Using Mixture Models for Collaborative Filtering

Jon Kleinberg*
Department of Computer Science
Cornell University, Ithaca, NY, 14853
kleinber@cs.cornell.edu

Mark Sandler
Department of Computer Science
Cornell University, Ithaca, NY 14853
sandler@cs.cornell.edu

ABSTRACT

A *collaborative filtering system* at an e-commerce site or similar service uses data about aggregate user behavior to make recommendations tailored to specific user interests. We develop recommendation algorithms with provable performance guarantees in a probabilistic *mixture model* for collaborative filtering proposed by Hoffman and Puzicha. We identify certain novel parameters of mixture models that are closely connected with the best achievable performance of a recommendation algorithm; we show that for any system in which these parameters are bounded, it is possible to give recommendations whose quality converges to optimal as the amount of data grows.

All our bounds depend on a new measure of independence that can be viewed as an L_1-analogue of the smallest singular value of a matrix. Using this, we introduce a technique based on generalized pseudoinverse matrices and linear programming for handling sets of high-dimensional vectors. We also show that standard approaches based on L_2 spectral methods are not strong enough to yield comparable results, thereby suggesting some inherent limitations of spectral analysis.

Categories and Subject Descriptors

F.2.2 [**Analysis of Algorithms and Problem Complexity**]: Non-numerical Algorithms and Problems; H.3.3 [**Information Storage and Retrieval**]: Clustering, Information Filtering

General Terms

Algorithms, theory

Keywords

Mixture models, latent class models, collaborative filtering, clustering, text classification, singular value decomposition, linear programming

*Supported in part by a David and Lucile Packard Foundation Fellowship and NSF ITR Grant IIS-0081334.

STOC'04, June 13–15, 2004, Chicago, Illinois, USA.

1. INTRODUCTION

Collaborative Filtering. A Web site or other on-line service that receives extensive traffic has the potential to analyze the resulting usage data for the benefit of its user population. One of the most common applications of such analysis is *collaborative filtering*: a Web site offering items for sale or download can analyze the aggregate decisions of the whole population, and then make *recommendations* to individual users of further items that they are likely to be interested in. The recommendations made to a specific user are thus based not just on his or her own previous actions, but also on *collaborative information* — the information collected from other users in the system. Perhaps the most well-known example of collaborative filtering in a practical setting is Amazon's purchase recommendations [9], which is based on rules of the form "users who are interested in item X are also likely to be interested in item Y." This is a simple but highly visible example of the notion; a wide range of more elaborate schemes have been studied and implemented as well, based on more extensive profiles of users and more subtle notions of similarity among items (see e.g. [11]).

Given the extensive experimental work in this area, there has been relatively little theoretical analysis of the problem of collaborative filtering. In particular, Hofmann and Puzicha have proposed a highly expressive probabilistic *mixture model* for collaborative filtering [6], but previous work has left open a large gap between the general form of this model and the limited special cases in which one can obtain algorithms with provable guarantees [7, 8].

In this paper, we provide the first recommendation algorithms with strong provable performance guarantees in a large and natural sub-class of mixture models. Focusing on a sub-class of the set of all mixture models is necessary, since it is known that collaborative filtering algorithms cannot achieve good performance in all instances of the mixture model [7]. Given this, we identify a novel parameter of mixture models that, in a fairly precise sense, "controls" the extent to which recommendation algorithms can achieve near-optimal performance, and we quantify our results in terms of this parameter, obtaining strong bounds whenever it is bounded away from 0.

In a line of work that parallels the use of mixture models for this problem, Azar et al. and Drineas et al. have considered a formalism in which user behavior follows a latent

linear model [2, 4]. This work is not directly comparable to ours, both because of these differences in the underlying generative model, as well as differences in the objective function and the way in which data is gathered from users. We discuss this comparison further below, focusing on the relationship between the spectral methods employed by [2, 4] and the mixture model parameters we develop here.

We now define the underlying mixture model that we use here, and then describe our results.

Mixture models. *Mixture models* have a long history in statistics and machine learning [10]; for our purposes, we cast the description in terms of Hofmann and Puzicha's mixture model formulation of collaborative filtering [6].

To define the model, we imagine a system with a set of M items (e.g. books) that are available for sale to a set of N *users*. Clearly if a user's interest in one item were unrelated to her interest in any other, there would be no hope of making recommendations; so it is necessary to posit some underlying generative process by which users select items. We therefore assume that there is a latent set of k clusters, which we can think of as the "genres" that users may be interested in.

Formally, each cluster c is a *distribution* over all the items, assigning probability $w(i|c)$ to each item i. These are the probabilities with which a user seeking something in genre c will choose each of the items; for example, if c corresponds to "computer books," then the distribution specifies that readers seeking computer books will choose *The Art of Computer Programming* with probability x, *The Mythical Man-Month* with probability y, and so on. Note that each cluster assigns a probability to *each* item, so these can be heavily overlapping clusters. (For example, the *The Mythical Man-Month* might also have a large probability in a cluster c' corresponding to "management.") The set of all probabilities induced by all clusters will be represented in a $M \times k$ *weight matrix* W, whose (i, c) entry is simply the probability $w(i|c)$.

Dually, each user u is represented by a distribution over clusters, with her probability (or preference) for cluster c denoted by $p(c|u)$. This reflects the fact that, at different times, the same user can be seeking items of different genres. These probabilities are encoded in a $k \times N$ *preference matrix* P.

For each user u, we now construct a history of s prior selections in the following natural way. For each of s iterations, user u does the following: first she selects a genre c with probability $p(c|u)$, and then she selects an item i with probability $w(i|c)$. For example, a user might first select *The Mythical Man-Month* because she was looking for something in the genre "management"; then select *The Art of Computer Programming* because she was looking for something in the genre "computer books"; and finally select *2001: A Space Odyssey* because she was looking for something in the genre "science fiction."

We thus have a model with underlying parameters (the weight matrix and preference matrix), and these generate a history of selections for each user. Finally, we need to formalize the goal in making recommendations.

Kumar et al. [8] proposed the following objective function: the system should recommend a single item i_u to each user u, and the *utility* of this recommendation is simply the probability that user u would have selected i_u herself. Since i_u could potentially have been selected as part of each of the k clusters, this probability is

$$\sum_{c \in \mathcal{C}} p(c|u) w(i_u|c), \tag{1}$$

where $\mathcal{C}$ is the set of clusters. The goal is to maximize the total utility of all recommendations. Clearly, if the system knew the full weight and preference matrices, then it could achieve the obvious optimum: recommending the item to each user for which the expression in Equation (1) is maximized. Kumar et al. proposed investigating the performance of recommendation algorithms relative to this optimum for two variants of the problem, depending on which parameters are unknown:

- *Semi-omniscient algorithms*, which know the weight matrix but not the preference matrix. This corresponds to a setting in which the operators of the collaborative filtering system have done some initial modeling of relationships among the items, but do not know anything about the user population. As we will see, in the full mixture model even this is quite challenging.
- *The Strong Benchmark*, in which the system knows neither the weight matrix nor the preference matrix.

Finally, we briefly discuss the relative sizes of the parameters under consideration. Algorithms that only begin making good recommendations after a user has selected an enormous number of items are clearly of limited interest; we want the number s of selections made by each user to remain bounded independently of the total number of items. On the other hand, it seems natural that if the number of items grows, then more and more users may be needed to gain sufficient information about the structure of the items. Thus, we parametrize the mixture model so that the number of selections s required from a single user may depend on the number of clusters k and the performance guarantee we are seeking, but is bounded independently of the number of items M and the number of users N; and the number of users we require in order to achieve good performance may grow as a function of the number of items M.

The mixture model is thus a very expressive framework for representing the collaborative filtering problem: although items are grouped into genres, these genres can overlap arbitrarily, and items can have partial membership in many different genres. Similarly, different selections by a single user might require different "explanations" in terms of these genres.

The expressiveness of the mixture model also poses a problem, since it has been shown that no algorithm can give near-optimal recommendations in all instances of the mixture model [7]. The only positive results to date have been for the special case in which the distributions induced by the clusters have *disjoint* support [7, 8] — in other words, each item belongs to a single cluster, and so there is no real "mixture" taking place. Our goal here is to find a much more general setting in which it is possible to design effective algorithms, and to do this we identify two further parameters of the mixture model. We show that when these parameters are both bounded, strong performance guarantees can be obtained; and both parameters are necessary in the sense that bounding either one alone does not suffice.

Our Results. Our first main result is a polynomial-time, semi-omniscient recommendation algorithm: given access to

the weight matrix and to a sufficient number of selections per user, the algorithm provides recommendations of utility at least $(1-\varepsilon)$ times optimal with probability at least $1-\delta$. The number of selections required per user is a function of ε, δ, the number of clusters k, and the two additional parameters alluded to above:

- *Cluster imbalance.* For each cluster c, consider the largest probability w_c that it assigns to any single item. We define $\mathbf{w}_+ = \max_{c\in\mathcal{C}} w_c$ and $\mathbf{w}_- = \min_{c\in\mathcal{C}} w_c$, and we call the ratio $\mathcal{W} = \frac{\mathbf{w}_+}{\mathbf{w}_-}$ the *cluster imbalance.*
- *Cluster independence.* We define

$$\Gamma = \min_{x\neq 0} \frac{|Wx|_1}{|x|_1} \tag{2}$$

  as a measure of linear independence between clusters. It is easy to show that if the cluster distributions have disjoint support (as in [7, 8]), then $\Gamma = 1$; on the other hand, if the distributions induced by the clusters are not linearly independent, then $\Gamma = 0$.

As we will show in the next section bounding $\mathcal{W}$ from above and Γ away from zero is natural in a sense we make precise below; roughly, any system in which these parameters are not bounded is *unstable*, and can be modified through the addition of a bounded number of items to one in which good recommendations are not possible.

Our second main result concerns the strong benchmark. Here we provide an algorithm that, given a sufficient number of users relative to the number of items, and a sufficient number of selections per user, provides recommendations of utility at least $(1-\varepsilon)$ times optimal with probability at least $1-\delta$. The number of selections needed per user is a function of ε, δ, k, $\mathcal{W}$, Γ, and one additional parameter, an analogue to Γ for the preference matrix:

- *User non-degeneracy.* By analogy with Γ, we define $\Gamma_P = \min_{x\neq 0} \frac{|xP/N|_1}{|x|_1}$, which measures how redundant the user preferences are. For example, if this parameter is 0, it means that the collection of preferences of each user for a given cluster can be computed from a fixed linear combination of the user preferences for the other clusters. Note that the use of P/N in this formula brings the normalization of P more closely into alignment with that of W, on which we computed Γ; the point is that the sum of all entries in P (without normalization) is equal to N (since each of the N columns of P corresponds to a user and sums to 1), while the sum of all entries in W is $k \ll N$ (since each of the k columns of W corresponds to a cluster and sums to 1).

The strong benchmark is more challenging than the case of semi-omniscient algorithms, and our result here is correspondingly weaker in two respects. First, in contrast to $\mathcal{W}$ and Γ, we do not know whether bounding the parameter Γ_P away from 0 is in fact necessary for obtaining strong performance guarantees. Second, while the number of selections required per user is polynomial in ε and δ, it is exponential in the number of clusters k; thus, the result should best be viewed as applying to a fixed constant number of clusters. Eliminating both these restrictions is an interesting open question.

We believe that the role of the parameter Γ in the analysis is an interesting feature of these results. One can think of Γ as an L_1-analogue of the smallest singular value of the weight matrix W, since the smallest singular value would be obtained by replacing the 1-norm in Equation (2) by the 2-norm. The parameter Γ appears to be fairly novel in these types of analyses, however, and we believe it would be interesting to study it further in its own right. In the next section we argue that, for purposes of the results here, assuming an analogous bound on the smallest (L_2) singular value would be much weaker, since there are cases where this converges to 0 while Γ remains large. This is another point of comparison with the framework of [2, 4] (which, again, posit a different underlying model and objective function): in a sense that would be interesting to put on a deeper technical foundation, the Γ parameter appears to be naturally adapted to the mixture model in much the same way that the smallest singular value is adapted to the latent linear structure used in those papers.

Finally, while we have cast these results in the language of collaborative filtering, they can also be interpreted in terms of mixture models more generally. Given the relevance of mixture models to information retrieval, computer vision, and a number of problems in statistics [10] we expect there may be further applications of the techniques here. In Section 3.2 we present preliminary computational results for a potential application in text classification.

2. MIXTURE MODELS: OVERVIEW

The goal of this section is to build intuition behind the mixture model and establish some basic facts. It is organized as follows. In the first two subsections we explain the role of the parameters defined in the introduction, and also discuss the sense in which they are essential quantities in the performance of any recommendation algorithm. The third subsection provides a brief comparison of singular values and our L_1 analogue. We note that all the examples in this section apply even to the case of semi-omniscient algorithms.

Clusters imbalance. If two users each get optimal recommendations, what is the maximum possible ratio between the utilities of these recommendations? In other words, how different might the contribution of two different users be to the total utility function? Obviously every user has preference $\geq \frac{1}{k}$ for at least one cluster; hence if we simply recommend the heaviest item in that cluster we will get utility at least $\frac{\mathbf{w}_-}{k}$. On the other hand, the total utility of item i for user u is $\sum_{c\in\mathcal{C}} w(i|c)p(c|u) \leq \sum_{c\in\mathcal{C}} \mathbf{w}_+ p(c|u) = \mathbf{w}_+$ Therefore the ratio between the contributions of two different users is at most $k\frac{\mathbf{w}_+}{\mathbf{w}_-} = k\mathcal{W}$.

We summarize this in the following lemma.

LEMMA 2.1. *For every user there exists a recommendation of utility at least $\frac{\mathbf{w}_-}{k}$ and there is no recommendation of utility more than $\mathbf{w}_+$.*

It can be shown that for any fixed function $g(k)$, one can choose $\mathcal{W}$ large enough so that in any system with cluster imbalance at least $\mathcal{W}$, and users with appropriately chosen preferences each selecting $g(k)$ items, no algorithm can give recommendations better than $O(\frac{1}{k}OPT)$ with constant probability. In fact, this holds even in the simpler *weighted model* of [7], where the cluster distributions have disjoint support. We refer the reader to [7] for an example of this.

Cluster independence and the L_1 norm. It is not difficult to construct examples of systems where Γ is small, and

no good recommendation algorithm exists. We refer the reader to [7] for an example of this. One can ask whether it is the case that good recommendations are impossible in *every* system with a small value of Γ, but this is clearly too sweeping to be the case. Consider for example an instance with two clusters that induce exactly the same distribution over items. Here we have $\Gamma = 0$, but clearly one can simply treat the two clusters as a single cluster, and good recommendations will be possible.

A related general negative result does hold, however: any system in which Γ is small is highly "unstable," in the sense that adding a bounded number of items to it will produce a system in which no good recommendation algorithm exists. More precisely, we can show that every system which has $\Gamma \leq 1/s$, where s is the number of samples per user, can be augmented with $O(k)$ items, so that it becomes impossible to give recommendations that are better than a 2-approximation in the worst case. Thus, while it is possible to have $\Gamma = 0$ and still be able to give close to optimal recommendations, such an ability is always vulnerable to the addition of just a few items.

Spectral analysis. As noted above, our definition of independence between clusters is very similar to the definition of the smallest singular value of a rectangular matrix. Indeed $\Gamma = \min_{x \neq 0} \frac{||Wx||_1}{||x||_1}$, while the smallest singular value can be defined as $\lambda = \min_{x \neq 0} \frac{||Wx||_2}{||x||_2}$. Using standard norm inequalities we immediately have $\frac{\Gamma}{\sqrt{M}} \leq \lambda \leq \Gamma\sqrt{k}$. Both inequalities are tight, but the number of clusters k is small in comparison with the total number of items M. Thus, to within a term that depends only on k, bounds expressed in terms of $\frac{1}{\Gamma}$ cannot be weaker than those expressed in terms of $\frac{1}{\lambda}$. But things can be much weaker in the opposite direction. The example in Appendix A provides a family of systems in which, as the number of items grows, Γ remains bounded by a constant while λ approaches zero. This shows a concrete sense in which bounds depending on $\frac{1}{\lambda}$ can be strictly weaker than those based on $\frac{1}{\Gamma}$.

3. A SEMI-OMNISCIENT ALGORITHM

There are a few notational conventions to which we will adhere in this and next sections:

- All items, users and clusters are numbered starting from 1. We use i and j to denote items, c and d to denote clusters, and u and v to denote users. We will also use these letters to denote matrix indices and unless specifically stated otherwise, they will "type-check" with the meaning of the index. We use capital calligraphic letters $\mathcal{I}$, $\mathcal{U}$ and $\mathcal{C}$ to denote collections of items, users and clusters respectively.

3.1 Discussion.

Our goal in this section is to give good recommendations in the case when the weight matrix W is known. For this, our analysis will need to compare two vectors (over the space of all items) associated with each user u: the *utility vector* $\mathbf{u}$, whose i^{th} entry is the probability that u will choose item i; and (after u has made s choices) the *selection vector* $\tilde{\mathbf{u}}$, whose i^{th} entry is the number of times that item i was selected in the s samples, divided by s. (Note that $\tilde{\mathbf{u}}$ is an extremely sparse vector, with almost all entries equal to 0.) Now, if we knew the utility vector, we would just recommend the entry with largest value; thus, we wish to show that we can closely approximate this value so as to make a near-optimal recommendation.

We begin with the following simple lemma.

LEMMA 3.1. *For an arbitrary user u with selection and utility vectors $\tilde{\mathbf{u}}$ and $\mathbf{u}$ respectively, and for any vector v such that $||v||_\infty < B$, if we have $s > \frac{B^2}{\varepsilon^2\delta}$ selections from this user then* $\Pr\left[|v^T\tilde{\mathbf{u}} - v^T\mathbf{u}| > \varepsilon\right] < \delta$

Proof. Indeed, we have

$$\tilde{\mathbf{u}} = \frac{1}{s}\sum_{l=1}^{s} \tilde{\mathbf{u}}_l,$$

where $\tilde{\mathbf{u}}_l$ denotes the indicator vector for the l^{th} selection. So

$$v^T\tilde{\mathbf{u}} = \frac{1}{s}\sum_{l=1}^{s} v^T\tilde{\mathbf{u}}_l,$$

where the terms in the sum are independent random variables (as user selections are independent from each other) drawn from the same distribution, and $|v^T\tilde{\mathbf{u}}_l| < B$. Therefore the variance of $v^T\tilde{\mathbf{u}}$ is at most $\frac{1}{s}B\sqrt{s}$ and hence by Chebyshev's inequality $\Pr\left[|v^T\tilde{\mathbf{u}} - v^t\mathbf{u}| > \varepsilon\right] < \frac{B^2}{\varepsilon^2 s} < \delta$. ■

In other words, this lemma shows that despite the sparseness of $\tilde{\mathbf{u}}$, we can use it to compute $v^T\mathbf{u}$ for any vector v whose coordinates have bounded absolute value.

The following is just a re-formulation of the lemma above.

COROLLARY 3.2. *Given an arbitrary user u making s selections, with selection and utility vectors $\tilde{\mathbf{u}}$ and $\mathbf{u}$, any vector v such that $||v||_\infty < B$, and any δ, we have* $\Pr\left[|v^T\tilde{\mathbf{u}} - v^T\mathbf{u}| > \frac{B}{\sqrt{s\delta}}\right] < \delta$.

The rest of our argument is based on the idea of generalized pseudoinverse matrices. For an arbitrary $M \times k$ weight matrix W of rank k, we call a $k \times M$ matrix W' a *generalized pseudoinverse*[1] of W if $W' \times W = I$. If $M = k$ then such a matrix is unique and it is simply W^{-1}. If $M > k$, then there can be infinitely many generalized pseudoinverses. We are interested in the one for which the largest absolute value of any entry is as small as possible. The following example illustrates how we intend to use such a matrix. Suppose there is a user u with selection and utility vectors $\tilde{\mathbf{u}}$ and $\mathbf{u}$. Obviously $\mathbf{u}$ is in the range of W (i.e. there exists y such that $W\mathrm{y} = \mathbf{u}$). Therefore

$$W(W'\mathbf{u}) = WW'(W\mathrm{y}) = W\mathrm{y} = \mathbf{u}.$$

Say W and W' have all elements bounded by constants w_+ and γ; then by lemma 3.1 and the Union Bound, it follows that $\frac{k^3\gamma^2}{\varepsilon^2\delta}$ selections are sufficient to have

$$||W'\tilde{\mathbf{u}} - W'\mathbf{u}||_\infty < \frac{\varepsilon}{k},$$

with probability at least $1 - \delta$. Therefore

$$||W(W'\tilde{\mathbf{u}} - W'\mathbf{u})||_\infty < w_+\varepsilon,$$

[1]We note that the standard notion of the *pseudoinverse matrix* from linear algebra is a particular instance of the generalized pseudoinverse as defined here, and different from the particular instances we will be considering

or equivalently

$$||WW'\tilde{\mathbf{u}} - \mathbf{u}||_\infty < w_+\varepsilon, \tag{3}$$

so we can reconstruct $\mathbf{u}$ with component-wise error at most $w_+\varepsilon$. We will make this more concrete after we establish the existence of a generalized pseudoinverse in which all entries are bounded.

THEOREM 3.3. *For any $M \times k$ matrix $W = \{w_{ic}\}$ such that $\Gamma = \min_{x \neq 0} \frac{|Wx|_1}{|x|_1} > 0$, the following holds:*

1. *There exists a generalized pseudoinverse $B = \{b_{cj}\}$ such that $\max |b_{cj}| < \frac{1}{\Gamma}$.*
2. *The generalized pseudoinverse matrix B minimizing $\max |b_{cj}|$ can be found in polynomial time.*

Proof. For the second part, the matrix $B = \{b_{cj}\}$ can be found by solving the following linear program:

$$\begin{cases} \sum_i b_{ci} w_{id} = \delta_{cd} & \text{for } 1 \leq c, d \leq k \\ -\gamma \leq b_{ci} \leq \gamma & \text{for } 1 \leq c \leq k,\ 1 \leq j \leq M \\ \min \gamma \end{cases},$$

where $\delta_{cd} = 1$ when $c = d$ and is equal to 0 otherwise. To prove the first part it suffices to show that the following system of linear inequalities is feasible for $\gamma \geq 1/\Gamma$.

$$\begin{cases} \sum_{i=1}^M b_{ci} w_{id} = \delta_{cd} & \text{for } 1 \leq c, d \leq k \\ -\gamma \leq b_{ci} \leq \gamma & \text{for } 1 \leq i \leq M,\ 1 \leq c \leq k \end{cases} \tag{4}$$

Obviously this system has a solution if and only if the following system has a solution for every c.

$$\begin{cases} \sum_{i=1}^M x_i w_{id} = \delta_{cd} & \text{for } 1 \leq d \leq k \\ -\gamma \leq x_i \leq \gamma & \text{for } 1 \leq i \leq M \end{cases} \tag{5}$$

Now we introduce additional variables y_i and z_i such that $y_i + z_i = 2\gamma$ and $x_i = y_i - \gamma = \gamma - z_i$. For simplicity we use vector notation $Y = (y_1, \ldots, y_M)$ and $Z = (z_1, \ldots, z_M)$ and rewrite the system in vector form:

$$\begin{cases} (Y - \vec{\gamma}, Z - \vec{\gamma}) \begin{pmatrix} W & I \\ -W & I \end{pmatrix} = (2\delta_c, \vec{0}) \\ Y \geq 0, Z \geq 0 \end{cases}, \tag{6}$$

where I is the $M \times M$ identity matrix, δ_c is the c-th row of the $k \times k$ identity matrix, and $\vec{\gamma}$ is the M-dimensional vector of the form $(\gamma, \gamma, \ldots \gamma)$. Simplifying, we have:

$$\begin{cases} (Y, Z) \begin{pmatrix} W & I \\ -W & I \end{pmatrix} = (2\delta_c, 2\vec{\gamma}) \\ Y \geq 0, Z \geq 0 \end{cases} \tag{7}$$

By Farkas's lemma this system has a solution if and only if the following dual system is infeasible.

$$\begin{cases} \begin{pmatrix} W & I \\ -W & I \end{pmatrix} \begin{pmatrix} V \\ U \end{pmatrix} \leq \vec{0} \\ (2\delta_c, 2\vec{\gamma}) \begin{pmatrix} V \\ U \end{pmatrix} > 0 \end{cases} \tag{8}$$

By expanding the first inequality we immediately have $U \leq WV \leq -U$, and hence $U \leq 0$. Therefore $||WV||_1 \leq ||U||_1 = -\sum_i u_i$ and thus

$$v_c \leq ||V||_1 \leq \frac{1}{\Gamma}||WV||_1 \leq -\frac{1}{\Gamma}\sum_{i=1}^M u_i.$$

Substituting this into second inequality we have:

$$(2\delta_c, 2\vec{\gamma}) \begin{pmatrix} V \\ U \end{pmatrix} \leq (-\frac{2}{\Gamma} + 2\gamma) \sum_i u_i \tag{9}$$

But if $\gamma \geq 1/\Gamma$, the right hand side is non-positive, and thus both constraints of (8) cannot be satisfied simultaneously; therefore for $\gamma \geq 1/\Gamma$ and every j the system (5) is feasible, and hence the desired generalized pseudoinverse B exists. ■

By the theorem, $\max w'_{ij} \leq \frac{1}{\Gamma}$, so substituting $\frac{1}{\Gamma}$ for γ in the discussion preceding (3), we have

$$||W(W'\tilde{\mathbf{u}}) - \mathbf{u}||_\infty < \mathbf{w}_+\epsilon.$$

But we know the maximal utility for every user is at least $\frac{\mathbf{w}_-}{k}$, so if we take $\epsilon = \frac{\varepsilon}{k\mathcal{W}}$, we get a recommendation of $(1 - \varepsilon)$ times the optimal total utility.

Now for completeness we present the full algorithm.

ALGORITHM 1 (SEMI-OMNISCIENT ALGORITHM).
Input: Weight matrix W, ε, δ, and for each user u a selection vector $\tilde{\mathbf{u}}$ with at least $\frac{k^5\mathcal{W}^2}{(\varepsilon\Gamma)^2\delta}$ selections.

Output: An approximately best recommendation for user u.

Description:

1 *Compute W' using the linear program of Theorem 3.3.*
2 *For user u, compute $\overline{\mathbf{u}} = WW'\tilde{\mathbf{u}}$ and recommend an item i which maximizes $\overline{\mathbf{u}}_i$.*

The correctness of this algorithm follows immediately from Theorem 3.3 and Lemmas 2.1 and 3.1.

3.2 Preliminary computational results

One application of the algorithm described in this section is to the problem of supervised text classification. To adapt the framework to this problem, we take the 'users' to be the documents, the 'items' to be all possible terms in the documents, and the 'clusters' to be the possible topics.

We implemented the algorithm and tested it on the newsgroup_20 dataset, which consists of 20000 messages from 20 different newsgroups. We used half of the messages to construct the term distribution for each newsgroup, and the other half to test the algorithm. The training part consists of computing the term distribution for every topic (this forms the matrix W in our analysis), followed by computing the generalized pseudoinverse W'. Now, given a new document with term vector $\tilde{\mathbf{u}}$, we compute a relevance to each topic by simply calculating the vector $\tilde{p} = W'\tilde{\mathbf{u}}$. We classify the document to be in topic c if $p_c \approx ||\tilde{p}||_\infty$.

While the results of this study are only preliminary, they appear promising relative to other approaches in this area (see e.g. [3]). Given that our algorithm computes, for each document, a distribution over all topics, it may also be useful for cases in which one wants to explicitly represent the partial relevance of a document to several topics simultaneously.

4. STRONG BENCHMARK

Our semi-omniscient algorithm was based on a sequence of facts that we recapitulate here at an informal level:

- If all the entries in an $k \times M$ matrix B have bounded absolute value, then $B\tilde{\mathbf{u}} \approx B\mathbf{u}$
- If the utility vector of a user u is in the range of a matrix A, then $AA'\mathbf{u} = \mathbf{u}$, and hence, possibly, $\mathbf{u} \approx AA'\tilde{\mathbf{u}}$
- Every utility vector is in the range of the weight matrix W, and all entries of W' have absolute value bounded by $\frac{1}{\Gamma}$.

Essentially, in our analysis, we only used the fact that the weight matrix W satisfies the first two of these points. In this section we consider the *strong benchmark* — the problem of making recommendations when even the weight matrix W is not known. Our goal is to to show that, despite lacking knowledge of W, we can build a matrix A that can be used instead of W. The rest of this section is organized as follows. First we provide our algorithm, which is fairly simple and intuitive; we devote the rest of the section to the analysis of the algorithm.

4.1 Algorithm

First we give two simple definitions:

DEFINITION 1 (CORRELATION MATRIX). *Let $\tilde{\mathcal{P}}_{ij}$ denote the fraction of all users whose first two selections are i and j respectively, and let $\mathbf{E}\left[\tilde{\mathcal{P}}_{ij}\right]$ denote the expected fraction of users with this property (where the expectation is computed with respect to the true weight and preference matrices). The $M \times M$ matrix $\tilde{\mathcal{P}} = \{\tilde{\mathcal{P}}_{ij}\}$ is called the* observed correlation matrix*, and the matrix $\mathcal{P} = \{\mathbf{E}\left[\tilde{\mathcal{P}}_{ij}\right]\}$ is called the* correlation matrix*.*

Obviously the matrix $\mathcal{P}$ is symmetric, $\sum_{ij} \tilde{\mathcal{P}}_{ij} = \sum_{ij} \mathcal{P}_{ij} = 1$, and $\mathcal{P} = W\frac{PP^T}{N}W^T$. We use $\mathcal{P}_i$ to denote the i-th row of the correlation matrix $\mathcal{P}$.

Note that to simplify our analysis we have only used the first two selections from every user; an implicit point of the analysis to follow is that this is sufficient to determine the necessary relationships among items. The plan is to carefully choose k columns of $\tilde{\mathcal{P}}$ to form the desired matrix A.

The second definition extends the notion of cluster independence to the setting of arbitrary matrices.

DEFINITION 2 (INDEPENDENCE COEFFICIENT). *We define the independence coefficient of a collection of vectors $(x_1, x_2, \ldots x_l)$ to be*

$$\min_{|\alpha_1|+\cdots+|\alpha_l|=1} ||\sum_i \alpha_i x_i||_1.$$

We define three functions. $\gamma_r(P)$ is the independence coefficient of the rows of P. $\gamma_c(W)$ is the independence coefficient of columns of W. The function $\gamma(x_1, x_2, \ldots x_l)$ over the collection of vectors $(x_1, x_2, \ldots x_l)$ is defined as independence coefficient of the vectors $\frac{x_1}{||x_1||_1}, \frac{x_2}{||x_2||_1}, \ldots \frac{x_l}{||x_l||_1}$.

Now we present the algorithm.

ALGORITHM 2.

Input: User selections, ε, δ.

Output: Recommendation i_u for each user u.

Description:

1. *Build the observed correlation matrix $\tilde{\mathcal{P}}$.*
2. *Find k columns of $\tilde{\mathcal{P}}$, $\tilde{\mathcal{P}}_{i_1}, \tilde{\mathcal{P}}_{i_2}, \ldots \tilde{\mathcal{P}}_{i_k}$, such that $||\tilde{\mathcal{P}}_{i_c}||_1 \geq \frac{\varepsilon}{N^{1/4}}$ for each $1 \leq c \leq k$, and the matrix A defined as*

$$A = \left(\tilde{\mathcal{P}}_{i_1}/||\tilde{\mathcal{P}}_{i_1}||_1, \ldots, \tilde{\mathcal{P}}_{i_k}/||\tilde{\mathcal{P}}_{i_k}||_1\right)$$

   has a column independence coefficient that is as large as possible.[2]
3. *For each user u with selection vector $\tilde{\mathbf{u}}$, compute $\bar{\mathbf{u}} = AA'\tilde{\mathbf{u}}$ and recommend the item i which maximizes utility in $\bar{\mathbf{u}}$*

Note that most of the computing time is spent in step 2 of the algorithm. Once this is done, we can make recommendations to users very quickly.

4.2 Analysis of the algorithm

Our analysis consists of two theorems. The first theorem guarantees that the matrix A found by the algorithm will have large independence coefficient and small maximal element. Then we give a few results bounding the sampling error. Finally we state and prove the main result of this section, showing that the algorithm makes good recommendations.

Before we continue we introduce some additional notation. All items which have total weight $w_i = \sum_{c \in \mathcal{C}} w(i|c) \leq \frac{\varepsilon\Gamma}{2M}$ (with respect to the true weight matrix W) are called *inessential*, reflecting the fact that the total aggregate weight of all such items combined is less than $\frac{\varepsilon\Gamma}{2}$. We denote the set of inessential items by $\mathcal{I}_0$. Correspondingly we call every item in $\mathcal{I}_1 = \mathcal{I} - \mathcal{I}_0$ an essential item.

Weight matrix. Extending the terminology used thus far, we call an arbitrary $M \times k$ matrix A a *weight matrix* if it has only nonnegative elements, and all of its columns are normalized (in the 1-norm). To prevent confusion, the matrix W will be referred to as the *true weight matrix*. For a weight matrix A, we use the same notation that we introduced earlier for the true weight matrix W. For example $a(i|c)$ denotes the element in the i-th row and c-th column. In addition we introduce a few additional symbols. Let A_c denote the c-th *column* of matrix A (corresponding to the probability distribution for cluster c) and let $\mathbf{a}_i$ denote the normalized (in 1-norm) i-th row of A (we will call this the *item affiliation vector*). Also let $a_i = \sum_c a(i|c)$ denote the total weight of item i (across all clusters).

Preference matrix. We call an arbitrary $k \times M$ matrix P a *preference matrix* if it has only nonnegative entries and all of its columns are normalized in the 1-norm. It is important to note that while W and P^T have the same dimensions, their normalization is different. Let P_c denote the normalized (in 1-norm) c-th *row* of P (this is the vector of user utilities over cluster c), and let $\mathbf{p}_u$ denote the u-th column of matrix P (the preference vector for user u).

Distance function. For a collection of vectors $(x_1, \ldots x_l)$, we denote by x_{-i} the collection of all vectors but x_i. We define $d_{\min}(x_1, x_2 \ldots x_l) = \min_i d(x_i, x_{-i})$, where $d(x_i, x_{-i})$ is the L_1 distance between x_i and subspace spanned by x_{-i}.

The rest of the analysis consists of two parts: first we prove that both A and A' have their elements bounded by

[2]While this suggests an exponential running time, in the analysis below we show that it can be replaced with a step that is implementable in polynomial time.

functions of $\mathcal{W}$, Γ and Γ_P, and then we will prove that these bounds are sufficient.

LEMMA 4.1. *For any $k \times N$ matrix P such that P/N has row independence at least Γ_P, the matrix $(\frac{PP^T}{N})^{-1}$ has the property that the absolute value of all entries is bounded by $\frac{1}{\Gamma_P^2}$. Moreover, $\gamma_r(\frac{PP^T}{N}) \geq \frac{\Gamma_P^2}{k}$*

Proof. It suffices to prove that the smallest eigenvalue of PP^T is at least $N\Gamma_P^2$. Indeed, for any vector x whose L_2-norm is equal to 1, we have:

$$\begin{aligned} ||PP^Tx||_2 &\geq (x^TPP^Tx) = ||P^Tx||_2^2 \geq \frac{||P^Tx||_1^2}{N} \geq \\ &\geq \frac{(N\Gamma_P||x||_1)^2}{N} \geq N\Gamma_P^2 \end{aligned}$$

which gives us the first part of the lemma. For the second part we just note that for any $k \times k$ matrix Q we have $\gamma_r(Q) \geq \frac{\max_{ij} Q_{ij}^{-1}}{k}$ ■

THEOREM 4.2 (BOUNDS ON A). *The matrix A found in step 2 of Algorithm 2 has the property that*

(a) the absolute values of all entries are bounded by $2\mathcal{W}$ and

(b) A has independence coefficient at least

$$\gamma_c(A) \geq \frac{\Gamma^k\Gamma_P^2}{2(2k+1)^{k-1}}. \tag{10}$$

We split the proof of this theorem into several lemmas.

First we want to bound the independence coefficient of A. Recall that for both the true weight matrix W and for P, we have made the assumptions that $\gamma_c(W)$ and $\gamma_r(P/N)$ respectively are bounded away from zero.

LEMMA 4.3. *If $\gamma_c(W) \geq \Gamma$, then for any $k-1$ vectors $\mathbf{X} = (x_1, x_2, \dots x_{k-1})$, there exists an essential item i such that $d(\mathbf{w}_i, \mathbf{X}) \geq \frac{\Gamma}{2k}$*

Proof. Suppose it is not the case; then for all $i \in \mathcal{I}_1$, we have $d(\mathbf{w}_i, \mathbf{X}) < \frac{\Gamma}{2k}$. For an item i, let $x(i)$ denote a vector which achieves this minimum distance. Since the subspace $\mathbf{X}$ has dimension at most $k-1$, there exists a vector $\mathbf{x}^\perp$, with $||\mathbf{x}^\perp||_1 = 1$, that is orthogonal to $\mathbf{X}$. By the definition of $\gamma_c(W)$ we have $Wx^\perp \geq \Gamma$, but on the other hand we have

$$\begin{aligned} W\mathbf{x}^\perp &= \textstyle\sum_{i\in\mathcal{I}} |(\mathbf{x}^\perp\mathbf{w}_i)|w_i \leq \sum_{i\in\mathcal{I}_0} \frac{\varepsilon\Gamma}{2M} + \\ &+ \textstyle\sum_{i\in\mathcal{I}_1} \Big[|\Big(\mathbf{x}^\perp\big(\mathbf{w}_i - x(i)\big)\Big)|w_i\Big] < \\ &< M\frac{\varepsilon\Gamma}{2M} + \frac{\Gamma}{k}|\textstyle\sum_{i\in\mathcal{I}_1} w_i| \leq \Gamma. \end{aligned}$$

leading us to a contradiction. ■

LEMMA 4.4. *Let $\gamma_c(W) \geq \Gamma$, and let $\mathcal{I}' = i_1, i_2, \dots, i_t$, be a subset of items, where $t < k$, with weight vectors $x_1, \dots, x_t$, satisfying $\gamma(x_1, x_2, \dots, x_t) \geq a$. Then $\mathcal{I}'$ can be augmented with an essential item j having weight vector x_{t+1} such that*

$$\gamma(x_1, x_2, \dots x_{t+1}) \geq a\frac{\Gamma}{\Gamma + 2k} \tag{11}$$

Proof. By Lemma 4.3, we can always choose an item j so that

$$d(\mathbf{w}_j, \{x_1, x_2, \dots x_t\}) \geq \frac{\Gamma}{2k}. \tag{12}$$

Now our claim is that this item j satisfies (11). For the sake of contradiction suppose it does not; then let

$$||\sum_{i=1,2,\dots t+1} \alpha_i x_i||_1 < a\frac{\Gamma}{\Gamma + 2k},$$

where $x_{t+1} = \mathbf{w}_j$. Obviously if $\alpha_{t+1} \leq a\frac{2k}{\Gamma+2k}$, then we contradict the independence of $x_1, \dots x_t$:

$$||\sum_{i=1,\dots t} \alpha_i x_i||_1 < a\frac{\Gamma}{\Gamma + 2k} + a\frac{2k}{\Gamma + 2k} = a.$$

On the other hand, if $\alpha_{t+1} > a\frac{2k}{\Gamma+2k}$, then we have

$$||\sum_{i=1,2,\dots t} \frac{\alpha_i}{\alpha_{t+1}x_i} + x_{t+1}||_1 < \frac{a\Gamma}{(\Gamma + 2k)\alpha_{t+1}} < \frac{\Gamma}{2k},$$

which obviously contradicts (12). ■

This lemma has an obvious corollary:

COROLLARY 4.5. *Let $\gamma_c(W) > \Gamma$. Then there always exists a subset of essential items $i_1, i_2, \dots i_k$, such that $\gamma(w_{i_1}, \dots w_{i_k}) \geq \left[\frac{\Gamma}{2k+1}\right]^{k-1}$*

From here, our next major goal is to show the existence of k sufficiently independent columns in the matrix $\mathcal{P}$. Before we continue we prove the following simple result.

LEMMA 4.6. *Let $\gamma_c(W) \geq \Gamma$ and $\gamma_r(P/N) \geq \Gamma_P$. Then there are k columns $i_1, i_2, \dots, i_k$ of matrix $\mathcal{P}$ such that*

$$\gamma(\mathcal{P}_{i_1}, \dots \mathcal{P}_{i_k}) \geq \frac{\Gamma^k}{(2k+1)^{k-1}}\Gamma_P^2. \tag{13}$$

Moreover items $i_1, \dots, i_k$ are essential.

Proof. By Corollary 4.5 there exists a set of essential items $i_1, \dots, i_k$, such that $\gamma(w_{i_1}, \dots w_{i_k}) \geq \left[\frac{\Gamma}{2k+1}\right]^{k-1}$ We show that this set satisfies (13). It is sufficient to show that for any $v = (v_1, \dots v_k)$ with $||v||_1 = 1$, we have $\mathcal{P}y \geq \frac{\Gamma^k}{(2k+1)^{k-1}}\Gamma_P^2$, where y is defined as follows:

$$y_j = \begin{cases} \frac{v_l}{||\mathcal{P}_l||_1} & \text{if } j = i_l \text{ for some } l \\ 0 & \text{otherwise} \end{cases}$$

Given our assumption that $\gamma_r(P/N) \geq \Gamma_P$, and since items are essential, we have $\mathcal{P}_l > 0$, so the definition above is valid. From our assumption on $i_1, \dots, i_k$ it immediately follows that

$$||W^Ty||_1 \geq [\frac{\Gamma}{2k+1}]^{k-1} \times \sum_l \frac{|v_l|w_l}{||\mathcal{P}_l||_1}$$

But,

$$||\mathcal{P}_l||_1 = \sum_c w(l|c)\frac{\sum_u p(u|c)}{N} \leq \sum_c w(l|c) = w_l,$$

and therefore we can rewrite the above bound as:

$$||W^Ty||_1 \geq [\frac{\Gamma}{2k+1}]^{k-1}\sum_l |v_l| \geq [\frac{\Gamma}{2k+1}]^{k-1}. \tag{14}$$

Now, recall the definition of $\mathcal{P} = W\frac{PP^T}{N}W^T$. Therefore

$$\mathcal{P}y \geq \Gamma||\frac{PP^T}{N}W^Ty||_1 \geq \Gamma\Gamma_P^2||W^Ty||_1 \geq \frac{\Gamma^k\Gamma_P^2}{(2k+1)^{k-1}}$$

where the first and second inequalities follow from the lemma's assumption of large $\gamma_c(W)$ and $\gamma_r(P/N)$, together with Lemma 4.1. The third inequality follows from (14), and this concludes the proof. ■

The algorithm only has access to the observed correlation matrix $\tilde{\mathcal{P}}$, not the true correlation matrix $\mathcal{P}$. We now must show that, with sufficient data, these two matrices are very close to one another. The following lemma is an immediate consequence of tail inequalities:

LEMMA 4.7. *For any fixed ε and δ, and given enough users, we have*

$$\max_{ij} |\mathcal{P}(i,j) - \tilde{\mathcal{P}}(i,j)| < \frac{\varepsilon}{N^{1/4}}, \tag{15}$$

with probability at least $1-\delta$.

Proof. For any item i and any λ, we can apply Chernoff bounds to obtain

$$\Pr\left[\tilde{\mathcal{P}}(i,j) - \mathcal{P}(i,j) \geq \lambda \mathcal{P}(i,j)\right] \leq \left[\frac{e^{\lambda}}{(1+\lambda)^{1+\lambda}}\right]^{\mathcal{P}(i,j)N}$$

and

$$\Pr\left[\tilde{\mathcal{P}}(i,j) - \mathcal{P}(i,j) \leq -\lambda \mathcal{P}(i,j)\right] \leq e^{-\frac{\lambda^2 \mathcal{P}(i,j)N}{2}}.$$

Note that these bounds hold for any values of N and λ. Now, if $\mathcal{P}(i,j) \leq N^{-1/3}$, then substituting $\lambda = \frac{\varepsilon N^{-1/4}}{\mathcal{P}(i,j)} \geq \varepsilon N^{1/12}$ gives us the desired bounds. If on the contrary $\mathcal{P}(i,j) \geq N^{-1/3}$, then recalling that $\mathcal{P}(i,j) \leq 1$ and taking $\lambda = \varepsilon N^{-1/4}$ we have $\frac{\varepsilon}{N^{1/4}} \geq \lambda \mathcal{P}(i,j)$, and hence

$$\Pr\left[\left|\tilde{\mathcal{P}}(i,j) - \mathcal{P}(i,j)\right| \geq \frac{\varepsilon}{N^{1/4}}\right] \leq e^{\left(-\frac{\lambda^2 \mathcal{P}(i,j)N}{4}\right)} \leq e^{-\frac{\varepsilon^2 N^{1/6}}{4}}$$

Note that the probability of wrong estimation decreases exponentially as N grows; therefore if we take N large enough we can apply union bounds and hence we can ensure that $\mathcal{P}(i,j)$ is estimated correctly for all items with high probability. ■

A similar result holds for most subsets of normalized columns of $\tilde{\mathcal{P}}$ and $\mathcal{P}$:

COROLLARY 4.8. *Let $i_1, i_2, \ldots, i_k$ be a collection of items such that $||\tilde{\mathcal{P}}_{i_c}||_1 \geq \frac{\varepsilon}{N^{1/4}}$ and let matrices A and B be comprised of normalized columns $\tilde{\mathcal{P}}_{i_1}, \tilde{\mathcal{P}}_{i_2}, \ldots \tilde{\mathcal{P}}_{i_k}$ and $\mathcal{P}_{i_1}, \mathcal{P}_{i_2}, \ldots \mathcal{P}_{i_k}$ respectively. For any fixed ϵ and δ and given enough users we have:*

$$\max_{i,c} |A_{ic} - B_{ic}| \leq \epsilon \tag{16}$$

with probability at least $1-\delta$.

Proof. This can be immediately achieved by using lemma 4.7 with $\varepsilon = \epsilon^2/2$, and using tail inequalities to bound difference between $||\tilde{\mathcal{P}}_i||_1$ and $||\mathcal{P}_i||_1$. ■

LEMMA 4.9 (EQUIVALENCE OF $\mathcal{P}$ AND $\tilde{\mathcal{P}}$). *Suppose $\mathcal{P}$ has a subset of independent columns with independence coefficient at least a, and all items corresponding to this subset are essential. Then given enough users, with probability at least $(1-\delta)$ the same subset in $\tilde{\mathcal{P}}$ is also independent, with independence coefficient at least $a/2$. It also holds in the opposite direction: if some subset of columns in $\tilde{\mathcal{P}}$ is independent, the same subset in $\mathcal{P}$ has independence coefficient at least half of that with probability $1-\delta$.*

Proof. Suppose that $\gamma(\tilde{\mathcal{P}}_{i_1}, \tilde{\mathcal{P}}_{i_2}, \ldots \tilde{\mathcal{P}}_{i_k}) \geq \varepsilon$, and all items $i_1, \ldots i_k$ are essential. We introduce two $M \times k$ matrices A and B which are formed by normalized columns $\tilde{\mathcal{P}}_{i_1}, \ldots \tilde{\mathcal{P}}_{i_k}$ and $\mathcal{P}_{i_1}, \ldots, \mathcal{P}_{i_k}$ respectively. We have to prove that $\gamma(B) \geq a$ implies $\gamma(A) \geq a/2$ with high probability.

This is equivalent to showing that for all v with $||v||_1 = 1$, we have $||Av||_1 \geq \frac{a}{2}$. It suffices to show that $||(A-B)v||_1 \leq \frac{a}{2}$, which in turn can be achieved by having

$$\max_{i,c} |B_{ic} - A_{ic}| \leq \frac{a}{2Mk}.$$

By Corollary 4.8 the last inequality holds if we have enough users. The proof for the other direction is completely symmetric. ■

Now, we want to bound maximal element of matrix A. The following lemma is immediate.

LEMMA 4.10. *For any vector v which is a convex combination of W_1, W_2, ..., W_k we have $\frac{\mathbf{w}_-}{k} \leq ||v||_\infty \leq \mathbf{w}_+$.*

COROLLARY 4.11. *If we have enough users, then for any normalized column v of $\tilde{\mathcal{P}}$, considered during step 2 of the algoirthm we have $\frac{\mathbf{w}_-}{2k} \leq ||v||_\infty \leq 2\mathbf{w}_+$, with high probability.*

Proof. Indeed we have

$$\mathcal{P} = \frac{WPP^TW^T}{N},$$

and since elements of P and W are non-negative, each column of $\mathcal{P}$ is a convex combination of columns of W. The result for $\tilde{\mathcal{P}}$ follows immediately from corollary 4.8, by taking $\varepsilon = \frac{\mathbf{w}_-}{2k}$ ■

Now we are ready to prove Theorem 4.2.

Proof of Thm 4.2. Part (a) holds because of Corollary 4.11. Now we prove part (b), for which it suffices to show that as the number of users N grows, all essential items will be considered during step 2 of the algorithm with high probability. Combining this fact and Lemmas 4.6 and 4.9 yields the desired result.

Indeed, any essential item i has total weight at least $\frac{\varepsilon\Gamma}{2M}$, and therefore there is at least one cluster c such that $w(i|c) \geq \frac{\varepsilon\Gamma}{2kM}$. Now, because $\gamma_r(P/N) = \Gamma_P$, each cluster has total probability weight at least $\Gamma_P N$, and so the expected number of times item i is selected is at least $N\frac{\varepsilon\Gamma\Gamma_P}{2kM}$. Thus

$$\mathbf{E}\left[||N\tilde{\mathcal{P}}_i||_1\right] \geq \frac{N\varepsilon\Gamma\Gamma_P}{2kM},$$

and since none of the parameters above depend on N, we can apply tail and union bounds to show that if N is large enough then $||\tilde{\mathcal{P}}_i||_1 \geq \frac{\varepsilon}{N^{1/4}}$ holds for each essential item i with high probability.

■

Recall that when we initially presented Algorithm 2, we noted that an exponential search for the k-tuple of columns with maximum independence coefficient was not actually necessary. One can now see the reason for this: the proof of Lemma 4.4 shows that we can apply a greedy algorithm similar to the one used there to build a matrix A with essentially the same results.

Now we have to show that the bounds we have obtained are sufficient. Observe that we cannot directly use the analysis of Section 3 here, since our user utility vectors are not truly in the range of A, but rather are close to it.

First we bound the different kinds of error incurred because of sampling error.

LEMMA 4.12. *For the matrix A found in step 2 of Algorithm 2, and for any fixed ε, δ, the following holds with probability at least $1-\delta$, provided that we have sufficiently many users with two selections per user:*

$$\max_{ij} |\left((AA' - I)\tilde{\mathcal{P}}\right)_{ij}| \leq \varepsilon$$

The number of users needed is a function of ε, δ, Γ, Γ_P and M.

Proof. Define matrix B in exactly the same way as in Lemma 4.9. By Lemma 4.9, we have $\gamma(B) > \gamma(A)/2$ with high probability. If this holds, then $BB'\mathcal{P} = \mathcal{P}$ (because $\mathcal{P}$ is a rank-k matrix, and all columns are linear combinations of columns of B). Therefore every column of $\mathcal{P}$, say $\mathcal{P}_i$, can be represented as a product of B and a k-dimensional vector $q_i = B'\mathcal{P}_i$; obviously $||q_i||_\infty \leq \frac{1}{\gamma_c(B)}$.

Now the rest is easy:

$$\tilde{\mathcal{P}}_i = \mathcal{P}_i + \varepsilon = Bq_i + \varepsilon = (A + \mathcal{E})q_i + \varepsilon = (Aq_i) + (\varepsilon + \mathcal{E}q_i),$$

where ε and $\mathcal{E}$ are vector and matrix error terms whose elements can be upper-bounded using Lemma 4.7 and Corollary 4.8. We have

$$\begin{aligned} AA'\tilde{\mathcal{P}}_i = &\quad AA'(Aq_i + (\varepsilon + \mathcal{E}q_i)) = Aq_i + AA'(\varepsilon + \mathcal{E}q_i) \\ = &\quad \tilde{\mathcal{P}}_i + (\varepsilon + \mathcal{E}q_i)(AA' - I). \end{aligned}$$

If we upper-bound each entry in ε and $\mathcal{E}$ by $\varepsilon_1 < \frac{\varepsilon[\gamma_c(A)]^2}{4M^2} \leq \frac{\varepsilon\gamma_c(B)\gamma_c(A)}{2M^2}$, assuming enough users as required by Lemma 4.7 and Corollary 4.8, then the total error term in this equation will be less than ε; hence

$$\max_{ij} |(AA'\tilde{\mathcal{P}} - \tilde{\mathcal{P}})_{ij}| \leq \varepsilon$$

■

Now, we are ready to formulate and prove the main theorem of this section.

THEOREM 4.13. *Assuming that the system contains enough users, Algorithm 2 gives a $(1-\varepsilon)$-optimal recommendation with high probability for any user u who made at least $s \geq \frac{k}{\delta[\Gamma'\varepsilon/(8k^2\mathcal{W})]^2}$ selections, where Γ' is the independence coefficient of A and $\Gamma' \geq \frac{\Gamma^k\Gamma_P^2}{2(2k+1)^{k-1}}$.*

Proof. For this proof $||x||$ denotes the L_∞ norm of x. Clearly every user u has at least one item of utility $\frac{\mathbf{w}_-}{k}$; hence it suffices to prove that $\mathbf{u}$ is estimated by $\bar{\mathbf{u}} = AA'\tilde{\mathbf{u}}$ such that

$$||\mathbf{u} - \bar{\mathbf{u}}|| < \frac{\varepsilon\mathbf{w}_-}{k}. \tag{17}$$

The recommended item will be at most $\frac{\varepsilon\mathbf{w}_-}{k}$ away from optimal and hence will be $(1-\varepsilon)$-optimal.

The proof consists of two parts: first we prove that the utility vector of a user u can be represented as $\mathbf{u} = \mathcal{P}v$, with $||v|| \leq \frac{k^2}{\Gamma\Gamma_P^2}$; and second we substitute this expression for $\mathbf{u}$ into $||\mathbf{u} - \bar{\mathbf{u}}||$ and finish the analysis.

Indeed we have $\mathcal{P} = W\frac{PP^T}{N}W^T$ and $\mathbf{u} = W\mathbf{p}$. Now, W^T is a $k \times M$ matrix, so $W^TW'^T = I$. Therefore we have the following:

$$\begin{aligned} \mathbf{u} &= W\mathbf{p} = W\tfrac{PP^T}{N}W^TW'^T\left(\tfrac{PP^T}{N}\right)^{-1}\mathbf{p} = \\ &= \mathcal{P}\left[W'\left(\tfrac{1}{N}PP'^T\right)^{-1}\mathbf{p}\right], \end{aligned}$$

where the existence of $(PP^T)^{-1}$ follows from Lemma 4.1. Moreover, the elements of $(\frac{PP^T}{N})^{-1}$ are bounded by $\frac{1}{\Gamma_P^2}$; therefore we have:

$$||v|| = ||W'^T\left(\frac{PP'^T}{N}\right)^{-1}\mathbf{p}|| \leq \frac{k^2}{\Gamma\Gamma_P^2}. \tag{18}$$

Now we substitute $u = \mathcal{P}v$ into the left-hand side of (17):

$$\begin{aligned} ||\bar{\mathbf{u}} - \mathbf{u}|| &= ||AA'\tilde{\mathbf{u}} - \mathbf{u}|| \leq \\ &\leq ||AA'\tilde{\mathbf{u}} - AA'\mathbf{u}|| + ||AA'\mathbf{u} - \mathbf{u}|| \leq \\ &\leq ||A(A'\tilde{\mathbf{u}} - A'\mathbf{u}))|| + ||(AA' - I)\mathcal{P}\mathbf{v}|| \end{aligned} \tag{19}$$

To bound the first term we use the fact that the absolute values of all entries in A' are bounded by $1/\Gamma'$. Applying Lemma 3.1 and the union bound, we have $||A'\tilde{\mathbf{u}} - A'\mathbf{u}|| < \frac{\varepsilon}{4k^2\mathcal{W}}$ with probability at least $1-\delta$. Substituting this we have

$$||A(A'\tilde{\mathbf{u}} - A'\mathbf{u})|| \leq \mathbf{w}_+\frac{\varepsilon}{4k\mathcal{W}} \leq \varepsilon\frac{\mathbf{w}_-}{2k}.$$

To bound the second term, we write

$$\begin{aligned} ||(AA' - I)(\mathcal{P}\mathbf{v})|| = &\quad ||(AA' - I)[(\tilde{\mathcal{P}} + \mathcal{E})\mathbf{v}]|| \\ \leq &\quad ||(AA' - I)\tilde{\mathcal{P}}\mathbf{v}|| + ||(AA' - I)\mathcal{E}\mathbf{v}|| \end{aligned} \tag{20}$$

where $\mathcal{E} = \mathcal{P} - \tilde{\mathcal{P}}$.

Now fix $\varepsilon_1 = \varepsilon\frac{\Gamma\Gamma_P^2}{2M^2k^2}$ and, provided we have enough users, apply Lemmas 4.7 and 4.12 so that we have

$$\begin{aligned} &\max_{ij} |\left((AA' - I)\tilde{\mathcal{P}}\right)_{ij}| \leq \varepsilon_1 \\ &\max_{ij} |\mathcal{E}_{ij}| \leq \varepsilon_1 \end{aligned}$$

with high probability. Therefore we can bound the expression in (20) by $\frac{\varepsilon}{2Mk} \leq \frac{\varepsilon\mathbf{w}_-}{2k}$ Thus the whole expression in (19) can be upper bounded by $\frac{\varepsilon\mathbf{w}_-}{k}$, as desired.

Note that the first term of (19) is an error introduced by insufficient sampling, while the second is an error introduced by an insufficient number of users. ■

5. NOTES AND OPEN PROBLEMS

We have shown how to obtain provably good recommendations for a mixture model with unknown parameters, provided the parameters $\mathcal{W}$, Γ, and Γ_P are bounded. While bounding Γ_P appears to be a relatively mild assumption in most potential applications of this model, we do not know of a concrete sense in which it is a necessary assumption; it is an interesting open question to determine whether good recommendations can still be found when this parameter is not bounded.

As discussed above, the definition of Γ raises the prospect of defining an L_1 analogue of the singular values of a matrix. Just as Γ plays the role of the smallest singular value, we can define the L_1 analogue of the i-th singular value:

$$\Gamma_i(W) = \min_{\dim\Omega = i}\max_{x\in\Omega}\frac{||Wx||_1}{||x||_1}$$

If W is a weight matrix then we clearly have $\Gamma = \Gamma_1 \leq \Gamma_2 \leq \cdots \leq \Gamma_k = 1$. It would be interesting to explore properties of these values; for example, can we define a useful analogue of the full singular value decomposition, but with respect to L_1 norm?

Finally, it would be interesting to explore trade-offs between the amount of data used by these types of recommendation algorithms and the performance guarantees they achieve. Our algorithms have a running time that is polynomial in the amount of data; but for the strong benchmark, the amount of data needed is exponential in some of the parameters. One would like to know whether this bound can be made polynomial, or whether perhaps it is possible to establish a lower bound. Further, while our goal has been to obtain $(1-\varepsilon)$-approximations for arbitrarily small ε, one can consider the amounts of data and computation required for weaker guarantees. For example, simply recommending the most popular item to everyone is an $\Omega(1/k)$-approximation, with enough users but with just one selection per user. How much data is required if we want a $(1/b)$-approximation for $b < k$?

Acknowledgment. The authors would like to thank Frank McSherry; discussions with him about spectral analysis and the use of correlation matrices provided part of the motivation for this work.

6. REFERENCES

[1] J. Breese, D. Heckerman, C. Kadie "Empirical Analysis of Predictive Algorithms for Collaborative Filtering," *In Proc. 14th Conference on Uncertainty in Artificial Intelligence*, 1998

[2] Y. Azar, A. Fiat, A. Karlin, F. McSherry, J. Saia "Spectral analysis of data" *Proc. ACM Symposium on Theory of Computing*, 2000

[3] L. D. Baker, A. K. McCallum "Distributional Clustering of Words for Text Categorization" *In Proc. ACM SIGIR Intl. Conf. Information Retrieval*, 1998

[4] P. Drineas, I. Kerendis, P. Raghavan "Competitive Recommender Systems" *Proc. ACM Symposium on Theory of Computing, 2002*

[5] G. H. Golub, C.F. Van Loan, *Matrix Computations* (3rd edition), Johns Hopkins University Press, 1996.

[6] T. Hofmann, J. Puzicha, "Latent Class Models for Collaborative Filtering," *Proc. International Joint Conference in Artificial Intelligence*, 1999.

[7] J. Kleinberg, M. Sandler, "Convergent Algorithms for Collaborative Filtering, " *Proc. ACM Conference on Electronic Commerce*, 2003.

[8] S.R. Kumar, P. Raghavan, S. Rajagopalan, A. Tomkins, "Recommendation systems: A probabilistic analysis," *Proc. IEEE Symposium on Foundations of Computer Science*, 1998.

[9] G. Linden, B. Smith, J. York, "Amazon.com Recommendations: Item-to-Item Collaborative Filtering," *IEEE Internet Computing*, Jan./Feb. 2003.

[10] Geoffrey McLachlan, David Peel. *Finite Mixture Models*. Wiley, 2000.

[11] P. Resnick, H. Varian, "Recommender systems," *Communications of the ACM*, 40(1997). Introduction to a special issue on collaborative filtering.

APPENDIX

A. SPECTRAL ANALYSIS: EXAMPLE

Fix some small θ, say $\theta = 0.1$, and pick some large m. Say we have $2m + m^\theta - 1$ items and two clusters, and let $r = 1 - m^{-\theta} \approx 1$. We define clusters as follows:

$$\left(\frac{2}{m^{2\theta}}, \underbrace{\frac{1}{m^{2\theta}}, \ldots, \frac{1}{m^{2\theta}}}_{m^\theta - 2 \text{ items}}, \underbrace{\frac{r}{m}, \ldots, \frac{r}{m}}_{m \text{ items}}, \underbrace{0,\ 0, \ldots, 0}_{m \text{ items}}\right)$$

$$\left(\underbrace{\frac{1}{m^{2\theta}}, \ldots, \frac{1}{m^{2\theta}}}_{m^\theta - 2 \text{ items}}, \frac{2}{m^{2\theta}}, \underbrace{0,\ 0, \ldots, 0,}_{m \text{ items}}, \underbrace{\frac{r}{m}, \ldots, \frac{r}{m}}_{m \text{ items}}\right)$$

We assume that there are $N/2$ users who each only like the first cluster, and $N/2$ users who each only like the second cluster. Obviously each user wants to get recommended an item with weight $2/m^{2\theta}$ in the cluster he likes, and these items are different for different clusters, so it is important to be able to distinguish between these different types of users.

In both clusters $1 - m^{-\theta}$ of the weight is concentrated on disjoint items; therefore it is easy to distinguish between different types of users. Easy calculations show that in this system $\Gamma > 0.9$ and $\mathcal{W} = 1$ for any sufficiently large m, and hence the algorithms we develop below will give good recommendations using only $f(\varepsilon, \delta)$ samples, for some function f. On the other hand, for spectral analysis, we consider the matrix $W' = (W_1', W_2')$ comprised of the weight vectors normalized with respect to the L_2 norm. (Without this normalization, it is even easier to construct a bad example for the smallest singular value.) Then the least singular value of W' can be bounded by:

$$\lambda \leq ||W_1' - W_2'||_2 \leq m^{\frac{3\theta}{2}} ||W_1 - W_2||_2 = O(-m^{\frac{\theta}{2}}),$$

which converges to 0 as m grows. Thus, any bound on the amount of data needed that is based on $1/\lambda$ will be increasing unboundedly with m, even though the actual amount of data needed (and the amount computed from a bound involving Γ) remains constant with m.