A Lower-Bound on the Number of Rankings Required in Recommender Systems Using Collaborativ Filtering

Peter Marbach,
Dept. of Computer Science, University of Toronto
Email: marbach@cs.tornto.edu

***Abstract*— We consider the situation where users rank items from a given set, and each user ranks only a (small) subset of all items. We assume that users can be classified into C classes, and users in a given class c have the same ranking for all items. For this situation we are interested in the following question. As a function of the number of users N in a given class c and the numbers of items I_N to be ranked, how many rankings m_N per user are needed in order to be able to correctly identify all user in class c? This question is of interest because correctly identifying all users in a class allows to accurately predict the ranking of an item by a given user that the user has not ranked, but that was ranked by another user in the same class. This is exactly the goal recommender systems using collaborative filtering. Therefore, being able to answer the above questions allows us to characterize how much data (i.e. how many rankings per user) is required by a recommender system using collaborative filtering to accurately predict user-item ranking pairs.**

We study the above question using a random graph model. Even though the resulting random graph is not a Erdos-Renyi graph, this allows us to use for our analysis similar techniques that have been developed for the analysis of Erdos-Renyi graphs.

I. INTRODUCTION

Consider the situation where users rank items (such as books, movies, etc.), and suppose that each user typically only ranks a small subset of all possible items. We assume that rankings take on a value on the interval $[0, 1]$ indicating how much the user likes (values) a given item. Recommender systems using collaborative filtering then try to accurately predict the ranking of an item by a given user that the user has not ranked, but that was ranked by other users.

The above problem has many applications. For example, Amazon.com uses a recommender system to suggest to customers books based on the book rankings that customers have provided. Similarly, the online movie rental company Netflix uses a recommender system to suggest movies to customers. Recommender systems can potentially also be used in the context social networking applications that have become increasingly popular.

Theoretical work on recommender systems that use collaborative filtering have focused on the analysis of specific algorithms. In particular, there has been interested in characterizing the number of rankings that a given algorithm requires in order to correctly predict (with high probability) rankings of all user-item pairs. In this paper, we take another point of view. Rather than focusing on a specific algorithm, we investigate the following question: "What is the minimal number of rankings that any algorithm requires in order to provide accurate predictions?".

Due to space constraints, we state several of our results without providing a proof.

II. PROBLEM FORMULATION

Clearly, in order to be able to predict the rankings for a given user based on the rankings on other users, there has to be some correlation between the users' ranking. There are several possible approaches to model such a correlation. For our analysis, we consider a model proposed by Awerbuch et al. in [2] which assumes that users can be classified into a fixed number of C classes, and users in the same class have the same ranking for all items. For a given set $\mathcal{I}$ of I items, let $r_c = (r_c(1), ..., r_c(I)) \in [0, 1]^I$ be the ranking vector of users in class c. The model allows for ranking vectors of different classes to "overlap", i.e. we can have that $r_{c'}(i) = r_{c''}(i)$, $c' \neq c''$, for some items i. For this model, we are interested in deriving a lower bound on the number of rankings required to correctly all users in a given class c.

As we are interested in a lower-bound on the number of required rankings, we assume in this paper *complete separation* among the rankings of different classes, i.e. if $c' \neq c''$ then we have $r_{c'}(i) \neq r_{c''}(i)$, $i = 1, ..., I$. Complete separation makes the ranking prediction problem simpler, and providing a lower bound on the number of rankings per user needed in order to accurately predict all user-item ranking pairs for this case will also provide a lower bound for the general case.

For the case of complete separation, it suffices to consider a single class and we formulate the problem considered in this paper as follows. Consider a given class c and let N_c be the number of users in this class. Let I_{N_c} be the number total number of items to be ranked. Suppose that each user ranks exactly m_{N_c} items that are chosen at random and uniformly over all items, independent of the choices of all other users. Then we are interested in the following question: "What is the minimal number m_{N_c} of rankings required in order to correctly identify all users in class c in the limit as N_c approaches infinity?". Note that we allow I_{N_c} and m_{N_c} to depend on N_c, i.e. the number of items to be ranked and the number of ranking that each user makes can grow as the user population N_c grows.

III. Random Graph Model

To study the above question, we consider the following random graph model to which we refer as the *user graph*. Let $\mathcal{N}$ be the set of the N users (over all classes) that rank items. Each user $u \in \mathcal{N}$ represents a node in the graph. There exists an edge between two users (nodes) u and v if, and only if, there exists an item i that has been ranked by u and v, and both u and v have assigned the same ranking to item i. As we assume complete separation of the ranking between two different classes, the existence of an edge between two users is given by the following condition. Let S_u be the set of items that have been ranked by user u. Then there exists an edge between two users u and v if and only if both u and v belong to the same class and $S_u \cap S_v \neq \emptyset$. Let V_N be the resulting set of edges, and let $G_N = (\mathcal{N}, \mathcal{V}_N)$ be the resulting graph. The following lemma follows immediately from the above definition.

Lemma 1: Let $G' = (\mathcal{N}', \mathcal{V}')$ be a connected component of the graph G_N. Then all users in $\mathcal{N}'$ belong to the same class.

Using the the above defined user graph, we derive in the following a necessary condition for any algorithm to correctly identify all users in a given class c. To do that we use the following adversary model. After a given algorithm has decided on how cluster users into classes based on the available ranking information (i.e. given the user graph G_N), the adversary decides on all the missing rankings (i.e. the rankings for all user-item pairs for which no prior ranking is available) where these rankings have to be consistent with the above model, i.e. all users that belong to the same component in G_N have the same ranking vector (see Lemma 1). We then have the following impossibility result.

Lemma 2: Suppose that the graph G_N has two disconnected components G_1 and G_2, and let $\mathcal{I}_1$ ($\mathcal{I}_2$) be the set of items ranked by users in G_1 (G_2). If $\mathcal{I}_1 \cap \mathcal{I}_2 = \emptyset$ then there exists no algorithm that can correctly predict whether the users in G_1 and G_2 belong to the same class, or not.

Proof: Let $r \in [0,1]^{I_N}$ be a ranking vector such that for all items $i \in \mathcal{I}_1$ the ranking $r(i)$ is as the ranking given by users in the graph G_1, and for all items $i \in \mathcal{I}_2$ the ranking $r(i)$ is as the ranking given by users in the graph G_2. In addition, let $r_1 \in [0,1]^{I_N}$ be a ranking vector such that for all items $i \in \mathcal{I}_1$ the ranking $r(i)$ is as the ranking given by users in the graph G_1, and for all items $i \in \mathcal{I}_N \backslash \mathcal{I}_1$ the ranking $r_1(i)$ is different from all rankings given by users $u \in \mathcal{N}$ to item i. Similarly, let $r_2 \in [0,1]^{I_N}$ be a ranking vector such that for all items $i \in \mathcal{I}_2$ the ranking $r(i)$ is as the ranking given by users in the graph G_2, and for all items $i \in \mathcal{I}_N \backslash \mathcal{I}_2$ the ranking $r_2(i)$ is different from all rankings given by users $u \in \mathcal{N}$ to item i. Suppose that a given algorithm decides that the users in G_1 and G_2 belong to the same class, then an adversary can assign ranking vector r_1 to the users in G_1 and ranking vector r_2 to the users in G_2, making the decision of the algorithm the wrong decision. Similarly, if the algorithm decides to that the users in G_1 and G_2 belong to different classes, then an adversary can assign assign the ranking vector r to users in both G_1 and G_2. ■

The following necessary condition for an algorithm to be able to correctly identify all users in a given class c is obtained immediately from the above lemma. We use the following notation. Let $\mathcal{N}_c$ be the set of users in class c, let $\mathcal{V}_{c,N}$ be the set of edges between users of class c in G_N, and let $G_{c,N} = (\mathcal{N}_c, \mathcal{V}_{c,N})$.

Corollary 1: A necessary condition for an algorithm to be able to correctly identify all users in a given class c is that the graph $G_{c,N}$ is fully connected.

Corollary 1 states that a necessary condition for being to identify all users in class c is that the corresponding user graph formed is fully connected. Therefore, studying the question of when it is possible to correctly identify all users in class c can be recast as deriving conditions on the number of rankings m_N required for the graph $G_{c,N}$ to be fully connected.

IV. Main Results

Consider a fixed class c, and in order to simplify notation let N (instead of N_c) denote the number of of users in class c. Recall that I_N is the total number of items to be ranked, and that m_N is the number of rankings each user provides where items are uniformly and independently chose over the set of all possible items. The following theorem provides necessary and sufficient conditions for the probability that the graph $G_{c,N}$ is fully connected to converge to 1 as N approaches infinity.

Theorem 1: Let P_N the the probability that the graph $G_{c,N}$ is fully connected. If

$$\frac{Nm_N^2}{I_N} = \omega(logN),$$

then we have $\lim_{N\to\infty} P_N = 1$. If

$$\frac{Nm_N^2}{I_N} = logN + a + o(1)$$

for a positive constant a, then we have

$$\lim_{N\to\infty} P_N \leq 1 - e^{e^{-a}}.$$

In the following we prove Theorem 1 by considering separately the following three cases: the *many-user case* where we have many more users than items and

$$\lim_{N\to\infty} \frac{N}{I_N logI_N} = \infty,$$

the *balanced case* where the number of users and items are of the same order and we have

$$\lim_{N\to\infty} \frac{N_c}{I_N} = b$$

for a positive constant b, and the *many-item case* where we have many more items than users and

$$\lim_{N\to\infty} \frac{Nm_N}{I_N} = 0.$$

V. Related Work

To the best of our knowledge there is no prior working on providing a lower-bound on the number of rankings required in recommender systems using collaborative filtering. However, there is prior work on the analysis of specific algorithms as we discuss below.

The above ranking model, i.e. the assumption that users can be clustered into different classes and the users in the same class have all the same ranking vector, has been proposed by Awerbuch et al. in [2] for the analysis of algorithms for interactive recommender systems. In interactive recommender systems, users actively probe items and share their rankings with all other users by posting them on a public billboard. As probing incurs a cost, the goal of in interactive recommender systems is for each user to learn its full ranking vector with minimal cost, i.e. with probing as few items as possible. An algorithm for interactive recommender systems decides (in a distributed manner) which and how many items a user probes. Note that this situation is different from the one considered in this paper: whereas in interactive recommender systems the question is *how to sample*, the problem that we consider here is *how to interpret* prior ranking information. In [2], Auwerbuch et al. propose and analyze two algorithms in which each user has to roughly probe $O((1 + I_N/N)logN)$, and $O((I_N/N)logN + 1)$, items respectively, in order to discover (with high probability) their full ranking vector. Alon et al. extend in [3] the work by Auwerbuch et al. to a more general ranking model that no longer assumes that users can be clustered can be clustered into different classes and the users in the same class have all the same ranking vector.

Kleinberg and Sandler consider in [4], [5] collaborative filtering for a slightly different recommender system than the system above, where the goal is to predict for each user the item that the user values the highest. Assuming a probabilistic mixture model for how valuations are generated, Kleinberg and Sandler show that it is possible to give recommendations whose quality converge to optimal as the amount of data grows, given that the model parameters are bounded. In their model, Kleinberg and Sandler assume that the number of items to be ranked is fixed, but the number of users and number of rankings per user increases.

Papadimitriou et al. [6] and Azar et al. [7] study recommender systems for which it is assumed that the full user-item matrix has a fixed rank as the number of users increases. For this case, they show that methods based on the singular-value decomposition are able (under suitable assumptions) to accurately predict the full user-item ranking matrix as the number of users approaches infinity given that each user ranks at least a constant fraction of all items.

VI. Probability that an Edge Exists in the User Graph

In this section, we derive the probability that an edge between two users in the user graph exists in the random graph $G_{c,N}$ that we defined in the previous section.

Lemma 3: Let $p(I_N, m_M)$ be the probability that there exists an edge between two users in the graph $G_{c,N}$. If

$$\lim_{N\to\infty} \frac{m_N^2}{I_N} = 0$$

then we have that

$$p(I_N, m_N) = \frac{m_N^2}{I_N} + o(\frac{m_N^2}{I_N}).$$

Proof: Here we only provide a rough sketch of the proof. Let S_u be the set of items that have been ranked by a user u in $G_{c,N}$. The probability $p(I_N, m_N)$ that there exists an edge between two users u, v in $G_{c,N}$ is then given by

$$p(I_N, m_N) = P(S_u \cap S_v \neq \emptyset) \quad = \quad 1 - P(S_u \cap S_v = \emptyset)$$

By assumption, for N large enough we have $m_N \leq \frac{I_N}{2}$ and

$$\begin{aligned} P(S_u \cap S_v = \emptyset) &= \frac{C_{I_N - m_N}^{m_N}}{C_{I_N}^{m_N}} = \frac{\frac{(I_N - m_N)!}{m_N!(I_N - 2m_N)!}}{\frac{I_N!}{m_N!(I_N - m_N)!}} \\ &= \frac{(I_N - m_N) \cdot ... \cdot (I_N - 2m_N + 1)}{I_N \cdot ... \cdot (I_N - m_N + 1)}, \end{aligned}$$

where for positive integers n, m, $n \geq m$, we have

$$C_n^m = \frac{n!}{m!(n-m)!}.$$

We then have

$$\begin{aligned} P(S_u \cap S_v = \emptyset) &\approx \left(\frac{I_N - m_N}{I_N}\right)^{m_N} = \left(1 - \frac{m_N}{I_N}\right)^{m_N} \\ &\approx e^{-\frac{m^2}{I_N}}. \end{aligned}$$

We then obtain

$$p(I_N, m_N) = 1 - P(S_u \cap S_v) \approx 1 - e^{\frac{-m^2}{I}} \approx \frac{m_N^2}{I_N}.$$

■

Using the above lemma, the conditions in Theorem 1 are the same as the conditions that a Erdos-Renyi $G(N, p(I_N, m_N)$ random graph is fully connected. However, the graph $G_{c,N}$ is not a Erdos-Renyi graph as the probabilities that edges exist are not independent of each other. To see this note the following. Consider a user u and suppose that this user has m_N neighbours in $G_{c,N}$ such that there exists no edge between any of these neighbours. Now consider another user $v \in G_{c,N}$. Given that v does not have an edge to any of the neighbour nodes of node u, the probability that there exists an edge between u and v is equal to 0, as each user makes m_N rankings.

VII. The Many-User Case

In this section we prove Theorem 1. To do that we first consider the three cases of a many-user, balanced, and many-item case separately. We conclude the section with a discussion of the general case. Recall that we focus on our analysis on a fixed class, where N is the total number of users in this class. Whenever we refer in the following to "users" then we mean users in this class.

The many-user case is the easiest to analyze. For this case where

$$\lim_{N\to\infty} \frac{N}{I_N log I_N} = \infty$$

we show that the probability that the graph $G_{c,N}$ is fully connected converges to 1 as N approaches infinity, as long as each user ranks at least 2 items, i.e. we have $m_N \geq 2$.

To do that we consider the *item graph* denoted by K_N. Each item $i \in \mathcal{I}_N$ represents a node in K_N. There exists an edge in K_N between two item $i, j \in \mathcal{I}_N$ if and only if there exists a user u that t ranks both i and j.

The following lemma makes a connection between the user graph $G_{c,N}$ and the above defined item graph. It states that if $G_{c,N}$ is not fully connected then there exists two non-empty sets $A, B \subset I_N$ such that $A \cap B = \emptyset$ and $A \cup B = I_N$, and all user either rank items exclusively in set A or B.

Lemma 4: Suppose that the item graph K_N is fully connect, i.e. for all non-empty subsets $A, B \subset I_N$, such that $A \cap B = \emptyset$ and $A \cup B = I_N$ we have that there exists a user u that ranks an item in set A and B. Then the user graph $G_{c,N}$ is fully connected.

Proof: Let us consider two users u and v. We have to show that under the above conditions there exists a path from u at v in $G_{c,N}$. Suppose that no such path exists, and let S^1 be the set of all users that can be reached from user u. Furthermore, let I^1 be the set of all items that are ranked by users in S^1. Note that there will exist a path from u to v if we have that $I^1 = I_N$; hence $I^1 \neq I_N$. Set $A = I^1$ and $B = I_N \backslash I^1$.

By assumption, there exists a user w that ranks an item in the set A and B. This implies that user w ranks an item that has also been ranked by at least one user in S^1, and there exists a path from u to w. Therefore, user w is in S^1. However, this contracts our assumption that A contains all the items ranked by users in S^1, and the result follows. ■

The next lemma provides a necessary condition for the item graph to be fully connected.

Lemma 5: Let P_N be the probability that the item graph K_N is fully connected. If

$$\lim_{N\to\infty} \frac{N}{I_N log I_N} > \infty,$$

then we have $\lim_{N\to\infty} P_N = 1$.

Proof: Here we provide a sketch of the proof. Without loss of generality, we can assume that $m_N = 2$. In this case, each user creates one edge in the item graph K_N (i.e. the edge between the two items that the user ranks), where each of the $I_N(I_N - 1)$ possible edge is equally likely to be created by a given user independent of the choices of the other users. Note however that not all of the edges created by users lead to distinct edges, i.e. two (or more) users may rank the same two items and hence duplicate an edge (but this will happen with a small probability if I_N is large). However, as by assumption we have that $\lim_{N\to\infty} \frac{N}{log N I_N} > \infty$, one can show that for every constant c we have that the users create at least $I_N log I_N + c$ distinct edges with probability 1. The result the follows using standard results for random graphs [1]. ■

Combining Lemma 4 and 5, we immediately obtain the following result.

Corollary 2: Let P_N be the probability that the user graph $G_{c,N}$ is fully connected. If

$$\lim_{N\to\infty} \frac{N}{I_N log I_N} > \infty,$$

then we have $\lim_{N\to\infty} P_N = 1$.

VIII. THE BALANCED CASE.

Next we consider the balanced case where

$$\lim_{N\to\infty} \frac{N}{I_N} = b$$

for some positive constant b. We distinguish between two cases for this situation, namely the two cases where

$$\lim_{N\to\infty} \frac{N m_N}{I_N log I_N} = \infty$$

and

$$\lim_{N\to\infty} \frac{N m_N}{I_N log I_N} < \infty.$$

Suppose that $\lim_{N\to\infty} \frac{N m_N}{I_N log I_N} = \infty$. Furthermore, suppose that rather than choosing m_N items at random and uniformly over all items, each user will choose m_N times an item out of all possible items, where items are chosen independently and uniformly over all possible items. Note that in this case, a user may choose a given item several times (although this will happen with a small probability if I_N is large), and the total number of items a user ranks is strictly less than m_N. We then immediately have the following lemma which we state without proof.

Lemma 6: Let P'_N be the probability that the graph $G_{c,N}$ is fully connected for the case where each user choose m_N times an item with possible repetitions, and let P_N be the probability that the graph $G_{c,N}$ is fully connected for the case where each user chooses m_N distinct items. If $lim_{N\to\infty} P'_N = 1$ then we have that $lim_{N\to\infty} P_N = 1$.

Assuming that users choose $m_N = 2l$ items as given above, we create an item graph as follows. Each user creates $l = m_N/2$ edges in the item graph K_N, where the lth edge is the edge between the items chosen at stage $(2l-1)$ and $2l$. Note that because of the above assumption, the l edges are chosen independently and each edge in the item graph is equally likely to be chosen. Using this fact, we obtain the following result.

Lemma 7: Let P_N be the probability that the item graph K_N is fully connected. If $\lim_{N\to\infty} N/I_N = b$ for some positive constant b and

$$\lim_{N\to\infty} \frac{N m_N}{log N I_N} > \infty,$$

then we have that $\lim_{N\to\infty} P_N = 1$.

The proof for the above lemma is similar to the proof of Lemma 5. Combining Lemma 7 with Lemma 4, we immediately obtain the following result.

Corollary 3: Let P_N be the probability that the user graph $G_{c,N}$ is fully connected. If $\lim_{N\to\infty} N/I_N = b$ for some positive constant b and

$$\lim_{N\to\infty} \frac{Nm_N}{logNI_N} > \infty,$$

then we have that $\lim_{N\to\infty} P_N = 1$.

For the second case where

$$\lim_{N\to\infty} \frac{Nm_N}{I_N logI_N} < \infty$$

we obtain the following result.

Theorem 2: Let P_N the the probability that the graph $G_{c,N}$ is fully connected, and suppose that $\lim_{N\to\infty} N/I_N = b$ for some positive constant b. If

$$\frac{Nm_N^2}{I_N} = \omega(logN),$$

then we have $\lim_{N\to\infty} P_N = 1$. If

$$\frac{Nm_N^2}{I_N} = logN + a + o(1)$$

for a positive constant a, then we have

$$\lim_{N\to\infty} P_N \leq 1 - e^{e^{-a}}.$$

Whereas the proofs of the above lemmas was rather straightforward, the proof of Theorem 2 is more involved. The proof is of independent interest as it makes an interesting extension of the proof-techniques for Erdos-Renyi random graphs, to random graphs were edges are not created independently but tend to be clustered (see the comments in the next paragraph). We provide an outline of the proof in the appendix.

Note that each user creates $m_N(m_N-1) \approx m_N^2$ edges in the item graph K_N. Therefore, the conditions for full connectivity in Theorem 2 are similar to the condition for fully connectivity for the random graph $G(I_N, Nm_N^2)$ where in a graph consisting of I_N nodes a total of Nm_N^2 edges are created at random and uniformly over all possible edge selections. Note however that the graph K_N is not a $G(I_N, Nm_N^2)$ random graph. To see this, note that the $m_N(m_N-1)$ created by a given user form a clique in K_N consisting of the m_N items chosen by this user.

IX. The Many-Item Case

For the many-item case, we have the following result.

Theorem 3: Let P_N the the probability that the graph $G_{c,N}$ is fully connected, and suppose that $\lim_{N\to\infty} \frac{Nm_N}{I_N} = 0$. If

$$\frac{Nm_N^2}{I_N} = \omega(logN),$$

then we have $\lim_{N\to\infty} P_N = 1$. If

$$\frac{Nm_N^2}{I_N} = logN + a + o(1)$$

for a positive constant a, then we have

$$\lim_{N\to\infty} P_N \leq 1 - e^{e^{-a}}.$$

The proof for this case is an extension of the standard proof techniques for Erdos-Renyi graphs [1].

X. The General Case

The above discussion does not fully cover all situations for which Theorem 1 applies. In particular, it does not cover all the situations for which there exist positive constants b_1 and b_2 such that

$$\lim_{N\to\infty} \frac{N}{I_N} \leq logI_N + b_1$$

and

$$\lim_{N\to\infty} \frac{Nm_N}{I_N} \geq b_2.$$

Theorem 1 for these cases can be shown using the same line of argument used to prove Theorem 2.

XI. Conclusions

We derived a lower-bound on the number of rankings required by recommender systems using collaborative filtering for a ranking model where it is assumed that users can be clustered into different classes, and users in the same class have all the same ranking vector. To derive a lower-bound for this ranking model, we considered the special case of complete separation: users in different classes have different rankings for all items, i.e. if $c' \neq c''$ then we have $r_{c'}(i) = r_{c''}(i)$, $i = 1, ..., I$. Our analysis is based on a random graph model on the set of all users.

We do not show that lower-bound on the number of rankings per user that we derive in this paper is tight, i.e. we did not show that there exists an algorithm that is able to accurately predict the full user-item ranking matrix given that each users ranks this minimal number of items. This question is on-going work, and preliminary results indicate that such an algorithm might indeed exists. For the derivation of such an algorithm we do not use the assumption of complete separation, but allow the ranking vector of different classes to “overlap”.

Besides the lower-bound on the number of rankings, an important contribution of the paper is to show that recommender systems using collaborative filtering can be studied using a random graph model. This allows to apply the rich literature, and wealth of results, on random graphs to this problem. We are not aware of prior work in the literature that explicitly makes this connection.

XII. Acknowledgments

I would like to thank Francois Baccelli, Marc Lelarge, and Laurent Massoulie for fruitful discussions, comments, and suggestions, on this problem.

References

[1] B. Bollobas, *Random Graphs,* Cambridge University Press, 2nd edition, 2001.

[2] B. Awwerbuch, Y. Azar, Z. Lotker, B. Patt-Shamir, and M. Tuttle, *Collaborate with Strangers to find Own Preferences*, Proc. 16th ACM-SIAM Symposium on Discrete-Algorithms (SODA), pages 1174-1183, 2005.

[3] N. Alon, B. Awerbuch, Y. Azar, B. Patt-Shamir, *Tell me who I am: An Interactive Recommendation System*, Proc. 18th ACM symposium on Parallelism in algorithms and architectures (SPAA ’06), pages 1–10, 2005.

[4] J. Kleinberg, M. Sandler, *Convergent Algorithms for Collaborative Filtering*, Proc. 4th ACM Conference on Electronic Commerce, 2003

[5] J. Kleinberg, M. Sandler, *Using Mixture Models for Collaborative Filtering*, Proc. 36th ACM Symposium on Theory of Computing, 2004

[6] C.H. Papadimitriou, P. Raghavan, H. Tamaki, S. Vempala, *Latent Semantic Indexing: A Probabilistic Analysis,* Proc. 17th ACM Symp. on Principles of Database Systems (PODS), pages 159–168, 1998.

[7] Y. Azar, A. Fiat, A.R. Karlin, F. McSherry, J. Saia, *Spectral Analysis of Data*, Proc. 33rd ACM Symposium on Theory of Computing, pages 619 – 626, 2001