Recommending under Threat of Manipulation

7

April 24, 2013

Abstract

Collaborative filtering accomplishes automated personalized content recommendation but by relying on anonymous user feedback is vulnerable to manipulation. Taking measures to improve robustness to outliers seeks to increase resistance against such manipulation. However, this detrimentally impacts recommendation quality even in the absence of such attacks. By taking a competitive game perspective on collaborative filtering a strategy that strikes the desired trade-off between recommendation quality and resistance to manipulation can be found. Simulation on a 1M rating MovieLens dataset of varied size content push attacks is performed on a weighted alternating least squares recommender system running different levels of outlier trimming. These simulation results are used to assemble games where the utilities depend in addition on the weight the recommender puts on preventing manipulation compared to maintaining recommendation quality and how costly increasing attack size is to the attacker. The maxmin strategy for the recommender and present Nash Equilibria in these games are analysed. Results suggest that alternating recommendation between trimmed and non-trimmed methods serves to decrease the effectiveness of manipulation attacks while minimizing the impact on recommendation quality.

1 Introduction

Recommender systems have become an important part of modern e-commerce systems. With continually increasing available content, whether in the form of products or information, these systems perform the important role of present content personalized to each user's preferences. Recommender systems that operate on the ideas of collaborative filtering are of importance due to being content independent, needing only user feedback on the content to make recommendations.

Unfortunately relying on anonymous user feedback to make recommendations is open to attempts by users to manipulate what content the system recommends to other users. As noted by [4] such attacks have been found to occur on commercial systems. This prompts the need of systems to be sufficiently resilient to such attacks. Considered here is the handling of malicious ratings by way of outlier detection integrated into the learning performed by a model based recommender system.

While ignoring ratings that appear to be outliers may help reduce vulnerability to manipulation it can also have the undesirable effect of reducing recommendation quality. This results from possibly mistakenly treating real ratings as being manipulative. As such, it is important to take into account how much baseline recommendation quality is given up to achieve resistance to manipulation. When deciding on this trade-off it is also worthwhile considering the attacker's perspective. If it is costly for malicious users to be inserted into the system less aggressive attack counter measures by the recommender may suffice.

Attack simulation on a 1M rating MovieLens dataset using a recommender system running outlier trimming is performed to identify the plausible impact on the recommendations made. By varying attack size and amount of outlier trimming performed an outcome table with the results for each action combination is constructed. By varying how important it is to system to avoid being manipulated and how costly it is to the attacker to launch larger attacks the outcomes are used to generate a game corresponding to each of these preference configurations. These games are then analysed for the recommender system maxmin strategy and searched for Nash Equilibria. The strategies identified give a means of choosing how the recommender system should be configured in such situations.

The rest of this paper is organized as follows: background on robust recommender systems and the game theory that is applied is presented in Section 2. The recommender system used is described in Section 3. In Section 4 the game formulation is described. The experiments preformed and a discussion of their results is in Section 5. Finally, the paper is concluded with a summary of the findings as well as remaining open considerations.

2 Background

As recommender systems see increasing use individuals who seek to promote particular content see increasing possible benefit of successfully manipulating such systems. An attacks is performed by inserting fake users into the system and having these fake users give feedback on content in such a way that the system more frequently recommends the target content than it would have otherwise. There are a variety of different designs for the fake user ratings discussed in the literature [7] [4]. Focused on in this paper is the Average attack where each attack user rates the target content the highest possible and in addition provides filler ratings. These filler ratings are on some number of randomly chosen content and with rating values chosen from normal distributions that have mean and standard deviation according to the mean and standard deviation of ratings of all other users on the content. The Average attack has been shown to be significantly more effective at shifting system recommendations than using simply random rating values for filler ratings [5].

With in the category of recommender systems based on collaborative filtering two prominent approaches are those based on nearest neighbour and matrix factorization [8]. Nearest neighbour has been shown consistently be more vulnerable to manipulation leading to matrix factorization being chosen as the base system on which changes are made to improve robustness [6][2]. These changes involve reducing the weight given to outliers according to M-estimators [6] or ignoring (trimming) those ratings that have greatest residual error in the model [2]. In this paper a recommender running outlier trimming is considered as this was shown to result in greater improvement in resistance to manipulation.

In the resulting recommendation game of pitting an attacker against the system of interest is according to how important deterring attacks is and how costly are attacks to perform what strategy should the recommender system employ. While an action corresponds to the system choosing a specific configuration a strategy can involve the system randomizing between various configurations. The system's maxmin strategy is the strategy that maximizes expected utility in the case that the attacker does what is worst for the system's objective. This, however, is overly pessimistic as the attacker is expected to incur some cost launching an attack and is only trying to boost a particular item and doesn't care about the system's overall recommendation quality. To consider what strategies should be applied under these different but not completely opposite objectives the Nash Equilibrium solution concept can be applied. The strategies at a Nash Equilibrium have the property that both agents best respond to what the other is playing making it so that neither can shift the outcome in their favour even if they knew what choice the opponent made. Importantly, by Nash's Theorem there is always a Nash Equilibrium in a finite game however there may be many of them which introduces the difficulty of which to pick.

3 Trimmed Weighted ALS Recommender

The following describes the recommender system that is evaluated. Ratings are standardized to have zero mean and standard deviation one. Trimmed Weighted Alternating Least Squares (T-ALS), shown in Algorithm 1, is applied to factorize the standardized rating matrix $R^{m \times n}$. The factors, $P^{f \times m}$, $Q^{f \times n}$, once computed are used predict missing ratings via $P^T Q$ and undoing the standardization. Residuals are computed by $e_{u,i} = |R_{u,i} - P_u^T Q_i|$. The parameters used are, number of features, $f = 15$, regularization, $\lambda = 10$, and iterations, $itrs = 20$. The remaining parameter γ, which will be referred to as the h/n fraction, adjusts the fraction of all ratings relevant to a particular user or content, n, to those that are used, h. This impacts how the system handles ratings that appear to be outliers. This algorithm is based on Least Trimmed Squares Matrix Factorization presented by [2]. The differences being instead of performing stochastic gradient descent alternating least squares is used and rating trimming is performed on each content and user entry individually instead of trimming being based on the largest residuals over all ratings.

Algorithm 1 T-ALS

```
Randomly Initallize $P, Q$.
for itr = 1:itrs do
    for u = 1:m do
        Update $W$ s.t. if $R_{u,i}$ is rated and its residual is in the bottom $\gamma$ fraction of all residuals on ratings in $R_i$ $W_{i,i} = 1$, otherwise $W_{i,i} = 0$.
        $P_u \leftarrow (QWQ^T + \lambda I)^{-1} QWR_u$
    for i = 1:n do
        Update $W$ s.t. if $R_{u,i}$ is rated and its residual is in the bottom $\gamma$ fraction of all residuals on ratings in $R_u$ $W_{u,u} = 1$, otherwise $W_{u,u} = 0$.
        $Q_i \leftarrow (PWP^T + \lambda I)^{-1} PWR_i$
```

4 Gaming Recommendations

By formulating the result of an attack on a recommender system as a game it is then possible to then reason about what strategies should be employed. The system's action space corresponds to choosing the h/n fraction. The attacker's action space is what size of attack to launch. To simplify the game the available actions are restricted to specific values for h/n fraction $\in \{1, 0.8, 0.7, 0.6, 0.5\}$ and the attack size $\in \{1\%, 2\%, 4\%, 6\%, 10\%\}$, where percent is in terms of all users in the system. This restriction to a smaller action space is expected to still be representative of the full game due to the outcomes are seen to vary gradually with changing of either action value. The outcome of an action profile is specified by the prediction shift on the targeted content and the Mean Absolute Error (MAE) of the recommender on a test set. Prediction shift evaluates how much the recommender's predictions of ratings on the target item are changed by the attack with respect to what they were with no attack. Prediction shift for a push attack is defined in Equation 1 where A_i is the set of users that have not rated content i, $\hat{r}_{i,u}$ is the original prediction and $\hat{r}^A_{i,u}$ is it after the attack. The calculation for MAE is also shown in Equation 1 and is a standard metric for evaluating recommendation quality. T is the set of all pairs (u, i) where user u has rated content i in the dataset and $r_{u,i}$ is the true rating.

$$\text{Prediction Shift} = \frac{\sum_{u \in A_i} (\hat{r}^A_{i,u} - \hat{r}_{i,u})}{|A_i|} \qquad \text{MAE} = \frac{\sum_{(u,i) \in T} |\hat{r}^A_{i,u} - r_{u,i}|}{|T|} \tag{1}$$

The utilities achieved by the recommender system and the attacker when (h/n fraction, attack size) is played and outcome (Shift, MAE) results also depends on how important preventing attacks is to the system with respect to MAE as well as how costly increasing attack size is to the attacker. α and β control these two trade-offs respectively giving the following utility functions:

- Recommender System Utility = α (- Shift) + (1 - α) (- MAE)
- Attacker Utility = β (Shift) + (1 - β) (- attack size)

By running simulations the action profile outcomes can be approximated giving a set of games, one for each (α, β) preference configuration. For each of these games the recommender's maxmin strategy can be found. Playing a maxmin strategy may be sensible for a recommender to do considering it doesn't know whom the attacker will be and so doesn't know if they may behave irrationally. Also while the game is not zero-sum the utility received from prediction shift is opposite for the recommender system compared to the attacker. In addition, the maxmin strategy is comparatively easy to compute in a two-player game. It can be identified by creating a new game where the attacker's utility is the exact opposite of the recommender's and solving for the Nash Equilibrium in this resulting game. Since this is a two-player zero-sum game a Nash Equilibrium can be found efficiently by a linear programming formulation and by the Maximin Theorem all Nash Equilibria of the game have the same utility. While the maxmin strategy is safe there is possibly room to do better especially when the cost of launching large attacks is significant. By seeking a Nash Equilibrium in the original game a strategy that takes into consideration the attacker's perspective may be found. Specifically, assuming the attacker is rational they wont employ a strategy if they achieve better utility using a different strategy. However, finding a Nash Equilibrium is much harder to do computationally, residing in the complexity class PPAD-complete [3], plus there may be multiple Nash Equilibria not all of which give the same utility.

5 Experiments and Discussion

To put these ideas to work attack simulation is performed against the recommender system described in Section 3 running on the Movie Lens[1] dataset. This dataset contains approximately 1M ratings on a 1-5 scale provided by 6040 users on 3952 movies. This is the same data set as that used by [2] and [6] in analysing their proposed robust recommender systems. 10% of the ratings in the dataset are randomly sampled and placed in a test set to evaluated the recommender's MAE on while the other 90% form the training set.

5.1 Attack Simulation

For 4 different randomly selected target movies the outcome of each (attacker, recommender) action profile is determined by simulation. Attack users are injected into the training set each of which applies an Average attack using a rating fill size of 3% of movies in the system. This fill size is consistent with that used by [2] and [6] and is close to the average number of movies rated by a user in the dataset. Figure 1 reports the average prediction shift that was caused on the attacked movie for each of these action profiles. The MAE of the recommender was not found to be noticeable impacted by attack size so in Figure 2 it is graphed only with respect to the h/n fraction. All of the mean results are also reported in Table 1.

[1] Available at: http://www.grouplens.org/node/73

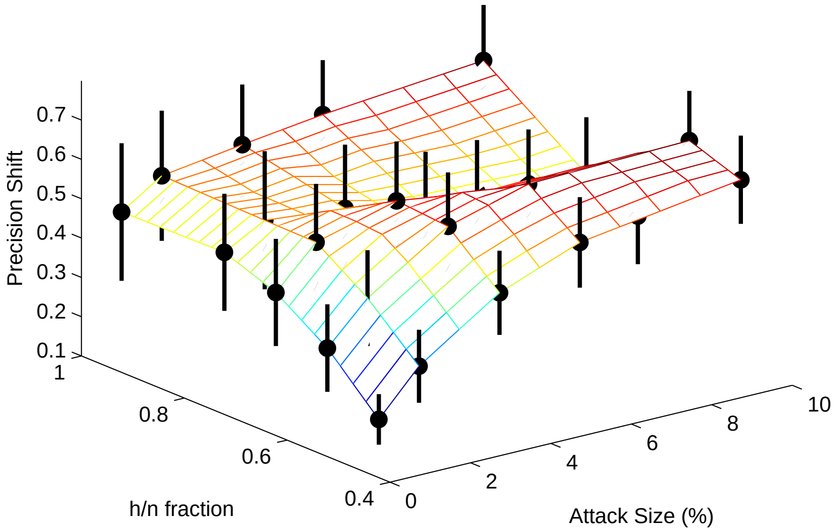

Figure 1: Prediction Shift of targeted movie

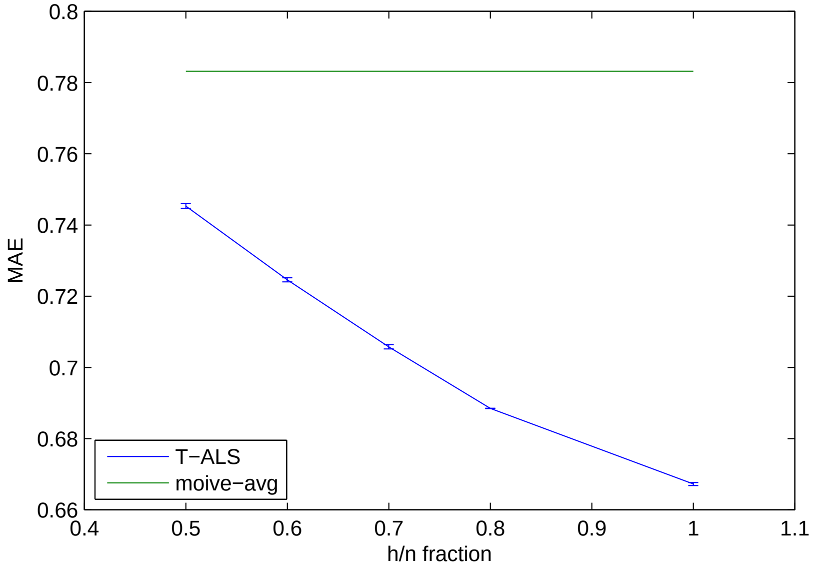

Figure 2: Recommender System MAE on Test set

The prediction shift results are rather surprising. It was expected that increasing outlier removal would reduce precision shift for all attack sizes. Instead, it is observed that for moderate amounts of outlier removal precision shift is increased for small attack sizes but decreased for large attack sizes. So much so that smaller attacks appear to become more effective. For the highest levels of outlier removal a drop-off in the effectiveness is seen in small attack sizes but large attacks become much more effective. These complex responses could be the result of whether or not the system notices that there are malicious users and if there are enough of them if it is tricked into thinking they are the real users.

The system MAE not being effected by the Average attacks is consistent with the findings of [6]. MAE becoming worse as the fraction of the ratings that are used is decreased is also expected. To give perspective to the significance of the worsening of MAE the MAE achieved by simply predicting the movie average is plotted in addition to the MAE of T-ALS in Figure 2. It is unlikely that MAE degradation to a value much above 0.7 is acceptable.

		Recommeder System				
		1	0.8	0.7	0.6	0.5
Attacker	1	0.44, 0.67	0.45, 0.69	0.40, 0.71	0.31, 0.73	0.18, 0.75
	2	0.51, 0.67	0.50, 0.69	0.50, 0.71	0.41, 0.73	0.29, 0.75
	4	0.54, 0.67	0.48, 0.69	0.56, 0.71	0.55, 0.73	0.43, 0.75
	6	0.57, 0.67	0.44, 0.69	0.53, 0.71	0.60, 0.72	0.51, 0.75
	10	0.61, 0.67	0.42, 0.69	0.37, 0.71	0.62, 0.72	0.57, 0.74

Table 1: (Shift,MAE) Outcomes: Attacker actions are in terms of attack size percent of total number of users and Recommend System actions are h/n fraction values

5.2 Game Strategies

Nash Equilibria for the games considered are searched for using a Matlab implementation [2] of the Nash Equilibrium finding algorithm presented in [1]. As discussed in Section 4 the utilities of both the recommender system and the attacker are both composed of two terms that are traded between by preference weights. For analysis of the recommender's maxmin strategy only the recommender's preference weight is relevant since the attacker's utility is not needed. The maxmin games are zero-sum with the Attacker's utility set to the negative of that of the recommender. The recommender maxmin strategies for the games that result from the setting of the preference weight, α, in range [0,1] discretized at 51 uniformly distributed values are shown in Figure 3. The actions A1-A5 correspond to the 5 setting for the h/n fraction from largest to smallest. Colour corresponds to with what probability each action is played.

[2] http://www.mathworks.com/matlabcentral/fileexchange/27837-n-person-game/content/npg/npg.m

The strategies found are interesting but not particularly surprising. No trimming (A1) is performed if only MAE performance is important, and mixing between A1 and A2 is beneficial when there is some value to preventing prediction shift. When the system just cares about preventing manipulation the second and last setting are mixed between to handle one being effective at preventing small attacks and the other at preventing large attacks.

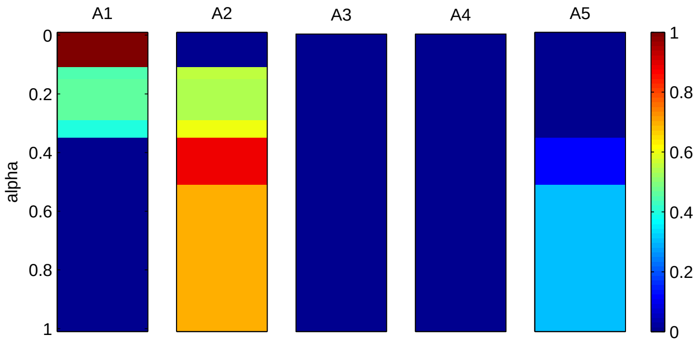

Figure 3: Recommender System maxmin strategies

To consider what strategies are played at the Nash Equilibria of the original game the Attacker's preference weight will now also be of importance. Both the recommender's preference weight, α, and the attacker's preference weight, β, are each in the range [0,1] and discretized at 51 uniformly distributed values. For each (α, β) a game is generated and on which the Nash Equilibrium finder is run. The strategies found for both the recommender and the attacker are displayed in Figure 4. Note that each point corresponds to one game of which there are 2601. The recommender is referred to as p1 the attacker as p2. The actions A1-A5 for the recommender are in order of decreasing h/n fraction and the actions A1-A2 for the attacker are in order of increasing attack size. As in figure 3 colour corresponsive to with what probability each action is played.

The noise in the otherwise visible strategy patterns is the result of a combination of the Nash finder sometimes failing to find a Nash Equilibrium resulting in a non-equilibrium strategy being reported and the presence of multiple equilibria in the game. Ignoring the noise and focusing on the patterns present while there are similarities between the recommender equilibrium strategies to the maxmin strategies there are also some notable differences. If the attacker has a high cost (low β) associated with increasing attack size there is no benefit for the recommender to use A2 (which is only effective at stopping large attacks) and so unlike in the maxmin strategy there is transition point between always playing A1 to always player A5 depending on how important preventing prediction shift is to the recommender. For when the recommender cares mostly about MAE (low α) only for when the attacker has a low cost associated with large attacks (high β) does it become worthwhile mixing A2 in with A1.

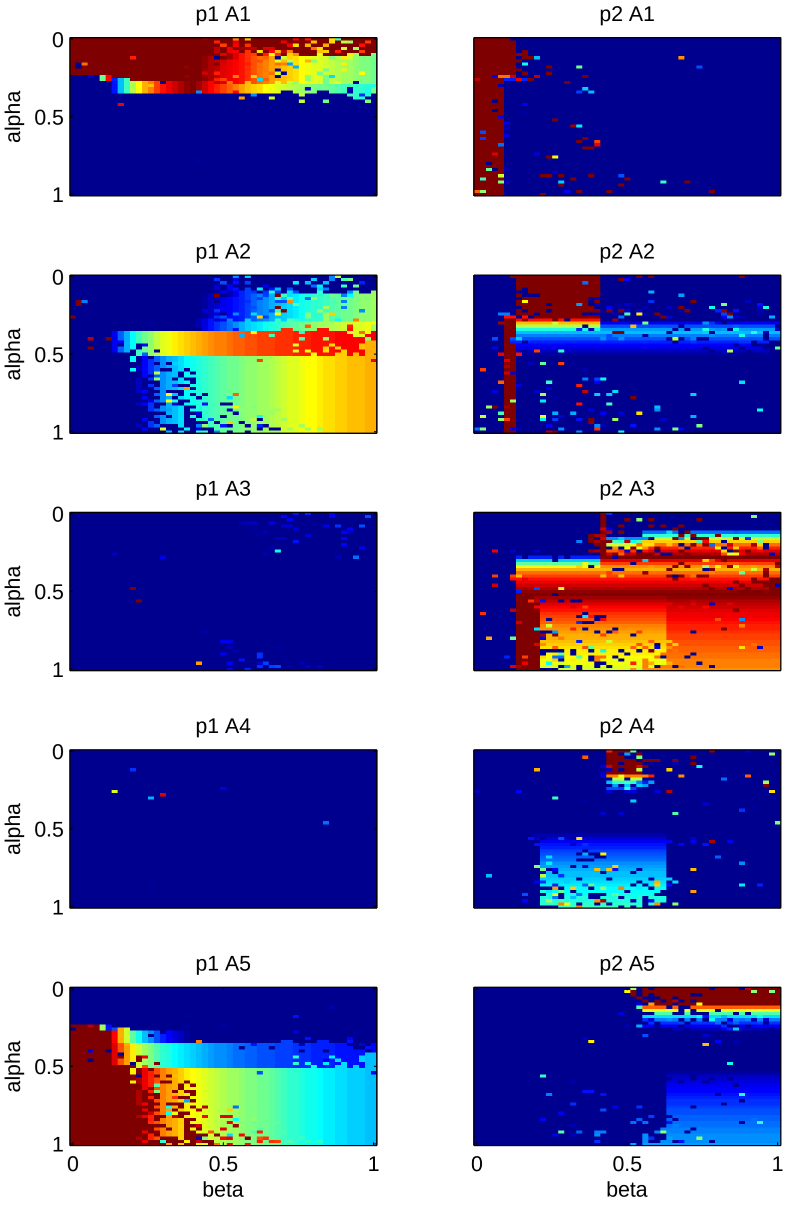

Figure 4: Strategies found by Nash Equilibrium finder

6 Conclusions and Future Directions

The use of attack simulation provides a means of approximating how a recommender system will be affected by attempts to manipulate it. By simulating alternate configuration settings for a recommender system it is possible to identify the trade-off between resistance to manipulation and avoiding degrading recommendation quality. Once an importance is chosen for these two objectives the trade-off can be implemented either using the maxmin strategy, in cases where little is known about the attacker and the worst case is to be avoided, or the Nash Equilibrium strategy, which can take advantage of knowledge about the attacker's costs and goals. For a sensible trade-off that puts most weight on MAE and some on preventing manipulations cycling between recommending based on a system with a small trim and no trim maximizes recommender utility. How frequently the trimmed version is to be applied depends on how easy it is for the attacker to insert fake users into the system. An important possible extension is to incorporate approximate knowledge of the game into the strategy reasoning. There are many sources of uncertainty that may be considered a few are: simulation results have significant variability, uncertainty in how costly attacks are and how important shifting recommendations is to attackers, and asymmetry in uncertainty about the system with the attacker have a much cruder or even incorrect picture of how the recommender system performs under attack.

References

[1] B. Chatterjee. An optimization formulation to compute nash equilibrium in finite games. In *Methods and Models in Computer Science, 2009. ICM2CS 2009. Proceeding of International Conference on*, pages 1–5, 2009.

[2] Z. Cheng and N. Hurley. Robust collaborative recommendation by least trimmed squares matrix factorization. In *Proceedings of the 2010 22nd IEEE International Conference on Tools with Artificial Intelligence - Volume 02*, ICTAI '10, pages 105–112, Washington, DC, USA, 2010. IEEE Computer Society.

[3] C. Daskalakis, P. W. Goldberg, and C. H. Papadimitriou. The complexity of computing a nash equilibrium. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, STOC '06, pages 71–78, New York, NY, USA, 2006. ACM.

[4] I. Gunes, C. Kaleli, A. Bilge, and H. Polat. Shilling attacks against recommender systems: a comprehensive survey. *Artificial Intelligence Review*, pages 1–33, Nov. 2012.

[5] S. K. Lam and J. Riedl. Shilling recommender systems for fun and profit. In *Proceedings of the 13th international conference on World Wide Web*, WWW '04, pages 393–402, New York, NY, USA, 2004. ACM.

[6] B. Mehta, T. Hofmann, and W. Nejdl. Robust collaborative filtering. In *Proceedings of the 2007 ACM conference on Recommender systems*, RecSys '07, pages 49–56, New York, NY, USA, 2007. ACM.

[7] B. Mobasher, R. Burke, R. Bhaumik, and C. Williams. Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness. *ACM Trans. Internet Technol.*, 7(4), Oct. 2007.

[8] F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors. *Recommender Systems Handbook*. Springer, 2011.