Collaborative Filtering with Low Regret

by

Luis F. Voloch

ARCHIVES

Submitted to the Department of Electrical Engineering and
Computer Science

in partial fulfillment of the requirements for the degree of
Master of Science in Electrical Engineering and Computer Science at the
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2015

© Massachusetts Institute of Technology 2015. All rights reserved.

Signature redacted

Author.

Luis F. Voloch

Department of Electrical Engineering and Computer Science

May 20, 2015

Signature redacted

Certified by

.....
Devavrat Shah

Associate Professor of Electrical Engineering

Thesis Supervisor

Signature redacted

Accepted by.

.....
J O U Leslie A. Kolodziejski
Chair, Department Committee on Graduate Students

Collaborative Filtering with Low Regret

by Luis F. Voloach

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of Master of Science in Electrical Engineering and Computer Science

Abstract

Collaborative filtering (CF) is a widely used technique in recommendation systems where recommendations are provided in a content-agnostic manner, and there are two main paradigms in neighborhood-based CF: the user-user paradigm and the item-item paradigm. To recommend to a user in the user-user paradigm, one first looks for similar users, and then recommends items liked by those similar users. In the item-item paradigm, in contrast, items similar to those liked by the user are found and subsequently recommended. Much empirical evidence exists for the success of the item-item paradigm (Linden et al., 2003; Koren and Bell, 2011), and in this thesis, motivated to understand reasons behind this, we study its theoretical performance and prove guarantees.

We work under a generic model where the population of items is represented by a distribution over $\{-1, +1\}^N$, with a binary string of length N associated with each item to represent which of the N users likes (+1) or dislikes (−1) the item. As the first main result, we show that a simple algorithm following item-item paradigm achieves a regret (which captures the number of poor recommendations over T time steps) that is sublinear and scales as $\tilde{O}(T^{\frac{d+1}{d+2}})$, where d is the doubling dimension of the item space. As the second main result we show that the cold-start time (which is the first time after which quality recommendations can be given) of this algorithm is $\tilde{O}(\frac{1}{\nu})$, where ν is the typical fraction of items that users like.

This thesis advances the state of the art on many fronts. First, our cold-start bound differs from that of Bresler et al. (2014) for user-user paradigm, where the cold-start time increases with number of items. Second, our regret bound is similar to those obtained in multi-armed bandits (surveyed in Bubeck and Cesa-Bianchi (2012)) when the arms belong to general spaces (Kleinberg et al., 2013; Bubeck et al., 2011). This is despite the notable differences that in our setting: (a) recommending the same item twice to a given user is not allowed, unlike in bandits where arms can be pulled twice; and (b) the distance function for the underlying metric space is not known in our setting. Finally, our mixture assumptions differ from earlier works, cf. (Kleinberg and Sandler, 2004; Dabeer, 2013; Bresler et al., 2014), that assume “gap” between mixture components. We circumvent gap conditions by instead using the doubling dimension of the item space.

Thesis Adviser: Devavrat Shah

Title: Jamieson Associate Professor

Acknowledgements

To my adviser, Devavrat Shah: thank you for your support and brilliant guidance in all matters relating academics, research, and to this thesis. Vastly more importantly, however, thank you for your friendship, which I can easily say is what has kept me going in grad school.

To Guy Bresler, with whom I had the immense pleasure of working and spending time, and also without whom this thesis would not have been the same: your passion, perseverance, incredible intellect, and modesty are a constant inspiration for me.

To the Jacobs Fellowship and Draper Fellowship: thank you for the support in my first two years.

To the larger MIT community, and in particular the EECS Department, Math Department, and Sloan: thank you for being an amazing academic environment.

To my parents, to whom I owe *everything*: thank you for your unconditional love, and for always believing in me.

Luis Filipe Voloch,
June 2015

Contents

Abstract	3
Acknowledgements	4
List of Figures	9
1 Introduction	11
1.1 Background	11
1.2 Organization of Thesis	12
1.3 Model	12
1.4 Objectives of a Recommendation Algorithm	13
1.5 Overview of Main Results	15
1.6 Related Works	16
1.6.1 Multi-armed Bandits	16
1.6.2 Matrix Factorization	18
1.6.3 Other Related Works	19
2 Structure in Data	21
2.1 Need for Structure	21
2.2 Item Types	23
2.3 Doubling Dimension	24
2.4 Low Rank and Doubling Dimension	27
3 Description of Algorithm	31
3.1 Exploit	33
3.2 Explore	33
4 Correctness of Explore	37

4.1	Guarantees for SIMILAR	37
4.2	Making the Partition	41
4.3	Sufficient Exploration	47
5	Regret Analysis	49
5.1	Quick Recommendations Lemma	49
5.2	Main Results	56
6	Conclusion	63
6.1	Discussion	63
6.1.1	Two roles of Doubling Dimension	64
6.1.2	Decision Making and Learning with Mixtures	64
6.1.3	Explore-Exploit	65
6.2	Future Works	65
A	Appendix	67
A.1	Chernoff Bound	67
A.2	Empirical Doubling Dimension Experiments	67
	Bibliography	69

List of Figures

1.1	Regret Behavior of ITEM-ITEM-CF. Between $T_{\text{cold-start}}$ and $T_{\text{max},N}$ the algorithm achieves sublinear expected regret $\tilde{O}(T^{\frac{d+1}{d+2}})$. After $T_{\text{max},N}$ the expected regret plateaus at a constant slope. The duration of the sublinear regime increases and the asymptotic linear slope decreases with the number of users.	16
2.1	Observed Clustering of Users and Items (from Bresler et al. (2014)). This is the densest subset of users (rows) and items (columns), where the darker spots indicate likes and the lighter spots indicate dislikes. One can see that the items and users can be grouped in relatively few types, where items of the same type tend to be liked by users of the same type.	24
2.2	Jester Doubling Dimensions	28
2.3	MovieLens Doubling Dimensions	28
3.1	Partition resulting from exploration in the previous epoch	32
3.2	During exploitation, items in P_k are recommended only when from a block when a user likes an item in the block	32
6.1	Expected Regret of ITEM-ITEM-CF, as proven in Theorems 5.1 and 5.2. . . .	63

Introduction

■ 1.1 Background

Whenever a business contains a large collection of items for sale, it is of interest to help customers find the items that are of most interest to them. Before the creation and widespread adoption of the Internet, this was done by trained store salesmen, who can recommend items based on experience and the customers' revealed preferences.

After the creation of the Internet, this "recommendation system" has been largely taken off the hands of trained salesmen and is now largely handled by automated, statistically driven policies. For many companies, the efficacy of their recommendation systems stands at the core of their business, where Amazon and Netflix are particularly prominent examples.

A natural and clever first idea in designing an automated recommendation system is to use content specific data. In this spirit, one may use similar words in the title and book's cover, or a user's age and geographic location as inputs to recommendation heuristics. This type of recommendation system, where content-specific data is used, is called content filtering.

In contrast to content filtering, a technique called collaborative filtering (CF) provides recommendation in a content-agnostic way. CF works by exploiting patterns in general purchase or usage data. For instance, if 90% of users agree on two items (that is, 90% of users either like both items or dislike both items), a CF algorithm may recommend the second item after a user has expressed positive feedback for the first item.

The term collaborative filtering was coined in Goldberg et al. (1992), and this technique is used in virtually all recommendation systems. There are two main paradigms in neighborhood-based collaborative filtering: the user-user paradigm and the item-item paradigm. To recommend to a user in the user-user paradigm, one first looks for similar users, and then recommends items liked by those similar users. In the item-item paradigm, in contrast, items similar to those liked by the user are found and subsequently recom-

mended. Much empirical evidence exists that the item-item paradigm performs well in many cases (Linden et al., 2003; Koren and Bell, 2011), and in this thesis, motivated to understand reasons behind this, we introduce a mathematical model and formally analyze the performance with respect to this model of a simple, intuitive algorithm (that we call ITEM-ITEM-CF and describe in chapter 3) that follows the item-item paradigm.

■ 1.2 Organization of Thesis

The rest of this thesis is organized as follows. For the remainder of chapter 1 we formally introduce the model, give an informal overview of the main results, and discuss related works. In chapter 2 describe the assumptions that are made in light of theoretical lower bounds and empirical observations. In chapter 3 we describe our algorithm and in chapter 4 we prove the correctness of its main set of routines. In chapter 5 we put all the pieces together and give our main results pertaining to the performance of ITEM-ITEM-CF. In chapter 6 we further discuss our results and list some future works.

■ 1.3 Model

Recommendation systems typically operate in an online setting, meaning that when a user logs into an virtual store (such as Amazon), a decision must be made immediately regarding which recommendations to make to that user. We model recommendation requests as follows: time is discrete and at each time step t a uniformly random user $U_t \in \{1, \dots, N\}$ is chosen to be provided with a recommendation¹ $I_t \in \mathcal{I}$, where $\mathcal{I}$ is the universe of items. Furthermore, we impose the constraint that the recommendation algorithm may not recommend the same item to the same user twice². For each time t , after being recommended item I_t , user U_t reveals the true preference L_{U_t, I_t} for that item, which may be +1 (like) or -1 (dislike).

For each recommendation, the algorithm may either recommend an item that has been previously recommended to other users (in which case it has some information about whether or not at least one user likes the item) or recommend a new item from $\mathcal{I}$.

For example, consider the toy example where we have $N = 3$ users. Suppose that

¹This is equivalent to the model where each user is equipped with an independent *exponential clock* of the same rate, and a recommendation is requested by a user when his clock ticks.

²The purpose of this work is to capture how well a recommendation algorithm can suggest *new* items to users rather the ones for which the users' preferences are already known. It is for this reason that we impose such a constraint. Of course, it also captures the setting where one is unlikely to buy the same book or watch the same movie twice.

we recommend item 1 to user 2, which then reveals that s/he likes it, i.e. $L_{2,1} = 1$. Then the observed rating matrix L_{obs} and the (hidden) rating matrix L this time might be

$$L = \begin{pmatrix} +1 \\ +1 \\ -1 \end{pmatrix}, L_{obs} = \begin{pmatrix} ? \\ +1 \\ ? \end{pmatrix}. \quad (1.1)$$

Since each item is either liked (+1) or disliked (−1) by a given user, then each item is associated with a binary string of length N encoding likes/dislikes of the N users for it, and we call this binary string the *type* of i .

We are interested in the situation where there are many items, and will assume that $\mathcal{I}$ is infinite. Therefore, from the N users' perspective, the population of items can be represented as a distribution, denoted by μ , over $\{-1, +1\}^N$. That is, when the algorithm wishes to recommend an item that hasn't yet been recommended, the item's type is drawn from this distribution in an i.i.d. manner³.

For instance, if

$$\mu(x) = \begin{cases} 1/2 & \text{if } x = [+1, +1, -1]^T \\ 1/2 & \text{if } x = [+1, -1, -1]^T, \end{cases} \quad (1.2)$$

and if the second recommendation was to user 3 was a new item (which turned out to be of type $[+1, -1, -1]^T$) then the rating matrices L and L_{obs} after that recommendation would be

$$L = \begin{pmatrix} +1 & +1 \\ +1 & -1 \\ -1 & -1 \end{pmatrix}, L_{obs} = \begin{pmatrix} ? & ? \\ +1 & ? \\ ? & -1 \end{pmatrix}. \quad (1.3)$$

Hence, we may think of μ as a way of adding columns to the rating matrix when new items are recommended, where the new column is distributed according to μ , and we will refer to an item space by its distribution μ .

■ 1.4 Objectives of a Recommendation Algorithm

Given the setting above, we are interested in showing that an algorithm based on item-item collaborative filtering works well. A standard way to formally analyze online algorithms is to look at its regret. The regret of a recommendation algorithm $\mathcal{A}$ is the

³Note that such a model is quite general as we are imposing no constraints on how users like items, other than the restriction that preferences are binary (i.e. ± 1).

difference in reward relative to an all-knowing algorithm that makes no bad recommendations. In our case, the regret after T time steps can be formally defined as

$$\mathcal{R}(T) \triangleq \frac{1}{N} \sum_{t=1}^{T \cdot N} \frac{1}{2} (1 - L_{U_t, I_t}), \quad (1.4)$$

as long as each user likes a positive fraction of items under μ , which will always be the case. At time t the user U_t is desiring a recommendation, I_t is the recommended item, the variable $L_{u,i}$ is equal to $+1$ (resp. -1) if u likes (resp. dislikes) the item i , and N is the number of users⁴. The regret $\mathcal{R}(T)$ captures the expected number of bad recommendations per user after having made an average of T recommendations per user (hence the scaling by N in the definition of $\mathcal{R}(T)$). Broadly, there are two main objectives in designing a successful recommendation algorithm:

- **Cold-Start Performance:** with no prior information, the algorithm should give reliable recommendations as quickly as possible. The *cold-start* performance of an recommendation algorithm $\mathcal{A}$ is measured by

$$T_{\text{cold-start}}^{(\mathcal{A})} = \min \left\{ T + \Gamma \text{ s.t. } T, \Gamma \geq 0, \mathbb{E}[\mathcal{R}(T + \Gamma + \Delta) - \mathcal{R}(T)] \leq 0.1(\Delta + \Gamma), \forall \Delta > 0 \right\}. \quad (1.5)$$

Intuitively this is the first time after which the algorithm's expected regret slope is bounded below 0.1. That is, from then on it can typically make a recommendation to a randomly chosen user and be wrong with probability at most 0.1. Whereas the choice of 0.1 is arbitrary, since we will assume that users tend to like only a small fraction of random items, one can think of the cold start time as the minimum time starting from when the algorithm can recommend significantly better than a random recommendation algorithm (which would make mostly bad recommendations). We want $T_{\text{cold-start}}$ to be as small as possible, and in particular we will see that it need not grow with N .

- **Improving Accuracy:** the algorithm should give increasingly more reliable recommendations as it gains more information about the users and items, and this is captured by having *sublinear expected regret*. Ideally, we want $\mathbb{E}[\mathcal{R}(T)]$ to be sublinear in T (i.e. $o(T)$) and as small as possible.

⁴For notational simplicity we omit $\mathcal{A}$ on the notation of the regret and cold-start time, as the algorithm of interest will be clear from context.

■ 1.5 Overview of Main Results

In this section we informally state our main results about the cold-start and improving accuracy of the algorithm ITEM-ITEM-CF (described in chapter 3), which is a simple, intuitive algorithm following the item-item paradigm.

Cold-Start Performance: We shall assume that each user likes at least $\nu > 0$ fraction of items. Then we have

- (Achieving Cold-Start Time, Theorem 5.2) The algorithm ITEM-ITEM-CF achieves $T_{\text{cold-start}}$ of $\tilde{O}(\frac{1}{\nu})$.

One can relate this to a biased coin intuition, where we expect approximately $1/p$ coin flips until one gets a head. Similarly, in our case, since users like approximately ν fraction of the items, it only takes $\tilde{O}(1/\nu)$ for the user to find an item that is liked, and after this point the algorithm can recommend similar items. In light of the cold-start result, we would like to note a *key asymmetry between item-item and user-user* collaborative filtering. It is much easier to compare two items than it is to compare two users. Whereas it takes a long time to make many recommendations to two particular users, comparing two items can be done in parallel by sampling different users.

Improving Accuracy: Our first result states that unless some assumption is made on the items, no algorithm can perform well.

- (Lower Bound: Necessity of Structure, Proposition 2.1) With no structure on the item space, it is not possible for any algorithm to achieve sublinear regret for any period of time.

We will henceforth assume that the doubling dimension (which we define and motivate later, but the reader may simply think of it as a measure of complexity of the space of items) d of the item space is $\mathcal{O}(1)$. Then

- (Sublinear Regime Regret Upper Bound, Theorem 5.1) the algorithm ITEM-ITEM-CF achieves sublinear expected regret $\tilde{O}(T^{\frac{d+1}{d+2}})$ until a certain time T_{max} , after which the expected regret plateaus at a constant slope,

where T_{max} increases with N and the slope decreases as a function with N , both of which illustrate the so-called collaborative gain⁵. Finally, we show that the

- (Lower Bound: Necessity of Linear Regime, Theorem 5.3) The asymptotic linear regime is unavoidable: every algorithm must have asymptotic regret $\Omega(T)$.

⁵By collaborative gain we mean the per-user performance improvement when we have many users, in comparison to the performance of only a single user using the system.

Figure 1.1: Regret Behavior of ITEM-ITEM-CF. Between $T_{\text{cold-start}}$ and $T_{\text{max},N}$ the algorithm achieves sublinear expected regret $\tilde{O}(T^{\frac{d+1}{d+2}})$. After $T_{\text{max},N}$ the expected regret plateaus at a constant slope. The duration of the sublinear regime increases and the asymptotic linear slope decreases with the number of users.

■ 1.6 Related Works

In this section we explore various related works. We begin by describing some relevant literature in multi-armed bandits, which deals with explore-exploit trade-offs similar to the ones a recommendation algorithm faces. We then, for completeness, describe matrix factorization methods of recommendation systems, which is another popular model of recommendation system. We conclude by discussing other relevant works.

■ 1.6.1 Multi-armed Bandits

The work on multi-armed bandits is also relevant to our problem. In the bandit setting, at each time t the user must choose which arm $i \in \mathcal{X}$ (hence multi-armed) to pull in a slot machine (bandit). After pulling an arm i , the user receives a reward which is distributed according to an i.i.d. random variable with mean r_i . The means $\{r_i\}$ are, however, unknown to the user, and hence the decision about which arm to pull depends on estimates of the means.

The goal of the user is to minimize the expected regret, which in this context the regret is defined as

$$\bar{\mathcal{R}}(T) = \mathbb{E} \left[\sum_{t=1}^T r^* - r_{I_t} \right], \quad (1.6)$$

where I_t is the random variable denoting the arm pulled at time t and $r^* \triangleq \sup_{i \in \mathcal{X}} r_i$ is the supremum of expected reward of all arms. Hence, the expected regret measures

difference in expected rewards in comparison with an oracle algorithm that always pulls the arm with the highest expected reward.

The survey by Bubeck and Cesa-Bianchi (2012) thoroughly covers many other variants of multi-armed bandits, and points to Thompson (1933) as the earliest works in bandits. The multi-armed bandit work that is most relevant to us, however, is when $\mathcal{X}$ is a very large set, in which case we can interpret arms as item, and pulling an arm as recommending an item to a user. In this case of a large $\mathcal{X}$ (possibly uncountably infinite), however, some structure on $\mathcal{X}$ must be assumed for any algorithm to achieve nontrivial regret (otherwise one is left with a needle in a haystack scenario).

The works of Kleinberg et al. (2013) and Bubeck et al. (2011) assume structure on $\mathcal{X}$ and prove strong upper and lower bounds on the regret. They assume that the set $\mathcal{X}$ is endowed with some geometry: Kleinberg et al. (2013) assumes that the arms are endowed with a metric, and Bubeck et al. (2011) 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 Lipschitz⁶ in the distance, and in the latter work the dissimilarity function constrains the slope of the reward around its maxima. In contrast, what is Lipschitz about our setting? Although slightly counter-intuitive since it involves many users, what is Lipschitz about our setting is the difference in reward when averaged among all users. That is

$$\mathbb{E}_u[|L_{u,i} - L_{u,j}|] \leq 2 \cdot \gamma_{i,j}, \quad (1.7)$$

where γ_{ij} is a distance between items, and will be defined in chapter 2.

Those two aforementioned works show that the regret of their algorithm is $\tilde{O}(T^{\frac{d'+1}{d'+2}})$, where d' is a weaker notion of the covering number of $\mathcal{X}$, and is closely related to the doubling dimension (which we define later) in the case of metric. Our regret bound proven in Theorem 5.1 is of the same form, despite two important additional restrictions: (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. Furthermore, unlike prior works, we quantify the cold-start problem.

⁶That is, they assume that $|\mathbb{E}[r_i] - \mathbb{E}[r_j]| \leq C \cdot d(i, j)$, where d is the distance metric between arms, and C is a constant.

■ 1.6.2 Matrix Factorization

Aside from neighborhood-based methods of collaborative filtering, matrix factorization is another branch of collaborative filtering that is widely used in practice. The winning team in the Netflix Challenge, for instance, used an algorithm based on matrix factorization, and due to its importance we will briefly cover some aspects of it here⁷.

Using the language of recommendations, the set-up for matrix factorization is that we have a (possibly incomplete) $n \times m$ matrix L of ratings (where rows represent users and columns represent rows) that we would like to factor as $L = AW$, where A is $n \times r$ and W is $r \times m$, and r is as small as possible. When L is complete, the singular value decomposition is known to provide the decomposition (cf. Ekstrand, Riedl, and Konstan (2011)). When L is incomplete, however, a popular approach in practice (cf. (Koren, Bell, and Volinsky, 2009)) is to use $r = 1$ and to find the decomposition by solving the regularized optimization problem

$$\arg \min_{A, W} \sum_{(u,i) \text{ observed}} \left(L_{u,i} - A_u^T W_i \right) + \lambda \left(\|A_u\|^2 + \|W_i\|^2 \right) \quad (1.8)$$

via, for instance, stochastic gradient descent (cf. (Zhang, 2004)). This, in effect, is looking for a sparse rank 1 factorization of L .

In many contexts, including that of recommendations, it also makes sense to want a nonnegative factorization. Intuitively, this enforces that we interpret each item as being composed of different attributes to various extents (rather than as a difference of attributes). Lee and Seung (2001) provided theoretical guarantees for fixed r , and Vavasis (2009) then showed that this problem is NP-hard with r as a variable. The first provable guarantee under nontrivial conditions came only later in Arora, Ge, Kannan, and Moitra (2012). They give an algorithm that runs in time that is polynomial in the m , n , and the inner rank, granted that L satisfies the so-called separability condition introduced by Donoho and Stodden (2003)⁸.

In Section 2.4 we compare a rank notion to that of doubling dimension, which is structural assumption that we use here.

⁷ As noted in Koren et al. (2009), matrix factorization techniques have some advantages over neighborhood-based methods, such as the ease of combining it with content-specific data and of including implicit feedback.

⁸Moitra (2014) surveys of all these and more results.

■ 1.6.3 Other Related Works

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 Bresler, Chen, and Shah (2014), via bandits in a wide variety of settings (for instance Alon, Cesa-Bianchi, Gentile, Mannor, Mansour, and Shamir (2014); Slivkins (2014); Cesa-Bianchi, Gentile, and Zappella (2013)), with focus on benefits to the cold-start problem Gentile, Li, and Zappella (2014); Caron and Bhagat (2013), and in practice (Das, Datar, Garg, and Rajaram, 2007; Bellogin and Parapar, 2012). In this thesis, in order to capture the power of a purely item-item collaborative filtering algorithms, we purposefully avoid using any user-user similarities.

The works of Hazan, Kale, and Shalev-Shwartz (2012); Cesa-Bianchi and Shamir (2013) on online learning and matrix completion are also of interest. In their case, however, the matrix entries revealed are not chosen by the algorithm and hence there is no explore-exploit trade-off as we intend to analyze here.

Kleinberg and Sandler (2004) consider collaborative filtering under a mixture model in the *offline* setting, and they make separation assumptions between the item types (called genres in their paper). Dabeer (2013) proves guarantees, but their analysis considers a moving horizon approximation, and the number of users types and item types is finite and are both known. Biau, Cadre, and Rouviere (2010) prove asymptotic consistency guarantees on estimating the ratings of unrecommended items.

A latent source model of user types is used by Bresler et al. (2014) to give performance guarantees for user-user collaborative filtering. The assumptions of user types and item types 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 thesis we assume a latent structure of items. Unlike the standard mixture model with "gap" between mixture components (as assumed in all the above mentioned works), our setup does not have any such gap condition. In contrast, our algorithm works with effectively the most generic model, and we establish the performance of the algorithm based on a notion of dimensionality of the item space.

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 typically be grouped in a few different types that tend to be liked by the same users. It is with this intuition and empirical observation in mind that the two main paradigms in neighborhood-based collaborative filtering, user-user and item-item, operate. In the user-user paradigm, the algorithm first looks for similar users, and then recommends items liked by those similar users. In contrast, in the item-item paradigm items similar to those liked by the user are found and subsequently recommended. In this chapter we will develop the appropriate tools to exploit with structure in data. First, in Section 2.1 we show that without structural assumptions, the expected regret of any online algorithm is linear at all periods of time. In Section 2.2, we then develop our intuition for the structure present in data and define a distance between types of items. In Section 2.3 we then define the precise structural assumption that we will make: that the item space has finite doubling dimension. Finally, in Section 2.4 we give an example of how to relate the concepts of doubling dimension and low rank.

■ 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. To be able to suggest good/likable items to users, there are two basic requirements. First, at the heart of any recommendation algorithm, the basic premise is the ability to achieve “collaborative gain”: in the item-item paradigm items similar to those liked by the user can be recommended, and in the user-user paradigm items liked by a similar user can be recommended. For any *any* algorithm to achieve non-trivial “collaborative gain”, it is necessary to impose some structural constraints on μ . To that end, we first define what an online algorithm is, and then state the natural result stating that, in the worst case where μ has little structure,

no online algorithm can do better than providing random recommendations.

Definition 2.1 (Online Recommendation Algorithm). A recommendation algorithm is called an *online* recommendation algorithm if the recommendation at time t depends only on U_t , and $\bigcup_{s < t} \{U_s, I_s, L_{U_s, I_s}, s\}$. That is, it has access only to the feedback received on previous recommendations.

That is, an online recommendation algorithm has no access to knowing whether an item is liked or disliked by a particular user prior to recommending it.

Proposition 2.1 (Lower Bound). *Under the uniform distribution μ over $\{-1, +1\}^N$, for all $T \geq 1$, the expected regret satisfies*

$$\mathbb{E}[\mathcal{R}(T)] \geq T/2$$

for any online recommendation algorithm. Conversely, the algorithm that recommends a random item at each time step achieves $\mathcal{R}(T) = \mathbb{E}[T/2]$.

Proof. We construct the desired item space by assigning $\mu(v) = 2^{-N}$ to each $v \in \{-1, 1\}^N$. That is, for every subset of users $\mathcal{U} \subset \{0, \dots, N-1\}$ there exists an item type $h_{\mathcal{U}}$ that is liked exactly by $\mathcal{U}$ and by no other users.

Now say that at time t we must make a recommendation to a user u , and let M_u be the set of items that have not yet been recommended to u . We will show that for each $i \in M_u$ we have $\mathbb{P}(L_{u,i} = +1 \mid H_t) = 1/2$, where H_t is the history up to time t of users' likes and dislikes. Let $\mathcal{R}_i$ be the set of users that have rated item i . Then, because the $N - |\mathcal{R}_i|$ other users may each like or dislike i , there are exactly $2^{N-|\mathcal{R}_i|}$ item types to which i can belong. Furthermore, since half of those item types are liked by u and half are not liked by u and since all item types are equally likely (by our construction), we conclude that $\mathbb{P}(L_{u,i} = +1 \mid H_t) = \frac{2^{N-|\mathcal{R}_i|}/2}{2^{N-|\mathcal{R}_i|}} = 1/2$.

By the argument above, we have constructed an item space for which no algorithm can do better than recommending wrong items in expectation half of the time. Now since $1 - L_{U_t, I_t} = 2$ when we make a bad recommendation at time t , we conclude that no algorithm can have expected regret lower than $\sum_{t=1}^T \mathbb{E}_2^1[1 - L_{U_t, I_t}] = \sum_{t=1}^T 1/2 = T/2$ over T recommendations.

Now consider the algorithm that recommends a random item at each time step. That is, at each time step t , we draw a new item I_t according to μ , and recommend I_t to U_t . Since μ is uniform over $\{-1, +1\}^N$, in particular the index corresponding to U_t is uniform over $\{-1, +1\}$ as well, and hence we have that $\mathbb{P}(L_{U_t, I_t} = -1) = 1/2$ for each

t . Since this is true for each time step, we conclude that this algorithm indeed 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 will assume throughout that

(A1) the item space, represented by μ , has doubling dimension at most d for a given $d \geq 0$,

where the doubling dimension is defined and motivated in Section 2.3.

Finally, if all users dislike all items or items are disliked by all users, then there is no way to recommend a likable item. Likewise, if users like too many items or some items are liked by too many users, then any regret benchmark becomes meaningless. To avoid such trivialities, we shall assume throughout that

(A2) each user likes a random item drawn from μ with probability $\in [\nu, 2\nu]$, and each item is liked by a fraction $\in [\nu, 2\nu]$ of the users, for a given $\nu \in (0, 1/4)$.

■ 2.2 Item Types

To more formally describe item types and the conditions on μ , let us first define a distance between the item types. We shall endow the N -dimensional hamming space, i.e. $\{-1, +1\}^N$, with the following normalized ℓ_1 metric: for any two item types $x, y \in \{-1, +1\}^N$, define distance between them as

$$\gamma_{x,y} \equiv \gamma(x, y) \triangleq \frac{1}{N} \sum_{k=1}^N \frac{1}{2} |x_k - y_k|. \quad (2.1)$$

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

One can plausibly go about capturing the empirical observation that items tend to belong to clusters that are liked by similar set of users is to assume that there are K different item types, and that the measure μ assigns positive mass to only K strings in $\{-1, +1\}^N$. Bresler et al. (2014) do this, but where the users (instead of items) come in clusters. In addition, they assume that the types are well-separated (that is, $\gamma_{x,y}$ is lower bounded for each two different types x and y with positive mass). This allows for clustering perfectly with high probability, which in turn leads to a small regret.

Figure 2.1: Observed Clustering of Users and Items (from Bresler et al. (2014)). This is the densest subset of users (rows) and items (columns), where the darker spots indicate likes and the lighter spots indicate dislikes. One can see that the items and users can be grouped in relatively few types, where items of the same type tend to be liked by users of the same type.

However, enforcing that the types are well-separated is counter-intuitive and not necessary for the following reason. If two types x and y are extremely close to each other, that should only make the problem easier and lead to a small regret, for if the algorithm mistakenly misclusters one type as the other, then the harm is small, as it will lead to extremely few bad recommendations.

It turns out that we can exploit this intuition when the item space has sufficient structure, as captured by a certain notion of dimensionality.

■ 2.3 Doubling Dimension

In order to capture this intuition that our mixture assumption should (i) give significant mass around item types and (ii) not have separation assumptions, we define a doubling dimension of μ , and then further discuss its advantages. Let $\mathcal{B}(x, r) = \{y \in \{-1, +1\}^N : \gamma_{x,y} \leq r\}$ be the ball of radius r around x with respect to metric γ^1 .

Definition 2.2. (Doubling Dimension) A distribution μ on $\{-1, +1\}^N$ is said to have dimension d if d is the least number such that for each $x \in \{-1, +1\}^N$ with $\mu(x) > 0$ we have

$$\sup_{r>0} \frac{\mu(\mathcal{B}(x, 2r))}{\mu(\mathcal{B}(x, r))} \leq 2^d. \quad (2.2)$$

Further, a measure that has finite doubling dimension is called a doubling measure.

¹We omit the dependence of the ball on γ throughout.

The above definition is a natural adaptation to probability measures on metric spaces of the well-known notion of doubling dimension for metric spaces (cf. (Heinonen, 2001; Har-Peled and Mendel, 2006; Dasgupta and Sinha, 2014)). As noted in, for instance Dasgupta and Sinha (2014), this is equivalent to enforcing that $\mu(B_Y(x, \alpha r)) \leq \alpha^d \cdot \mu(B_Y(x, r))$ 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.

Despite its simplicity, measures of low doubling dimension capture the observed clustering phenomena. Proposition 2.2 below, which follows directly from the definition, shows that a small doubling dimension ensures that the balls around any item type must have a significant mass.

Proposition 2.2. *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*

$$\mu(\mathcal{B}(x, r)) \geq r^d. \quad (2.3)$$

Proof. Using the fact that $B(x, 1) = \{-1, +1\}^N$ for any $x \in \{-1, +1\}^N$ and the definition of doubling dimension we immediately get $1 = \mu(B(x, 1)) \leq \left(\frac{1}{r}\right)^d \cdot \mu(\mathcal{B}(x, r))$, which proves the proposition. ■

Doubling measures also induce many other nice properties on the item space, but let us first define an ε -net for an item space.

Definition 2.1 (ε -net). *An ε -net an item space μ is a set $\mathcal{C}$ of items such that for a random i drawn according to μ , with probability 1 there exists $c \in \mathcal{C}$ such that $\gamma_{i,c} \leq \varepsilon$. Furthermore, $\gamma_{i,j} > \varepsilon/2$ for each two items $i, j \in \mathcal{C}$.*

Proposition 2.3 below shows that, given an ε -net of items, there can only be a small number of net items close to any given item.

Proposition 2.3. *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 $\mu(B(c_j, r + \frac{5}{4}\epsilon))$ to show that we could pack at most $m_{c_j} \left(\frac{4}{\epsilon}\right)^d \left(\frac{4r+5\epsilon}{4\epsilon}\right)^d$ 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 $\epsilon/2$ apart, and hence the balls of radius $\epsilon/4$ 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

$$K \cdot \left(\frac{\epsilon}{4}\right)^d \leq \sum_{c \in \mathcal{C}_j} \mu(B(c, \epsilon/4)) = \mu\left(\bigcup_{c \in \mathcal{C}_j} B(c, \epsilon/4)\right) \leq \mu(B(c_j, r + 5\epsilon/4)), \quad (2.4)$$

where the first inequality is due to Proposition 2.2 and the last inequality is due to $\bigcup_{c \in \mathcal{C}_j} B(c, \epsilon/4) \subset B(c_j, r + 5\epsilon/4)$. Using the bound from eq. (★) we arrive at

$$K \leq \left(\frac{4}{\epsilon}\right)^d \cdot m_{c_j} \cdot \left(\frac{r + 5\epsilon/4}{\epsilon}\right)^d. \quad (2.5)$$

as we wished. ■

To further illustrate and gain intuition for doubling dimension, let us consider an example of doubling dimension of an item space.

Example 2.1. 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 $\mu(B(x, \alpha)) \leq 1$, and $\mu(B(x, \alpha/2)) \geq w$, we have that

$$d = \max_{x \in \{-1, +1\}^N} \sup_r \frac{\mu(B(x, r))}{\mu(B(x, r/2))} \leq \frac{\mu(B(x, \alpha))}{\mu(B(x, \alpha/2))} \leq \log_2 \left(\frac{1}{w}\right) = \log_2 1/w. \quad (2.6)$$

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

$$d = \max_{x \in \{-1, +1\}^N} \sup_r \frac{\mu(B(x, r))}{\mu(B(x, r/2))} \leq \frac{1}{1/K} = \log_2 K. \quad (2.7)$$

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 Bresler et al. (2014) (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 2.2. 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

$$\mathbb{P}\left(\bigcup_{ij} \{v_{ij} \notin [.4, .6]\}\right) \leq \sum_{ij} \mathbb{P}\left(v_{ij} \notin [.4, .6]\right) \leq \binom{K}{2} \exp(-\Theta(N)), \quad (2.8)$$

where the first inequality is due to a union bound, and the second to a Chernoff bound. Hence, with high probability we get that

$$d \geq \frac{\mu(\mathcal{B}(x, .7))}{\mu(\mathcal{B}(x, .35))} \underset{\text{w.h.p.}}{=} \log_2 \frac{1}{1/K} = \log_2 K. \quad (2.9)$$

By Example 2.1 we also have that $d \leq \log_2(K)$, and hence we can conclude that with high probability we have that $d = \log_2(K)$.

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 Dataset² and for the MovieLens 1M Dataset³. For the MovieLens dataset we consider the only movies that have been rated by at least 750 users (to ensure some density).

In both datasets we calculated the empirical doubling dimension d_i (that is, the smallest d_i such that $\mu(\mathcal{B}(i, 2r)) \leq 2^{d_i} \mu(\mathcal{B}(i, r))$ for each r) around each item i . Under a simple noise assumption, figs. 2.2 and 2.3 show that all the d_i tend to be small. The appendix A.2 describes the precise experiments.

■ 2.4 Low Rank and Doubling Dimension

As mentioned in Section 1.6, a common assumption behind matrix factorization assumptions is that the matrix has low rank. In this section we would like to draw a connection between rank properties of the rating matrix L and the doubling dimension of the item space *induced* by L . In particular, we will show that low doubling dimension is a weaker requirement than low right binary rank of L . To that end, we will first define what we mean by the item space induced by a rating matrix.

Definition 2.3. (Item Space induced by Rating Matrix) Let L be an $N \times M$ binary rating

²Goldberg et al. (2001), and data available on <http://goldberg.berkeley.edu/jester-data/>

³Riedl and Konstan (1998), and data available on <http://grouplens.org/datasets/movielens/>

Figure 2.2: Jester Doubling Dimensions

Figure 2.3: MovieLens Doubling Dimensions

matrix. Then the item space μ induced by L is defined by

$$\mu(x) = \frac{1}{M} \sum_{j=1}^M \mathbf{1}_{L_j=x}, \text{ for each } x \in \{-1, +1\}^N. \quad (2.10)$$

where L_j is the j^{th} column of L .

That is, the item space induced by a rating matrix assigns mass to item types according to the empirical frequency of the item type in L . For instance, the matrix

$$L = \begin{pmatrix} -1 & -1 & -1 & +1 \\ -1 & +1 & -1 & +1 \\ +1 & -1 & +1 & +1 \end{pmatrix} \quad (2.11)$$

has its induced distribution

$$\mu_L(x) = \begin{cases} 1/2 & \text{if } x = [-1, -1, +1]^T \\ 1/4 & \text{if } x = [-1, +1, -1]^T \\ 1/4 & \text{if } x = [+1, +1, +1]^T. \end{cases} \quad (2.12)$$

The example below shows that there are rating matrices of high binary rank, but whose corresponding doubling dimension is constant (at most 2). Consider the $N \times N$ matrix L_N whose j^{th} column's first j entries are $+1$ and the remaining -1 . That is, each of its columns different from its adjacent columns in exactly one entry. For instance, for

$N = 4$ we have

$$L_4 = \begin{pmatrix} +1 & +1 & +1 & +1 \\ -1 & +1 & +1 & +1 \\ -1 & -1 & +1 & +1 \\ -1 & -1 & -1 & +1 \end{pmatrix}. \quad (2.13)$$

The matrix L_N clearly has rank N . However, the doubling dimension of its canonical item space is at most 2. We can see that because for each x and $r \in \{0, \dots, N-1\}$

$$\frac{1+r}{N} \leq \mu \left(\mathcal{B} \left(x, \frac{r}{N} \right) \right) \leq \frac{1+2r}{N}, \quad (2.14)$$

which we can use to in turn conclude that

$$d \leq \max_x \max_r \log_2 \left(\frac{\mu \left(\mathcal{B} \left(x, \frac{2r}{N} \right) \right)}{\mu \left(\mathcal{B} \left(x, \frac{r}{N} \right) \right)} \right) \leq \log_2 \left(\frac{\frac{1+4r}{N}}{\frac{1+r}{N}} \right) \leq \log_2 2^2 = 2. \quad (2.15)$$

Hence, this shows that the doubling dimension of the induced item space can be substantially smaller than the rank of the rating matrix. Since the rank is a form of count of the number of item types, this again reinforces the fact that the number of types is not particularly important, but the geometric structure between them is.

Description of Algorithm

In this chapter we describe the algorithm ITEM-ITEM-CF. The algorithm carries out a certain procedure over increasingly larger epochs (blocks of time), where the epochs are denoted by $\tau \in \mathbb{N}^+$. The epochs are of increasing time length, and in each epoch the algorithm uses a careful balance of “Explore” and “Exploit” steps. Further, the algorithm begins with a period which is purely exploratory and aimed at the cold-start problem, and shortly after it the algorithm can already give meaningful recommendations to users.

In the “Explore” steps of epoch τ , a partition of items is produced that will be used in the subsequent epoch. More precisely, in the epoch τ , the “Explore” recommendations construct a partition $\{P_k^{(\tau+1)}\}$ of a set of items. In this partition we want that if two items i and j are in the same block $P_k^{(\tau+1)}$, then $\gamma_{ij} \leq \varepsilon_{\tau+1}$, where the ε_τ is defined below and is the target precision at which the algorithm will construct the partition.

In the “Exploit” recommendations of epoch τ , the partition $\{P_k^{(\tau)}\}$ produced in the previous epoch is used for providing recommendations to users. The algorithm follows a simple policy, where recommendations to a user u are made as follows: u samples a random item i from a random block $P_k^{(\tau)}$, and only if u likes i then the rest of $P_k^{(\tau)}$ is recommended in future “exploit” steps. After all of $P_k^{(\tau)}$ has been recommended, the algorithm will use the future “exploit” steps for u to have u sample random items in random blocks until u likes another item j in $P_{k'}^{(\tau)}$. The algorithm will then recommend the rest of $P_{k'}^{(\tau)}$, and repeat the procedure until the epoch ends or it runs out of items, in which case it makes random recommendations. The pseudo-code of algorithm is as follows.

Figure 3.1: Partition resulting from exploration in the previous epoch

Figure 3.2: During exploitation, items in P_k are recommended only when from a block when a user likes an item in the block

ITEM-ITEM-CF(N)

1 **Algorithm Parameters:**

$$\varepsilon_N = \left(\frac{2^{5d+18}}{\nu} \cdot 630(2d+11)(d+2)^4 \frac{1}{N} \right)^{\frac{1}{d+5}}, C = \frac{\nu}{47} \frac{1}{20}$$

$$\varepsilon_\tau = \max\left(\frac{1}{2^\tau}, \varepsilon_N\right) \cdot C, \text{ for } \tau \geq 1 \text{ (target accuracy for epoch)}$$

$$M_\tau = \frac{2^{\max(3.5d, 8)}}{\nu} \frac{(3d+1)}{\varepsilon_\tau^{d+2}} \ln\left(\frac{2}{\varepsilon_\tau}\right), \text{ for } \tau \geq 1 \text{ (number of items introduced in epoch)}$$

$$D_\tau = \frac{\nu}{2} M_\tau, \text{ for } \tau \geq 1 \text{ (duration of epoch)}$$

2 **Cold-Start Epoch:**

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

3 **Subsequent Epochs:**

for $\tau \geq 1$

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

do $u = \text{random user}$

w.p. $1 - \varepsilon_\tau$: *exploit* the partition $\{P_k^{(\tau)}\}$ to give precise recommendation to u (described in Section 3.1)

w.p. ε_τ : give an *explore* recommendation to u to construct the partition $\{P_k^{(\tau+1)}\}$ (described in Section 3.2)

■ 3.1 Exploit

Given the random user U_t at time t in some epoch τ , ITEM-ITEM-CF provides exploit recommendations by using the partition $\{P_k^{(\tau)}\}$ in an intuitive way: whenever the user has liked an item, the algorithm recommends the rest of this item's block (consisting of hopefully similar items) to U_t in this user's subsequent exploit steps.

More formally, the exploit steps to U_t during epoch τ have U_t sample a uniformly random item i from a uniformly random block $P_k^{(\tau)}$ that U_t hasn't yet sampled. If U_t likes i , then the algorithm recommends the remaining items of $P_k^{(\tau)}$ in subsequent exploit steps to U_t . If there are no further items to recommend to U_t (i.e. all of $P_k^{(\tau)}$ has been recommended, or U_t hasn't yet liked any items), then the algorithm has U_t sample another random item from a random block that U_t hasn't yet sampled. If there are no more blocks to be sampled (which we prove rarely happens), then the algorithm recommends an arbitrary item that has not yet been recommended to U_t .

■ 3.2 Explore

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 $y_{ij} \leq \varepsilon_{\tau+1}$. We later prove that this can be done by executing the routine 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).

For 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.

We now describe MAKE-PARTITION. This routine has two main steps:

- (create ε -net): finds a set of items $\mathcal{C}$ such that with high probability for an item i drawn from the item space there is an item $j \in \mathcal{C}$ with $y_{ij} \leq \varepsilon_{\tau+1}$.
- (create partition): using $\mathcal{C}$ as a starting point, the algorithm creates a partition $\{P_k\}$ of items such that with high probability all items in the same block are $\varepsilon_{\tau+1}$ close to each other.

We now describe MAKE-PARTITION in more detail, and the pseudocode is below (where the routines SIMILAR and GET-NET are described later). First it finds a net for the item space (using the subroutine GET-NET), and then partitions a set of items using the net as the starting point. For each of the M items sampled, MAKE-PARTITION finds an item $i \in \mathcal{C}$ that is similar to the item j sampled, and assigns j to the partition block P_i of items similar to i (in case there is more than one such item i , the algorithm breaks ties 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 is important in Lemma 5.1 in ensuring a good cold-start performance.

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  $i \in \mathcal{C}$ 
4      do let  $P_i = \emptyset$ 
5  for each  $j \in \mathcal{M}$ 
6      do  $S_j = \{i \mid \text{SIMILAR}(i, j, 0.6\varepsilon, \frac{\delta}{4M|\mathcal{C}|}) \text{ returns TRUE}\}$ 
7          if  $|S_j| > 0$  then
8               $P_i = P_i \cup j$ , for  $i$  chosen u.r. from  $S_j$ 
9  if  $|P_i| > 1/\varepsilon$  for any  $i$ , partition  $P_i$  into blocks of size at most  $\frac{1}{2\varepsilon}$ 
10 return  $\{P_i\}$ 
```

It is crucial that blocks in the partition are not too small. This is important 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 the 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 2.3, which shows that there are not many items in the net very close to any given item j .

The subroutine SIMILAR is used throughout MAKE-PARTITION, and it aims at deciding whether most users agree on two items i and j (that is, whether most users like or dislike i and j together). It does so by sampling many random users and estimating the fraction that disagree on the two items.

SIMILAR($i, j, \varepsilon, \delta$)

```

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

```

The subroutine GET-NET below is a natural greedy procedure for obtaining 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(ε, δ)

```

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

```


Correctness of Explore

In this chapter we will prove that with high probability the procedure MAKE-PARTITION indeed produces a partition of similar items during each epoch. In Section 4.1 we prove that SIMILAR succeeds in deciding whether two items are close to each other. In Section 4.2 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 Section 4.3 we prove that with high probability during any given epoch there will be enough explore recommendations.

■ 4.1 Guarantees for SIMILAR

The procedure SIMILAR (which we restate below for convenience) is used throughout GET-NET and MAKE-PARTITION, and it is aimed at testing whether two items are approximately ε -close to each other.

```

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       $X_u = \mathbb{I}\{L_{u,i} \neq L_{u,j}\}$ 
5  if  $\frac{1}{q_{\varepsilon, \delta}} \sum_{\text{sampled } u} X_u \geq 0.9\varepsilon$  then
6      return FALSE
7  if  $\frac{1}{q_{\varepsilon, \delta}} \sum_{\text{sampled } u} X_u < 0.9\varepsilon$  then
8      return TRUE

```

The Lemma below shows that given two items i and j , SIMILAR indeed succeeds in telling the items are similar when $\gamma_{i,j} \leq 0.8\varepsilon$, and that the items are not similar when $\gamma_{i,j} \geq \varepsilon$.

Lemma 4.1. *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 $\mathbb{P}(S_{ij}) \geq 1 - \delta$, and*

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

Proof. Let us begin with case (i), where $\gamma_{ij} \leq 0.8\varepsilon$. 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 $\mathbb{E}\left(\sum_{n=1}^{q_{\varepsilon,\delta}} \mathbb{I}(A_n)\right) \leq 0.8\varepsilon q_{\varepsilon,\delta}$. Then, by the Chernoff Bound (stated in Theorem A.1), we get

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

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

$$\mathbb{P}(S_{ij}^c \mid \gamma_{ij} \leq 0.8\varepsilon) = \mathbb{P}\left(\sum_{n=1}^{q_{\varepsilon,\delta}} \mathbb{I}(A_n) \geq 0.9\varepsilon q_{\varepsilon,\delta}\right) \leq \delta.$$

Let us now consider case (ii), where $\gamma_{ij} \in [k\varepsilon, (k+1)\varepsilon]$. 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

$$\mathbb{P}(S_{ij} \mid \gamma_{ij} \in [k\varepsilon, (k+1)\varepsilon]) = \mathbb{P}\left(\sum_{n=1}^{q_{\varepsilon,\delta}} \mathbb{I}(A_n) \leq 0.9\varepsilon q_{\varepsilon,\delta}\right) \quad (4.1)$$

$$\leq \mathbb{P}\left(\sum_{n=1}^{q_{\varepsilon,\delta}} \mathbb{I}(A_n) \leq (1 - .1) \mathbb{E}\left(\sum_{n=1}^{q_{\varepsilon,\delta}} \mathbb{I}(A_n)\right)\right) \quad (4.2)$$

$$\leq \exp\left(-\frac{.1^2}{2 + .1} \mathbb{E}\left(\sum_{n=1}^{q_{\varepsilon,\delta}} \mathbb{I}(A_n)\right)\right) \leq \exp\left(-\frac{1}{210} q_{\varepsilon,\delta} k \varepsilon\right). \quad (4.3)$$

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

$$\exp\left(-\frac{1}{210} q_{\varepsilon,\delta} k \varepsilon\right) \leq \frac{\delta}{4} \left(\frac{1}{4k}\right)^d \frac{1}{k^2}. \quad (4.4)$$

By taking natural log of both sides we get

$$\frac{1}{210} q_{\varepsilon, \delta} k \varepsilon \geq d \ln(4k) + \ln\left(\frac{4k^2}{\delta}\right), \quad (4.5)$$

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

$$\frac{1}{210} q_{\varepsilon, \delta} k \varepsilon \geq (d+1) \ln\left(\frac{16k^3}{\delta}\right). \quad (4.6)$$

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

$$\frac{1}{210} q_{\varepsilon, \delta} k \varepsilon \geq \frac{1}{210} 630 \frac{d+1}{\varepsilon} \ln\left(\frac{1}{\delta}\right) k \varepsilon = 3(d+1)k \ln\left(\frac{1}{\delta}\right). \quad (4.7)$$

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

$$\mathbb{P}(S_{ij} \mid \gamma_{ij} \in [k\varepsilon, (k+1)\varepsilon]) = \mathbb{P}\left(\sum_{n=1}^q \mathbb{I}(A_n) \leq 0.9\varepsilon q_{\varepsilon, \delta}\right) \leq \frac{\delta}{4} \left(\frac{1}{4k}\right)^d \frac{1}{k^2}, \quad (4.8)$$

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 4.2 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 4.2. *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_{ij} be the event that $\text{SIMILAR}(i, J, \varepsilon, \delta)$ returns TRUE. Then we have $\mathbb{P}(\gamma_{i,J} \geq \varepsilon \mid S_{ij}) \leq \delta$.*

Proof. By Bayes' rule we get

$$\mathbb{P}(\gamma_{i,J} \geq \varepsilon \mid S_{ij}) = \frac{\mathbb{P}(\gamma_{i,J} \geq \varepsilon, S_{ij})}{\mathbb{P}(\gamma_{i,J} \geq \varepsilon, S_{ij}) + \mathbb{P}(\gamma_{i,J} < \varepsilon, S_{ij})}, \quad (4.9)$$

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

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

holds, we get

$$\mathbb{P}(\gamma_{i,J} \geq \varepsilon \mid S_{ij}) = \frac{1}{1 + \frac{\mathbb{P}(\gamma_{i,J} < \varepsilon, S_{ij})}{\mathbb{P}(\gamma_{i,J} \geq \varepsilon, S_{ij})}} \leq \frac{1}{1 + \frac{1-\delta}{\delta}} = \delta. \quad (4.10)$$

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

$$\mathbb{P}(\gamma_{ij} < \varepsilon, S_{ij}) \geq \mathbb{P}(S_{ij} \mid \gamma_{ij} \leq \varepsilon/2) \mu(\mathcal{B}(i, \varepsilon/2)) \geq \mathbb{P}(S_{ij} \mid \gamma_{ij} = \varepsilon/2) \mu(\mathcal{B}(i, \varepsilon/2)), \quad (4.11)$$

and

$$\mathbb{P}(\gamma_{ij} \geq \varepsilon, S_{ij}) = \sum_{k=0}^{\lceil \log_2(\frac{1}{\varepsilon}) \rceil} \mathbb{P}(S_{ij} \mid \gamma_{ij} \in [2^k \varepsilon, 2^{k+1} \varepsilon)) \mu(\mathcal{B}(i, 2^{k+1} \varepsilon) - \mathcal{B}(i, 2^k \varepsilon)) \quad (4.12)$$

$$\leq \sum_{k=0}^{\lceil \log_2(\frac{1}{\varepsilon}) \rceil} \mathbb{P}(S_{ij} \mid \gamma_{ij} = 2^k \varepsilon) \mu(\mathcal{B}(i, 2^{k+1} \varepsilon)). \quad (4.13)$$

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 4.1 we get that

$$\mathbb{P}(S_{ij} \mid \gamma_{ij} = \varepsilon/2) \mu(\mathcal{B}(i, \varepsilon/2)) \geq (1 - \delta) p. \quad (4.14)$$

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

$$\mathbb{P}(\gamma_{ij} \geq \varepsilon, S_{ij}) \leq \sum_{k=0}^{\lceil \log_2(\frac{1}{\varepsilon}) \rceil} \mathbb{P}(S_{ij} \mid \gamma_{ij} = 2^k \varepsilon) \mu(\mathcal{B}_\gamma(i, 2^{k+1} \varepsilon)) \quad (4.15)$$

$$\leq \sum_{k=0}^{\lceil \log_2(\frac{1}{\varepsilon}) \rceil} \mathbb{P}(S_{ij} \mid \gamma_{ij} = 2^k \varepsilon) (2^{k+2})^d p. \quad (4.16)$$

We now use the second half of Lemma 4.1, and arrive at

$$\mathbb{P}(\gamma_{ij} \geq \varepsilon, S_{ij}) \leq \sum_{k=0}^{\lceil \log_2(\frac{1}{\varepsilon}) \rceil} \left(\frac{\delta}{4} \left(\frac{1}{4 \cdot 2^k} \right)^d \frac{1}{2^{2k}} \right) (2^{k+2})^d p \leq p \frac{\delta}{4} \sum_{k=0}^{\infty} \frac{1}{2^{2k}} \leq p \frac{\delta}{2}. \quad (4.17)$$

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

$$\delta \mathbb{P}(S_{ij}, \gamma_{ij} < \varepsilon) \geq \delta p (1 - \delta) \geq \frac{\delta}{2} p (1 - \delta) \geq (1 - \delta) \mathbb{P}(S_{ij}, \gamma_{ij} \geq \varepsilon), \quad (4.18)$$

which completes the proof. ■

■ 4.2 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` and restated below, succeeds at producing an ε -net of items with high probability.

`GET-NET`(ε, δ)

```

1   $\mathcal{C} = \emptyset$ 
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   $\text{COUNT} = 0$ 
4  while  $\text{COUNT} \leq \text{MAX-WAIT}$  and  $|\mathcal{C}| < \text{MAX-SIZE}$ 
5      do draw item  $i$  from  $\mu$ 
6          if SIMILAR( $i, j, \varepsilon, \delta'$ ) for any  $j \in \mathcal{C}$  then
7               $\text{COUNT} = \text{COUNT} + 1$ 
8          else  $\mathcal{C} = \mathcal{C} \cup i$ 
9               $\text{COUNT} = 0$ 
10 return  $\mathcal{C}$ 
```

Lemma 4.3. *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 $\{S_{i,j}, \gamma_{ij} > 0.5\varepsilon\} \cup \{S_{i,j}^c, \gamma_{ij} < 0.8\varepsilon\}$, 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) $\mathbb{P}(P^c \mid E^c) = 0$, and
- (C) $\mathbb{P}(C^c \mid E^c) \leq \delta/2$,

which together show that GET-NET(ε, δ) 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 4.1 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 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

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

and we will show that

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

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

which together prove (C).

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

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

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

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

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

Proof of (C₂): Consider now the case in which $|C_{final}| < \text{MAX-SIZE}$. 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 $i \in C_r$ but the algorithm nevertheless terminated and returned 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 C_r . Then $C^c \subset \bigcup_{r=0}^{|C_{final}|} T_r$, and hence

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

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 C_r , then the ball $B(j, \varepsilon/5)$ must have significant mass which is all also not close to any item in 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 C_r was sampled, the event T_r happens only if j is judged to be similar to some $c \in C_r$. 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. In

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  $i \in \mathcal{C}$ 
4      do let  $P_i = \emptyset$ 
5  for each  $j \in \mathcal{M}$ 
6      do  $S_j = \{i \mid \text{SIMILAR}(i, j, 0.6\varepsilon, \frac{\delta}{4M|\mathcal{C}|}) \text{ returns TRUE}\}$ 
7          if  $|S_j| > 0$  then
8               $P_i = P_i \cup j$ , for  $i$  chosen u.r. from  $S_j$ 
9  if  $|P_i| > 1/\varepsilon$  for any  $i$ , partition  $P_i$  into blocks of size at most  $\frac{1}{2\varepsilon}$ 
10 return  $\{P_i\}$ 
```

We now prove that with high probability the procedure MAKE-PARTITION (restated above for convenience) 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 4.4. *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 $y_{i,j} \leq 1.2 \cdot \varepsilon$,*
- (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 4.3, 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 $S_{i,j}$ is the event that routine $\text{SIMILAR}(i, j, \varepsilon/2, \delta/4M^2)$ 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 some block P_k has size less than $\frac{1}{2\varepsilon}$, and let $B = \bigcup_k B_k$. 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

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

and showing

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

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

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

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

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 $\frac{1}{2\varepsilon}$ before we split the large blocks at the end of the algorithm. 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} . Even though the algorithm does break up large blocks at the end, never does it make a block of size above $\frac{1}{2\varepsilon}$ into smaller than that. Hence

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

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

$$\mathbb{P}(X_{c,n} \mid C^c, E^c) \geq (\varepsilon/12)^d \quad (4.19)$$

allows us to prove (A), and we then prove eq. (4.19). Note that the $\{X_{c,n}\}_n$ are independent, and hence the sum $\sum_{n=1}^M \mathbb{I}\{X_{c,n}\}$ 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

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

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

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

as we wished.

Proof of eq. (4.19): We can lower bound $\mathbb{P}(X_{c,n} \mid C^c, E^c)$ as

$$\mathbb{P}(X_{c,n} \mid C^c, E^c) \geq \mathbb{P}(X_{c,n} \mid C^c, E^c, \gamma_{c,j_n} \leq \varepsilon/2) \mathbb{P}_{j_n}(\gamma_{c,j_n} \leq \varepsilon/2), \quad (*)$$

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) $\text{SIMILAR}(c, j_n, \varepsilon/2, \delta/(4M|\mathcal{C}|))$ 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 $\text{SIMILAR}(c, j_n, \varepsilon/2, \delta/(4M|\mathcal{C}|))$ returns TRUE, and hence $\mathbb{P}(X_{c,n} \mid C^c, E^c, \gamma_{c,j_n} \leq \varepsilon/2) = 1/K$, where K is the number of items in $\mathcal{C}$ for which SIMILAR also returned TRUE. By Proposition 2.3 (with r as $\frac{6}{5}\varepsilon/2$ and ε as $\varepsilon/2$ in the claim), we get that $K \leq \left(\frac{8}{\varepsilon}\right)^d \mu(B(c, \varepsilon/2)) \left(\frac{5}{4} + \frac{6}{5}\right)^d \leq \left(\frac{12}{\varepsilon}\right)^d \mu(B(c, \varepsilon/2))$. Now noting that in eq. (*) $\mathbb{P}_{j_n}(\gamma_{c,j_n} \leq \varepsilon/2) = \mu(B(c, \varepsilon/2))$, we arrive at

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

which proves eq. (4.19) 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 $\mathbb{P}(B^c \mid C^c, E^c) = 0$.

Proof of (C): By Lemma 4.1, $\mathbb{P}(E_c, j) \leq \frac{\delta}{4} \frac{1}{(4/\varepsilon)^d} \frac{1}{M}$ 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 4.3.

Proof of (D): This follows directly from Lemma 4.3. ■

■ 4.3 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 4.5. *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

$$MP(\tau + 1) \leq \left(\frac{8}{\varepsilon_{\tau+1}} \right)^{d+1} 4 \cdot 630 (d + 1)^3 M_{\tau+1} \ln^2 \left(\frac{8}{\varepsilon_{\tau+1}} \right) \ln(M_{\tau+1}) \quad (4.20)$$

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

$$\frac{4}{\varepsilon_\tau} \frac{1}{M_\tau} MP(\tau+1) \leq \left(\frac{16}{\varepsilon_\tau}\right)^{d+1} 4 \cdot 630 (d+1)^3 2^{d+2} \ln^2\left(\frac{16}{\varepsilon_\tau}\right) \ln\left(\frac{2^{3d+13}}{\nu \varepsilon_\tau^{d+1}} \ln(1/\varepsilon_\tau)\right) \quad (4.21)$$

$$\leq 2^{5d+12} \cdot 630 (d+1)^3 \left(\frac{1}{\varepsilon_\tau}\right)^{d+2} \ln^2\left(\frac{16}{\varepsilon_\tau}\right) \ln\left(\frac{2^{3d+13}}{\nu \varepsilon_\tau^{d+2}}\right). \quad (4.22)$$

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 $\mathbb{P}\left(\text{Bin}(D_\tau N, \varepsilon_\tau) \leq \frac{1}{2} \varepsilon_\tau D_\tau N\right)$ 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 $D_\tau \geq \frac{9}{\varepsilon_\tau} \ln\left(\frac{1}{\varepsilon_{\tau+1}}\right)$. ■

Regret Analysis

■ 5.1 Quick Recommendations Lemma

In chapter 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 5.1 (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*

$$\mathbb{E}[\mathcal{R}^{(\tau)}(T)] \leq \frac{47\varepsilon_\tau}{v} T \quad (5.1)$$

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

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

$$\mathbb{E}[\mathcal{R}^{(\tau)}(T)] \leq \varepsilon_\tau T + \mathbb{E}[\mathcal{R}'^{(\tau)}(T)]. \quad (5.2)$$

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 $\mathcal{E}_\tau$ be the event that the partition $\{P_k\}$ to be used during epoch τ satisfies the conditions specified in Lemma 4.4 (with $M = M_\tau$, $\varepsilon = \varepsilon_\tau$, $\delta = \varepsilon_\tau$). Then we get

$$\mathbb{E}[\mathcal{R}^{(\tau)}(T)] \leq \varepsilon_\tau T + \mathbb{E}[\mathcal{R}'^{(\tau)}(T)] \leq 2\varepsilon_\tau T + \mathbb{E}[\mathcal{R}'^{(\tau)}(T) \mid \mathcal{E}_\tau], \quad (5.3)$$

where the last inequality is due to Lemma 4.4, which guarantees that $\mathbb{P}(\mathcal{E}_\tau) \leq \varepsilon_\tau$.

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

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

where $\mathcal{R}'^{(\tau)}_u(T)$ 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 $8\frac{T}{D_\tau} |\{P_k^{(\tau)}\}|$ different blocks from $\{P_k\}$ 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). Then we get

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

We dedicate Lemma 5.3 to showing that $\frac{1}{N} \sum_u \mathbb{P}(g_{u,T}^c | \mathcal{E}_\tau) \leq \frac{5}{v} \varepsilon_\tau$. It hence suffices to show that for each $T > T_{\min_\tau}$ we have

$$\frac{1}{N} \sum_u \mathbb{E} [\mathcal{R}'^{(\tau)}_u(T) | \mathcal{E}_\tau, g_{u,T}] \leq \frac{40}{v} \varepsilon_\tau T, \quad (\star)$$

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

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

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

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

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 $\mathbb{E} [W_{u,k,T} | \mathcal{E}_\tau, g_{u,T}, s_{u,k,T}^c] \mathbb{P}(s_{u,k,T}^c | \mathcal{E}_\tau, g_{u,T})$ is absent from the expression above is because $\mathbb{E} [W_{u,k,T} | \mathcal{E}_\tau, g_{u,T}, s_{u,k,T}^c] = 0$ 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 $8 \frac{T}{D_\tau} |\{P_k^{(\tau)}\}|$ blocks and hence

$$\mathbb{P}(s_{u,k,T}^c \mid \varepsilon_\tau, g_{u,T}) \leq \frac{8 \frac{T}{D_\tau} |\{P_k^{(\tau)}\}|}{|\{P_k^{(\tau)}\}|} = 8 \cdot \frac{T}{D_\tau}. \quad (5.4)$$

We will now use this bound and we get

$$\frac{1}{N} \sum_u \mathbb{E}[\mathcal{R}_u^{(\tau)}(T) \mid \varepsilon_\tau, g_{u,T}] \leq \frac{1}{N} \frac{8 \cdot T}{D_\tau} \sum_u \sum_k \mathbb{E}[W_{u,k,T} \mid \varepsilon_\tau, g_{u,T}, s_{u,k,T}], \quad (5.5)$$

which we can in turn reduce to

$$\stackrel{\text{Lemma 5.2}}{\leq} \frac{1}{N} \frac{8 \cdot T}{D_\tau} \sum_u \sum_k 1.2(1 + \varepsilon_\tau |P_k|)N \stackrel{\text{Lemma 4.4}}{\leq} 20 \varepsilon_\tau T \frac{M_\tau}{D_\tau} \stackrel{D_\tau = \frac{v}{2} M_\tau}{=} \frac{40}{v} \varepsilon_\tau T, \quad (5.6)$$

where the first inequality is a simple application of Lemma 5.2, the second inequality follows from Lemma 4.4, which guarantees that $|P_k| \geq 1/\varepsilon_\tau$, and the remaining from arithmetic. This completes the proof of eq. (★), as we wished. ■

The lemma below was used in Lemma 5.1. 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 5.2 (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*

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

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 also 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

$$\mathbb{E} \left[\sum_u A_{u,k} + W_{u,k} | s_{u,k} \right] = \underbrace{\sum_u \frac{(|P_k| - \ell_{u,k})}{|P_k|}}_{\leq N} + 2 \sum_u \frac{\ell_{u,k} (|P_k| - \ell_{u,k})}{|P_k|}, \quad (5.7)$$

which in turn at most

$$N + \frac{2}{|P_k|} \sum_{\ell=0}^{|P_k|} n_k(\ell) \cdot \ell (|P_k| - \ell), \quad (5.8)$$

where $n_k(\ell)$ is the number of users who like exactly ℓ item in P_k .

We will now show that $\sum_{\ell=0}^{|P_k|} n_k(\ell) \cdot \ell (|P_k| - \ell) \leq \binom{|P_k|}{2} \varepsilon N$, which will conclude the proof. We can now use the property that all items in P_k are within ε of each other, and we arrive at

$$\frac{2}{|P_k|} \sum_{\ell=0}^{|P_k|} n_k(\ell) \cdot \ell (|P_k| - \ell) \stackrel{\text{by definition}}{=} \frac{2}{|P_k|} \sum_u \sum_{i,j \in P_k} \mathbb{I}\{L_{u,i} \neq L_{u,j}\} \quad (5.9)$$

$$= \frac{2}{|P_k|} \sum_{i,j \in P_k} \underbrace{\sum_u \mathbb{I}\{L_{u,i} \neq L_{u,j}\}}_{= \gamma_{ij} \cdot N} \leq \frac{2}{|P_k|} \sum_{i,j \in P_k} \varepsilon N \leq \frac{2}{|P_k|} \binom{|P_k|}{2} \varepsilon N \leq \varepsilon |P_k| N. \quad (5.10)$$

Hence, putting it all together we get

$$\mathbb{E} \left[\sum_u A_{u,k} + W_{u,k} | s_{u,k} \right] = \underbrace{\sum_u \frac{(|P_k| - \ell_{u,k})}{|P_k|}}_{\leq N} + \underbrace{2 \sum_u \frac{\ell_{u,k} (|P_k| - \ell_{u,k})}{|P_k|}}_{\varepsilon |P_k| N} = (\varepsilon |P_k| + 1) N, \quad (5.11)$$

as desired. $\blacksquare$

The lemma below was also needed in the proof of Lemma 5.1.

Lemma 5.3 (Auxiliary Claim). *Consider an arbitrary epoch τ , and let $g_{u,T}$ be the event that by the $(TN)^{th}$ exploit recommendation of epoch τ user u has tried at most $8\frac{T}{D_\tau}|\{P_k^{(\tau)}\}|$ blocks from the partition $\{P_k\}$ constructed during the $\text{MAKE-PARTITION}(M_\tau, \varepsilon_\tau, \varepsilon_\tau)$ of the previous epoch, and let $\mathcal{E}_\tau$ be the event that $\{P_k\}$ satisfies the conditions specified in Lemma 4.4. Then*

$$\frac{1}{N} \sum_u \mathbb{P}(g_{u,T}^c | \mathcal{E}_\tau) \leq \frac{5}{v} \varepsilon_\tau$$

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

Proof. 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 $0.9vM_\tau$ among the items in $\{P_k\}$. Furthermore, let $H_{u,T}$ be the event that by the TN^{th} exploit recommendation there are still at least $\frac{v}{5}M_\tau$ 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 $T \in (\frac{1}{v} \ln(\frac{1}{\varepsilon_\tau v}), D_\tau]$

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

$$(B) \quad \frac{1}{N} \sum_u \mathbb{P}(H_{u,T}^c | N_{u,T}, \mathcal{E}_\tau) \leq \frac{3}{v} \varepsilon_\tau$$

$$(C) \quad \mathbb{P}(N_{u,T}^c | \mathcal{E}_\tau) \leq \varepsilon_\tau,$$

from which the lemma follows.

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

$$\{g_{u,T}^c, \mathcal{E}_\tau, N_{u,T}, H_{u,T}\} \subseteq \left\{ \sum_{n=1}^{8\frac{T}{D_\tau}|\{P_k^{(\tau)}\}|} |P_{k_n}| \cdot \mathbb{I}\{\ell_{u,k,T}\} \leq 1.1T, \mathcal{E}_\tau, N_{u,T}, H_{u,T} \right\}, \quad (5.12)$$

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

$$\mathbb{P}(g_{u,T}^c | \mathcal{E}_\tau, N_{u,T}, H_{u,T}) \leq \mathbb{P}\left(\sum_{n=1}^{8\frac{T}{D_\tau}|\{P_k^{(\tau)}\}|} |P_{k_n}| \cdot \mathbb{I}\{\ell_{u,k,T}\} \leq 1.1T \mid \mathcal{E}_\tau, N_{u,T}, H_{u,T} \right), \quad (5.13)$$

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

First, note that by conditioning on $\mathcal{E}_\tau$, we are guaranteed that $|P_{k_n}| \geq 1/\varepsilon_\tau$ by Lemma 4.4. Hence

$$\mathbb{P} \left(\sum_{n=1}^{8 \frac{T}{D_\tau} |\{P_k^{(r)}\}|} |P_{k_n}| \cdot \mathbb{I}\{\ell_{u,k,T}\} \leq 1.1T \mid \mathcal{E}_\tau, H_{u,T}, N_{u,T} \right) \leq \mathbb{P} \left(\sum_{n=1}^{8 \frac{T}{D_\tau} |\{P_k^{(r)}\}|} \mathbb{I}\{\ell_{u,k,T}\} \leq 1.1T \varepsilon_\tau \mid \mathcal{E}_\tau, H_{u,T}, N_{u,T} \right). \quad (5.14)$$

We will now show that

(A₁) For each n , $\mathbb{P}(\ell_{u,k_n,T} \mid \mathcal{E}_\tau, H_{u,T}, N_{u,T})$ 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

$$\mathbb{P} \left(\sum_{n=1}^{16 \frac{T}{D_\tau} \left(\frac{12}{\varepsilon}\right)^d} X_n < 1.1T \frac{2}{M_\tau} \frac{12^d}{\varepsilon_\tau^{d+1}} \log(2/\varepsilon_\tau) \right) \leq C_a \varepsilon_\tau \quad (5.15)$$

Proof of (A₁): Since we are conditioning on $H_{u,T}$, there are still $0.2\nu M_\tau$ liked in unsampled blocks, and since we are conditioning on $\mathcal{E}_\tau$, there are at most $2 \frac{12^d}{\varepsilon_\tau^{d+1}} \log(2/\varepsilon_\tau)$ blocks, each of size at most $\frac{\varepsilon_\tau^{d+1}}{12^d} \frac{1}{\log(2/\varepsilon_\tau)}$. It is intuitive to see 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 $\mathcal{E}_\tau$, there are at most $2 \frac{12^d}{\varepsilon_\tau^{d+1}} \log(2/\varepsilon_\tau)$ blocks, and since we can “fit” $0.2\nu M_\tau$ in at most $0.2\nu M_\tau / (M_\tau (\varepsilon_\tau/12)^d) = 0.2\nu (12/\varepsilon_\tau)^d$ blocks. Since, which we can conclude that

$$\mathbb{P}(\ell_{u,k_n,T} \mid \mathcal{E}_\tau, H_{u,T}, N_{u,T}) \geq \frac{0.2\nu M_\tau}{M_\tau (\varepsilon_\tau/12)^d} \frac{(12/\varepsilon_\tau)^d}{2} = 0.1\nu. \quad (5.16)$$

Proof of (A₂): Let $\{X_n\}$ be a set of independent Bernoulli random variables with parameter 0.1ν . Then

$$\mathbb{P} \left(\sum_{n=1}^{8 \frac{T}{D_\tau} |\{P_k^{(r)}\}|} X_n < 1.1T \varepsilon_\tau \right) \leq \mathbb{P} \left(\sum_{n=1}^{8 \frac{T}{D_\tau} |\{P_k^{(r)}\}|} X_n < (1 - 0.25) \mathbb{E} \left[\sum_{n=1}^{8 \frac{T}{D_\tau} |\{P_k^{(r)}\}|} X_n \right] \right), \quad (5.17)$$

where the inequality is due to $D_\tau \triangleq \frac{\nu}{2}M_\tau$. Now by the Chernoff bound we get

$$\exp\left(-\frac{0.25^2}{2+0.25}\mathbb{E}\left[\sum_{n=1}^{\lceil 8\frac{T}{D_\tau}|\{P_k^{(\tau)}\}|\rceil} X_n\right]\right) \leq \exp\left(-\frac{0.8\nu}{18}\frac{T}{D_\tau}|\{P_k^{(\tau)}\}|\right). \quad (5.18)$$

By Lemma 4.4 we know that $|\{P_k^{(\tau)}\}| \geq M_\tau/\varepsilon_\tau$, and again since $D_\tau \triangleq \frac{\nu}{2}M_\tau$ we get

$$\mathbb{P}\left(\sum_{n=1}^{\lceil 8\frac{T}{D_\tau}|\{P_k^{(\tau)}\}|\rceil} X_n < 1.1T\varepsilon_\tau\right) \leq \exp\left(-\frac{12}{\varepsilon_\tau}T\right) \leq \varepsilon_\tau$$

where the last inequality is due to $T > \frac{12}{\varepsilon_\tau} \ln(1/\varepsilon_\tau)$.

Proof of (b): First, for each u it is clear that $\mathbb{P}(H_{u,T}^c \mid N_{u,T}) \leq \mathbb{P}(H_{u,D_\tau}^c \mid N_{u,D_\tau})$ holds, since by end of the epoch the user will have explored the most. Now note that by conditioning on N_{u,D_τ} (which is the event that user u has been recommended at most $1.05T$ items *and* that at least $0.55\nu M_\tau$ items in the partition are likable to u), then the event H_{u,D_τ}^c can happen only if

$$\sum_k A_{k,u} \mathbb{I}\{s_{u,k,D_\tau}\} \geq 0.4\nu M_\tau,$$

and hence

$$\mathbb{P}(H_{u,D_\tau}^c \mid N_{u,D_\tau})$$

Note that since we are conditioning on $N_{u,T}$, the event that at least $0.9\nu M$ items are likable to u and that u is recommended at most $1.05\nu M_\tau$ items during epoch τ , then H_{u,D_τ} can only happen if $\sum_k A_{k,u} \mathbb{I}\{s_{u,k,D_\tau}\} \geq 0.35\nu M_\tau$, where $s_{u,k,T}$ is the event that user u has sampled an item from block P_k by the TN^{th} recommendation of epoch τ . We will hence show that

$$\frac{1}{N} \sum_u \mathbb{P}\left(\sum_k A_{k,u} \mathbb{I}\{s_{u,k,D_\tau}\} \geq 0.35\nu M_\tau \mid \mathcal{E}_\tau, N_{u,T}\right) \leq \frac{3}{\nu} \varepsilon_\tau. \quad (5.19)$$

By Markov's inequality we get that

$$\mathbb{P}\left(\sum_k A_{k,u} \mathbb{I}\{s_{u,k,D_\tau}\} \geq 0.35\nu M_\tau \mid \mathcal{E}_\tau, N_{u,T}\right) \leq \frac{\mathbb{E}[\sum_k A_{k,u} \mid \mathcal{E}_\tau, s_{u,k,D_\tau}]}{0.35\nu M_\tau}, \quad (5.20)$$

where in the last inequality the conditioning on $N_{u,T}$ was removed because $A_{u,k}$ depends

only on $s_{u,k,T}$ and the properties of P_k . Hence, we get that

$$\begin{aligned} \frac{1}{N} \sum_u \mathbb{P} \left(\sum_k A_{k,u} \mathbb{I}\{s_{u,k,D_\tau}\} \geq 0.35\nu M_\tau \mid \mathcal{E}_\tau, N_{u,T} \right) &\leq \frac{1}{N} \sum_u \frac{\mathbb{E}[\sum_k A_{k,u} \mid \mathcal{E}_\tau, s_{u,k,D_\tau}]}{0.35\nu M_\tau} \\ &= \frac{1}{N} \frac{\sum_k \sum_u \mathbb{E}[A_{k,u} \mid \mathcal{E}_\tau, s_{u,k,D_\tau}]}{0.35\nu M_\tau} \leq \frac{\sum_k |P_k| \varepsilon_\tau N}{0.35\nu M_\tau N} \leq \frac{3}{\nu} \varepsilon_\tau, \end{aligned} \quad (5.21)$$

as desired, and where the penultimate inequality is due to Lemma 5.2.

Proof of (c): The event $N_{u,T}^c$ happens whenever a user u has been recommended more than $0.55\nu M_\tau$ 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

$$\mathbb{P} \left(\left| \text{Bin} \left(ND_\tau, \frac{1}{N} \right) \right| \geq 1.1D_\tau \right) \leq \exp \left(-\frac{.1^2}{2+0.1} D_\tau \right), \quad (5.22)$$

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:

$$\mathbb{P}(\text{Bin}(M_\tau, \nu) < 0.9\nu M_\tau) \leq \exp \left(-\frac{.1^2}{2+0.1} \frac{1}{2} M_\tau \nu \right) \leq \varepsilon_\tau/2, \quad (5.23)$$

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

■ 5.2 Main Results

In the previous section we proved Lemma 5.1, 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 5.1 (Regret Upper Bound of ITEM-ITEM-CF). Suppose assumptions (A1) – (A2) are satisfied. Then ITEM-ITEM-CF achieves expected regret

$$\mathbb{E}[\mathcal{R}(T)] \leq \begin{cases} T_{\min} + \alpha(T - T_{\min})^{\frac{d+1}{d+2}} \log_2(T - T_{\min}) & T_{\min} < T \leq T_{\max} \\ \beta + \varepsilon_N(T - T_{\max}) & T > T_{\max} \end{cases}, \quad (5.24)$$

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

The reader is directed to the proof below for the exact constants.

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

$$MP(1) \triangleq \left(\frac{8}{\varepsilon_1} \right)^{d+1} 4 \cdot 630 (d+1)^3 M_1 \ln^2 \left(\frac{8}{\varepsilon_1} \right) \ln(M_1) \quad (5.25)$$

recommendations (by Lemma 4.4), 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 $\frac{C}{2} \frac{1}{2^\tau}$, until it plateaus at ε_N when $\frac{C}{2} \frac{1}{2^\tau} \leq \varepsilon_N$. Hence

$$\tau^* \triangleq \lceil \log_2 \frac{C/2}{\varepsilon_N} \rceil \quad (5.26)$$

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

$$T_{MP(1)} + \sum_{\tau=1}^{\tau^*-1} D_\tau \geq g(v, d) N^{\frac{d}{d+5}} \triangleq T_{max}. \quad (5.27)$$

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

$$\tau^* \geq \frac{1}{d+5} \log_2 \left(\left(\frac{v}{47} \right)^{d+5} \frac{v}{630(2d+11)(d+2)^4} \frac{1}{2^{5d+18}} \cdot N \right), \quad (5.28)$$

and hence

$$T_{MP(1)} + \sum_{\tau=1}^{\tau^*-1} D_\tau \geq \sum_{\tau=1}^{\tau^*-1} D_\tau = \sum_{\tau=1}^{\tau^*-1} \frac{v}{2} M_\tau. \quad (5.29)$$

Now recall that $D_\tau = \frac{v}{2} M_\tau$ and $M_\tau \triangleq C_M \frac{1}{\varepsilon_\tau^{d+2}} \ln(\frac{2}{\varepsilon_\tau})$, where $C_M = \frac{2^{\max(3.5d, 8)}}{v} (3d+1)$, and for $\tau \leq \tau^*$ we have $\varepsilon_\tau = C_\varepsilon / 2^\tau$, where $C_\varepsilon = v/(47 \cdot 20)$. Then we get

$$\begin{aligned} T_{MP(1)} + \sum_{\tau=1}^{\tau^*-1} D_\tau &\geq \frac{v}{2} C_M (1/C_\varepsilon)^{d+2} \sum_{\tau=1}^{\tau^*-1} 2^{\tau(d+2)} \geq \frac{v}{4} C_M (1/C_\varepsilon)^{d+2} 2^{\tau^*(d+2)} \\ &\geq \underbrace{\frac{v}{4} C_M (1/C_\varepsilon)^{d+2} \left(\frac{v}{630(2d+11)(d+2)^4} \frac{1}{2^{5d+18}} \right)^{\frac{d+2}{d+5}}}_{g(v, d)} \cdot N^{\frac{d+2}{d+5}} \triangleq T_{max}, \end{aligned} \quad (5.30)$$

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 5.1, for $T \in [T_{min}, T_{max}]$, where $T_{min} = T_{MP} + T_{min,1}$, the expected regret satisfies

$$\mathcal{R}(T) - T_{MP} \leq \frac{47}{v} \sum_{\tau=1}^{\tau(T)} \varepsilon_\tau D_\tau, \quad (5.31)$$

which we can further bound as

$$\mathcal{R}(T) - T_{MP} \leq \frac{C_M}{2} \log_2 \left(\frac{2^{\tau(T)}}{2C_\varepsilon} \right) \sum_{\tau=1}^{\tau(T)} 2^{\tau(d+1)} \leq \frac{C_M}{2} \log_2 \left(\frac{2^{\tau(T)}}{2C_\varepsilon} \right) 2^{(\tau(T)+1)(d+1)}. \quad (5.32)$$

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

$$\begin{aligned} \mathcal{R}(T) &\leq T_{MP} + \frac{C_M}{2} \log_2 \left(\frac{1}{C_\varepsilon(d+2)} \frac{1}{\log(2/C_\varepsilon)} \frac{T - T_{MP}}{C_M} \right) 2^{2(d+1)} 2^{(\tau(T)+1)(d+1)} \quad (5.33) \\ &\leq T_{MP} + \underbrace{\frac{C_M}{2} \log_2 \left(\frac{1}{C_\varepsilon(d+2)} \frac{1}{\log(2/C_\varepsilon)} \frac{1}{C_M} \right)}_{\triangleq C'} 2^{4(d+1)} \log_2(T - T_{MP}) 2^{\frac{d+1}{d+2} \log_2 \left(\frac{T - T_{MP}}{C_M} \frac{1}{\log(2/C_\varepsilon)} \right)} \\ &\leq T_{MP} + \underbrace{C' \left(\frac{1}{C_M} \frac{1}{\log(2/\varepsilon)} \right)^{\frac{d+1}{d+2}}}_{\triangleq \alpha} (T - T_{MP})^{\frac{d+1}{d+2}} \log_2(T - T_{MP}), \end{aligned}$$

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

The case $T > T_{max}$ now follows. Recall that by Lemma 5.1 we get

$$\mathcal{R}(T) \leq T_{MP} + \frac{47}{v} \sum_{\tau=1}^{\tau(T)} \varepsilon_\tau D_\tau, \quad (5.34)$$

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

$$\mathcal{R}(T) \leq T_{MP} + \frac{47}{v} \sum_{\tau=1}^{\tau^*-1} \varepsilon_\tau D_\tau + \frac{47}{v} \sum_{\tau=\tau^*}^{\tau(T)} \varepsilon_\tau D_\tau \quad (5.35)$$

$$\leq \underbrace{T_{MP} + \alpha T_{max}^{\frac{d+1}{d+2}} \log_2(T_{max})}_{\triangleq \beta} + \varepsilon_N(T - T_{max}), \quad (5.36)$$

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

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 $\Delta > 0$ we have $\mathbb{E}[\mathcal{R}^{(\mathcal{A})}(T + \Gamma + \Delta) - \mathcal{R}^{(\mathcal{A})}(T)] \leq 0.1(\Delta + \Gamma)$.

Theorem 5.2 (Cold-Start Performance). Suppose assumptions (A1)–(A2) are satisfied. Then the algorithm ITEM-ITEM-CF has cold-start time $T_{\text{cold-start}} = \frac{f(v,d)}{N} + \tilde{O}(1/v)$.

Proof. First recall the usual definitions: $D_\tau = \frac{v}{2}M_\tau$, $T_\tau = T_{MP} + \sum_{\tau' < \tau} D_{\tau'}$, and $T_{MP} = f(v, d)$, where $f(v, d)$ is the number of recommendations required for the initial MAKE-PARTITION call (as stated in Lemma 4.4), and $T_{\min, \tau} = \frac{12}{\varepsilon_\tau} \ln\left(\frac{1}{\varepsilon_\tau}\right)$ (as stated in Lemma 5.1).

We will show how

(i) $\mathbb{E}[\mathcal{R}(T_{MP} + T_{\min, 1} + \Delta) - \mathcal{R}(T_{MP})] \leq 0.1(T_{\min, 1} + \Delta)$, 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) $\mathbb{E}[\mathcal{R}(T_\tau + \Delta) - \mathcal{R}(T_\tau)] \leq 0.1(\Delta + D_{\tau-1})$, for $T \leq D_\tau$ and $\tau \geq 2$

allow us to prove the theorem, where (i) follows directly from Lemma 5.1, and (ii) will be proved momentarily.

Let τ^* be the epoch to which $T_{\text{cs}} + T$ belongs and let $\Delta = T_{MP} + T_{\min, 1} + T - T_{\tau^*}$. Then (i) and (ii) we can get the desired result for the subsequent epochs $\tau^* > 1$ as

$$\begin{aligned} \mathbb{E}[\mathcal{R}(T_{MP} + T_{\min, 1} + T) - \mathcal{R}(T_{MP})] &\stackrel{\Delta \triangleq T_{\text{cs}} + T - T_{\tau^*}}{\leq} \underbrace{\mathbb{E}[\mathcal{R}(T_2) - \mathcal{R}(T_{MP})]}_{\leq 0.1 \cdot D_1 \text{ by (A)}} \quad (5.37) \\ &+ \mathbf{1}_{\tau^* \geq 3} \left(\sum_{\tau=2}^{\tau^*-1} \underbrace{\mathbb{E}[\mathcal{R}(T_\tau) - \mathcal{R}(T_{\tau-1})]}_{\leq 0.1(D_\tau + D_{\tau-1}) \text{ by (B)}} \right) + \underbrace{\mathbb{E}[\mathcal{R}(T_{\tau^*} + \Delta) - \mathcal{R}(T_{\tau^*})]}_{\leq 0.1(\Delta + D_{\tau^*-1}) \text{ by (B)}} \\ &\leq 0.1 \left(\Delta + 2 \sum_{\tau=1}^{\tau^*-1} D_\tau \right) \leq 0.2 \cdot T, \end{aligned}$$

and hence $T_{\text{cold-start}}$ is at most $T_{MP} + T_{\min, 1} = f(v, d)/N + \tilde{O}(1/v)$, as desired.

Proof of (ii): Lemma 5.1 tells us that for $\Delta \in (T_{\min, \tau}, D_\tau)$ we have that $\mathbb{E}[\mathcal{R}(T_\tau + \Delta) - \mathcal{R}(T_\tau)] \leq \frac{47}{v} \varepsilon_\tau \Delta$, and since $\mathbb{E}[\mathcal{R}(T_\tau + T_{\min, \tau}) - \mathcal{R}(T_\tau)] \leq \frac{47}{v} \varepsilon_\tau T_{\min, \tau}$ we get that, for $\Delta < D_\tau$,

that

$$\mathbb{E}[\mathcal{R}(T_\tau + \Delta) - \mathcal{R}(T_\tau)] \leq \frac{47}{v} \varepsilon_\tau T_{\min, \tau} + \underbrace{\frac{47}{v} \varepsilon_\tau \Delta}_{\leq 0.05 \text{ for } \tau \geq 2}. \quad (5.38)$$

Furthermore $T_{\min, \tau} \triangleq \frac{12}{\varepsilon_\tau} \ln\left(\frac{1}{\varepsilon_\tau}\right) \leq 0.05 \frac{300}{\varepsilon_\tau} \ln\left(\frac{1}{\varepsilon_\tau}\right) = 0.05 D_{\tau-1}$, and thus

$$\mathbb{E}[\mathcal{R}(T_\tau + \Delta) - \mathcal{R}(T_\tau)] \leq 0.05 (\Delta + D_{\tau-1}), \quad (5.39)$$

as we wished. ■

Hence, the algorithm ITEM-ITEM-CF has cold-start time not increasing with N and d . Finally, the regret bound in Theorem 5.1 has an asymptotic linear regime. The result below shows that with a finite number of users such linear regime is unavoidable.

Theorem 5.3 (Asymptotic Linear Regime is unavoidable). Consider an item space μ satisfying assumptions (A1)–(A2). Then any online algorithm must have expected asymptotic regret $\mathbb{E}[\mathcal{R}(T)] \geq c(v, d)T$.

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

$$\mathbb{E}[\mathcal{R}(T)] = \frac{1}{N} \mathbb{E} \left[\sum_{t=1}^{TN} \frac{1}{2} (1 - L_{U_t, I_t}) \right] \quad (5.40)$$

$$= \frac{1}{N} \mathbb{E} \left[\sum_{k=1}^{k_T} \sum_{t=1}^{TN} \frac{1}{2} \mathbb{I}\{I_t = i_k\} (1 - L_{U_t, i_k}) \right] \geq \frac{1}{N} \mathbb{E} \left[\sum_{k=1}^{k_T} \frac{1}{2} (1 - L_{U_{T_k}, i_k}) \right], \quad (5.41)$$

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 $\mathbb{E} \left[\frac{1}{2} (1 - L_{U_{T_k}, i_k}) \right] \geq 1 - 2v$, 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

$$\mathbb{E}[\mathcal{R}(T)] \geq \frac{1 - 2v}{N} k_T. \quad (5.42)$$

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

$$\mathbb{E}[\mathcal{R}(T)] \geq \underbrace{\frac{1-2\nu}{N}}_{c(\nu,d)}, \quad (5.43)$$

as we wished. ■

Conclusion

In this section we further discuss our results and provide suggestions for future work in the subject.

■ 6.1 Discussion

In this thesis we provided a formal expected regret analysis of ITEM-ITEM-CF, a simple recommendation algorithm following the item-item paradigm in collaborative filtering. We first proved that unless some structural assumption is made, no online recommendation algorithm can have sublinear expected for any period of time.

We then motivated using the doubling dimension d of the item space as a measure of how structured the data is, and showed that the algorithm achieves expected regret $\tilde{O}\left(T^{\frac{d+1}{d+2}}\right)$ for a period of time that increases with the number of users. Furthermore, we prove that the asymptotic linear regime, following the sublinear regime, is unavoidable.

Figure 6.1: Expected Regret of ITEM-ITEM-CF, as proven in Theorems 5.1 and 5.2.

■ 6.1.1 Two roles of Doubling Dimension

In Section 2.1 we make the assumption that μ has small doubling dimension. This is crucial in the proofs for two distinct reasons. First, it guarantees that the ε -net *grows slowly* as ε -decreases (polynomially in $1/\varepsilon$). This is important in Lemma 4.4 for ensuring that the blocks of ε similar items are large enough and the reward for exploration pays off (as when the algorithm finds an item liked by the user, it can now recommend the many other items in the block). It is in this “slowly-growing ε -net” sense that doubling dimension/covering numbers are used, for instance, as Kleinberg, Slivkins, and Upfal (2013); Bubeck, Munos, Stoltz, and Szepesvari (2011). Second, the doubling dimension ensures that SIMILAR, which is a random projection, works with high probability (as proved in Lemma 4.2) in preserving relative distances. It is in this *random projection preserves relative distances* sense that it is used, for instance, in Dasgupta and Sinha (2014).

■ 6.1.2 Decision Making and Learning with Mixtures

We would now like to discuss how separation assumptions play a role in decision making and learning, and we will compare our work to some of the progress in learning Gaussian Mixture Models (GMM).

In parameter estimation of a mixture of two n -dimensional Gaussians

$$\mathcal{N}_{\text{mix}}(x; w_1, \mu_1, \mu_2, \Sigma_1, \Sigma_2) \sim w_1 \mathcal{N}(x; \mu_1, \Sigma_1) + (1 - w_1) \mathcal{N}(x; \mu_2, \Sigma_2), \quad (6.1)$$

one would like to draw enough samples from $\mathcal{N}_{\text{mix}}$ and then estimate the parameters $w_1, \mu_1, \mu_2, \Sigma_1$, and Σ_2 . This line of work was initiated in Dasgupta (1999), where a separation assumption of the form $\|\mu_1 - \mu_2\| \geq \tilde{\Omega}(\sqrt{n})$ was made. More relaxed separation assumptions were then made, where, for instance, Arora and Kannan (2005) achieved $\|\mu_1 - \mu_2\| \geq \tilde{\Omega}(n^{1/4})$. This sequence of works, and the sequence subsequently described sequence is nicely surveyed in Moitra (2014).

This line of work is in contrast to the later papers that, instead of parameter estimation, look for a normal approximation $\tilde{\mathcal{M}}$ to $\mathcal{M}$ in total variation distance (called proper learning). Kalai, Moitra, and Valiant (2010) and Belkin and Sinha (2010) independently first have guarantees for gaussian proper learning without separation assumptions, and Li and Schmidt (2015) gave the first such guarantee in the agnostic setting (when $\mathcal{N}_{\text{mix}}$ may be not exactly a Gaussian).

An important lesson from the contrast of these two lines of work (with vs. without separation assumptions) is that when one wishes to learn an approximation in total

variation distance (which is enough for the making many decision based on distribution), then one may do away with separation assumptions. A similar contrast exists in our work and that of Bresler et al. (2014). In their work, prior to recommending they first cluster users, and require a separation between users types to enable perfect clustering. In our work, we do not require perfect clustering, and show that approximately learning item types suffices to provide quality recommendations.

■ 6.1.3 Explore-Exploit

Algorithmically, our work also clearly highlights an exploration-exploitation trade-off, as we spend part of the time explicitly exploring the item space to make sure that item similarities are calculated based on sufficient data. This theoretical point relates to the practical difficulty associated with calculating similarities based on too little data. In practice (Koren and Bell (2011) mention this), an arbitrary penalty term is placed to make sure that inaccurate similarities are not being considered.

■ 6.2 Future Works

Our algorithm exclusively exploits item-item similarities. It would be desirable to have a matching lower bound for such algorithms, in the spirit of lower bound for multi-armed bandits in metric spaces provided in Kleinberg, Slivkins, and Upfal (2013) and more generally in Bubeck, Munos, Stoltz, and Szepesvari (2011). Furthermore, many practitioners use a hybrid of user-user and item-item paradigms (Wang, De Vries, and Reinders, 2006; Verstrepen and Goethals, 2014), 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 have had good results (Melville, Mooney, and Nagarajan, 2002). 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.

Appendix

■ A.1 Chernoff Bound

The following is a standard version of Chernoff Bound (McDiarmid, 1998).

Theorem A.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 $\varepsilon \geq 0$,

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

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

■ A.2 Empirical Doubling Dimension Experiments

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. 2.2 and 2.3 then show the histogram of the $\{d_i\}$.

Bibliography

- 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.
- Sanjeev Arora and Ravi Kannan. Learning mixtures of separated nonspherical gaussians. *The Annals of Applied Probability*, 15(1A):69–92, 2005.
- Sanjeev Arora, Rong Ge, Ravindran Kannan, and Ankur Moitra. Computing a nonnegative matrix factorization—provably. In *STOC*, 2012.
- Mikhail Belkin and Kaushik Sinha. Polynomial learning of distribution families. *FOCS*, 2010.
- Alejandro Bellogin and Javier Parapar. Using graph partitioning techniques for neighbour selection in user-based collaborative filtering. In *RecSys*, 2012.
- Gerard Biau, Benoit Cadre, and Laurent Rouviere. Statistical analysis of k-nearest neighbor collaborative recommendation. *The Annals of Statistics*, 38(3):1568–1592, 2010.
- Guy Bresler, George Chen, and Devavrat Shah. A latent source model for online collaborative filtering. In *NIPS*, 2014.
- Sébastien Bubeck and Nicolo Cesa-Bianchi. Regret analysis of stochastic and non-stochastic multi-armed bandit problems. *arXiv preprint arXiv:1204.5721*, 2012.
- Sébastien Bubeck, Rémi Munos, Gilles Stoltz, and Csaba Szepesvari. X-armed bandits. *JMLR*, 2011.
- 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.

- Nicolò Cesa-Bianchi and Ohad Shamir. Efficient transductive online learning via randomized rounding. In *Empirical Inference*, pages 177–194. Springer, 2013.
- Nicolo Cesa-Bianchi, Claudio Gentile, and Giovanni Zappella. A gang of bandits. In *NIPS*, 2013.
- Onkar Dabeer. Adaptive collaborating filtering: The low noise regime. In *ISIT*, 2013.
- Abhinandan S Das, Mayur Datar, Ashutosh Garg, and Shyam Rajaram. Google news personalization: scalable online collaborative filtering. In *WWW*, 2007.
- Sanjoy Dasgupta. Learning mixtures of gaussians. In *FOCS*, 1999.
- Sanjoy Dasgupta and Kaushik Sinha. Randomized partition trees for nearest neighbor search. *Algorithmica*, 2014.
- David Donoho and Victoria Stodden. When does non-negative matrix factorization give a correct decomposition into parts? In *NIPS*, 2003.
- Michael D Ekstrand, John T Riedl, and Joseph A Konstan. Collaborative filtering recommender systems. *Foundations and Trends in Human-Computer Interaction*, 4(2): 81–173, 2011.
- Claudio Gentile, Shuai Li, and Giovanni Zappella. Online clustering of bandits. *ICML*, 2014.
- David Goldberg, David Nichols, Brian M. Oki, and Douglas Terry. Using collaborative filtering to weave an information tapestry. *Commun. ACM*, 1992.
- Ken Goldberg, Theresa Roeder, Dhruv Gupta, and Chris Perkins. Eigentaste: A constant time collaborative filtering algorithm. *Information Retrieval*, 2001.
- Sariel 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.
- Elad Hazan, Satyen Kale, and Shai Shalev-Shwartz. Near-optimal algorithms for online matrix prediction. *COLT*, 2012.
- Juha Heinonen. *Lectures on analysis on metric spaces*. Springer, 2001.
- Adam Tauman Kalai, Ankur Moitra, and Gregory Valiant. Efficiently learning mixtures of two gaussians. In *STOC*, 2010.

- Jon Kleinberg and Mark Sandler. Using mixture models for collaborative filtering. In *STOC*, 2004.
- Robert Kleinberg, Aleksandrs Slivkins, and Eli Upfal. Bandits and experts in metric spaces. *arXiv preprint arXiv:1312.1277*, 2013.
- Yehuda Koren and Robert Bell. Advances in collaborative filtering. In *Recommender Systems Handbook*, pages 145–186. Springer US, 2011.
- Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *IEEE*, 2009.
- Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *NIPS*, 2001.
- Jerry Li and Ludwig Schmidt. A nearly optimal and agnostic algorithm for properly learning a mixture of k gaussians, for any constant k . *Manuscript*, 2015.
- Greg Linden, Brent Smith, and Jeremy York. Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, 7(1):76–80, 2003.
- Colin McDiarmid. Concentration. In *Probabilistic methods for algorithmic discrete mathematics*. Springer, 1998.
- Prem Melville, Raymond J Mooney, and Ramadass Nagarajan. Content-boosted collaborative filtering for improved recommendations. In *AAAI/IAAI*, 2002.
- Ankur Moitra. Algorithmic aspects of machine learning. *Online Manuscript*, 2014.
- J Riedl and J Konstan. Movielens dataset, 1998.
- Aleksandrs Slivkins. Contextual bandits with similarity information. *The Journal of Machine Learning Research*, 2014.
- William R Thompson. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika*, pages 285–294, 1933.
- Stephen A Vavasis. On the complexity of nonnegative matrix factorization. *SIAM Journal on Optimization*, 2009.
- Koen Verstrepen and Bart Goethals. Unifying nearest neighbors collaborative filtering. In *Recsys*, 2014.

Jun Wang, Arjen De Vries, and Marcel Reinders. Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *ACM SIGIR*, 2006.

Tong Zhang. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *ICML*, 2004.