A DeepWalk-Based Approach to Defend Profile Injection Attack in Recommendation System

Xu Gao, Wenjia Niu, Jingjing Liu, Tong Chen, Yingxiao Xiang, Xiaoxuan Bai, Jiqiang Liu

A DeepWalk-based Approach to Defend Profile Injection Attack in Recommendation System

Xu Gao, Wenjia Niu*, Yingdi Wang, Tong Chen, Yingxiao Xiang, Jingjing Liu, Xiaoxuan Bai, and Jiqiang Liu

Beijing Key Laboratory of Security and Privacy in Intelligent Transportation Beijing Jiaotong University,3 Shangyuan Village, Haidian District, Beijing 100044,China

{niuwj,17120360}@bjtu.edu.cn

Abstract. In the open social networks, the analysis of user data after the injection attack has a great impact on the recommendation system. K-Nearest Neighbor-based collaborative filtering algorithms are very vulnerable to this attack. Another recommendation algorithm based on probabilistic latent semantic analysis has relatively accurate recommendation, but it is not very stable and robust against attacks on the overall user data of the recommendation system. In this paper, we propose to use to DeepWalk the user network processing, while taking advantage of the user profile feature time series to consider the user's behavior over time, the algorithm also analyzes the stability and robustness of DeepWalk and user profile. The results show that especially the DeepWalk-based approach can achieve comparable recommendation accuracy.

Keywords: deepword, user profile, social representation, injection attack

1   Introduction

Recently, the recommendation system has been applied to different aspects of life, not only video or music websites that have a large amount of data and users, but also uses a recommendation system to analyze user data and behaviors for various aspects of life. Algorithm analysis recommends relevant resources to users. The recommendation system uses information based on past users and recommends by comparing the behavior of new users with similarities of other known users. In a configuration file injection attack, an attacker interacts with a recommendation system and constructs multiple configuration files related to the virtual identity in order to interfere with the system's output.

It can be clearly seen why collaborative filtering is vulnerable to these attacks through analysis. User-based collaborative filtering makes recommendations by finding users with similar profiles by collecting configuration files, and assuming that the user profile represents many different personal preferences. The configuration file may also contain some abnormal data. These abnormal data may be viewed by the system as

*corresponding author: niuwj@bjtu.edu.cn

normal users and result in biased recommendations. This is the phenomenon found in DeepWalk.

DeepWalk is a method of associating users with users and expressing them in a simple way. This paper mainly proposes a method of implicitly expressing nodes in a network. This method encodes the network relations of nodes in a continuous vector space. In fact, it maps each node in a network into a low-dimensional one. vector. In simple terms, a vector is used to represent each node in the network, and it is hoped that these vectors can express the relationships in the nodes in the network, that is, the nodes in the original network that the closer the relationship is, the corresponding vector is in its space. The closer the middle distance.

The potential role of the DeepWalk-based approach is to distinguish and link the injection profile from the original file and combine it into a low-latitude network vector. However, with the completion of the network establishment, the difference between the injection configuration file and the normal file cannot be intuitively distinguished.

The main contribution of this paper is to demonstrate that the model-based algorithm, especially based on DeepWalk algorithm, is significantly more robust against injection attacks than the PLSA-based algorithm. In addition, the method in the text uses user profile time series and digital image technology to further distinguish and complete recommendation system for accurate recommendation.

For the first time, this paper combines user profile with DeepWalk network nodes. This new expression can well absorb the advantages of the two methods, and at the same time it can play a good role in different aspects. This is a good use of this point. In the training network, a random walk is used to establish a node-node-associated network, and it is possible to provide a reduced-dimension network. This network can be easily and clearly defined in the network. The similar nodes are found by the Euclidean distance algorithm. The found nodes build the time series by the algorithm of the user's portrait.

Another innovation is that after defining the time series of user profile, not only basic methods can be used to process time series, but also sequence vectors can be converted into two-dimensional images. The processing of two-dimensional images can be converted into pictures for deep learning processing. The vector values are associated with grayscale images. Machine vision techniques are used to clean the data, and the images are classified and trained. The training methods can use convolutional nerves. Network, this method has obtained relatively good results in image processing, and we can also use these new technologies and user profile technology combined to handle many problems in the recommendation system.

2   Related Work

In this section, we review two algorithms for anti-injection attacks, based not only on the K-means clustering algorithms, but also on the related documents of the user profile sequences in order to deepen the understanding of the paper.

DeepWalk

Previous studies have begun to examine the vulnerability of different recommended technologies(H. Liu et al 2014; G. Stringhini 2013), such as collaborative filtering, in the face of attacks known as "shillings". We use more descriptive phrases "data injection attacks" because the promotion of a specific product is just one way of this attack. In a profile injection attack, the attacker interacts with a collaborative recommendation system to build multiple configuration files related to the virtual identity in order to bias the system output.

It is easy to see why collaborative filtering is vulnerable to these attacks. The deepwalk-based collaborative filtering algorithm collects user profiles, assuming that the user profile represents the preferences of many different individuals and makes recommendations by finding peers with similar profiles. If the profile database contains biased data (for example, some profiles assign higher ratings to specific items), these biased profiles may be considered as genuine user's companions and lead to biased recommendations. This is exactly the effect found in the DeepWalk related paper in (M. Jiang 2016) and (K. Thomas et al 2013).

DeepWalk uses local information obtained from truncated random walks. Potential performance in multiple multi-label network classification tasks on social networks such as BlogCatalog, Flickr and YouTube. The results show that DeepWalk beats the challenging baselines that allow global observation of the network, especially in the presence of missing information(K. Lee, J. Caverlee, and S. Webb 2010). When the marker data is scarce, DeepWalk representation can provide a F1 score that is 10% higher than the competition method. In some experiments, DeepWalk performance outperformed all benchmark methods while using 60% less training data in (G. Stringhini, C. Kruegel, and G. Vigna 2010). DeepWalk is also extensible. It is an online learning algorithm that can build useful incremental results and can be parallelized. These features make it suitable for a wide range of real-world applications, such as network classification and anomaly detection.

DNA-inspired Behavioral Modeling

In this work, we use DNA heuristic modeling to analyze the time series of certain related activities. With regard to some algorithms in this area, there are some papers whose philosophical background is similar. In fact, these papers focus on the different characteristics and dimensions of the behavioral characteristics, and the general-purpose ones do not study one group as a group.

In papers 24 and 25, the connectivity pattern in the large graph was studied, and unexpected behavior emerged. After accidental behavior has factual data with lockstep features, for example, a large number of followers are connected to the same set of followers, the author depicts the correspondence between the lock walk in the social graph and the dense blocks in the graph's adjacency matrix(L. Bergroth, H. Hakonen, and T. Raita 2000). We borrowed this social connection between user groups.

We believe intuitively that if the collective online behavior occurs once, then the behavior is not necessarily deceptive. On the contrary, if this collective action recurs

over time, especially in response to similar incidents, it may represent an abnormal activity. Specifically, this work focuses on forwarding activities, defining features for the retweets thread characterization, and proposes a method for capturing simultaneous fraud.

3 Methods

3.1 Model Preparation

A profile injection attack against a collaborative recommendation system consists of multiple attack profiles added to the real user profile database. The goal of push attacks is to increase the system's predictive rating for a particular user's target project. The attack type is a method of building an attack summary based on the recommendation system, rating database, product and/or user knowledge.

The specific attack type defines the method of assigning scores to filled items and target items. A filled item set represents a randomly selected set of items in the database that are assigned ratings in the attack profile. In some types of attacks, a subset of items may be pre-selected for precise impact. Target items in push attacks are usually given the maximum allowable level.

Random attacks and average attacks are the basic types of attacks and are further summarized. In both cases, the fill of the attack profile is assigned a random rating. For random attacks, the ratings are distributed around the average of the global ratings. For the average attack, the ratings are distributed around the individual averages of each fill item.
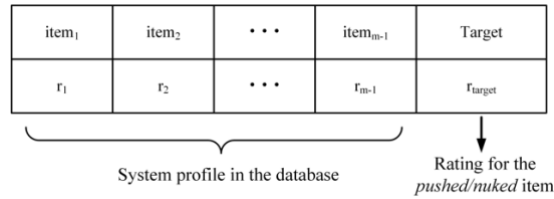
item$_1$	item$_2$	$\cdots$	item$_{m-1}$	Target
r$_1$	r$_2$	$\cdots$	r$_{m-1}$	r$_{target}$

System profile in the database   Rating for the *pushed/nuked* item

Fig. 1. General form of an attack profile

In fact, average attacks are more effective than random attacks. However, it needs more knowledge about the distribution of system ratings. The reason why this knowledge cost is minimized is that the average attack can be quite successful with a small set of filled items, and a random attack usually must score every item in the database to be effective.

The extension of random attacks, ad hoc attacks are almost as effective as ordinary attacks. The goal of rogue attacks is to associate the target item with a small number of frequently rated items. This facilitates feature distribution: a few items will get the most ratings. In the wagon attack, a small number of frequently rated items are selected along with the same set of random fills. Attack profiles provide maximum ratings for items that have high visibility and are therefore likely to be similar to a large number of users.

In our experiment, we used the public Movie-Lens 100K dataset1. The data set was rated 100,000 by 943 users for 1682 movies. All ratings are integer values between 1 and 5. Our data includes all users who rated at least 20 movies.

For the attack experiment, the entire data set was divided into training set and test set. Typically, the test set contains a sample of 50 user profiles that reflect the overall distribution of the user, including the number of movies seen and the ratings provided. When evaluating users within a segment, segmented attacks are an exception. In this case, the test set only contains user profiles that evaluate each movie in four or five. If there are more than 50 subdivided users, random sample 50 is used as the test set. The rest of the user profile after deleting the test set is designated as the training set. All collaborative filtering models and attack profiles were built from the training set and separated from the test set.

This group of attacks consisted of 50 films and their score distribution matched the overall score distribution of all films. Each movie will be attacked as a separate test and the results will be aggregated. In each case, some attack profiles are generated and inserted into the training set, and any existing scores of the attacked videos in the test set are temporarily deleted.

3.2 Model Based on DeepWalk

In this paper, there are a large number of user configuration files in the recommendation system. We mix the original data files in the data set with the injected attack files and use the DeepWalk algorithm to build the entire configuration file on the network. In simple terms, it is through random walks. The method of walking traverses the local structure of the network.

For the user profile in the recommendation system, we use the DeepWalk algorithm to build the user file of the user and the attacker into a network. $W = (w_1, w_2, w_3, \cdots, w_{n-1}, w_n)$ indicates the sequence of user profiles in the network.

The algorithm mainly consists of two parts: a random walker and an update process. The random walk generator randomly selects network nodes and generates a fixed-length random walk sequence. Each node generates a random walk sequence of a corresponding length. In this paper, we use the layered skipGram algorithm to update the node representation. This article uses the layered softmax method to train.

3.3 The Time Queue of User Profile

The idea of user profile is widely used in various aspects such as algorithms and research. This method can analyze specific problems, find out their representative features, and abstract them into vector representations. However, this kind of vector representation also has its defect type. Simply considering the vector representation does not accurately solve the problem in this paper.

We use the classification method based on the time queue of user profile to represent the specific behavior of a user or configuration file. This method of representation has two advantages. It can accurately represent the time characteristics of specific behaviors, so that it can reflect deeper features, and the use of user profile can express

abstract things visually. It is very important for understanding and scientific analysis. s help. The specific feature representation method is as follows. The data set is subjected to data cleaning and related features are extracted.

We have also used time series to analyze the behavioral similarity between users and users. The processing methods here have been proposed in many previous studies, such as abstracting time series into images, training them with time, and obtaining the final results. We can use the longest common subsequence of time series to compare the similarity between time series. This method is intuitive, and we use this method to build a network based on deepwalk, and then compare the network nodes that are close to the attack point. Then through the analysis of its time series, to prevent the attack of the injected node.

4   Experiment

4.1   Network Correlation and Characteristics

The traditional graph-based representation, such as $G = (V, E)$, uses different symbols to name different user profiles, and uses the storage structure of the adjacency matrix to represent the relationship between different users. This method of representation also has drawbacks. There is no relationship among most nodes in long-tail distribution, so the adjacency matrix is very sparse, which is not conducive to storage calculations. Skip-Gram's model is shown in Figure 2.
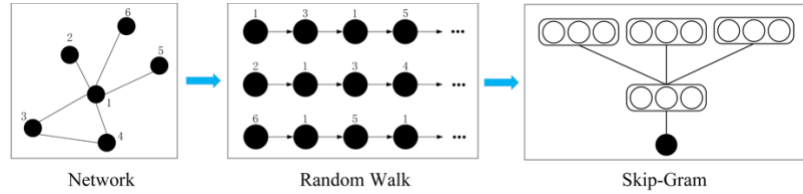

Fig. 2. The model of Random walk

Another method for representing graphs is network identification learning, which also becomes a graph embedding method. Nodes in a network are represented by low-dimensional, dense, and real-valued vectors. Nodes may have intrinsic relationships that facilitate computer storage. Feature extraction, adaptive, and can project heterogeneous information into the same low-dimensional space for downstream calculations.
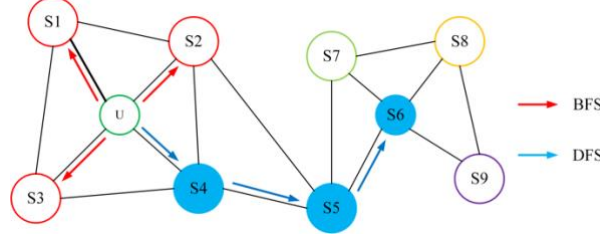
Fig. 3. Two ways to traverse graphs

In Figure 3, SkipGram method is applied to the training of the whole network to represent the nodes S in the network. According to SkipGram idea, the most important thing is to define the context, that is, the neighborhood. Random walk to get the neighboring nodes of graph or network. At the same time when constructing the network, two concepts were used, namely first-order similarity and second-order similarity. The first-order similarity represents the joint probability of directly connected nodes, as follows:

$$p_1(v_{i,}v_j) = \frac{1}{1+\exp(-\vec{u}_i^T \cdot \vec{u}_j)} \tag{1}$$

V stands for the node in the network and u stands for the embedding of the node. The above expression means that the more similar the two nodes are, the larger the inner product is, and the larger the value after sigmoid mapping is, that is, the greater the weight of the two nodes is connected. The second-order similarity uses conditional probabilities between nodes connected by other intermediate nodes, as follows:

$$p_2(v_i|v_j) = \frac{1}{\sum_{k=1}^{|V|} \exp(\vec{u}_i^T \cdot \vec{u}_j)} \tag{2}$$

The goal of the model is to make the similarity between the front and back nodes in the network unchanged. If the nodes indicate that the two nodes are similar before learning, then the two nodes that intruded represent the similarity of the vectors. In this paper, KL divergence is used to measure the distance between two probability distributions. The joint probability of experience between nodes is:

$$\hat{p}_1(i,j) = \frac{w_{ij}}{W} \tag{3}$$

$$O_1 = -\sum_{(i,j)\epsilon E} w_{ij} \log p_1(v_i, v_j) \tag{4}$$

The specific calculation method firstly selects the network nodes randomly and generates a fixed-length random walk sequence. This sequence is analogized to the user's profile property, and the skip-gram model is used to learn the distributed representation of nodes. If a network node obeys a power law distribution, then the number of occurrences of nodes in a random walk sequence obeys the same distribution. In this model, depth and breadth can also be used in the same way. The innovation lies in improving the strategy of random walk, achieving balance in the two

walk strategies, taking into account local and macro information, and having a high degree of adaptation. Sex. The following figure shows the specific steps for building a network through this model.

4.2 Analysis

We established the network between nodes through DeepWalk. After that, we selected the node associated with the injected node in the network and extracted it. We used the abstract feature of the user's portrait to analyze the user's image behavior sequence at a certain time period. Features include time, viewing style, and corresponding scores.
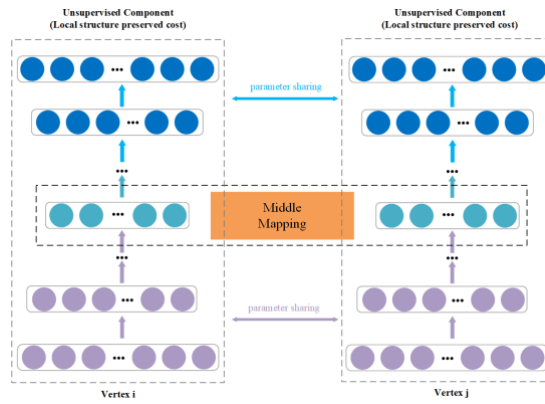

Fig. 4. The framework of the semi-supervised deep model of user profile

Figure 4 is a training model for user profile similarity calculations, including supervised and unsupervised ways to train the model. We pass through the analysis of these features through two kinds of analysis. One is the matching of the simplest longest subsequence, and the other is the abstraction of features into image grayscale values. The group sequence is converted into a two-dimensional image. Through the training and matching of the images, the similarity between the associated points is found and the points with the larger differences are eliminated. There are many ready-made methods for this method, for example, Deep learning, machine learning, and more.

We first compare the accuracy of the user's portrait and the algorithm based on the DeepWalk model. To detect the accuracy of the algorithm and assist in adjusting the recommended algorithm, we use the Mean Absolute Error (MAE) metric. MAE is a statistical method that compares predicted values with actual user ratings. In all cases, cross-validation is performed on the entire data set and no attack profile is injected.
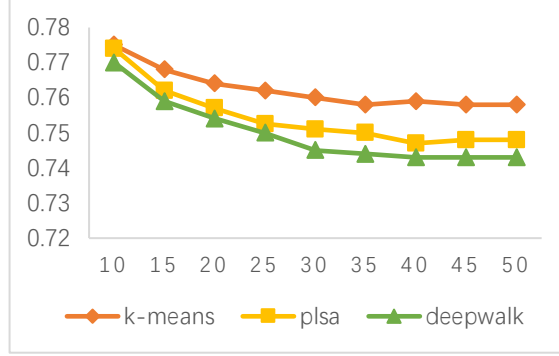
Fig. 5. Comparison of MAE

In Figure 5, we use the user to obtain the optimal results for the neighbourhood size of the algorithm proposed in this paper. For model-based algorithms, we use user-specific features to segment to get the most favorable results. In all cases, we will filter out features with a similarity score less than 0.1. For PLSA, we observed that the optimal threshold $\mu = 0.035$. Second, we use time, type, and scores to comprehensively measure similarity features.

As shown in the figure, k-means and PLSA are not as accurate as Deepwalk. PLSA is more accurate than k-means. For the rest of the experiments, we applied 30 user segments for k-means and PLSA. Although more user groups lead to improved MAE, 30 seems to be the key to reducing regression. In order to build a model, the larger part requires a considerable amount of processing time, and DeepWalk makes time acceptable by abstracting the network relationships.

For each profile injection attack, we track the attack size and padding size. The size of the attack is the number of injected attack rules and is measured as a percentage of the pre-attack training set. There are approximately 1000 users in the database, so a 1% attack size corresponds to approximately 10 attack signatures added to the system. Fill size is the number of fill ratings for a particular attack profile and is measured as a percentage of the total number of movies. There are approximately 1700 movies in the database, so a fill of 15% corresponds to approximately 200 fill ratings in each attack profile. We use the metrics of predictive transfer and hit rate to measure the relative performance of various attack models.
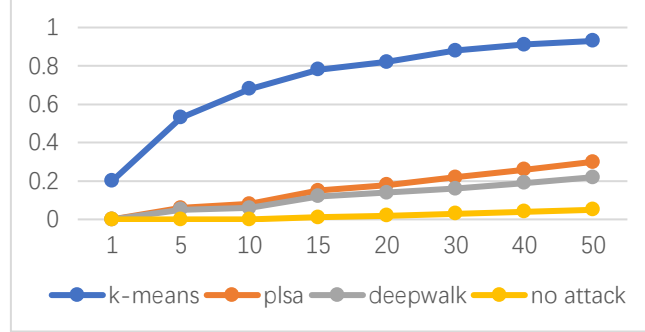
Fig. 6. Average attack hit ratio at 5% filler and 15% attack

Figure 6 presents hit ratio for an average attack using a 5% filler size and 15% attack size. To assess the sensitivity of the filler size, we tested 5%, 10%, 25% and 100% of the fill items on each type of attack. 100% fillers are listed as potential benchmarks for attack. However, from the perspective of the attacker, it is unlikely to be practical. Collaboratively filtering rating databases is often very rare, so the attack characteristics of rating each product are very conspicuous. Of particular interest is the smaller filler size. An attack with few fill-in items is difficult to detect.

5   Conclusion

In this article, we have demonstrated the robustness and stability of the approach based on Deep-walk and user profile. Specifically, we focused on a recommendation algorithm based on deep-walk-based user behavior analysis in order to find similar user segments compared to online users' profiles to generate recommendations. This algorithm models and analyzes all user data information, abstracts the data and compares the similarities between similar configuration files, while eliminating most of the impact of the biased attack profile.

Acknowledge

References

1. H. Liu et al., "U*ncovering deception in social media*" Social Network Analysis and Mining, vol. 4, no. 1, pp. 1–2, 2014.
2. G. Stringhini, G. Wang, M. Egele, C. Kruegel, G. Vigna, H. Zheng, and B. Y. Zhao, "*Follow the green: growth and dynamics in Twit- ter follower markets*" in Internet Measurement Conference (IMC). ACM, 2013, pp. 163–176.
3. M. Jiang, P. Cui, A. Beutel, C. Faloutsos, and S. Yang, "*Catching synchronized behaviors in large networks: A graph mining ap- proach*" ACM Trans. on Knowledge Discovery from Data, vol. 10, no. 4, 2016.
4. K. Thomas, D. McCoy, C. Grier, A. Kolcz, and V. Paxson, "*Traf- ficking fraudulent accounts: The role of the underground market in Twitter spam and abuse*" in 22nd USENIX Security Symposium, 2013, pp. 195–210.
5. K. Lee, J. Caverlee, and S. Webb, "*Uncovering social spammers: social honeypots machine learning*" in 33rd Research and Devel- opment in Information Retrieval. ACM, 2010, pp. 435–442.
6. G. Stringhini, C. Kruegel, and G. Vigna, "*Detecting spammers on social networks*" in 26th Annual Computer Security Applications Conference (ACSAC). ACM, 2010, pp. 1–9.
7. S. Fortunato, "*Community detection in graphs*" Physics Reports, vol. 486, no. 3, pp. 75–174, 2010.
8. Z. Yang, C. Wilson, X. Wang, T. Gao, B. Y. Zhao, and Y. Dai, "*Uncovering social network sybils in the wild*" Trans. Knowledge Discovery from Data, vol. 8, no. 1, 2014, special issue TKDD-CASIN.
9. L. Bergroth, H. Hakonen, and T. Raita, "*A survey of longest common subsequence algorithms*" in String Processing and Infor- mation Retrieval, 2000. SPIRE 2000. Proceedings. Seventh International Symposium on. IEEE, 2000, pp. 39–48.
10. D. P. Kroese, T. Brereton, T. Taimre, and Z. I. Botev, "*Why the Monte Carlo method is so important today*" Wiley Interdisciplinary Reviews: Computational Statistics, vol. 6, no. 6, pp. 386–392, 2014.
11. D. M. W. Powers, "*Evaluation: from Precision, Recall and F- Measure to ROC, informedness, markedness and correlation*" International Journal of Machine Learning Technologies, vol. 2, no. 1, pp. 37–63, 2011.