International journal of basic and applied research
www.pragatipublication.com
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

A Metric Learning-Based Approach to Identify Professional Malicious Users in Recommender Systems

Mr. Y. Subrahmanyam
1 Assistant Professor, Department of CSE, Malla Reddy College of Engineering for Women., Maisammaguda., Medchal., TS, India

Abstract:
A number of e-commerce platforms have made extensive use of recommender systems in an effort to enhance the user experience. Users' faith in such systems is waning, however, as security risks like phoney reviews and ratings start to sprout up. An example of an assault that targets recommender systems is the data poisoning attack. The attacker pretends to be a real user on the system and tries to change the suggested goods by adding bogus ratings. Many countermeasures have been suggested, however the most of them need awareness of both true and fraudulent ratings. Therefore, we provide recommender systems that can withstand data poisoning without human intervention.

1 INTRODUCTION

When it comes to modern e-commerce platforms, recommender systems are crucial to the success of the user experience. The recommender system allows users to quickly discover products that may be of their preference. On the other hand, user ratings are used to choose suggested goods. Security issues, such phoney reviews and ratings, are a cause for worry in a setting where everyone may post harmful info. Data poisoning attacks on recommender systems are the main topic of this article. The data poisoning assault alters suggested products by flooding the system with false ratings. Defense strategies are still in their early stages, despite several suggested assaults. The majority of current defensive strategies need background information on actual and/or fabricated ratings. As a result, we provide recommender systems that can withstand data poisoning without human intervention.

1.1 Prior Experience

A data poisoning assault on a recommender system primarily aims to increase or decrease the popularity of certain goods. Consequently, three assault models were presented—random, average, and bandwagon—and the topic of how to create ratings of hostile individuals with minimal information was addressed in (Burke et al., 2005).
An further attack paradigm, obfuscated assaults, was offered by (Williams et al., 2006), which further complicates the task of detecting malevolent individuals.
There are two main types of defense strategies: supervised and non-supervised. Detectors in a supervised environment learn to distinguish between genuine and malicious users based on a predefined set of ratings. Wu et al., 2011; Burke et al., 2006; Mobasher et al., 2007; and Chirita et al., 2005 are among the publications that have contributed to the development of this model's many aspects. Despite the excellent precision achieved by these detectors, the circumstances are impractical. Obtaining rated data with a label, for example, is challenging in real-world use situations. In contrast, an unlabeled rating data set is used to create a detector via non-supervised approaches. In this research, we refer to the PCA-based method—which was originally developed by Mehta (2007)—as the first detector of its kind. The obfuscated attack is one example of an advanced assault that may target non-supervised algorithms (Li et al., 2016).

Part 1.2: What We Did

International journal of basic and applied research
www.pragatipublication.com
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

Our proposed knowledgeless protection approach outperforms the current state of the art. We have summarized our key contribution as follows:
_ A trim-learning-based matrix factorization method is designed. When trying to forecast ratings that have not been seen, matrix factorization is the way to go.
One approach to learning that is resistant to data poisoning in linear regression is trim learning, which was suggested by Jagielski et al. (2018). Our learning system effectively safeguards against false ratings by using the principle of trim learning in conjunction with matrix factorization. Using a real-world dataset, we show how our suggested technique works. Our suggested approach is compared to another existing detection method on the assumption of four forms of data poisoning assaults. Based on the findings, our suggested strategy significantly increases resistance to data poisoning.

2 PRELIMINARIES

2.1 Recommender Systems
We zero in on matrix factorization-based collaborative filtering, which is extensively used in practical recommender systems (Koren et al., 2009). The number of things in the system is n, and the number of users is m.

Consider a rating matrix with many missing components, M 2 Rm_n. W stands for the set of observable indices. A user's rating for the jth item is indicated by Mi; j. Two latent factors, U 2 Rm_k and V 2 Rk_n, are produced using matrix factorization. Here, k _ min(m;n) is a positive integer. The user matrix U and the item matrix V are the terms used to describe them. By resolving the following optimization issue, these are produced:

$$\min_{\mathbf{U},\mathbf{V}} \sum_{(i,j)\in\Omega} (\mathbf{M}_{i,j} - \mathbf{u}_i\mathbf{v}_j)^2 + \lambda(||\mathbf{U}||^2 + ||\mathbf{V}||^2), \quad (1)$$

where l (_ 0) is a regularization parameter and ui and vj are the ith row of U and the jth column of V, respectively. The Frobenius norm is denoted as jj _ jj.

The recommender system uses the formula bM = UV to estimate the values of the items in M that are absent. The anticipated ratings are used to choose the recommended goods.

As optimization techniques, alternating minimization and stochastic gradient descent (SGD) are well-known for solving Eq. (1). For sparse matrices like rating matrix M, the SGD approach is quicker and simpler to implement, thus we build our learning algorithm around it. Every rating in the training set is iterated over by the algorithm. If we use a rating as an example, we can say that the prediction error is

$$e_{i,j} := \mathbf{M}_{i,j} - \mathbf{u}_i\mathbf{v}_j. \quad (2)$$

$$\mathbf{u}_i \leftarrow \mathbf{u}_i + \eta \cdot (e_{i,j} \cdot \mathbf{u}_i - \lambda \cdot \mathbf{v}_j) \quad (3)$$

$$\mathbf{v}_j \leftarrow \mathbf{v}_j + \eta \cdot (e_{i,j} \cdot \mathbf{v}_i - \lambda \cdot \mathbf{u}_i), \quad (4)$$

2.2 Data Poisoning Attacks
An attacker may introduce false ratings into a recommender system by creating several user accounts and launching a data poisoning assault. With m0 being the number of malevolent users, let M0 2 Rm0_n be a fabricated rating matrix introduced into the recommender system. The size of an assault is defined as a := m0=m.

International journal of basic and applied research
www.pragatipublication.com
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

Our main emphasis is on a kind of assault known as a push attack, which aims to increase the popularity of certain products. A malevolent user's push attack ratings are composed of three parts:

_ Item Targets: These are the things that the enemy is trying to boost in popularity. The greatest possible rate is given.

The purpose of the filler objects is to make detection more challenging. Measured as a ratio of the number of filler items to n, Filler Size b is an important factor to consider.

Unrated Items: The bulk of a data point is taken up by the remaining unrated items.

Both the 2005 and 2006 studies by Burke et al. provide four methods for assigning rating levels to filler items:

The filler items are ranked at random around the total average value in random attacks.

The filler items are rated at random around the average rating value of each item for typical attacks.

_ Attacks on the Bandwagon: Some filler items get the highest ratings because they are the most popular. All the remaining filler items are given ratings that are randomly distributed around the average rating. The paper's tests allocate half of the filler items to the most popular items.

_ Obfuscated Attacks: Randomly picked from a pool of popular things, all of the filler items are rated close to the average overall value. 2.3 Trim Learning

One learning approach that may withstand data poisoning is the trim learning, which was suggested by Jagielski et al. (2018). Assume D is a training set with N good samples and 0 bad ones. Assume that the model's parameters define a loss function L. q. The following is a formal definition of the optimization issue by Jagielski et al.:

$$\min_{\theta, I} L(\mathcal{D}^I, \theta)$$
$$\text{s.t. } I \subset [N+N'], |I| = N, \qquad (5)$$

Jagielski et al. provided an optimization algorithm to solve Eq. (5) for linear regression. We apply this concept to matrix factorization, and develop a learning algorithm robust to data poisoning for a recommender system.

3 METHOD

By merging matrix factorization with trim learning, we provide a recommender system learning strategy that is resistant to data poisoning. In this part, we create a strong matrix factorization technique and describe an optimization issue that our approach can solve.

3.1.1 Prompting the Issue

Because the enemy has a hard time mimicking genuine users, their actions are statistically distinct from those of good users. Consequently, there will be more training loss for malevolent users than for lawful ones. Furthermore, training loss is increased for both legitimate and criminal users due to tainted products, including target items and filler things. Consequently, we develop a method to train a model while simultaneously removing harmful users and things from the system.

International journal of basic and applied research
www.pragatipublication.com
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

In order to achieve this goal, we formulate the following optimization problem for our suggested method:

$$\min_{\mathbf{U},\mathbf{V},I_u,I_v} L(\mathbf{M},\mathbf{U},\mathbf{V},I_u,I_v)$$

$$= \sum_{(i,j)\in\Omega^{I_u\times I_v}} (\mathbf{M}_{i,j}-\mathbf{u}_i\mathbf{v}_j)^2+\lambda(||\mathbf{U}||^2+||\mathbf{V}||^2)$$

$$\text{s.t. } I_u\subset[m],\ I_v\subset[n],\ |I_u|=m^*,\ |I_v|=n^*,\quad (6)$$

where subsets of items and users are represented by Iv and Iu, respectively.

m_ is the number of users, while n_ is the number of things that remain untrimmed. In M, WIu_Iv is a collection of elements that correspond to users Iu and items Iv. These elements were solved with V_ (or U_) fixed. Alternative minimization may also be used to solve Iu and Iv. Our suggested method's methodology is summarized in methodology 1.

Algorithm 1: Trim matrix factorization.

Input : $\mathbf{M}, m, n, k, m^*, n^*, \alpha, \beta, \lambda, \eta$
Output: $\mathbf{U}^*, \mathbf{V}^*, I_u, I_v$

Initialize $I_u^{(0)}, I_v^{(0)}$.
$t \leftarrow 0$
repeat
  $\mathbf{U}^{*(t)}, \mathbf{V}^{*(t)} \leftarrow$
    $\arg\min_{\mathbf{U}^*,\mathbf{V}^*} L(\mathbf{M},\mathbf{U}^*,\mathbf{V}^*,I_u^{(t)},I_v^{(t)})$
  $I_u^{c(t)} \leftarrow [m]\setminus I_u^{(t)}, I_v^{c(t)} \leftarrow [n]\setminus I_v^{(t)}$
  $\mathbf{U}^{c(t)} \leftarrow$
    $\arg\min_{\mathbf{U}^c} L(\mathbf{M},\mathbf{U}^c,\mathbf{V}^{*(t)},I_u^{c(t)},I_v^{(t)})$
  $\mathbf{V}^{c(t)} \leftarrow$
    $\arg\min_{\mathbf{V}^c} L(\mathbf{M},\mathbf{U}^{*(t)},\mathbf{V}^c,I_u^{(t)},I_v^{c(t)})$
  $\mathbf{U}^{(t)} \leftarrow \text{concatenate}(\mathbf{U}^{*(t)},\mathbf{U}^{c(t)})$
  $\mathbf{V}^{(t)} \leftarrow \text{concatenate}(\mathbf{V}^{*(t)},\mathbf{V}^{c(t)})$
  $I_u^{(t+1)}, I_v^{(t+1)} \leftarrow$
    $\arg\min_{I_u,I_v} L(\mathbf{M},\mathbf{U}^{(t)},\mathbf{V}^{(t)},I_u,I_v,)$
  $t \leftarrow t+1$
until $I_u^{(t)} = I_u^{(t-1)} \wedge I_v^{(t)} = I_v^{(t-1)}$
return $\mathbf{U}^{*(t)}, \mathbf{V}^{*(t)}, I_u^{(t)}, I_v^{(t)}$

4 EXPERIMENTS

4.1 Setup

For this study, we relied on the MovieLens 1M dataset, which was created by GroupLens Research in 2016. As mentioned in Section 2.2, we executed four distinct assaults.

International journal of basic and applied research
www.pragatipublication.com
ISSN 2249-3352 (P) 2278-0505 (E)
Cosmos Impact Factor-5.86

You may choose an assault size of 0.05, 0.1, 15, or 2 seconds.

At random, one object was chosen to be the target. For each value of b, we randomly chose filler items, and the filler size was set to 0:05, 0:1, 0:15, or 0:2. In order to stabilize the findings, we ran each assault 10 times for each parameter value. Our suggested method's parameters were set to k = 32, m_ = m, n_ = (1 - b)n - 1, l = 0:01, and h = 0:02.

As a measure for our suggested approach, detection rate is defined here. The detection rate may be calculated by dividing the total number of harmful users by the number of trimmed malicious users. The average value of the detection rate was used to assess the efficacy of our suggested technique. To keep things fair, we used the same statistic to assess how well the PCA-based technique (Mehta, 2007) performed.

Our suggested method's detection rates (b = 0:1) are shown in Table 1.

Attack size	0.05	0.1	0.15	0.2
Random	0.887	0.941	0.963	0.974
Average	0.880	0.938	0.960	0.970
Bandwagon	0.793	0.859	0.897	0.918
Obfuscated	0.827	0.921	0.938	0.956

Table 2: Detection rates of PCA based method (b = 0:1).

Attack size	0.05	0.1	0.15	0.2
Random	0.0	0.010	0.029	0.121
Average	0.667	0.754	0.758	0.762
Bandwagon	0.733	0.770	0.802	0.780
Obfuscated	0.0	0.111	0.323	0.475

Table 3: Detection rates of our proposed method (a = 0:1).

Filler size	0.05	0.1	0.15	0.2
Random	0.761	0.941	0.993	1.0
Average	0.757	0.938	0.980	1.0
Bandwagon	0.318	0.859	0.948	0.987
Obfuscated	0.333	0.921	0.970	1.0

4.2 Evaluation Results

Our suggested method's detection rate and the PCA-based method's detection rate at b = 0:1 are shown in Tables 1 and 2, respectively. The outcomes for a = 0:1 are shown in Tables 3 and 4. It is shown that, across all parameters, our suggested strategy outperforms the PCA-based method in identification rate. That is why our suggested approach is so much more resistant to data poisoning.

5 CONCLUSIONS

Recommender systems that are resistant to data poisoning were suggested in this research. Combining matrix factorization with trim learning is our suggested approach. While removing harmful individuals and infected things, the system learns a recommendation model.

Experimental findings demonstrated a significant improvement in resilience to data poisoning when using our suggested strategy.

We will theoretically analyze our suggested strategy and do more tests with different real-world datasets in the future. In addition, we will extend the idea of our suggested technique to recommender systems' more complex learning processes.

REFERENCES

Burke, R., Mobasher, B., and Bhaumik, R. (2005). Limited knowledge shilling attacks in collaborative filtering systems. In Proceedings of the 3rd IJCAI Workshop in Intelligent Techniques for Personalization.

Burke, R., Mobasher, B., Williams, C., and Bhaumik, R. (2006). Classification features for attack detection in collaborative recommender systems. In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

Chirita, P., Nejdl, W., and Zamfir, C. (2005). Preventing shilling attacks in online recommender systems. In Proceedings of the 7th ACM international workshop on Web information and data management (WIDM), pages 67–74.

GroupLens Research (2016). MovieLens 1M Dataset. http://grouplens.org/datasets/movielens/.

Jagielski, M., Oprea, A., Biggio, B., Liu, C., Nita-Rotaru, C., and Li, B. (2018). Manipulating machine learning: Poisoning attacks and countermeasures for regression learning. In Proceedings of the 39th IEEE Symposium on Security and Privacy (S&P).

Koren, Y., Bell, R., and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. Computer and Information Science, 42(8):30–37.

Li, W., Gao, M., Li, H., and Zeng, J. (2016). Shilling attack detection in recommender systems via selecting patterns analysis. IEICE Transactions on Information and Systems, E99-10(10).

Mehta, B. (2007). Unsupervised shilling detection for collaborative filtering. In Proceedings of the 22nd AAAI Conference on Artificial Intelligence.

Mobasher, B., Burke, R., Bhaumik, R., and Williams, C. (2007). Toward trustworthy recommender systems: An analysis of attack models and algorithm robustness. ACM Transactions on Internet Technology (TOIT), 7(23).

Williams, C., Mobasher, B., Burke, R. D., Sandvig, J. J., and Bhaumik, R. (2006). Detection of obfuscated attacks in collaborative recommender systems. In Proceedings of the ECAI 2006 Workshop on Recommender Systems.

Wu, Z., Cao, J., Mao, B., and Wang, Y. (2011). Semi-SAD: Applying semi-supervised learning to shilling attack detection. In Proceedings of the 5th ACM Conference on Recommender Systems (RecSys).

Wu, Z., Wu, J., Cao, J., and Tao, D. (2012). HySAD: A semi-supervised hybrid shilling attack detector for trustworthy product recommendation. In Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.