- [26] Ken Goldberg, Theresa Roeder, Dhruv Gupta, and Chris Perkins. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, 2001.
- [27] J Riedl and J Konstan. Movielens dataset, 1998.
- [28] Colin McDiarmid. Concentration. In *Probabilistic methods for algorithmic discrete mathematics*. Springer, 1998.

A Correctness of Explore

This section of the Appendix establishes correctness of the explore procedure as well as some of its properties that will be utilized for establishing the main result of the paper. Concretely, we will prove that with high probability the procedure MAKE-PARTITION produces a partition of similar items during each epoch. To that end, in appendix A.1, we prove that SIMILAR succeeds in deciding whether two items are close to each other. In appendix A.2 we prove that the procedure GET-NET succeeds in finding a set of items that is an ε -net for μ . We then put all the pieces together and prove that MAKE-PARTITION, the routine at which the explore recommendations are aimed at completing, succeeds in creating a partition of similar items. Finally, in appendix A.3 we prove that with high probability during any given epoch there will be enough explore recommendations.

A.1 Guarantees for SIMILAR

The procedure SIMILAR is used throughout GET-NET and MAKE-PARTITION. It tests whether two items are approximately ε -close to each other.

Lemma A.1. *Let i and j be arbitrary items, $\delta, \varepsilon \in (0, 1)$, and $S_{i,j}$ be the event that SIMILAR($i, j, \varepsilon, \delta$) returns TRUE. Then we have that*

- (i) if $\gamma_{i,j} \leq 0.8\varepsilon$, then $\mathbb{P}(S_{i,j}) \geq 1 - \delta$, and
- (ii) if $\gamma_{i,j} \in [k\varepsilon, (k+1)\varepsilon]$ where $k \in \{1, \dots, \lfloor \frac{1}{\varepsilon} \rfloor\}$, then $\mathbb{P}(S_{i,j}) \leq \frac{\delta}{4} \left(\frac{1}{4k}\right)^d \frac{1}{k^2}$.

Proof. Let us begin with case (i), where $\gamma_{i,j} \leq 0.8\varepsilon$. Let A_n be the event that the n^{th} randomly chosen user disagrees on i and j (i.e. that user likes exactly one of i and j), and note that $\mathbb{E}(\sum_{n=1}^{q_{\varepsilon,\delta}} \mathbf{1}_{A_n}) \leq 0.8\varepsilon q_{\varepsilon,\delta}$. Then, by the Chernoff Bound (stated in Theorem E.1), we get

$$\begin{aligned} \mathbb{P}(S_{i,j}^c \mid \gamma_{i,j} \leq 0.8\varepsilon) &= \mathbb{P}\left(\sum_{n=1}^{q_{\varepsilon,\delta}} \mathbf{1}_{A_n} \geq 0.9\varepsilon q_{\varepsilon,\delta}\right) \leq \mathbb{P}\left(\sum_{n=1}^{q_{\varepsilon,\delta}} \mathbf{1}_{A_n} \geq (1 + .1) \mathbb{E}\left(\sum_{n=1}^{q_{\varepsilon,\delta}} \mathbf{1}_{A_n}\right)\right) \\ &\leq \exp\left(-\frac{.1^2}{2 + .1} \mathbb{E}\left(\sum_{n=1}^{q_{\varepsilon,\delta}} \mathbf{1}_{A_n}\right)\right) \leq \exp\left(-\frac{1}{210} 0.8\varepsilon q_{\varepsilon,\delta}\right). \end{aligned} \quad (3)$$

Now, since $q_{\varepsilon,\delta} = \lceil 630 \frac{d+1}{\varepsilon} \ln(\frac{1}{\delta}) \rceil \geq 210 \cdot \frac{5}{4} \frac{1}{\varepsilon} \ln(\frac{1}{\delta})$, we get

$$\mathbb{P}(S_{i,j}^c \mid \gamma_{i,j} \leq 0.8\varepsilon) = \mathbb{P}\left(\sum_{n=1}^{q_{\varepsilon,\delta}} \mathbf{1}_{A_n} \geq 0.9\varepsilon q_{\varepsilon,\delta}\right) \leq \delta. \quad (4)$$

Let us now consider case (ii), where $\gamma_{i,j} \in [k\varepsilon, (k+1)\varepsilon]$. As before, let A_n be the event that the n^{th} randomly chosen user disagrees on i and j . Then, since $k \geq 1$ and again by the Chernoff bound we get

$$\begin{aligned} \mathbb{P}(S_{i,j} \mid \gamma_{i,j} \in [k\varepsilon, (k+1)\varepsilon]) &= \mathbb{P}\left(\sum_{n=1}^{q_{\varepsilon,\delta}} \mathbf{1}_{A_n} \leq 0.9\varepsilon q_{\varepsilon,\delta}\right) \\ &\leq \mathbb{P}\left(\sum_{n=1}^{q_{\varepsilon,\delta}} \mathbf{1}_{A_n} \leq (1 - .1) \mathbb{E}\left(\sum_{n=1}^{q_{\varepsilon,\delta}} \mathbf{1}_{A_n}\right)\right) \\ &\leq \exp\left(-\frac{.1^2}{2 + .1} \mathbb{E}\left(\sum_{n=1}^{q_{\varepsilon,\delta}} \mathbf{1}_{A_n}\right)\right) \leq \exp\left(-\frac{1}{210} q_{\varepsilon,\delta} k\varepsilon\right). \end{aligned} \quad (5)$$

In order to get the desired conditions we need to show that

$$\exp\left(-\frac{1}{210} q_{\varepsilon,\delta} k\varepsilon\right) \leq \frac{\delta}{4} \left(\frac{1}{4k}\right)^d \frac{1}{k^2}. \quad (6)$$

By taking natural log of both sides we get

$$\frac{1}{210}q_{\varepsilon,\delta}k\varepsilon \geq d \ln(4k) + \ln\left(\frac{4k^2}{\delta}\right), \quad (7)$$

which is in turn at most $(d+1) \ln\left(\frac{16k^3}{\delta}\right)$. Hence, it suffices to show that

$$\frac{1}{210}q_{\varepsilon,\delta}k\varepsilon \geq (d+1) \ln\left(\frac{16k^3}{\delta}\right). \quad (8)$$

However, since we have $q_{\varepsilon,\delta} \geq 630 \frac{d+1}{\varepsilon} \ln\left(\frac{1}{\delta}\right)$, we get

$$\frac{1}{210}q_{\varepsilon,\delta}k\varepsilon \geq \frac{1}{210}630 \frac{d+1}{\varepsilon} \ln\left(\frac{1}{\delta}\right) k\varepsilon = 3(d+1)k \ln\left(\frac{1}{\delta}\right). \quad (9)$$

Now since $3k \geq \ln(16k^3)$ for $k \geq 1$, we conclude that

$$\mathbb{P}(S_{ij} \mid \gamma_{ij} \in [k\varepsilon, (k+1)\varepsilon]) = \mathbb{P}\left(\sum_{n=1}^q \mathbf{1}_{A_n} \leq 0.9\varepsilon q_{\varepsilon,\delta}\right) \leq \frac{\delta}{4} \left(\frac{1}{4k}\right)^d \frac{1}{k^2}, \quad (10)$$

as desired. ■

In the Lemma above we showed that given two items i and j , the routine SIMILAR can tell that the items are ε -close when $\gamma_{i,j} \leq 0.8\varepsilon$, and it can tell that they are not when $\gamma_{i,j} \geq \varepsilon$. Furthermore, the Lemma states that the probability of a false-positive decreases extremely fast as the items get farther apart. The Lemma below shows that, when one of the items is drawn from μ , SIMILAR still works and that the false positive rate is small, despite the possibility that the it may be much more likely to draw an item that is far from i . In Lemma A.2 below we use the doubling dimension of μ for the first time, and in this context the doubling dimension guarantees that SIMILAR (which is a random projection) preserves relative distances.

Lemma A.2. *Let i be an arbitrary item, let J be a randomly drawn item from an item space μ of doubling dimension d , and let $S_{i,J}$ be the event that SIMILAR($i, J, \varepsilon, \delta$) returns TRUE. Then we have $\mathbb{P}(\gamma_{i,J} \geq \varepsilon \mid S_{i,J}) \leq \delta$.*

Proof. By Bayes' rule we get

$$\mathbb{P}(\gamma_{i,J} \geq \varepsilon \mid S_{i,J}) = \frac{\mathbb{P}(\gamma_{i,J} \geq \varepsilon, S_{i,J})}{\mathbb{P}(\gamma_{i,J} \geq \varepsilon, S_{i,J}) + \mathbb{P}(\gamma_{i,J} < \varepsilon, S_{i,J})}, \quad (11)$$

where the probability is with the respect to the random choice of J and the random users in SIMILAR. Now if

$$\delta \mathbb{P}(S_{i,J}, \gamma_{i,J} < \varepsilon) \geq (1 - \delta) \mathbb{P}(S_{i,J}, \gamma_{i,J} \geq \varepsilon) \quad (\star)$$

holds, we get

$$\mathbb{P}(\gamma_{i,J} \geq \varepsilon \mid S_{i,J}) = \frac{1}{1 + \frac{\mathbb{P}(\gamma_{i,J} < \varepsilon, S_{i,J})}{\mathbb{P}(\gamma_{i,J} \geq \varepsilon, S_{i,J})}} \leq \frac{1}{1 + \frac{1-\delta}{\delta}} = \delta. \quad (12)$$

Hence, it suffices to show (\star) . Recall that $\mathcal{B}(i, r)$ is the ball of radius r centered at i , and note that

$$\mathbb{P}(\gamma_{i,J} < \varepsilon, S_{i,J}) \geq \mathbb{P}(S_{i,J} \mid \gamma_{i,J} \leq \varepsilon/2) \mu(\mathcal{B}(i, \varepsilon/2)), \quad (13)$$

and

$$\mathbb{P}(\gamma_{i,J} \geq \varepsilon, S_{i,J}) = \sum_{k=0}^{\lceil \log_2(\frac{1}{\varepsilon}) \rceil} \mathbb{P}(S_{i,J} \mid \gamma_{i,J} \in [2^k \varepsilon, 2^{k+1} \varepsilon]) \mu(\mathcal{B}(i, 2^{k+1} \varepsilon) - \mathcal{B}(i, 2^k \varepsilon)) \quad (14)$$

$$\leq \sum_{k=0}^{\lceil \log_2(\frac{1}{\varepsilon}) \rceil} \mathbb{P}(S_{iJ} \mid \gamma_{iJ} \in [2^k \varepsilon, 2^{k+1} \varepsilon)) \mu(\mathcal{B}(i, 2^{k+1} \varepsilon)). \quad (15)$$

Let us first lower bound $\mathbb{P}(S_{iJ}, \gamma_{iJ} \leq \varepsilon/2) \mu(\mathcal{B}(i, \varepsilon/2))$. Let $p \triangleq \mu(\mathcal{B}(i, \varepsilon/2))$. Then by Lemma A.1 we get that

$$\mathbb{P}(S_{iJ} \mid \gamma_{iJ} \leq \varepsilon/2) \mu(\mathcal{B}(i, \varepsilon/2)) \geq (1 - \delta) p. \quad (16)$$

We will now upper bound $\mathbb{P}(S_{iJ}, \gamma_{iJ} \geq \varepsilon)$. Using the doubling dimension of the item space, which implies that $\mu(\mathcal{B}_\gamma(i, 2^{k+1} \varepsilon)) \leq (2^{k+2})^d p$, we also get that

$$\mathbb{P}(\gamma_{iJ} \geq \varepsilon, S_{iJ}) \leq \sum_{k=0}^{\lceil \log_2(\frac{1}{\varepsilon}) \rceil} \mathbb{P}(S_{iJ} \mid \gamma_{iJ} \in [2^k \varepsilon, 2^{k+1} \varepsilon)) \mu(\mathcal{B}_\gamma(i, 2^{k+1} \varepsilon)) \quad (17)$$

$$\leq \sum_{k=0}^{\lceil \log_2(\frac{1}{\varepsilon}) \rceil} \mathbb{P}(S_{iJ} \mid \gamma_{iJ} \in [2^k \varepsilon, 2^{k+1} \varepsilon)) (2^{k+2})^d p. \quad (18)$$

We now use the second half of Lemma A.1, and arrive at

$$\mathbb{P}(\gamma_{iJ} \geq \varepsilon, S_{iJ}) \leq \sum_{k=0}^{\lceil \log_2(\frac{1}{\varepsilon}) \rceil} \left(\frac{\delta}{4} \left(\frac{1}{4 \cdot 2^k} \right)^d \frac{1}{2^{2k}} \right) (2^{k+2})^d p \leq p \frac{\delta}{4} \sum_{k=0}^{\infty} \frac{1}{2^{2k}} \leq p \frac{\delta}{2}. \quad (19)$$

We can now check that indeed sufficient condition from eq. (*) is satisfied:

$$\delta \mathbb{P}(S_{iJ}, \gamma_{iJ} < \varepsilon) \geq \delta p (1 - \delta) \geq \frac{\delta}{2} p (1 - \delta) \geq (1 - \delta) \mathbb{P}(S_{iJ}, \gamma_{iJ} \geq \varepsilon), \quad (20)$$

which completes the proof. ■

A.2 Making the Partition

In the previous section we proved that the procedure SIMILAR works well in deciding whether two items are similar to each other at some desired precision. In this section, we will prove that with SIMILAR as a building block can partition items into blocks of similar items.

We will begin by proving that the subroutine GET-NET, used in the beginning of MAKE-PARTITION, succeeds at producing an ε -net of items with high probability.

Lemma A.3. *With probability at least $1 - \delta$ the routine GET-NET(M, ε, δ) returns an ε -net for μ that contains at most $(\frac{4}{\varepsilon})^d$ items.*

Proof. Let us first settle some notation. Let \mathcal{C}_{final} be the set returned GET-NET(ε, δ), let \mathcal{C}_r be the set \mathcal{C} when it had r items, and let \mathcal{M}_r be the set of random items drawn when \mathcal{C} had r items. Furthermore, denote by P be event that for each $i, j \in \mathcal{C}_{final}$ we have $\gamma_{i,j} \geq \varepsilon/2$, and C the event that for each item i there exists a $j \in \mathcal{C}_{final}$ such that $\gamma_{i,j} \leq \varepsilon$. Furthermore, let $E_{i,j}$ be the event $\{S_{i,j}, \gamma_{i,j} > 0.5\varepsilon\} \cup \{S_{i,j}^c, \gamma_{i,j} < 0.8\varepsilon\}$, and

$$E \triangleq \bigcup_{r=0}^{|\mathcal{C}_{final}|-1} \bigcup_{j \in \mathcal{M}_r} \bigcup_{c \in \mathcal{C}_r} E_{c,j}.$$

Intuitively, the event E happens when some call to SIMILAR returned an erroneous answer. We will show that

$$(A) \mathbb{P}(E) \leq \delta/2,$$

$$(B) \mathbb{P}(P^c | E^c) = 0, \text{ and}$$

$$(C) \mathbb{P}(C^c | E^c) \leq \delta/2,$$

which together show that $\text{GET-NET}(\varepsilon, \delta)$ returns an ε -net with probability at least $1 - \delta$.

Proof of (A): By a union bound we get

$$\mathbb{P}(E) \leq \sum_{r=0}^{|\mathcal{C}_{final}|-1} \sum_{j \in \mathcal{M}_r} \sum_{c \in \mathcal{C}_r} \mathbb{P}(E_{c,j}) \leq \sum_{r=0}^{\text{MAX-SIZE}} \sum_{j \in \mathcal{M}_r} \sum_{c \in \mathcal{C}_r} \mathbb{P}(E_{c,j}),$$

and since there are at most MAX-WAIT items in \mathcal{M}_r and at most MAX-SIZE items in \mathcal{C}_r we get

$$\mathbb{P}(E) \leq (\text{MAX-SIZE})^2 \cdot \text{MAX-WAIT} \cdot \mathbb{P}(E_{c,j}) \leq \delta/4.$$

where the last inequality follows since Lemma A.1 gives us that

$$\mathbb{P}(E_{c,j}) \leq \delta' = \delta / (4 \cdot \text{MAX-WAIT} \cdot \text{MAX-SIZE}^2).$$

Proof of (B): Note that if there are two items $i, j \in \mathcal{C}$ such that $\gamma_{i,j} \leq \varepsilon/2$, then this must have happened as a result of some erroneous response of $\text{SIMILAR}(i, j, 0.6\varepsilon, \delta')$. However, since we are conditioning on E^c no such erroneous response can occur.

Proof of (C): Let us consider the two cases, when $|\mathcal{C}_{final}| = \text{MAX-SIZE}$, and when $|\mathcal{C}_{final}| < \text{MAX-SIZE}$. Then

$$\mathbb{P}(C^c | E^c) \leq \mathbb{P}(C^c | E^c, |\mathcal{C}_{final}| = \text{MAX-SIZE}) + \mathbb{P}(C^c | E^c, |\mathcal{C}_{final}| < \text{MAX-SIZE}),$$

and we will show that

$$(C_1) \mathbb{P}(C^c | E^c, |\mathcal{C}_{final}| = \text{MAX-SIZE}) = 0, \text{ and}$$

$$(C_2) \mathbb{P}(C^c | E^c, |\mathcal{C}_{final}| < \text{MAX-SIZE}) = \delta/2,$$

which together prove (C).

Proof of (C₁): Note that since we are conditioning on E_c , which in turn implies that \mathcal{C}_{final} is a packing, we have that $\gamma_{ij} \geq \varepsilon/2$ for each $i, j \in \mathcal{C}_{final}$, we get that $B_\gamma(i, \varepsilon/4) \cap \mathcal{B}(j, \varepsilon/4) = \emptyset$ for each $i, j \in \mathcal{C}_{final}$ as well. Hence

$$\mu \left(\bigcup_{i \in \mathcal{C}_{final}} \mathcal{B}(i, \varepsilon/2) \right) \geq \mu \left(\bigcup_{i \in \mathcal{C}_{final}} \mathcal{B}(i, \varepsilon/4) \right) = \sum_{i \in \mathcal{C}_{final}} \mu(\mathcal{B}(i, \varepsilon/4)).$$

Now by the doubling dimension condition we get that $\mu(\mathcal{B}(i, \varepsilon/2)) \geq (\frac{\varepsilon}{4})^d$, and hence can conclude that

$$\mu \left(\bigcup_{i \in \mathcal{C}_{final}} \mathcal{B}(i, \varepsilon/2) \right) \geq \left(\frac{\varepsilon}{4}\right)^d \cdot \left(\frac{4}{\varepsilon}\right)^d = 1,$$

and hence for any item there exists an item in \mathcal{C}_{final} such that $\gamma_{i,j} \leq \varepsilon/2$.

Proof of (C₂): Consider now the case in which $|\mathcal{C}_{final}| < \text{MAX-SIZE}$. This means that at some iteration $r \in \{0, \dots, \text{MAX-SIZE} - 1\}$ of the while loop of there existed an item j such that $\gamma_{ij} > \varepsilon$ for each $i \in \mathcal{C}_r$ but the algorithm nevertheless terminated and returned \mathcal{C}_r . Let T_r be the event that

the algorithm terminated at round r while there still existed an item j which is not ε close to any item in \mathcal{C}_r . Then $C^c \subset \bigcup_{r=0}^{|\mathcal{C}_{final}|} T_r$, and hence

$$\mathbb{P}(C^c \mid E^c, |\mathcal{C}_{final}| < \text{MAX-SIZE}) \leq \sum_{r=0}^{|\mathcal{C}_{final}|} \mathbb{P}(T_r \mid E^c),$$

and so it suffices for C_2 to show that $\mathbb{P}(T_r \mid E^c) \leq \frac{\delta}{2} \frac{1}{\text{MAX-SIZE}}$ for each r .

To show that $\mathbb{P}(T_r \mid E^c) \leq \frac{\delta}{2} \frac{1}{\text{MAX-SIZE}}$, we will first observe that if there is an item j which is ε far from all of \mathcal{C}_r , then the ball $B(j, \varepsilon/5)$ must have significant mass which is all also not close to any item in \mathcal{C}_r . We will then conclude, by a standard coupon collector argument, that this mass is found with high probability.

By the doubling dimension condition, the ball $B(j, \varepsilon/5)$ must have mass at least $(\varepsilon/5)^d$. Let M_r be the event that no item in $B(j, \varepsilon/5)$ was sampled during the r^{th} of the while loop. Then

$$\mathbb{P}(T_r \mid E^c) \leq \mathbb{P}(T_r \mid E^c, M_r^c) + \mathbb{P}(M_r \mid E^c).$$

Note that given M_r^c , which implies that an item j which is at least 0.8ε away from all of \mathcal{C}_r was sampled, the event T_r happens only if j is judged to be similar to some $c \in \mathcal{C}_r$. However, since we're conditioning on E_c that cannot happen, and we get $\mathbb{P}(T_r \mid E^c, M_r^c) = 0$.

Finally, we will use the coupon collector argument and show that $\mathbb{P}(M_r \mid E^c) \leq \delta/(2 \cdot \text{MAX-SIZE})$. The event M_r , that no item in $B(j, \varepsilon/2)$ was sampled during the r^{th} iteration of the loop happens with probability at most

$$\left(1 - \left(\frac{\varepsilon}{5}\right)^d\right)^{\text{MAX-WAIT}} \leq \exp\left(-\left(\frac{\varepsilon}{5}\right)^d \text{MAX-WAIT}\right) \leq \frac{\delta}{2 \cdot \text{MAX-SIZE}},$$

as we wished. ■

It is now only left to prove that the main tool used during exploration, MAKE-PARTITION, indeed produces a partition of similar items.

We now prove that with high probability the procedure MAKE-PARTITION creates a partition of similar items. Furthermore, the additional properties stated in the Lemma, regarding the size of the blocks, will be crucial later in ensuring a quick cold-start performance.

Lemma A.4. *Let $\varepsilon, \delta \in (0, 1)$, and let $M \geq 12 \cdot \left(\frac{12}{\varepsilon}\right)^{d+1} \ln\left(\frac{2}{\delta} \left(\frac{8}{\varepsilon}\right)^d\right)$. Then with probability at least $1 - \delta$ the subroutine MAKE-PARTITION(M, ε, δ) returns a partition $\{P_k\}$ of a subset of M randomly drawn items such that*

- (i) *For each block P_k and $i, j \in P_k$ we have $\gamma_{i,j} \leq 1.2 \cdot \varepsilon$,*
- (ii) *Each block P_k contains at least $\frac{1}{2\varepsilon}$ items,*
- (iii) *Each block P_k contains at most $1/\varepsilon$ items,*
- (iv) *There are at most $2M\varepsilon$ blocks.*

Proof. We will show that properties (i) and (ii) hold with probability at least $1 - \delta$, and note that (iii) follows directly from the algorithm and that (iv) follows from (ii).

Let C be the event that the set \mathcal{C} returned by GET-NET is not an $\frac{\varepsilon}{2}$ -net for μ , and let \mathcal{M} be the set of M items sampled. Similarly to in the proof of Lemma A.3, let $E_{i,j}$ be the event $\{S_{i,j}, \gamma_{ij} > 0.6\varepsilon\} \cup \{S_{i,j}^c, \gamma_{ij} < 0.5\varepsilon\}$, where S_{ij} is the event that routine SIMILAR($i, j, 0.6\varepsilon, \delta/(4M|\mathcal{C}|)$) returns SIMILAR. Intuitively, the event $E_{i,j}$ happens when SIMILAR returns what it shouldn't have. Furthermore, let $E = \bigcup_{c \in \mathcal{C}} \bigcup_{j \in \mathcal{M}} E_{c,j}$.

Let F be the event that for some block P_k there exists $i, j \in P_k$ such that $\gamma_{ij} > 1.2\varepsilon$, and let B be the event that all blocks in the partition have size at least $\frac{1}{2\varepsilon}$. Therefore, $B^c = \bigcup_k B_k^c$, where B_k^c is the event that P_k has size less than $\frac{1}{2\varepsilon}$.

Since event F guarantees condition (i) and event B guarantees condition (ii) it suffices to show that

$$\mathbb{P}(F^c \cup B^c) \leq \delta.$$

We will do so by conditioning on C^c and E^c where after a couple of union bounds we arrive at

$$\mathbb{P}(F^c \cup B^c) \leq \mathbb{P}(F^c | C^c, E^c) + \mathbb{P}(B^c | C^c, E^c) + \mathbb{P}(C) + \mathbb{P}(E),$$

and showing

$$(A) \mathbb{P}(B^c | C^c, E^c) \leq \delta/2,$$

$$(B) \mathbb{P}(F^c | C^c, E^c) = 0,$$

$$(C) \mathbb{P}(E) \leq \delta/4, \text{ and}$$

$$(D) \mathbb{P}(C) \leq \delta/4$$

completes the proof.

Proof of (A): Note that $B^c = \bigcup_{c \in \mathcal{C}} B_c^c$, where B_c is the event that the block P_{k_c} constructed with $c \in \mathcal{C}$ as reference has size at least $\frac{1}{2\varepsilon}$.

Note that the event B_c happens whenever at least $\frac{1}{2}M \left(\frac{\varepsilon}{12}\right)^d$ (which is at least $1/\varepsilon$) items are added to P_{k_c} . This is because in this event, P_{k_c} will receive more than $1/\varepsilon$ items assigned to it and hence the algorithm will break it up in to smaller block but each of them will be of size at least $\frac{1}{2\varepsilon}$. Hence

$$\left\{ \sum_{n=1}^M \mathbf{1}_{X_{c,n}} > \frac{1}{2}M \left(\frac{\varepsilon}{12}\right)^d \right\} \subset B_c,$$

where $X_{c,n}$ is the event that the n^{th} item sampled ends up in P_{k_c} . We will show how

$$\mathbb{P}(X_{c,n} | C^c, E^c) \geq (\varepsilon/12)^d \tag{21}$$

allows us to prove (A), and we then prove eq. (21). Note that the $\{X_{c,n}\}_n$ are independent, and hence the sum $\sum_{n=1}^M \mathbf{1}_{X_{c,n}}$ stochastically dominates the sum $\sum_{n=1}^M Y_{c,n}$, where each $Y_{c,n}$ is an independent Bernoulli random variable of parameter $(\varepsilon/12)^d$. By the Chernoff bound we then get

$$\mathbb{P}(B_c^c | C^c, E^c) \leq \mathbb{P}\left(\sum_{n=1}^M Y_{c,n} \leq \frac{1}{2}M \left(\frac{\varepsilon}{12}\right)^d\right) \leq \exp\left(-\frac{1}{12}M \left(\frac{\varepsilon}{12}\right)^d\right) \leq \frac{\delta}{2} \left(\frac{\varepsilon}{8}\right)^d,$$

where the last inequality is due to $M \geq 12 \left(\frac{12}{\varepsilon}\right)^d \ln\left(\frac{2}{\delta} \left(\frac{8}{\varepsilon}\right)^d\right)$. Hence we arrive at

$$\mathbb{P}(B^c | C^c, E^c) \leq \sum_{c \in \mathcal{C}} \mathbb{P}(B_c^c | C^c, E^c) \leq \left(\frac{8}{\varepsilon}\right)^d \frac{\delta}{2} \left(\frac{\varepsilon}{8}\right)^d = \delta/2,$$

as we wished.

Proof of eq. (21): We can lower bound $\mathbb{P}(X_{c,n} | C^c, E^c)$ as

$$\mathbb{P}(X_{c,n} | C^c, E^c) \geq \mathbb{P}(X_{c,n} | C^c, E^c, \gamma_{c,j_n} \leq \varepsilon/2) \mathbb{P}_{j_n}(\gamma_{c,j_n} \leq \varepsilon/2), \tag{\star}$$

where $j_n \in \mathcal{M}$ is the n^{th} item drawn during MAKE-PARTITION.

Now note that the event $X_{c,n}$ occurs when (a) $\text{SIMILAR}(c, j_n, 0.6\varepsilon, \delta/(4M|\mathcal{C}|))$ returns TRUE, (b) c is chosen uniformly at random among the other items in \mathcal{C} that are also similar j_n .

Conditioning on E^c, C^c , and $\gamma_{c,j_n} \leq \varepsilon/2$ guarantees that $\text{SIMILAR}(c, j_n, 0.6\varepsilon, \delta/(4M|\mathcal{C}|))$ returns TRUE, and hence $\mathbb{P}(X_{c,n} | C^c, E^c, \gamma_{c,j_n} \leq \varepsilon/2) = 1/K$, where K is the number of items in \mathcal{C} for which SIMILAR also returned TRUE. By Proposition D.2 (with $r \leftarrow 0.6\varepsilon$ and $\varepsilon \leftarrow \varepsilon/2$ in the Proposition), we get that $K \leq (\frac{8}{\varepsilon})^d \mu(B(c, \varepsilon/2)) (\frac{5}{4} + \frac{6}{5})^d \leq (\frac{12}{\varepsilon})^d \mu(B(c, \varepsilon/2))$. Now noting that in eq. (★) $\mathbb{P}_{j_n}(\gamma_{c,j_n} \leq \varepsilon/2) = \mu(B(c, \varepsilon/2))$, we arrive at

$$\begin{aligned} \mathbb{P}(X_{c,n} | C^c, E^c) &\geq \mathbb{P}(X_{c,n} | C^c, E^c, \gamma_{c,j_n} \leq \varepsilon/2) \mathbb{P}_{j_n}(\gamma_{c,j_n} \leq \varepsilon/2) \\ &\geq \frac{1}{(\frac{12}{\varepsilon})^d \mu(B(c, \varepsilon/2))} \mu(B(c, \varepsilon/2)) = \left(\frac{\varepsilon}{12}\right)^d, \end{aligned} \quad (22)$$

which proves eq. (21) and hence (A).

Proof of (B): The event B^c happens when for some block P_k there are items $i, j \in P_k$ such that $\gamma_{ij} > 1.2\varepsilon$. Conditioning on C^c , however, this can only happen if $\gamma_{c_k,j} > 0.6 \cdot \varepsilon$ (or $\gamma_{c_k,i} > 0.6 \cdot \varepsilon$) but $\text{SIMILAR}(c_k, j, 0.5\varepsilon, \delta')$ returned SIMILAR nevertheless. Conditioning on E^c , however, that cannot happen and we get $\mathbb{P}(B^c | C^c, E^c) = 0$.

Proof of (C): By Lemma A.1, $\mathbb{P}(E_c, j) \leq \frac{\delta}{4} \frac{1}{(4/\varepsilon)^d} \frac{1}{M}$ and hence

$$\mathbb{P}(E) \leq \sum_{c \in \mathcal{C}} \sum_{j \in \mathcal{M}} \mathbb{P}(E_{i,j}) \leq \frac{\delta}{4} \frac{|\mathcal{C}|}{(4/\varepsilon)^d} \frac{M}{M} \leq \delta/4,$$

where the last inequality is due to $|\mathcal{C}| \leq (4/\varepsilon)^d$, as guaranteed in Lemma A.3.

Proof of (D): This follows directly from Lemma A.3. ■

A.3 Sufficient Exploration

During epoch τ the algorithm uses any given recommendation to be an explore recommendation with probability ε_τ . In the Lemma below, we should that during each epoch there are enough explore recommendations for the procedure MAKE-PARTITION to terminate.

Lemma A.5. *With probability at least $1 - \varepsilon_{\tau+1}$, during the τ^{th} epoch the algorithm has enough explore recommendations for MAKE-PARTITION $(M_{\tau+1}, \varepsilon_{\tau+1}, \varepsilon_{\tau+1})$ to terminate.*

Proof. It suffices to prove the following two facts:

- (A) the number of times *explore* is required for MAKE-PARTITION $(M_{\tau+1}, \varepsilon_{\tau+1}, \varepsilon_{\tau+1})$ to terminate is at most $\frac{1}{2}\varepsilon_\tau D_\tau N$ for each τ , and
- (B) with probability at least $1 - \varepsilon_{\tau+1}$ we have that *explore* will be called at least $\frac{1}{2}\varepsilon_\tau D_\tau$ times.

Proof of (A): Let us denote by $MP(\tau + 1)$ the number of explore calls required for the routine MAKE-PARTITION $(M_{\tau+1}, \varepsilon_{\tau+1}, \varepsilon_{\tau+1})$ to terminate. Then we want to show that $MP(\tau + 1) \leq \frac{1}{2}\varepsilon_\tau D_\tau N$, or equivalently that

$$\frac{2}{\varepsilon} MP(\tau + 1) / D_\tau \leq N. \quad (\star)$$

Note that (which we soon check) MAKE-PARTITION $(M_\tau, \varepsilon_\tau, \varepsilon_\tau)$ makes at most

$$MP(\tau + 1) \leq \left(\frac{8}{\varepsilon_{\tau+1}}\right)^{d+1} 4 \cdot 630 (d + 1)^3 M_{\tau+1} \ln^2 \left(\frac{8}{\varepsilon_{\tau+1}}\right) \ln(M_{\tau+1}) \quad (23)$$

recommendations, and since $M_\tau \triangleq \frac{2^{\max(8, 3.5d)}}{\nu} \frac{(3d+1)}{\varepsilon_\tau^{d+2}} \ln\left(\frac{2}{\varepsilon_\tau}\right)$ and $M_{\tau+1} \leq M_\tau \cdot 2^{d+2}$ we get

$$\frac{4}{\varepsilon_\tau} \frac{1}{M_\tau} MP(\tau+1) \leq \left(\frac{16}{\varepsilon_\tau}\right)^{d+1} 4 \cdot 630 (d+1)^3 2^{d+2} \ln^2\left(\frac{16}{\varepsilon_\tau}\right) \ln\left(\frac{2^{3d+13}}{\nu \varepsilon_\tau^{d+1}} \ln(1/\varepsilon_\tau)\right) \quad (24)$$

$$\leq 2^{5d+12} \cdot 630 (d+1)^3 \left(\frac{1}{\varepsilon_\tau}\right)^{d+2} \ln^2\left(\frac{16}{\varepsilon_\tau}\right) \ln\left(\frac{2^{3d+13}}{\nu \varepsilon_\tau^{d+2}}\right). \quad (25)$$

It is simple to show that $\ln^2\left(\frac{16}{\varepsilon_\tau}\right) \ln\left(\frac{2^{3d+13}}{\nu \varepsilon_\tau^{d+2}}\right) \leq \frac{2^6}{\nu} (2d+11)(d+2) \frac{1}{\varepsilon_\tau^3}$, which we can use to further bound $\frac{4}{\varepsilon_\tau} \frac{1}{M_\tau} MP(\tau+1)$ as

$$\begin{aligned} \frac{4}{\varepsilon_\tau} \frac{1}{M_\tau} MP(\tau+1) &\leq \left(2^{5d+12} \cdot 630 (d+1)^3 \left(\frac{1}{\varepsilon_\tau}\right)^{d+2}\right) \left(\frac{2^6}{\nu} (2d+11)(d+2) \frac{1}{\varepsilon_\tau^3}\right) \\ &= \frac{2^{5d+18}}{\nu} \cdot 630 (2d+11)(d+2)^4 \frac{1}{\varepsilon_\tau^{d+5}}, \end{aligned}$$

and since $\varepsilon_\tau \geq \varepsilon_N = \left(\frac{2^{5d+18}}{\nu} \cdot 630 (2d+11)(d+2)^4 \frac{1}{N}\right)^{\frac{1}{d+5}}$, we get that eq. (★) is satisfied and we are done.

Proof of (B): *explore* is called with probability ε_τ at each of the $D_\tau N$ recommendations of epoch τ . Hence, all we need to show is that $\mathbb{P}(\text{Bin}(D_\tau N, \varepsilon_\tau) \leq \frac{1}{2} \varepsilon_\tau D_\tau N)$ is at most $\varepsilon_{\tau+1}$. This follows from the Chernoff bound:

$$\begin{aligned} \mathbb{P}\left(\text{Bin}(D_\tau N, \varepsilon_\tau) < \frac{1}{2} \varepsilon_\tau D_\tau N\right) &= \mathbb{P}\left(\text{Bin}(D_\tau N, \varepsilon_\tau) < (1-0.5) \mathbb{E}[\text{Bin}(D_\tau N, \varepsilon_\tau)]\right) \\ &\leq \exp\left(-\frac{0.5^2}{2+0.5^2} \mathbb{E}[\text{Bin}(D_\tau N, \varepsilon_\tau)]\right) = \exp\left(-\frac{1}{9} \varepsilon_\tau D_\tau N\right) \leq \varepsilon_{\tau+1}, \end{aligned}$$

where the second to last inequality follows from $D_\tau \geq \frac{9}{\varepsilon_\tau} \ln\left(\frac{1}{\varepsilon_{\tau+1}}\right)$. ■

B Quick Recommendations Lemma

In Section 3 we described that the algorithm, which starts recommending to a user as soon as it knows of one item that the user likes. Below we show that indeed shortly after the beginning of the epoch the slope of the regret is small.

Lemma B.1 (Quick Recommendations Lemma). *For $\tau \geq 1$ let $\mathcal{R}^{(\tau)}(T) = \frac{1}{N} \sum_{t=T_\tau}^{T_\tau+TN} \frac{1}{2} (1 - L_{U_t, I_t})$ denote the number of bad recommendations made to users during the first TN recommendations of epoch τ . Then we have*

$$\mathbb{E}\left[\mathcal{R}^{(\tau)}(T)\right] \leq \frac{148\varepsilon_\tau}{\nu} T \quad (26)$$

whenever $T \in [T_{\min, \tau}, D_\tau]$ and where $T_{\min, \tau} \triangleq \frac{12}{\varepsilon_\tau} \ln\left(\frac{1}{\varepsilon_\tau}\right)$. For $T < T_{\min, \tau}$, we trivially have $\mathbb{E}\left[\mathcal{R}^{(\tau)}(T)\right] \leq T$.

Proof. Let $\mathcal{R}'^{(\tau)}(T)$ denote the number of bad recommendations made to users during the first TN exploit recommendations of epoch τ . Then, since the expected number of explore recommendations by time TN of epoch τ is $\varepsilon_\tau TN$, we get that

$$\mathbb{E}\left[\mathcal{R}^{(\tau)}(T)\right] \leq \varepsilon_\tau T + \mathbb{E}\left[\mathcal{R}'^{(\tau)}(T)\right]. \quad (27)$$

Furthermore, as described in the algorithm, during the epoch $\tau - 1$ the algorithm spends a small fraction of the recommendations, in the explore part, to create a partition $\{P_k\}$ (which we call

$\{P_k^{(\tau)}\}$ in the pseudocode) of M_τ random items to be exploited during epoch τ . Let \mathcal{E}_τ be the event that the partition $\{P_k^{(\tau)}\}$ to be used during epoch τ satisfies the conditions specified in Lemma A.4 (with $M = M_\tau \varepsilon = \varepsilon_\tau$, $\delta = \varepsilon_\tau$). Then we get

$$\mathbb{E} \left[\mathcal{R}^{(\tau)}(T) \right] \leq \varepsilon_\tau T + \mathbb{E} \left[\mathcal{R}'^{(\tau)}(T) \right] \leq 2\varepsilon_\tau T + \mathbb{E} \left[\mathcal{R}'^{(\tau)}(T) \mid \mathcal{E}_\tau \right], \quad (28)$$

where the last inequality is due to Lemma A.4, which guarantees that $\mathbb{P}(\mathcal{E}_\tau) \leq \varepsilon_\tau$.

For the remaining of the proof, we will show that $\mathbb{E} \left[\mathcal{R}'(T)^{(\tau)}(T) \right] \leq \frac{45}{\nu} \varepsilon_\tau T$. We will do so by first rewriting in terms of the number of bad exploit recommendations to each user

$$\mathbb{E} \left[\mathcal{R}'^{(\tau)} \mid \mathcal{E}_\tau \right] \leq \frac{1}{N} \mathbb{E} \left[\sum_u \mathcal{R}'^{(\tau)}(T) \mid \mathcal{E}_\tau \right],$$

where $\mathcal{R}'_u^{(\tau)}(T)$ is the number of bad recommendations made to user u during the first TN exploit recommendations of epoch τ . We will now bound the latter term by conditioning on a nice property of users (which we will characterize by the event $g_{u,T}$), and showing that this property holds for most users. Let $g_{u,T}$ be the event that user u has tried at most $16 \frac{T}{D_\tau} P^{(\tau)}$ blocks during the first TN recommendations of epoch τ (we omit τ in the notation of $g_{u,T}$ since it is clear from the context here). Here we use notation $P^{(\tau)} = |\{P_k^{(\tau)}\}|$ to denote the total number of blocks in the partition for epoch τ . Then we get

$$\frac{1}{N} \mathbb{E} \left[\sum_u \mathcal{R}'_u^{(\tau)}(T) \mid \mathcal{E}_\tau \right] \leq \frac{1}{N} \sum_u \mathbb{E} \left[\mathcal{R}'_u^{(\tau)}(T) \mid \mathcal{E}_\tau, g_{u,T} \right] + T \frac{1}{N} \sum_u \mathbb{P} (g_{u,T}^c \mid \mathcal{E}_\tau).$$

We dedicate Lemma B.3 to showing that $\frac{1}{N} \sum_u \mathbb{P} (g_{u,T}^c \mid \mathcal{E}_\tau) \leq \frac{42}{\nu} \varepsilon_\tau$. Hence, it suffices to show that for each $T > T_{\min, \tau}$ we have

$$\frac{1}{N} \sum_u \mathbb{E} \left[\mathcal{R}'_u^{(\tau)}(T) \mid \mathcal{E}_\tau, g_{u,T} \right] \leq \frac{104}{\nu} \varepsilon_\tau T, \quad (\star)$$

which we prove now. We will first rewrite the regret by summing over the number of bad recommendations due to each of the blocks as

$$\sum_u \mathbb{E} \left[\mathcal{R}'_u^{(\tau)}(T) \mid \mathcal{E}_\tau, g_{u,T} \right] = \sum_u \mathbb{E} \left[\sum_k W_{u,k,T} \mid \mathcal{E}_\tau, g_{u,T} \right],$$

where $W_{u,k,T}$ is the random variable denoting the number of bad exploit recommendations to user u from block P_k among the first TN exploit recommendations of epoch τ . We can further rewrite this as

$$\frac{1}{N} \sum_u \mathbb{E} \left[\sum_k W_{u,k,T} \mid \mathcal{E}_\tau, g_{u,T} \right] \leq \frac{1}{N} \sum_u \sum_k \mathbb{E} [W_{u,k,T} \mid \mathcal{E}_\tau, g_{u,T}, s_{u,k,T}] \mathbb{P} (s_{u,k,T} \mid \mathcal{E}_\tau, g_{u,T}), \quad (29)$$

where $s_{u,k,T}$ denotes the event that by time T user u has sampled an item from block P_k . Note that the reason why the natural term $\mathbb{E} \left[W_{u,k,T} \mid \mathcal{E}_\tau, g_{u,T}, s_{u,k,T}^c \right] \mathbb{P} (s_{u,k,T}^c \mid \mathcal{E}_\tau, g_{u,T})$ is absent from the expression above is because $\mathbb{E} \left[W_{u,k,T} \mid \mathcal{E}_\tau, g_{u,T}, s_{u,k,T}^c \right] = 0$ since the user hasn't sampled an item from the block.

Now note that by conditioning on $g_{u,T}$, we know that user u has sampled at most $16 \frac{T}{D_\tau} P^{(\tau)}$ blocks. Now given $g_{u,T}$ as well as \mathcal{E}_τ , the indices of the sampled blocks are not revealed. Let \tilde{K} be a random

variable that selects one of the indices of the blocks uniformly at random. Then, it follows that with respect to randomness in K ,

$$\mathbb{P}(s_{u,K,T} | \mathcal{E}_\tau, g_{u,T}) \leq \frac{16 \frac{T}{D_\tau} P^{(\tau)}}{P^{(\tau)}} = 16 \cdot \frac{T}{D_\tau}. \quad (30)$$

We can re-write (29) in this notation and apply the above discussed bound to obtain

$$\begin{aligned} \frac{1}{N} \sum_u \mathbb{E} \left[\sum_k W_{u,k,T} | \mathcal{E}_\tau, g_{u,T} \right] &\leq \frac{P^{(\tau)}}{N} \sum_u \mathbb{E} [W_{u,K,T} | \mathcal{E}_\tau, g_{u,T}, s_{u,K,T}] \mathbb{P}(s_{u,K,T} | \mathcal{E}_\tau, g_{u,T}) \\ &\leq \frac{P^{(\tau)}}{N} \sum_u \mathbb{E} [W_{u,K,T} | \mathcal{E}_\tau, g_{u,T}, s_{u,K,T}] \frac{16T}{D_\tau} \\ &= \frac{16T}{ND_\tau} \sum_u \sum_k \mathbb{E} [W_{u,k,T} | \mathcal{E}_\tau, g_{u,T}, s_{u,k,T}]. \end{aligned} \quad (31)$$

The right hand side above can be bounded as

$$\begin{aligned} &\stackrel{\text{Lemma B.2}}{\leq} \frac{16T}{ND_\tau} \left(\sum_k (1 + 1.2\varepsilon_\tau |P_k^{(\tau)}|) N \right) \\ &\stackrel{\text{Lemma A.4}}{\leq} 52\varepsilon_\tau T \frac{M_\tau}{D_\tau} \stackrel{D_\tau = \frac{\nu}{2} M_\tau}{=} \frac{104}{\nu} \varepsilon_\tau T. \end{aligned} \quad (32)$$

To see the above two inequalities, consider the following. The first inequality follows from Lemma B.2 by realizing that each block, $P_k^{(\tau)}$ corresponds to collection of items such that for any $i, j \in P_k^{(\tau)}$, we have $\gamma_{ij} < 1.2\varepsilon_\tau$. The second inequality can be argued as: from Lemma A.4, $1/2\varepsilon_\tau \leq |P_k^{(\tau)}| \leq 1/\varepsilon_\tau$ and hence

$$\begin{aligned} \sum_k (1 + 1.2\varepsilon_\tau |P_k^{(\tau)}|) &\leq \sum_k (3.2\varepsilon_\tau |P_k^{(\tau)}|) \\ &\leq 3.2\varepsilon_\tau \left(\sum_k |P_k^{(\tau)}| \right) \\ &= 3.2\varepsilon_\tau M_\tau. \end{aligned}$$

This, along with simple calculation, completes the proof of eq. (★). ■

The lemma below was used in Lemma B.1. Informally, it says that our recommendation policy, which recommends the whole block to a user after the user likes an item in the block, succeeds in finding most likable items to recommend and in not recommending many bad items.

Lemma B.2 (Partition Lemma). *Let P_k be a set of items such that for each $i, j \in P_k$ we have $\gamma_{ij} < \varepsilon$, and consider the usual recommendation policy that ITEM-ITEM-CF uses during its “exploit” steps (where when user u samples a random item $i \in_R P_k$, only if u likes i will u be recommended the remaining items). Let $s_{u,k}$ be the event that user u has sampled an item from P_k , let $W_{u,k}$ (W for wrong) denote the number of wrong recommendations made to u from P_k , and let $A_{u,k}$ (A for absent) denote the number of items in P_k that u likes that are not recommended to u . Then we get*

$$\sum_u \mathbb{E} [A_{u,k} + W_{u,k} | s_{u,k}] \leq (1 + \varepsilon |P_k|) N.$$

Proof. For each block P_k and user u , and let $\ell_{u,k} = |\{i \in P_k \mid L_{u,i} = +1\}|$ denote the number of items in P_k that u likes. Note that $\mathbb{E} [A_{u,k} | s_{u,k}] = \ell_{u,k} \cdot \frac{(|P_k| - \ell_{u,k})}{|P_k|}$ and $\mathbb{E} [W_{u,k} | s_{u,k}] =$

$(|P_k| - \ell_{u,k}) \cdot \frac{\ell_{u,k}}{|P_k|} + 1 \cdot \frac{(|P_k| - \ell_{u,k})}{|P_k|}$. This is because with probability $\ell_{u,k}/|P_k|$ user u will sample an item from P_k that u likes and will then be recommended $(|P_k| - \ell_{u,k})$ bad items, and with probability $(|P_k| - \ell_{u,k})/|P_k|$ the first item recommended to u is bad. Likewise, with probability $(|P_k| - \ell_{u,k})/|P_k|$ the user will sample an item that the user dislikes, and then fail to be recommended $\ell_{u,k}$ items that the user likes. Hence we have that

$$\mathbb{E} \left[\sum_u A_{u,k} + W_{u,k} \mid s_{u,k} \right] = \underbrace{\sum_u \frac{(|P_k| - \ell_{u,k})}{|P_k|}}_{\leq N} + 2 \sum_u \frac{\ell_{u,k} (|P_k| - \ell_{u,k})}{|P_k|}. \quad (33)$$

Now,

$$\begin{aligned} 2 \sum_u \frac{\ell_{u,k} (|P_k| - \ell_{u,k})}{|P_k|} &\stackrel{\text{by definition}}{=} \frac{2}{|P_k|} \sum_u \sum_{i,j \in P_k} \mathbf{1}_{L_{u,i} \neq L_{u,j}} \\ &= \frac{2}{|P_k|} \sum_{i,j \in P_k} \underbrace{\sum_u \mathbf{1}_{L_{u,i} \neq L_{u,j}}}_{=\gamma_{ij} \cdot N} \\ &\leq \underbrace{\sum_{\gamma_{ij} < \varepsilon}}_{\leq} \frac{2}{|P_k|} \sum_{i,j \in P_k} \varepsilon N \\ &\leq \frac{2}{|P_k|} \binom{|P_k|}{2} \varepsilon N \leq \varepsilon |P_k| N. \end{aligned} \quad (34)$$

Putting it all together we get

$$\mathbb{E} \left[\sum_u A_{u,k} + W_{u,k} \mid s_{u,k} \right] = \underbrace{\sum_u \frac{(|P_k| - \ell_{u,k})}{|P_k|}}_{\leq N} + \underbrace{2 \sum_u \frac{\ell_{u,k} (|P_k| - \ell_{u,k})}{|P_k|}}_{\varepsilon |P_k| N} = (\varepsilon |P_k| + 1) N. \quad (35)$$

The lemma below was also needed in the proof of Lemma B.1.

Lemma B.3 (Auxiliary Claim). *Consider an arbitrary epoch τ , and let $g_{u,T}$ be the event that by the $(TN)^{\text{th}}$ exploit recommendation of epoch τ user u has tried at most $16 \frac{T}{D_\tau} |P_k^{(\tau)}|$ blocks from the partition $\{P_k^{(\tau)}\}$ constructed during the MAKE-PARTITION($M_\tau, \varepsilon_\tau, \varepsilon_\tau$) of the previous epoch, and let \mathcal{E}_τ be the event that $\{P_k^{(\tau)}\}$ satisfies the conditions specified in Lemma A.4. Then*

$$\frac{1}{N} \sum_u \mathbb{P}(g_{u,T}^c \mid \mathcal{E}_\tau) \leq \frac{42}{\nu} \varepsilon_\tau$$

holds for any $T \in (\frac{12}{\varepsilon_\tau} \ln(1/\varepsilon_\tau), D_\tau]$.

Proof. Let us consider few definitions:

1. Let $N_{u,T}$ be the event that by the TN^{th} exploit recommendation user u has been recommended at most $1.1T$ items, and that u likes at least $0.9\nu M_\tau$ among the items in $\{P_k^{(\tau)}\}$.
2. Let $H_{u,T}$ be the event that by the TN^{th} exploit recommendation there are still at least $\frac{\nu}{5} M_\tau$ items liked by u in blocks that haven't been sampled by u .

Then we get

$$\mathbb{P}(g_{u,T}^c | \mathcal{E}_\tau) \leq \mathbb{P}(g_{u,T}^c | H_{u,T}, N_{u,T}, \mathcal{E}_\tau) + \mathbb{P}(H_{u,T}^c | N_{u,T}, \mathcal{E}_\tau) + \mathbb{P}(N_{u,T}^c | \mathcal{E}_\tau).$$

We will show that for $T \in (\frac{12}{\varepsilon_\tau} \ln(1/\varepsilon_\tau), D_\tau]$

- (A) $\mathbb{P}(g_{u,T}^c | H_{u,T}, N_{u,T}, \mathcal{E}_\tau) \leq \varepsilon_\tau$
- (B) $\frac{1}{N} \sum_u \mathbb{P}(H_{u,T}^c | N_{u,T}, \mathcal{E}_\tau) \leq \frac{40}{\nu} \varepsilon_\tau$
- (C) $\mathbb{P}(N_{u,T}^c | \mathcal{E}_\tau) \leq \varepsilon_\tau,$

from which the lemma follows.

Proof of (A): Note that by conditioning on $N_{u,T}$ we know that there were at least $0.2\nu M_\tau$ items likable to u for each $t \leq T$. Now let $\ell_{u,k}$ denote the event that u likes the first item sampled from block $P_k^{(\tau)}$. Then we have that

$$\left\{ g_{u,T}^c, \mathcal{E}_\tau, N_{u,T}, H_{u,T} \right\} \subseteq \left\{ \sum_{n=1}^{16 \frac{T}{D_\tau} |\{P_k^{(\tau)}\}|} |P_{k_n}| \cdot \mathbf{1}_{\ell_{u,k_n}} \leq 1.1T, \mathcal{E}_\tau, N_{u,T}, H_{u,T} \right\}, \quad (36)$$

where P_{k_n} is the n^{th} block sampled by u . This in turn gives us

$$\mathbb{P}\left(g_{u,T}^c | \mathcal{E}_\tau, N_{u,T}, H_{u,T}\right) \leq \mathbb{P}\left(\sum_{n=1}^{16 \frac{T}{D_\tau} |\{P_k^{(\tau)}\}|} |P_{k_n}| \cdot \mathbf{1}_{\ell_{u,k_n}} \leq 1.1T | \mathcal{E}_\tau, N_{u,T}, H_{u,T}\right), \quad (37)$$

and we will now prove that the latter is at most ε_τ .

First, note that by conditioning on \mathcal{E}_τ , we are guaranteed that $|P_{k_n}| \geq 1/2\varepsilon_\tau$ by Lemma A.4. Hence

$$\begin{aligned} & \mathbb{P}\left(\sum_{n=1}^{16 \frac{T}{D_\tau} |\{P_k^{(\tau)}\}|} |P_{k_n}| \cdot \mathbf{1}_{\ell_{u,k_n}} \leq 1.1T | \mathcal{E}_\tau, H_{u,T}, N_{u,T}\right) \\ & \leq \mathbb{P}\left(\sum_{n=1}^{16 \frac{T}{D_\tau} |\{P_k^{(\tau)}\}|} \mathbf{1}_{\ell_{u,k_n}} \leq 2.2T\varepsilon_\tau | \mathcal{E}_\tau, H_{u,T}, N_{u,T}\right). \end{aligned}$$

We will now show following two claims, which in turn implies (A) in light of the above discussion.

- (A₁) For each n , $\mathbb{P}(\ell_{u,k_n} | \mathcal{E}_\tau, H_{u,T}, N_{u,T})$ stochastically dominates a Bernoulli random variable with parameter 0.1ν .
- (A₂) Let $\{X_n\}$ be a set of independent Bernoulli random variables with parameter 0.1ν . Then

$$\mathbb{P}\left(\sum_{n=1}^{16 \frac{T}{D_\tau} |\{P_k^{(\tau)}\}|} X_n < 2.2T\varepsilon_\tau\right) \leq \varepsilon_\tau \quad (38)$$

Proof of (A₁): Since we are conditioning on $H_{u,T}$, there are still $0.2\nu M_\tau$ liked items in unsampled blocks, and since we are conditioning on \mathcal{E}_τ , there are at most $2M_\tau\varepsilon_\tau$ blocks and each of size at least $1/2\varepsilon_\tau$ and at most $1/\varepsilon_\tau$. It can be easily argued that the setting in which sampling a random block yields a likable item is least likely is when all likable items are in the largest blocks. By \mathcal{E}_τ , we can “fit” $0.2\nu M_\tau$ in at least $0.2\nu M_\tau / (1/\varepsilon_\tau) = 0.2\varepsilon_\tau \nu M_\tau$ blocks. Therefore,

$$\mathbb{P}(\ell_{u,k_n} | \mathcal{E}_\tau, H_{u,T}, N_{u,T}) \geq \frac{0.2\varepsilon_\tau \nu M_\tau}{2M_\tau \varepsilon_\tau} = 0.1\nu. \quad (39)$$

Proof of (A₂): Let $\{X_n\}$ be a set of independent Bernoulli random variables with parameter 0.1ν . Then the sum of $16 \frac{T}{D_\tau} |\{P_k^{(\tau)}\}|$ such i.i.d. random variables have average equal to

$$\begin{aligned} 1.6\nu \frac{T}{D_\tau} |\{P_k^{(\tau)}\}| &\geq 1.6\nu \frac{T}{D_\tau} M_\tau \varepsilon_\tau \\ &= 3.2T\varepsilon_\tau, \end{aligned} \quad (40)$$

where we have used the fact that since by \mathcal{E}_τ each block is of size at most $1/\varepsilon_\tau$ and number of items M_τ , we have at least $M_\tau \varepsilon_\tau$ blocks and $D_\tau = M_\tau \nu / 2$. Then

$$\mathbb{P} \left(\sum_{n=1}^{16 \frac{T}{D_\tau} |\{P_k^{(\tau)}\}|} X_n < 2.2T\varepsilon_\tau \right) \leq \mathbb{P} \left(\sum_{n=1}^{16 \frac{T}{D_\tau} |\{P_k^{(\tau)}\}|} X_n < (1 - 0.25) \mathbb{E} \left[\sum_{n=1}^{16 \frac{T}{D_\tau} |\{P_k^{(\tau)}\}|} X_n \right] \right), \quad (41)$$

where the inequality is due to (40). Now by the Chernoff bound we get

$$\exp \left(- \frac{0.25^2}{2 + 0.25} \mathbb{E} \left[\sum_{n=1}^{16 \frac{T}{D_\tau} |\{P_k^{(\tau)}\}|} X_n \right] \right) \leq \exp \left(- \frac{3.2T\varepsilon_\tau}{36} \right). \quad (42)$$

Therefore, if $T > \frac{12}{\varepsilon_\tau} \ln(1/\varepsilon_\tau)$, then the right hand side is less than ε_τ , as desired.

Proof of (B): First, for each u it is clear that $\mathbb{P}(H_{u,T}^c | N_{u,T}) \leq \mathbb{P}(H_{u,D_\tau}^c | N_{u,D_\tau})$ holds, since by end of the epoch the user will have explored the most (recall $T \leq D_\tau$). Recall that under event N_{u,D_τ} , user u has been recommended at most $1.1D_\tau$ items and that at least $0.9\nu M_\tau$ items in the partition are likable to u . For H_{u,D_τ}^c to happen, that is, for there to be at most $0.2\nu M_\tau$ at the end of $D_\tau N$ exploit recommendation (overall), it must be that at least

$$0.9\nu M_\tau - 0.2\nu M_\tau - 1.1D_\tau = 0.7\nu M_\tau - 0.55\nu M_\tau = 0.15\nu M_\tau. \quad (43)$$

many items liked by user u are “wasted”. Using notation from the proof of Lemma B.2, formally it can be written as

$$\sum_k A_{k,u} \mathbf{1}_{s_{u,k,D_\tau}} \geq 0.15\nu M_\tau. \quad (44)$$

By Markov’s inequality we get that

$$\begin{aligned} \mathbb{P} \left(\sum_k A_{k,u} \mathbf{1}_{s_{u,k,D_\tau}} \geq 0.15\nu M_\tau \mid \mathcal{E}_\tau, N_{u,T} \right) &\leq \mathbb{P} \left(\sum_k A_{k,u} \geq 0.15\nu M_\tau \mid \mathcal{E}_\tau, N_{u,T} \right) \\ &\stackrel{(a)}{\leq} \frac{\mathbb{P} \left(\sum_k A_{k,u} \geq 0.15\nu M_\tau \mid \mathcal{E}_\tau \right)}{\mathbb{P} \left(N_{u,T} \right)} \\ &\stackrel{(b)}{\leq} \frac{\mathbb{P} \left(\sum_k A_{k,u} \geq 0.15\nu M_\tau \mid \mathcal{E}_\tau \right)}{1 - \varepsilon_\tau} \\ &\leq \frac{2\mathbb{E} [\sum_k A_{k,u} \mid \mathcal{E}_\tau]}{0.15\nu M_\tau}, \end{aligned} \quad (45)$$

where inequality (a) uses the fact that $P(A|B) \leq P(A)/P(B)$; inequality (b) uses the fact (C) (we note that (B) is NOT used to prove (C) and hence there is no circularity of the argument); and last inequality uses the fact that $\varepsilon_\tau \leq 1/2$ for all τ . We note that effectively, we have assumed away that s_{u,k,D_τ} has happened for all k for the given user u . Therefore, we will utilize Lemma B.2 to bound

the right hand side as follows:

$$\begin{aligned}
\frac{1}{N} \sum_u \mathbb{P} \left(\sum_k A_{k,u} \mathbf{1}_{s_{u,k}, D_\tau} \geq 0.35\nu M_\tau \mid \mathcal{E}_\tau, N_{u,T} \right) &\leq \frac{1}{N} \sum_u \frac{2\mathbb{E}[\sum_k A_{k,u} \mid \mathcal{E}_\tau]}{0.15\nu M_\tau} \\
&\leq \frac{2}{0.15\nu M_\tau N} \left(\sum_k (1 + \varepsilon_\tau |P_k^T|) N \right) \\
&\stackrel{(a)}{\leq} \frac{2}{0.15\nu M_\tau} \left(\sum_k 3\varepsilon_\tau |P_k^T| \right) \\
&= \frac{40\varepsilon_\tau}{\nu M_\tau} \left(\sum_k |P_k^T| \right) \\
&= \frac{40\varepsilon_\tau}{\nu} \tag{46}
\end{aligned}$$

where (a) uses the fact that under \mathcal{E}_τ , $|P_k^T| \geq 1/2\varepsilon_\tau$ for all k , and we have used the fact that all partitions sum up to M_τ .

Proof of (C): The event $N_{u,T}^c$ happens whenever a user u has been recommended more than $1.1D_\tau = 0.55\nu M_\tau$ times by time T , or when the users likes less than $0.95\nu M_\tau$. The probability that user u has been recommended more than $0.55\nu M_\tau$ items by time T is greatest at $T = D_\tau$, and, by Chernoff bound, is

$$\mathbb{P} \left(\left| \text{Bin} \left(ND_\tau, \frac{1}{N} \right) \geq 1.1D_\tau \right) \leq \exp \left(-\frac{.1^2}{2 + 0.1} D_\tau \right), \tag{47}$$

which, since $D_\tau > 210 \ln(\frac{2}{\varepsilon_\tau})$, is at most $\varepsilon_\tau/2$.

The probability that a user u likes less than $0.9\nu M_\tau$ among the M_τ items can be also bounded using a Chernoff bound:

$$\mathbb{P} (\text{Bin} (M_\tau, \nu) < 0.9\nu M_\tau) \leq \exp \left(-\frac{.1^2}{2 + 0.1} \frac{1}{2} M_\tau \nu \right) \leq \varepsilon_\tau/2, \tag{48}$$

where the last inequality is due to $M_\tau \geq 210 \ln(\frac{2}{\varepsilon_\tau})$. ■

C Proof of Main Results

In the previous section we proved Lemma B.1, which states that shortly after the beginning of each epoch, the expected regret of the algorithm becomes small. This allows us to prove our main results below.

Theorem 5.2 (Regret Upper Bound of ITEM-ITEM-CF). *Suppose assumptions A1 and A2 are satisfied. Then ITEM-ITEM-CF achieves expected regret*

$$\mathbb{E} [\mathcal{R}(T)] \leq \begin{cases} T_{\min} + \alpha(\nu, d) \cdot (T - T_{\min})^{\frac{d+1}{d+2}} \log_2(T - T_{\min}) & T_{\min} < T \leq T_{\max} \\ \beta + \varepsilon_N (T - T_{\max}) & T > T_{\max} \end{cases}, \tag{2}$$

where $T_{\min} = \tilde{\mathcal{O}}(\frac{1}{\nu}) + \frac{f(d,\nu)}{N}$, $T_{\max} = g(\nu, d) N^{\frac{d+2}{d+5}}$, $\varepsilon_{N,d,\nu} = h(d, \nu) (\frac{1}{N})^{\frac{1}{d+5}}$, $\beta = T_{\min} + \alpha(\nu, d) \cdot (T_{\max} - T_{\min})^{\frac{d+1}{d+2}} \log_2(T_{\max} - T_{\min})$.

Proof. Recall that during the beginning of ITEM-ITEM-CF it runs the routine MAKE-PARTITION($M_1, \varepsilon_1, \varepsilon_1$). This consumes at most

$$MP(1) \triangleq \left(\frac{8}{\varepsilon_1} \right)^{d+1} 4 \cdot 630 (d+1)^3 M_1 \ln^2 \left(\frac{8}{\varepsilon_1} \right) \ln(M_1) \tag{49}$$

recommendations (by Lemma A.4), and hence finishes in at most $T_{MP} \triangleq MP(1)/N$ time steps. For this initial exploratory period $T \leq T_{MP}$ we will bound the regret with the trivial bound $\mathcal{R}(T) \leq T$.

Let us now deal with the regime between T_{min} and T_{max} . Recall that the target ε_τ used in the τ^{th} epoch is decreasing as $\frac{C}{2^\tau}$, until it plateaus at ε_N when $\frac{C}{2^\tau} \leq \varepsilon_N$, where $C = \frac{\nu}{148 \cdot 20}$. Hence

$$\tau^* \triangleq \lceil \log_2 \frac{C}{\varepsilon_N} \rceil \quad (50)$$

is the first epoch in which ε_N is used. For a function g defined later, we will show that

$$T_{MP(1)} + \sum_{\tau'=1}^{\tau^*-1} D_{\tau'} \geq g(\nu, d) N^{\frac{d}{d+5}} \triangleq T_{max}. \quad (51)$$

Now since $\varepsilon_N = \left(\frac{2^{5d+18}}{\nu} \cdot 630(2d+11)(d+2)^4 \frac{1}{N} \right)^{\frac{1}{d+5}}$, we get that

$$\tau^* \geq \frac{1}{d+5} \log_2 \left(\left(\frac{\nu}{148 \cdot 20} \right)^{d+5} \frac{\nu}{630(2d+11)(d+2)^4} \frac{1}{2^{5d+18}} \cdot N \right). \quad (52)$$

Also,

$$T_{MP(1)} + \sum_{\tau=1}^{\tau^*-1} D_\tau \geq \sum_{\tau=1}^{\tau^*-1} D_\tau = \sum_{\tau=1}^{\tau^*-1} \frac{\nu}{2} M_\tau, \quad (53)$$

where we used the fact that $D_\tau = \frac{\nu}{2} M_\tau$. Recall $M_\tau \triangleq C_M \frac{1}{\varepsilon_\tau^{d+2}} \ln(\frac{2}{\varepsilon_\tau})$, where $C_M = \frac{2^{\max(3.5d, 8)}}{\nu} (3d+1)$, and for $\tau \leq \tau^*$ we have $\varepsilon_\tau = C/2^\tau$, where $C = \nu/(148 \cdot 20)$. Then we get

$$\begin{aligned} T_{MP(1)} + \sum_{\tau=1}^{\tau^*-1} D_\tau &\geq \frac{\nu}{2} C_M (1/C)^{d+2} \sum_{\tau=1}^{\tau^*-1} 2^{\tau(d+2)} \\ &\geq \frac{\nu}{4} C_M (1/C)^{d+2} 2^{\tau^*(d+2)} \\ &\geq \frac{\nu}{4} C_M (1/C)^{d+2} \underbrace{\left(\frac{\nu}{630(2d+11)(d+2)^4} \frac{1}{2^{5d+18}} \right)^{\frac{d+2}{d+5}}}_{g(\nu, d)} \cdot N^{\frac{d+2}{d+5}} \triangleq T_{max}, \end{aligned} \quad (54)$$

as wished. Hence, between T_{min} and T_{max} the target ε_τ for the epochs is indeed halving for each subsequent epoch. Let $\tau(T)$ be the epoch of time T . Then, by Lemma B.1, for $T \in [T_{min}, T_{max}]$, where $T_{min} = T_{MP} + T_{min,1}$, the expected regret satisfies

$$\mathcal{R}(T) - T_{MP} \leq \frac{148}{\nu} \sum_{\tau=1}^{\tau(T)} \varepsilon_\tau D_\tau, \quad (55)$$

which we can further bound as

$$\begin{aligned} \mathcal{R}(T) - T_{MP} &\leq \frac{C_M}{2} \log_2 \left(\frac{2^{\tau(T)}}{2C} \right) \sum_{\tau=1}^{\tau(T)} 2^{\tau(d+1)} \\ &\leq \frac{C_M}{2} \log_2 \left(\frac{2^{\tau(T)}}{2C} \right) 2^{2(\tau(T)+1)(d+1)}. \end{aligned} \quad (56)$$

Now, since for $T > T_{min}$ the epoch $\tau(T)$ is at most $1 + \frac{1}{d+2} \log_2 \left(\frac{T - T_{MP}}{C_M} \frac{1}{\log(2/C)} \right)$, we get that

$$\mathcal{R}(T) \leq T_{MP} + \frac{C_M}{2} \log_2 \left(\frac{1}{C(d+2)} \frac{1}{\log(2/C)} \frac{T - T_{MP}}{C_M} \right) 2^{2(d+1)} 2^{2(\tau(T)+1)(d+1)} \quad (57)$$

$$\begin{aligned}
&\leq T_{MP} + \underbrace{\frac{C_M}{2} \log_2 \left(\frac{1}{C(d+2)} \frac{1}{\log(2/C)} \frac{1}{C_M} \right)}_{\triangleq C'} 2^{4(d+1)} \log_2 (T - T_{MP}) 2^{\frac{d+1}{d+2} \log_2 \left(\frac{T - T_{MP}}{C_M \log(2/C)} \right)} \\
&\leq T_{MP} + \underbrace{C' \left(\frac{1}{C_M \log(2/C)} \right)^{\frac{d+1}{d+2}}}_{\triangleq \alpha(\nu, d)} (T - T_{MP})^{\frac{d+1}{d+2}} \log_2 (T - T_{MP}),
\end{aligned}$$

as we wished, which completes the proof of the sublinear regret regime.

The case $T > T_{max}$ now follows. Recall that by Lemma B.1 we get

$$\mathcal{R}(T) \leq T_{MP} + \frac{148}{\nu} \sum_{\tau=1}^{\tau(T)} \varepsilon_\tau D_\tau, \quad (58)$$

which we can in turn split between before T_{max} and after T_{max} as

$$\mathcal{R}(T) \leq T_{MP} + \frac{148}{\nu} \sum_{\tau=1}^{\tau^*-1} \varepsilon_\tau D_\tau + \frac{148}{\nu} \sum_{\tau=\tau^*}^{\tau(T)} \varepsilon_\tau D_\tau \quad (59)$$

$$\leq \underbrace{T_{MP} + \alpha(\nu, d) T_{max}^{\frac{d+1}{d+2}} \log_2 (T_{max})}_{\triangleq \beta} + \varepsilon_N (T - T_{max}), \quad (60)$$

as claimed, and where the last inequality is due to the sublinear regime proved above. \blacksquare

We are now ready to bound the cold-start time of ITEM-ITEM-CF. Recall that cold-start time of a recommendation algorithm \mathcal{A} is defined as the least $T + \Gamma$ such that for all $\Delta > \Gamma$ we have $\mathbb{E} [\mathcal{R}^{(\mathcal{A})}(T + \Delta) - \mathcal{R}^{(\mathcal{A})}(T)] \leq 0.1\Delta$.

Theorem 5.1 (Cold-Start Performance). *Suppose assumptions A1 and A2 are satisfied. Then the algorithm ITEM-ITEM-CF has cold-start time $T_{\text{cold-start}} = \frac{f(\nu, d)}{N} + \tilde{\mathcal{O}}(1/\nu)$.*

Proof. First recall the usual definitions: $D_\tau = \frac{\nu}{2} M_\tau$, $T_\tau = T_{MP} + \sum_{\tau' < \tau} D_{\tau'}$, and $T_{MP} = f(\nu, d)/N$, where $f(\nu, d)$ is the number of recommendations required for the initial MAKE-PARTITION call (as stated in Lemma A.4), and $T_{min, \tau} = \frac{12}{\varepsilon_\tau} \ln \left(\frac{1}{\varepsilon_\tau} \right)$ (as stated in Lemma B.1). We will show the bound in the definition of cold-start time with $T = T_{MP}$ and $\Gamma = T_{min, 1}$, which implies $T_{\text{cold-start}} = T_{MP} + T_{min, 1}$.

To complete the proof, we shall establish the following two properties:

- (i) For any $\Delta > 0$, $\mathbb{E} [\mathcal{R}(T_{MP} + T_{min, 1} + \Delta) - \mathcal{R}(T_{MP})] \leq 0.1(T_{min, 1} + \Delta)$, for $T_{MP} + T_{min, 1} + \Delta \leq T_2$. This condition says that the desired property holds for times involving the first epoch, and
- (ii) $\mathbb{E} [\mathcal{R}(T_\tau + \Delta) - \mathcal{R}(T_\tau)] \leq 0.05(\Delta + D_{\tau-1})$, for $\Delta \leq D_\tau$ and $\tau \geq 2$.

Before we show how the above two properties imply the desired result, we note that (i) follows directly from Lemma B.1, and (ii) will be proved at the end.

Now let complete the proof using (i) and (ii). To that end, consider a time of the form $T_{\text{cold-start}} + \Delta = T_{MP} + T_{min, 1} + \Delta$, for any $\Delta > 0$. Let $\tau^* \geq 1$ be the epoch to which $T_{MP} + T_{min, 1} + \Delta$ belongs, i.e.

$$T_{\tau^*} < T_{MP} + T_{min, 1} + \Delta \leq T_{\tau^*+1}.$$

Define $t = T_{MP} + T_{min,1} + \Delta - T_{\tau^*} > 0$. We shall argue $\tau^* = 1$ and $\tau^* > 1$ separately. For $\tau^* = 1$, (i) implies the desired result. For $\tau^* > 1$, we use (ii) to argue it as follows:

$$\begin{aligned}
\mathbb{E} [\mathcal{R}(T_{MP} + T_{min,1} + \Delta) - \mathcal{R}(T_{MP})] &\stackrel{\underbrace{t=T_{cs}+\Delta-T_{\tau^*}}}{\leq} \underbrace{\mathbb{E} [\mathcal{R}(T_2) - \mathcal{R}(T_{MP})]}_{\leq 0.1 \cdot D_1 \text{ by (i)}} \\
&+ \mathbf{1}_{\tau^* \geq 3} \left(\sum_{\tau=2}^{\tau^*-1} \underbrace{\mathbb{E} [\mathcal{R}(T_\tau) - \mathcal{R}(T_{\tau-1})]}_{\leq 0.05(D_\tau + D_{\tau-1}) \text{ by (ii)}} \right) + \underbrace{\mathbb{E} [\mathcal{R}(T_{\tau^*} + t) - \mathcal{R}(T_{\tau^*})]}_{\leq 0.05(t + D_{\tau-1}) \text{ by (ii)}} \\
&\leq 0.05 \left(t + 2 \sum_{\tau=1}^{\tau^*-1} D_\tau \right) \\
&\leq 0.1 \cdot (\Delta + T_{min,1}). \tag{61}
\end{aligned}$$

This establishes the desired result that $T_{cold-start} = T_{MP} + T_{min,1} = f(\nu, d)/N + \tilde{\mathcal{O}}(1/\nu)$.

Proof of (ii): Now we argue the remaining property (ii). Lemma B.1 tells us that for $\Delta \in (T_{min,\tau}, D_\tau)$ we have that $\mathbb{E} [\mathcal{R}(T_\tau + \Delta) - \mathcal{R}(T_\tau)] \leq \frac{148}{\nu} \varepsilon_\tau \Delta$, i.e.

$$\mathbb{E} [\mathcal{R}(T_\tau + T_{min,\tau}) - \mathcal{R}(T_\tau)] \leq \frac{148}{\nu} \varepsilon_\tau T_{min,\tau}.$$

Thus, for $\Delta < D_\tau$, we have that

$$\mathbb{E} [\mathcal{R}(T_\tau + \Delta) - \mathcal{R}(T_\tau)] \leq \frac{148}{\nu} \varepsilon_\tau T_{min,\tau} + \underbrace{\frac{148}{\nu} \varepsilon_\tau \Delta}_{\leq 0.05 \text{ for } \tau \geq 2}. \tag{62}$$

In above we used the fact that for $\Delta < T_{min,1}$, $\mathcal{R}(T_\tau + \Delta) \leq \mathcal{R}(T_\tau + T_{min,1})$. Using the fact that $T_{min,\tau} \triangleq \frac{12}{\varepsilon_\tau} \ln\left(\frac{1}{\varepsilon_\tau}\right) \leq 0.05 \left(\frac{300}{\varepsilon_\tau} \ln\left(\frac{1}{\varepsilon_\tau}\right)\right) = 0.05 D_{\tau-1}$, we conclude

$$\mathbb{E} [\mathcal{R}(T_\tau + \Delta) - \mathcal{R}(T_\tau)] \leq 0.05 (\Delta + D_{\tau-1}). \tag{63}$$

■

Theorem 5.3 (Asymptotic linear regret is unavoidable). *Consider an item space μ satisfying assumptions A1 and A2. Then any online algorithm must have expected asymptotic regret $\mathbb{E} [\mathcal{R}(T)] \geq C(\nu, N) \cdot T$, where $C(\nu, N) = (1 - 2\nu)/N$.*

Proof. Let $\{i_1, \dots, i_{k_T}\}$ be the set of distinct items that have been recommended up to time TN . Then we have

$$\begin{aligned}
\mathbb{E} [\mathcal{R}(T)] &= \frac{1}{N} \mathbb{E} \left[\sum_{t=1}^{TN} \frac{1}{2} (1 - L_{U_t, I_t}) \right] = \frac{1}{N} \mathbb{E} \left[\sum_{k=1}^{k_T} \sum_{t=1}^{TN} \frac{1}{2} \mathbf{1}_{I_t=i_k} (1 - L_{U_t, i_k}) \right] \\
&\geq \frac{1}{N} \mathbb{E} \left[\sum_{k=1}^{k_T} \frac{1}{2} (1 - L_{U_{T_k}, i_k}) \right],
\end{aligned}$$

where T_k is the first time in which the item i_k is recommended to any user. Now note that for each k by (A2) we have that $\mathbb{E} \left[\frac{1}{2} (1 - L_{U_{T_k}, i_k}) \right] \geq 1 - 2\nu$, since when we have no prior information about i_k the best we can do is to recommend it to the user that likes the largest fraction of items. Hence we get

$$\mathbb{E} [\mathcal{R}(T)] \geq \frac{1 - 2\nu}{N} k_T. \tag{64}$$

Since each item can be recommended to each user at most once, we see that by the TN^{th} recommendation at least T different items must have been recommended (that is, $k_t \geq T$). We can then conclude that

$$\mathbb{E}[\mathcal{R}(T)] \geq \underbrace{\frac{1-2\nu}{N}}_{C(\nu,d)}, \quad (65)$$

as we wished. ■

D More on Doubling Dimension

In this section we provide examples of spaces with low doubling dimension, give some useful properties, and describe experiments indicating that doubling dimension is often small in practice.

Example D.1. Consider an item space μ over N users that assigns probability at least $w > 0$ to K distinct item types with separation at least $\sigma > 0$. Then, since $\mu(\mathcal{B}(x, \alpha)) \leq 1$, and $\mu(\mathcal{B}(x, \alpha/2)) \geq w$, we have that

$$d = \max_{x \in \{-1, +1\}^N} \sup_r \frac{\mu(\mathcal{B}(x, r))}{\mu(\mathcal{B}(x, r/2))} \leq \frac{\mu(\mathcal{B}(x, \alpha))}{\mu(\mathcal{B}(x, \alpha/2))} \leq \log_2 \left(\frac{1}{w} \right) = \log_2 1/w. \quad (66)$$

Similarly, if we only know that there are at most K equally likely item types we can bound the doubling dimension as

$$d = \max_{x \in \{-1, +1\}^N} \sup_r \frac{\mu(\mathcal{B}(x, r))}{\mu(\mathcal{B}(x, r/2))} \leq \frac{1}{1/K} = \log_2 K. \quad (67)$$

With the example above in mind, we would like to emphasize that doubling dimension assumptions are strictly more general than the style of assumptions made in [4] (finite K with separation assumptions) because (a) doubling measure require no separation assumptions (that is, two item types x and y that are arbitrarily close to each other can have positive mass) and (b) the number of types of positive mass is not bounded by a finite K anymore, but instead can grow with the number of users.

Example D.2. Consider an item space μ such that it assigns probability $1/K$ to K item types randomly uniformly drawn from $\{-1, +1\}^N$. Then, for each two item types i, j we have that

$$\mathbb{P} \left(\bigcup_{ij} \{\gamma_{ij} \notin [.4, .6]\} \right) \leq \sum_{ij} \mathbb{P} \left(\gamma_{ij} \notin [.4, .6] \right) \leq \binom{K}{2} \exp(-\Theta(N)), \quad (68)$$

where the first inequality is due to a union bound, and the second to a Chernoff bound. Hence, with high probability we get that $d \geq \frac{\mu(\mathcal{B}(x, .7))}{\mu(\mathcal{B}(x, .35))} \stackrel{w.h.p.}{=} \log_2 \frac{1}{1/K} = \log_2 K$. By Example D.1 we also have

that $d \leq \log_2(K)$, and hence we can conclude that with high probability we have that $d = \log_2(K)$.

Proposition D.1. Let μ be an item space for N users with doubling dimension d . Then for any item type $x \in \{-1, 1\}^N$ with $\mu(x) > 0$ we have

$$\mu(\mathcal{B}(x, r)) \geq r^d. \quad (69)$$

Proposition D.2. Let μ be an item space for N users with doubling dimension d , let \mathcal{C} be an ε -net for μ , let j be an arbitrary item, let $c_j \in \mathcal{C}$ be such that $\gamma_{j, c_j} < \varepsilon$, and let $m_{c_j} \triangleq \mu(\mathcal{B}(c_j, \varepsilon))$. Then, for each $r \in [\varepsilon/2, 1/2]$, there are at most $m_{c_j} \left(\frac{4}{\varepsilon}\right)^d \left(\frac{4r+5\varepsilon}{4\varepsilon}\right)^d$ items in \mathcal{C} within radius r of j .

Proof. By the doubling dimension of μ we get

$$\mu \left(\mathcal{B} \left(c_j, r + \frac{5}{4}\varepsilon \right) \right) \leq \mu(\mathcal{B}(c_j, \varepsilon)) \cdot \left(\frac{r + 5\varepsilon/4}{\varepsilon} \right)^d = m_{c_j} \cdot \left(\frac{r + 5\varepsilon/4}{\varepsilon} \right)^d. \quad (\star)$$

We will now use this bound on $\mu(\mathcal{B}(c_j, r + \frac{5}{4}\varepsilon))$ to show that we could pack at most $m_{c_j} \left(\frac{4}{\varepsilon}\right)^d \left(\frac{4r+5\varepsilon}{4\varepsilon}\right)^d$ items from \mathcal{C} within r of j .

Since \mathcal{C} is an ε -net, each two items $i, j \in \mathcal{C}$ are at least $\varepsilon/2$ apart, and hence the balls of radius $\varepsilon/4$ around each $i \in \mathcal{C}$ are disjoint. Say that there are $K \triangleq |\mathcal{C}_j|$ items \mathcal{C} within distance r of j , where $\mathcal{C}_j \triangleq \{c \in \mathcal{C} \mid \gamma_{c,j} \leq r\}$. Then we get

$$K \cdot \left(\frac{\varepsilon}{4}\right)^d \leq \sum_{c \in \mathcal{C}_j} \mu(\mathcal{B}(c, \varepsilon/4)) = \mu\left(\bigcup_{c \in \mathcal{C}_j} \mathcal{B}(c, \varepsilon/4)\right) \leq \mu(\mathcal{B}(c_j, r + 5\varepsilon/4)), \quad (70)$$

where the first inequality is due to Proposition D.1 and the last inequality is due to $\bigcup_{c \in \mathcal{C}_j} \mathcal{B}(c, \varepsilon/4) \subset \mathcal{B}(c_j, r + 5\varepsilon/4)$. Using the bound from eq. (*) we arrive at

$$K \leq \left(\frac{4}{\varepsilon}\right)^d \cdot m_{c_j} \cdot \left(\frac{r + 5\varepsilon/4}{\varepsilon}\right)^d. \quad (71)$$

as we wished. ■

Finally, we would like to note that doubling dimension is not only a “proof technique”: it can be estimated from data and tends to be small in practice. To illustrate this point, we calculate the doubling dimension on the Jester Jokes Dataset⁷ and for the MovieLens 1M Dataset⁸. For the MovieLens dataset we consider the only movies that have been rated by at least 750 users (to ensure some density).

The Jester dataset contains ratings of one hundred jokes by over seventy thousand users. The dataset is fairly dense (as the average number of ratings per user is over fifty), which makes it a great dataset for calculating the doubling dimension. For the MovieLens 1M Dataset we consider the only movies that have been rated by at least 750 users (to ensure some density).

The Jester ratings are in $[-10, 10]$, with an average of 2, so we make ratings greater than 2 a $R_{u,i} = +1$, and ratings at most 2 a $R_{u,i} = -1$. For the MovieLens 1M Dataset we make ratings 1, 2, 3 into -1 , and 4, 5 into $+1$. We then estimate the doubling dimension as follows:

- For each pair of items (i, j) , we calculate $\hat{d}_{i,j,\Delta}$ as fraction of users that agree on them, where the Δ subscript is put to denote our assumption that each entry has a noise probability of Δ (that is, $\mathbb{P}(R_{u,i} \neq L_{u,i}) = \Delta$), where R is the empirical ratings matrix and L is the true, noiseless, ratings matrix.
- Assuming that each entry has a noise probability of $\Delta = 0.20$, we estimate the true distance $d_{i,j}$ as the solution to $\hat{d}_{i,j,\Delta} = (1 - d_{ij})(2\Delta(1 - \Delta)) + d_{i,j}(\Delta^2 + (1 - \Delta)^2)$.
- For each item i and r in $\{0, \frac{1}{N}, \dots, \frac{N-1}{N}, 1\}$, let $N_{i,r}$ be the number of items such that $d_{i,j} \leq r$.
- For each item i let d_i be the least such that $N_{i,2r}/N_{i,r} \leq 2^{d_i}$ for each r in $\{0, \frac{1}{N}, \dots, \frac{1}{2}\}$.
- The figs. 1 and 2 show the histogram of the $\{d_i\}$.

⁷[26], and data available on <http://goldberg.berkeley.edu/jester-data/>

⁸[27], and data available on <http://grouplens.org/datasets/movielens/>

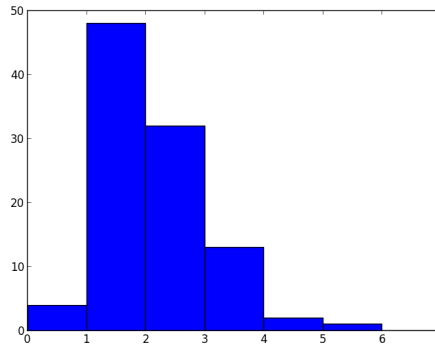


Figure 1: Jester Doubling Dimensions

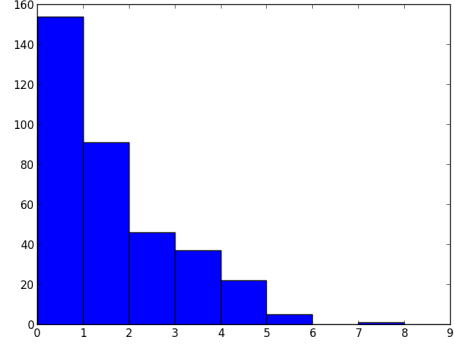


Figure 2: MovieLens Doubling Dimensions

E Chernoff Bound

The following is a standard version of Chernoff Bound [28] that we use throughout the paper.

Theorem E.1 (Chernoff Bound). *Let X_1, \dots, X_n be independent random variables that take value in $[0, 1]$. Let $X = \sum_{i=1}^n X_i$, and let $\bar{X} = \sum_{i=1}^n \mathbb{E}X_i$. Then, for any $\varepsilon \geq 0$,*

$$\mathbb{P}(X \geq (1 + \varepsilon) \bar{X}) \leq \exp\left(-\frac{\varepsilon^2}{2 + \varepsilon} \bar{X}\right), \text{ and}$$

$$\mathbb{P}(X \leq (1 - \varepsilon) \bar{X}) \leq \exp\left(-\frac{\varepsilon^2}{2} \bar{X}\right).$$