Adaptive Sampling and Learning in Recommendation Systems

by

Lingfang Zhu

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved November 2015 by the
Graduate Supervisory Committee:

Guoliang Xue, Chair
Jingrui He
Hanghang Tong

ARIZONA STATE UNIVERSITY

December 2015

ABSTRACT

This thesis studies recommendation systems and considers joint sampling and learning. Sampling in recommendation systems is to obtain users' ratings on specific items chosen by the recommendation platform, and learning is to infer the unknown ratings of users to items given the existing data. In this thesis, the problem is formulated as an adaptive matrix completion problem in which sampling is to reveal the unknown entries of a $U \times M$ matrix where U is the number of users, M is the number of items, and each entry of the $U \times M$ matrix represents the rating of a user to an item. In the literature, this matrix completion problem has been studied under a static setting, i.e., recovering the matrix based on a set of partial ratings. This thesis considers both sampling and learning, and proposes an adaptive algorithm. The algorithm adapts its sampling and learning based on the existing data. The idea is to sample items that reveal more information based on the previous sampling results and then learn based on clustering. Performance of the proposed algorithm has been evaluated using simulations.

TABLE OF CONTENTS

Page

LIST OF TABLES iii

LIST OF FIGURES iv

CHAPTER

1 INTRODUCTION 1

2 PROBLEM FORMULATION 4

3 METHODOLOGY 9

3.1 Two-User Hypothesis Testing 9

3.2 Cluster Assignment 15

3.3 Adaptive Sampling and Learning 18

4 PERFORMANCE EVALUATION 22

4.1 Case 1: Noiseless Reporting 22

4.2 Case 2: Noisy Reporting 23

5 CONCLUSIONS AND FUTURE WORK 26

REFERENCES 27

LIST OF TABLES

Table Page

3.1 Table of Notation 19

LIST OF FIGURES

Figure		Page
3.1	The Number of Required Samples versus the Error Bound	14
3.2	The Number of Required Samples versus the Flipping Probability	14
3.3	The Number of Required Samples versus the Percentage of Difference	15
3.4	The Number of Required Samples under the Proposed Algorithm and the Random Sampling	18
4.1	The Performance Comparison between Joint and Random.	23
4.2	The Sampling Rate versus the Cluster Size.	24
4.3	The Performance Comparison between Joint and Random.	24
4.4	The Sampling Rate versus the Cluster Size.	25

Chapter 1

INTRODUCTION

Modern recommendation systems, such as the recommendation systems used by Amazon and Netflix, involve a large number of users and a large number of items. The input of a recommendation system is the partial ratings of the items given by the users. For example, a user may rate a movie from 1 to 5. The output of a recommender system is a few items that each user would like. The common approach is to exploit the similarity among users and items to predict users' preference. Mathematically, the problem can be formulated as a matrix completion problem. Assume there are U users and M items, then the rating matrix is an $U \times M$ matrix whose entries are the ratings. Then to learn all the ratings, the problem is a matrix completion problem which is to recover all unknown entries from the known entries. The matrix completion problem has been extensively studied in the literature recently.

It is obvious that the problem is impossible to solve if the underlying matrix has no structure (i.e., can be an arbitrary matrix), so the existing studies focus on the cases where the matrix has structural properties that can be exploited. Two popular structural properties that have been utilized in the literature: (1) The first popular one is the low-rank assumption (Candès and Recht, 2009; Candès and Tao, 2010; Recht, 2011; Gross, 2011; Keshavan *et al.*, 2010) which assumes the matrix is a low rank matrix, i.e., the rank (denoted by K) is much smaller than U and M. This low-rank assumption implies each row (the ratings from a specific user) can be represented by a linear combination of K basis vectors. Therefore, if the known ratings are sufficient for us to recover the K basis vectors, then we can utilize them to recover the full matrix. Algorithms used to recover low-rank matrices include 1-1 norm

minimization (Candès and Recht, 2009; Candès and Tao, 2010; Recht, 2011; Gross, 2011; Keshavan *et al.*, 2010) and alternating minimization (Jain *et al.*, 2013). (2) The second popular assumption is the clustering assumption (Tomozei and Massoulié, 2014; Barman and Dabeer, 2012; Xu *et al.*, 2013; Zhu *et al.*, 2014) which assumes that users, or items or both form clusters. For example, when users are clustered, the users in the same cluster give the same rating to the same item. Assume users form K clusters and items form K clusters, then recovering the rating matrix is to recover a $K \times K$ matrix. Each entry of the $K \times K$ matrix represents the rating to a cluster of items from a cluster of users.

With these assumptions on the rating matrices, the fundamental limits and computationally efficient matrix completion algorithms have been studied in the literature. However, most models used so far took a static view, where the goal is to predict the unknown ratings as accurate as possible from a fixed set of revealed partial ratings. In other words, it is a single-shot optimization problem without considering sampling.

In practice, new ratings are added to the system every day, and new users and new items are added into the system constantly. After we recommend items to a user, if the user purchases the item, she/he may rate the item, which provides more ratings for future recommendation. The system can even offer free products (samples) to users to seek their feedback to obtain more ratings and enhance the performance of the system. Therefore, when we decide on which ratings to obtain by various methods. The focus of this thesis is on adaptive and dynamic matrix completion algorithms to develop efficient sampling and learning algorithms to recover the ratings matrix with a minimum number of samples. The remaining of the thesis is organized as follows: Chapter 2 presents the basic models and problem formulation, Chapter 3 presents analytical analysis and proposes the adaptive sampling and learning algo-

rithm, Chapter 4 presents the performance evaluation using simulations and Chapter 5 concludes this thesis.

Chapter 2

PROBLEM FORMULATION

In this thesis, we consider the model studied in Barman and Dabeer (2012); Xu *et al.* (2014); Zhu *et al.* (2014). In other words, the thesis focuses on the matrices with clustering structures instead of studying general low rank matrices. The reason the clustering assumption is chosen for this thesis is because it has been discovered in the literature Barman and Dabeer (2012); Xu *et al.* (2014); Zhu *et al.* (2014) that with the clustering structure, algorithms with much lower computational complexity can be developed to achieve better and more robust results. It is noted that both clustering and low-rank are modeling assumptions. Most real-world datasets are incomplete so both assumptions are difficult to be validated in practice. The only meaningful validation of the assumptions seems to compare the recommendation accuracy resulted from the algorithms derived under different assumptions to see which algorithm (assumption) yields the best recommendation accuracy in real world datasets. A recent study Zhu *et al.* (2014) has given favorable answer to the clustering assumption. As it becomes clear in the remaining parts of the thesis, the clustering structure can be exploited further in the dynamic sampling/recommendation setting to significant improve the system performance.

We next review the model used in Zhu *et al.* (2014). The notions used in Zhu *et al.* (2014) are adopted in this thesis as our algorithm extends the algorithm in Zhu *et al.* (2014) in a dynamic setting. The ratings to items from users are represented by a $U \times M$ matrix, where U is the number of users and M is the number of items. The rating matrix is denoted by $\mathbf{B}$. In Zhu *et al.* (2014), the authors considered both clustering and co-clustering cases, where in the clustering case, the users (or

the items) form K clusters. Each user (or item) belongs to one and only one cluster, so each user-cluster has U/K users, and each item-cluster has M/K items. In the model, the users in the same cluster give the same rating to the same item. The authors in Zhu *et al.* (2014) further considered the case where both users and items are cluster, called co-clustering. As a simple example, assume users are clustered and items are not. An example of a rating matrix $\mathbf{B}$ is then given below

$$\mathbf{B} = \left(\begin{array}{c|cccccc} & \text{item 1} & \text{item 2} & \text{item 3} & \text{item 4} & \text{item 5} & \text{item 6} \\ \text{user 1} & 5 & 1 & 2 & 4 & 1 & 1 \\ \text{user 2} & 5 & 1 & 2 & 4 & 1 & 1 \\ \text{user 3} & 5 & 1 & 2 & 4 & 1 & 1 \\ \text{user 4} & 1 & 3 & 5 & 5 & 2 & 5 \\ \text{user 5} & 1 & 3 & 5 & 5 & 2 & 5 \\ \text{user 6} & 1 & 3 & 5 & 5 & 2 & 5 \end{array}\right), \tag{2.1}$$

where the users are separated into two clusters, where cluster 1 includes users 1, 2, 3 and cluster 2 includes users 4, 5, 6.

The true rating matrix in practice is not available. The goal of the matrix completion problem is to recover this matrix from a sparse and noisy observed rating matrix, denoted by $\mathbf{R}$. Given the true rating matrix, the observed rating matrix is generated by removing majority of the ratings and then flip remaining entries with a certain probability. The following picture given in Zhu *et al.* (2014) illustrates the overall process.

$$\mathbf{B} \rightarrow \text{a noisy channel } \tilde{\mathbf{R}} \rightarrow \text{an erasure channel } \mathbf{R}.$$

In other words, the observed rating matrix $\mathbf{R}$ is generated by passing the true rating matrix $\mathbf{B}$ through a noisy channel (flipping the ratings) and then an erasure channel

(removing the ratings). An example of $\mathbf{R}$ generated from the $\mathbf{B}$ in (2.1) is given below

$$\mathbf{B} = \left(\begin{array}{c|cccccc} & \text{item 1} & \text{item 2} & \text{item 3} & \text{item 4} & \text{item 5} & \text{item 6} \\ \text{user 1} & 5 & \star & 1 & \star & \star & \star \\ \text{user 2} & 5 & 1 & \star & \star & \star & 1 \\ \text{user 3} & \star & \star & 3 & \star & 2 & 1 \\ \text{user 4} & \star & 3 & \star & 5 & \star & 5 \\ \text{user 5} & \star & 1 & 5 & \star & \star & \star \\ \text{user 6} & \star & \star & 5 & \star & \star & 5 \end{array}\right). \tag{2.2}$$

In this model, the erasure models the fact that only a small fraction of ratings are known to the recommendation platform and the random flipping models the fact that users may give inaccurate ratings in practice because of various reasons. A recommendation system requires less number of ratings is called more effective and can tolerate more errors is called more robust. Both effectiveness and robustness are important criterions for the design of a practical recommendation system.

A significant contribution of Zhu *et al.* (2014) is to take into account information rich and information sparse users in the recovering the underlying rating matrix, where an information-rich user is a user who rates βM movies on average, where β is a positive constant. Information-poor users, which are the majority, only rate $\log M$ movies each. It shows that with the existence of information-rich users in each cluster, the authors proved that the true rating matrix can be fully recovered when we have $\omega(MK \log M)$ noisy entries. The authors also proved that MK entries are necessary. This surprising result shows that the existence of heterogeneous users can significantly help us recover the rating matrix. Zhu *et al.* (2014) also showed that the existence of information-rich users in real-world datasets. The paper found that in MovieLens dataset, the number of users who rated more than 1,000 movies is 38 while the total number of users is 6,040, and 73% of users gave less than 200 ratings.

Now come back to the adaptive sampling question, the implication of results in Zhu *et al.* (2014) is that we should recommend items to users in such a way that we can quickly identify one information-rich user from each user cluster. Then for the remaining users, we only needs to assign them to the "right" cluster. The ratings can then be recovered by properly aggregating the observed ratings within a cluster (e.g.., using a majority voting for each item among observed ratings). There are a sequence of questions needed to be answered:

- **Question 1:** Given two users, how many co-rated items are needed to tell whether they are in the same cluster or not with a given accuracy? In other words, let $p_{e,W}$ denote the error probability of hypothesis testing on whether two users are in the same cluster, the problem is

$$\min_{W} p_{e,W} \leq \bar{p}$$

  for given a requirement on $\bar{p}$.

- **Question 2:** Now assume we identified an information-rich user for each cluster. Given a new user, how to sample the users' item to identify its cluster with a minimum number of samples? Let $C(x, W)$ denote the cluster the algorithm assigns user x to after sampling W ratings and $C(x)$ denote the actual cluster the user is in. The problem is to

$$\min_{W} \Pr\left(C(x, W) \neq C(x)\right) \leq \bar{p},$$

  where $\bar{p}$ is the requirement on the error probability.

- **Questions 3:** Finally, after we identify the initial cluster of all users, how to recover the rating matrix based on the known samples? For this question, we will leverage the algorithm in Zhu *et al.* (2014).

In the following chapter, the answers to each of the three questions above will be presented.

Chapter 3

METHODOLOGY

This thesis takes a significant step of answering the questions mentioned in the previous chapter. We will first consider the hypothesis testing problem for two users, then the cluster identification problems, and finally the profit maximization problem.

3.1 Two-User Hypothesis Testing

The first question is when given users u and v, how many corated items are needed to tell whether they are in the same cluster or not. Let us first consider a simple scenario where there is no flipping, i.e., assume the reported ratings are all accurate. In order to conduct some preliminary theoretical analysis to derive the intuition of this problem, we assume the ratings are binary $\{-1, +1\}$, and the following assumption is also made.

Assumption 1 *For two users in different clusters, at least β fraction of their ratings are different.*

Based on this assumption, we consider the following hypothesis testing problem.

Hypothesis Testing: Let $\mathcal{M}_c$ denote the set of items rated by both user u and user v. We want to know whether the two users are in the same cluster or not. In other words, the binary hypothesis testing is

- H_0 : users u and v are in the same cluster.
- H_1 : users u and v are not in the same cluster.

We consider the following hypothesis testing rule.

Hypothesis Testing Rule for Zero Flipping Probability: The two users are declared to be in the same cluster if both users agree on the ratings in $\mathcal{M}_c$; otherwise we declare they are in a different cluster.

Let A_0 denote the event that hypothesis H_0 is accepted and A_1 denote the event that hypothesis H_1 is accepted. The next lemma presents the type-I and type-II errors.

Lemma 1 *Assume the set of co-rated items are uniformly, randomly selected from all items. The hypothesis testing rule above is the maximum likelihood test. Furthermore,*

$$\Pr(A_1|H_0) = 0 \tag{3.1}$$

$$\Pr(A_0|H_1) \leq \frac{\binom{(1-\beta)M}{|\mathcal{M}_c|}}{\binom{M}{|\mathcal{M}_c|}}. \tag{3.2}$$

Proof: Equality (3.1) is obvious since given two users are in the same cluster, all of their ratings should agree. When the flipping probability is zero, A_1 will not occur. To obtain inequality (3.2), it is noted that given two users are in different clusters, they give different ratings to at least βM fraction of items. Therefore, the probability that none of the $|\mathcal{M}_c|$ randomly selected items are from the αM items is

$$\frac{\binom{(1-\beta)M}{|\mathcal{M}_c|}}{\binom{M}{|\mathcal{M}_c|}}.$$

We next prove that the hypothesis testing rule is the maximum likelihood testing. First given A_1

$$\Pr(A_1|H_1) = 1 > \Pr(A_1|H_0) = 0.$$

Second given A_0,

$$\Pr(A_0|H_0) = 1 > \Pr(A_1|H_1).$$

Therefore, the hypothesis testing rule is the maximum likelihood rule. □

Note that $\Pr(A_1|H_0)$ is the so called type-I error (also called false positive), and $\Pr(A_0|H_1)$ is the so called type-II error (also called false negative).

The next is to consider the scenario the flipping probability is not zero. Let p_f denote the flipping probability and assume $p_f < 0.5$. When the flipping problem is nonzero, even two users in the same cluster can have different observed ratings, where an observed rating is a probabilistically-flipped version of the user's true rating. Therefore, we may want to declare two users are in the same cluster when most of their ratings (even not all of them) agree.

Hypothesis Testing Rule for Non-zero Flipping Probability: The two users are declared to be in the same cluster if both users agree on at least ρ fraction of ratings in $\mathcal{M}_c$; otherwise we declare they are in a different cluster.

We now analyze the performance of this hypothesis testing rule. First, we have

$$\begin{aligned}\Pr(A_1|H_1) &= \sum_{m=0}^{|\mathcal{M}_c|} \frac{\binom{(1-\gamma)M}{|\mathcal{M}_c|-m}\binom{\gamma M}{m}}{\binom{M}{|\mathcal{M}_c|}} \times \\ &\quad \sum_{\kappa=\rho|\mathcal{M}_c|}^{|\mathcal{M}_c|} \sum_{z=\kappa-|\mathcal{M}_c|+m}^{\min\{\kappa,m\}} \binom{m}{z}\binom{|\mathcal{M}_c|-m}{\kappa-z} p_f^{m+\kappa-2z}(1-p_f)^{|\mathcal{M}_c|-m-\kappa+2z},\end{aligned}$$

where m is the number of items for which the true ratings of user u and v are different, κ is the number of different observed ratings, and z is the number of observed different ratings for which the true ratings are also different. Following a similar analysis, we have

$$\Pr(A_1|H_0) = \sum_{\kappa=\rho|\mathcal{M}_c|}^{|\mathcal{M}_c|} \left(\begin{array}{c} |\mathcal{M}_c| \\ \kappa \end{array} \right) \left(1 - p_f^2 - (1-p_f)^2\right)^{\kappa} \left(p_f^2 + (1-p_f)^2\right)^{|\mathcal{M}_c|-\kappa}.$$

Note that $\Pr(A_1|H_1)$ is complex and difficult to analyze, so we use the following calculation to approximate it. Given H_1, when an item is uniformly at random selected, the two observed ratings from users u and v are different with probability

$$q = \gamma\left(p_f^2 + (1-p_f)^2\right) + (1-\gamma)\left(1 - p_f^2 - (1-p_f)^2\right).$$

So we use the following approximation.

$$\Pr(A_1|H_1) \approx \sum_{\kappa=\rho|\mathcal{M}_c|}^{|\mathcal{M}_c|} \left(\begin{array}{c} |\mathcal{M}_c| \\ \kappa \end{array} \right) q^{\kappa}(1-q)^{|\mathcal{M}_c|-\kappa}.$$

From the results above, we can see that under both H_0 and H_1, the results of hypothesis testing are results of binomial random variables. Since a binomial random variable is the sum of i.i.d. Bernolli random variables, it is not difficult to see that according to law of large numbers, the value of the binomial random variable will concentrate around its mean. Let X be $B(|\mathcal{M}_c|, q')$, where $q' = 1 - q_f^2 - (1-q_f)^2$, then

$$\begin{aligned} E[X] &= |\mathcal{M}_c|q' \\ Var(X) &= |\mathcal{M}_c|q'(1-q'). \end{aligned}$$

Let Y be $B(|\mathcal{M}_c|, q)$, then

$$\begin{aligned} E[Y] &= |\mathcal{M}_c|q \\ Var(Y) &= |\mathcal{M}_c|q(1-q). \end{aligned}$$

Theorem 2 *Assume two users are uniformly at random selected from the M users. Given an upper bound $\bar{p}$ on the error probability, the minimum number of co-rated items needed is*

$$W = \arg\min_{W} \left\{ W : \min_{S} \frac{1}{K}(1 - F_X^{(W)}(S)) + \frac{K-1}{K} F_Y^{(W)}(S) \leq \bar{p} \right\},$$

where $F_X^{(W)}(\cdot)$ is the cumulative distribution function of X with $|\mathcal{M}_c| = W$, and $F_Y^{(W)}(\cdot)$ is the cumulative distribution function of Y.

Proof: Note that $1/K$ is the probability that the two users are in the same cluster and $1 - 1/K$ is the probability they are not in the same cluster. $1 - F_X^{(W)}(S)$ is the probability that the number of different observed ratings is more than S when sampling W items and when the two users are in the same cluster. $F_Y^{(W)}(S)$ is the probability that the number of different observed ratings is at most S when sampling W items and the two users are in the same cluster. These two are the type-I and type-II errors, respectively. Therefore,

$$\min_{S} \frac{1}{K}(1 - F_X^{(W)}(S)) + \frac{K-1}{K} F_Y^{(W)}(S)$$

is the minimum error probability by choosing the optimal threshold in hypothesis testing. □

The following figures show the value of W with choices of parameters. Figure 3.1 shows the case in which $p_f = 0.2$, which is the flipping probability, $\gamma = 0.5$, which is the fraction of ratings that are different, $K = 5$, which is the number of clusters. The simulation shows that to 35 samples are needed to achieve error probability 0.1 and 146 samples are needed to reduce the error probability to 0.01.

Figure 3.2 shows the case in which the flipping probability varied from 0.1 to 0.3. Similar to the first case, $\gamma = 0.5$, $K = 5$, and the upper bound on the error probability was chosen to be 0.05. From the figure, it can be seen that the number of

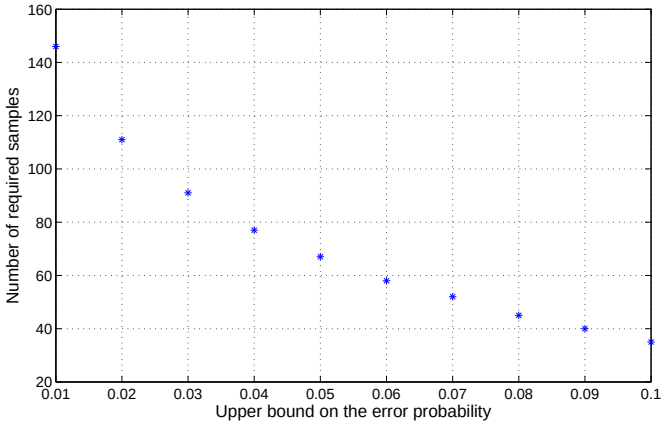

Figure 3.1: The Number of Required Samples versus the Error Bound

required samples is very sensitive the flipping probability. When the flipping probability increases from 0.1 to 0.3, the number of required samples increases from 19 to 348.

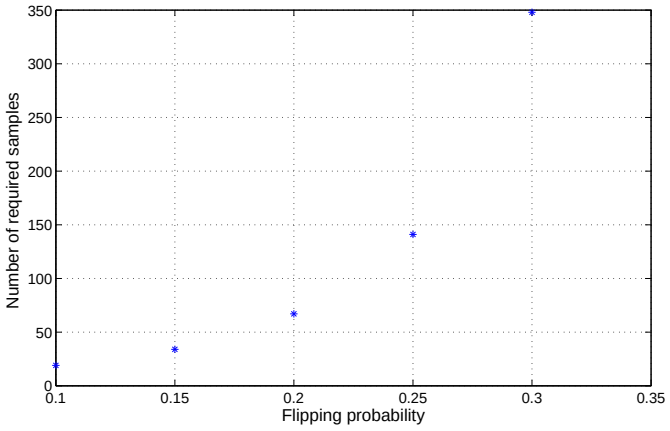

Figure 3.2: The Number of Required Samples versus the Flipping Probability

Figure 3.3 shows the case in which the percentage of different ratings among all ratings varied from 0.2 to 0.5. Similar to the first case, $p_f = 0.2$, $K = 5$, and the upper bound on the error probability was chosen to be 0.05. When γ increases from 0.2 to 0.5, the number of required samples decreases from 398 to 67.

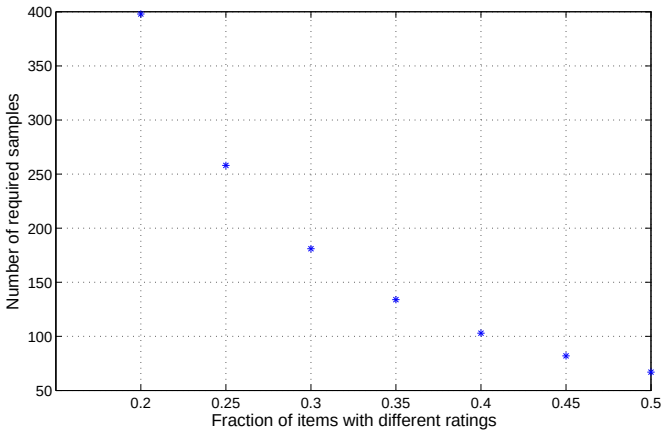

Figure 3.3: The Number of Required Samples versus the Percentage of Difference

3.2 Cluster Assignment

This section focuses on assigning a user to the "correct" cluster after identifying one information-rich user for each cluster. This is a somewhat difficult question to answer as it depends on the observed ratings of the information rich users. To obtain some analytical understand, the problem is formulated as follows. Assume K information rich users are given, one for each cluster. Without loss of generality, assume the users are indexed $1, 2, \cdots, K$. Furthermore, let $\mathcal{R}_k$ denote the set of observed ratings of user k. For convenience, $R_{um} = 0$ is used to denote that the rating to item m from user u is missing.

Again, we start from the case the flipping probability is zero, i.e., all observed ratings are true ratings. In this case, if there exists an item such that $R_{um} \neq R_{vm} \neq 0$, then users u and v are not in the same cluster. In other words, the possible clusters for user u can be reduced by eliminating information-rich users who have different ratings on the same item with user u. To apply this intuition, the following fast sampling algorithm is used to identify the cluster of a user.

Fast sampling algorithm:

Let $\mathcal{U}_t$ denote the remaining information users at iteration t, and $\mathcal{U}_0 = \{1, \cdots, K\}$. At tth iteration,

(i) Define

$$\begin{aligned} \tilde{U}_{1,m} &= \sum_{k \in \mathcal{U}_t} 1_{R_{km}=1} \\ \tilde{U}_{0,m} &= \sum_{k \in \mathcal{U}_t} 1_{R_{km}=0} \\ \tilde{U}_{-1,m} &= \sum_{k \in \mathcal{U}_t} 1_{R_{km}=-1} \end{aligned}$$

Select item m^* such that

$$\begin{aligned} m^* \in \arg\max_m & \frac{\tilde{U}_{1,m}}{2\tilde{U}_{1,m} + \tilde{U}_{0,m}} \log(2\tilde{U}_{1,m} + \tilde{U}_{0,m}) + \frac{\tilde{U}_{-1,m}}{2\tilde{U}_{-1,m} + \tilde{U}_{0,m}} \log(2\tilde{U}_{-1,m} + \tilde{U}_{0,m}) \quad (3.3) \\ & + \frac{\tilde{U}_{0,m}|\mathcal{U}_t|}{(2\tilde{U}_{1,m} + \tilde{U}_{0,m})(2\tilde{U}_{-1,m} + \tilde{U}_{0,m})} \log \frac{(2\tilde{U}_{1,m} + \tilde{U}_{0,m})(2\tilde{U}_{-1,m} + \tilde{U}_{0,m})}{|\mathcal{U}_t|}, \quad (3.4) \end{aligned}$$

i.e., select the item that provides the most information.

(ii) Set

$$\mathcal{U}_{t+1} = \mathcal{U}_t \setminus \{k : k \in \mathcal{U}_t \text{ and } R_{km^*} = -\text{sgn}(R_{um^*})\},$$

i.e., remove information-rich users that are not possibly in the same cluster with user u.

Equation (3.4) is motivated by the concept of entropy in information theory Cover and Thomas (1991). Define $\tilde{K} = \sum_{k \in \mathcal{U}_t, 1_{R_{km}=0}} B_{km}$, so $\tilde{K}$ is the number of "1" ratings among those unknown ratings for time m. It is easy to see that

$$E[\tilde{K}] = 0.5\tilde{U}_0$$

Note that the rating of the sampled item, say item m, is equally likely to be 1 or -1. Assume it is 1, then the probability that it is in the same cluster with information-rich user k with $R_{km} = 1$ is

$$p_1 = \frac{1}{\tilde{U}_{1,m} + 0.5\tilde{U}_{0,m}}.$$

The probability that it is in the same cluster with information-rich user k with $R_{km} = 0$ is

$$p_{10} = \frac{1}{2\tilde{U}_{1,m} + \tilde{U}_{0,m}}.$$

Similarly, assume it is -1, then the probability that it is in the same cluster with information-rich user k with $R_{km} = 1$ is

$$p_{-1} = \frac{1}{\tilde{U}_{-1,m} + 0.5\tilde{U}_{0,m}}.$$

The probability that it is in the same cluster with information-rich user k with $R_{km} = 0$ is

$$p_{-10} = \frac{1}{2\tilde{U}_{-1,m} + \tilde{U}_{0,m}}.$$

Therefore, the expected entropy after knowing item m is

$$\begin{aligned}
& -\tilde{U}_{1,m} 0.5 p_1 \log(0.5 p_1) - \tilde{U}_{-1,m} 0.5 p_{-1} \log(0.5 p_{-1}) \\
& -\tilde{U}_{0,m} \left((0.5 p_{10} + 0.5 p_{-10}) \log(0.5 p_{10} + 0.5 p_{-10}) \right) \\
= \; & \frac{\tilde{U}_{1,m}}{2\tilde{U}_{1,m} + \tilde{U}_{0,m}} \log(2\tilde{U}_{1,m} + \tilde{U}_{0,m}) + \frac{\tilde{U}_{-1,m}}{2\tilde{U}_{-1,m} + \tilde{U}_{0,m}} \log(2\tilde{U}_{-1,m} + \tilde{U}_{0,m}) \\
& + \frac{\tilde{U}_{0,m} |\mathcal{U}_t|}{(2\tilde{U}_{1,m} + \tilde{U}_{0,m})(2\tilde{U}_{-1,m} + \tilde{U}_{0,m})} \log \frac{(2\tilde{U}_{1,m} + \tilde{U}_{0,m})(2\tilde{U}_{-1,m} + \tilde{U}_{0,m})}{|\mathcal{U}_t|}.
\end{aligned}$$

Figure 3.4 shows the average number of required samples under the proposed algorithm and the random sampling algorithm when the number of clusters varied. From the figure, we can see that the proposed algorithm outperform the random sampling. Each cluster has 20 users in this experiment.

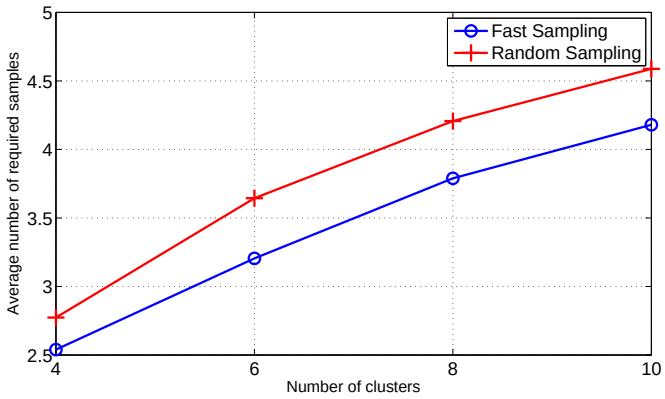

Figure 3.4: The Number of Required Samples under the Proposed Algorithm and the Random Sampling

For the case with flipping, we apply a similar algorithm for selecting the item for sampling. However, since there are errors in the reporting ratings, to decide whether a user is in a specific cluster (step (ii)), we can either apply the hypothesis testing results in the previous section or use the similarity measure in the next section. Step (i) of the algorithm remains to be the same.

3.3 Adaptive Sampling and Learning

At the clustering step, a variation of the user clustering for recommendation (UCR) proposed in Zhu *et al.* (2014) is used in this thesis. In Zhu *et al.* (2014), the authors defined the following concepts:

- Co-rating of users u and v : the number of items rated by both users.

$$\varphi_{u,v} = \sum_{m=1}^{M} \mathbf{1}_{r_{vm} \neq \star, r_{um} \neq \star}.$$

- Similarity of users u and v : the the number of items two users rate the same

minus the number of items they rate differently.

$$\begin{aligned}\sigma_{u,v} &= \sum_{m=1}^{M} \mathbf{1}_{r_{um}=r_{vm}\neq\star} - \sum_{m=1}^{M} \mathbf{1}_{r_{um}\neq r_{vm}, r_{vm}\neq\star, r_{um}\neq\star} \\ &= 2\sum_{m=1}^{M} \mathbf{1}_{r_{um}=r_{vm}\neq\star} - \varphi_{u,v}.\end{aligned}$$

- Normalized similarity:

$$\tilde{\sigma}_{u,v} = \frac{\sigma_{u,v}}{\varphi_{u,v}} = \frac{2\sum_{m=1}^{M} \mathbf{1}_{r_{um}=r_{vm}\neq\star}}{\varphi_{u,v}} - 1.$$

A summary of notation is presented in Table 3.1.

U	the number of users
M	the number of items
K	the number of clusters
$\mathbf{B}$	the true rating matrix
$\mathbf{R}$	the observed rating matrix
$\sigma_{u,v}$	the similarity between user u and user v
$\varphi_{u,v}$	the number of items co-rated by users u and v
$\tilde{\sigma}_{u,v}$	the normalized similarity between user u and user v

Table 3.1: Table of Notation

The following MUCR is a modified version of UCR proposed in Zhu *et al.* (2014), which will be used in the thesis for clustering.

Modified User Clustering for Recommendation (MUCR)

(i) For user u, the algorithm selects a user v who has the highest similarity to user u, i.e.,

$$v \in \arg\max_{w\neq u} \sigma_{u,w}.$$

(ii) The algorithm then selects $\frac{U}{K} - 2$ users in a descending order according to their normalized similarity to user v. Define $\mathcal{F}_u$ to be the set of the selected $\frac{U}{K} - 2$ users, user v and user u.

(iii) For each item m, the score of the item, denoted by s_{wm} for $w \in \mathcal{F}_u$, is determined by the sum of users' ratings in $\mathcal{F}_u$, i.e.,

$$s_{wm} = \sum_{v \in \mathcal{F}_u} r_{vm}.$$

Furthermore, let $R_{wm} = \text{sign}(s_{wm})$.

We next present the adaptive sampling and learning algorithm. We first introduce a popular measure in matrix completion with clustering structure is to user the cosine-based similarity. The similarity for user m and user n under the cosine-based measure is defined below

$$\text{similarity}(R(m), R(n)) = \frac{\sum_{i:R(m,i)\neq\star, R(n,i)\neq\star} R(m,i)R(n,i)}{\sqrt{\sum_{i:R(m,i)\neq\star} R^2(m,i)}\sqrt{\sum_{i:R(n,i)\neq\star} R^2(n,i)}},$$

where $R(m,i)$ is the rating user m gives to item i.

Adaptive Sampling and Learning

(1) Identify one information-rich users for each cluster. First define $\mathcal{RU} = \emptyset$. For $k = 1, \cdots, K$, repeat the following:

1.i Random select a user u and sample γM ratings from the user.

1.ii For there exists a user $w \in \mathcal{RU}$ such that $\text{similarity}(R(m), R(n)) > \alpha$, repeat step (1.i); otherwise, $\mathcal{RU} = \mathcal{RU} \cup \{u\}$.

(2) For each user u, continue to reveal its rating until

$$\max_k \Pr\left(R(u)|u \text{ is from cluster } k\right) > \beta.$$

(3) Apply MUCR to recover the matrix.

Chapter 4

PERFORMANCE EVALUATION

The performance of the proposed algorithm has been evaluated using synthetic data. Specifically, we randomly generated a true rating matrix $\mathbf{B}$. The adaptive sampling and learning algorithm is then applied. When a rating is sampled, it will be flipped according to a given flipping probability. In the synthetic data, it is assumed that b_{um} takes binary values $\{-1, +1\}$, where $b_{um} = +1$ means user u likes item m and $b_{um} = -1$ means user u does not like item m. It is further assumed that r_{um} takes values from $\{-1, 0, +1\}$ where 0 means the rating of user u to item m is unknown.

We considered the case with 100 users and 100 items, i.e., $\mathbf{B}$ is a 100×100 matrix. We assume each cluster has 20 users. The adaptive sampling and learning algorithm is compared with a random sampling algorithm with the same number sampled ratings. The error rate (the fraction of ratings that are different from the true ratings) is used as the performance metric.

4.1 Case 1: Noiseless Reporting

In the first set of simulations, it is assumed that the flipping probability is zero. When identifying information rich users, we reveal $\gamma = 0.8$ fraction of ratings. The results are shown in Figure 4.1, in which the red 'o'-line is under the joint sampling and learning (named Joint) and the blue '+' line is under the random sampling algorithm (named Random). It can be seen that Joint performs much better than Random. Figure 4.2 shows the sampling rate versus the cluster size, where the sampling rate is the fraction of ratings that were revealed.

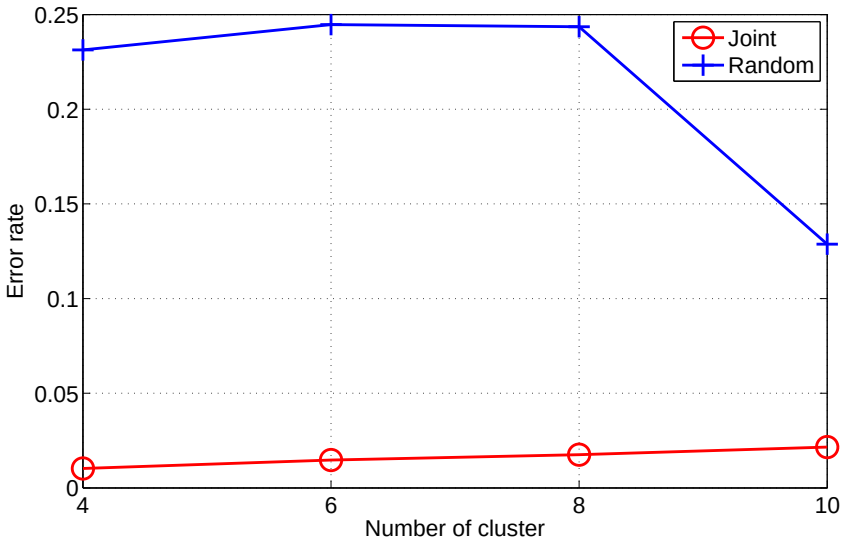

Figure 4.1: The Performance Comparison between Joint and Random.

4.2 Case 2: Noisy Reporting

In this case, the flipping probability is chosen to be 0.1. The results are shown in Figure 4.3. Again we can see that Join performs better than Random. It can be noted that the error rates are smaller than those without flipping. One possible reason is because the sampling rates are much here in this scenario (see Figure 4.4). So the number of observed ratings are much larger while the observations are noisy.

As a summary, the joint sampling and learning algorithm performs better than the random sampling, which significantly reduces the error rates when the same number of ratings were given.

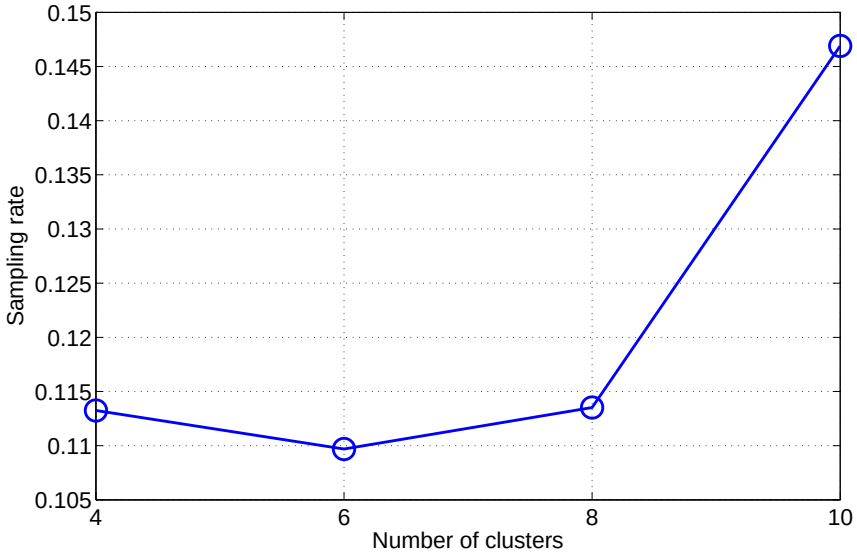

Figure 4.2: The Sampling Rate versus the Cluster Size.

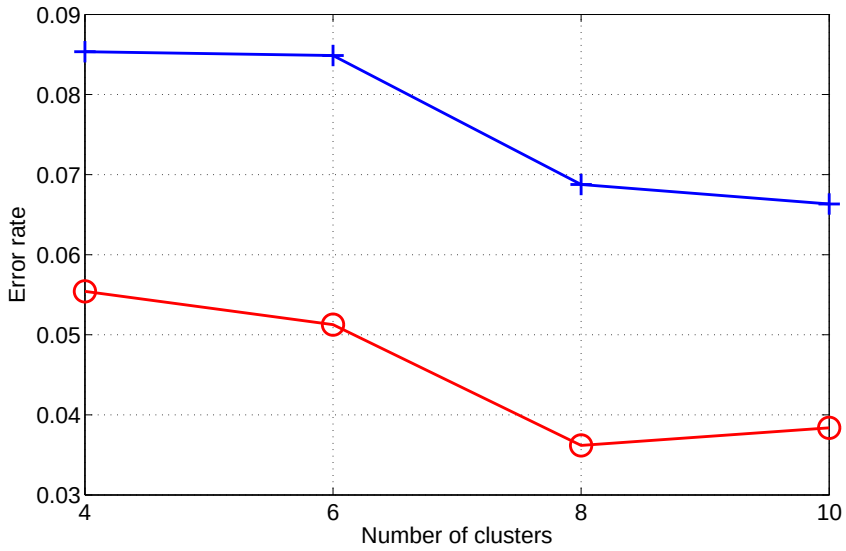

Figure 4.3: The Performance Comparison between Joint and Random.

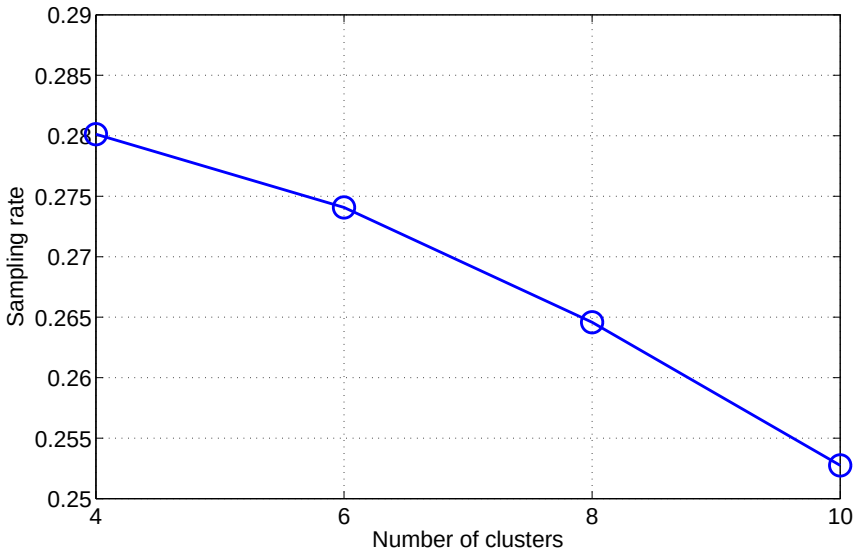

Figure 4.4: The Sampling Rate versus the Cluster Size.

Chapter 5

CONCLUSIONS AND FUTURE WORK

This thesis investigated adaptive sampling and learning in recommendation systems. The proposed algorithm adaptively selects samples to maximize the recover accuracy. One limitation of the proposed algorithm is that it fixes the number of users and the number items. It indeed would be interesting to investigate the case where the number of items changes (i.e., new items come to the market) and the number of users also change (i.e., new customers join the system).

REFERENCES

Barman, K. and O. Dabeer, "Analysis of a collaborative filter based on popularity amongst neighbors", IEEE Transactions on Information Theory (2012).

Candès, E. J. and B. Recht, "Exact matrix completion via convex optimization", Foundations of Computational mathematics **9**, 6, 717–772 (2009).

Candès, E. J. and T. Tao, "The power of convex relaxation: Near-optimal matrix completion", IEEE Transactions on Information Theory **56**, 5, 2053–2080 (2010).

Cover, T. M. and J. A. Thomas, *Elements of Information Theory* (Wiley, New York, NY, 1991).

Gross, D., "Recovering low-rank matrices from few coefficients in any basis", IEEE Transactions on Information Theory **57**, 3, 1548–1566 (2011).

Jain, P., P. Netrapalli and S. Sanghavi, "Low-rank matrix completion using alternating minimization", in "Proceedings of the 45th annual ACM symposium on Symposium on theory of computing", pp. 665–674 (2013).

Keshavan, R. H., A. Montanari and S. Oh, "Matrix completion from noisy entries", The Journal of Machine Learning Research **99**, 2057–2078 (2010).

Recht, B., "A simpler approach to matrix completion", The Journal of Machine Learning Research pp. 3413–3430 (2011).

Tomozei, D.-C. and L. Massoulié, "Distributed user profiling via spectral methods", Stochastic Systems **4**, 1–43, DOI: 10.1214/11-SSY036 (2014).

Xu, J., R. Wu, K. Zhu, B. Hajek, R. Srikant and L. Ying, "Exact block-constant rating matrix recovery from a few noisy observations", arXiv preprint arXiv:1310.0512 (2013).

Xu, J., R. Wu, K. Zhu, B. Hajek, R. Srikant and L. Ying, "Jointly clustering rows and columns of binary matrices: algorithms and trade-offs", in "Proc. Ann. ACM SIGMETRICS Conf.", pp. 29–41 (Austin, Texas, 2014).

Zhu, K., R. Wu, L. Ying and R. Srikant, "Collaborative filtering with information-rich and information-sparse entities", Machine Learning **97**, 1-2, 177–203 (2014).