Data Poisoning Attacks on Linked Data with Graph Regularization

by

Venkatesh Magham

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

Approved April 2019 by the
Graduate Supervisory Committee:

Huan Liu, Chair
Liang Wu
Hani Ben Amor

ARIZONA STATE UNIVERSITY

May 2019

ABSTRACT

Social media has become the norm of everyone for communication and also as a mainstream in everyday life. (1). The usage of social media has increased exponentially in the last decade. The myriads of Social media services such as Facebook, Twitter, Snapchat, and Instagram etc allow people to connect with their friends, and followers freely. The attackers who try to take advantage of this situation has also increased at an exponential rate. Every social media service has its own recommender systems and user profiling algorithms(2). These algorithms use users current information to make different recommendations. Often the data that is formed from social media services is Linked data as each item/user is usually linked with other users/items. Recommender systems due to their ubiquitous and prominent nature are prone to several forms of attacks(11). One of the major form of attacks is poisoning the training set data. As recommender systems use current user/item information as the training set to make recommendations, the attacker tries to modify the training set in such a way that the recommender system would benefit the attacker or give incorrect recommendations and hence failing in its basic functionality (10). Most existing training set attack algorithms work with "flat" attribute-value data which is typically assumed to be independent and identically distributed (i.i.d.) (4) . However, the i.i.d. assumption does not hold for social media data since it is inherently linked as described above. Usage of user-similarity with Graph Regularizer in morphing the training data produces best results to attacker (3). This thesis proves the same by demonstrating experiments on Collaborative Filtering with multiple datasets.

Whatever I am or whatever I have is all because of my parents. It's only fair that I dedicate this thesis to them.

ACKNOWLEDGMENTS

This work would not have been possible without the scholastic and continous support from Dr. Liang Wu and Dr. Huan Liu. I am especially indebted to Dr. Liang Wu, who has been supportive of this thesis and also my career goals. I am grateful to all those with whom I have had the pleasure to work during my last two years. I would like to thank Dr.Hani Ben Amor for taking his precious time and be on my committee. I thank all the DMML members who have helped me with this thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

Social media has become the most important aspect of everyday life. Nowadays, most communication is done through social media. Imagining a life without social media for example Facebook, Instagram, and Snapchat etc has become much harder. Number of active users in Facebook has increased from around 100 million in 2008 to more than 2 billion people in 2018; Instagram has almost a billion users now while it had only 90 million users five years back in 2013. The rate of increase is quite similar in other social media services such as Twitter and Snapchat from their inception. In each form of social media, the relation between users are similar with each user connected to multiple users as friends or followers.(1) This paper mainly focuses on Recommender systems and User profiling algorithms. Recommender systems use the relations between users on the social media service to suggest potential friends or followers in all of social media. On the other hand, User profiling algorithms make models of each user based on their activity on the social media service. User-profiling algorithms use these models for suggestions and recommendations.(6) In most of the algorithms, user-product relation is represented as a matrix with each value in the matrix representing affinity of user in the corresponding row to product of the corresponding column. As these matrices are incomplete, algorithms like collaborative-filtering try to fill them by using user similarity matching.(15) As recommender systems play a vital role, they are susceptible to different types of attacks. We are exploring one form of attack called data-poisoning where the attacker tries to morph the training data for recommender system and user profiling algorithms (10). In this type of attack, a malicious party creates a set of users with preferences in such

a way that, the recommender systems benefit the attacker. Data poisoning attacks could be of two types. In one case, recommender system might benefit one user or one product or in the other the system might make completely obsolete recommendations. In both cases, credibility of the model is lost.

Many existing algorithms working on graph linked data assume that the data is independent and identically distributed. But social media data is linked data in which each user is connected to other user and hence the assumption of i.i.d assumption on the data is no longer valid. The problem of linked data is very well described in *Tang et al*(3). In social media data, the users tend to form groups having much intra-connections in the group than inter-connections with users of other groups. Having this type of linkage makes the i.i.d assumption obsolete.

We present a systematic approach to computing near-optimal data poisoning attacks for factorization- based collaborative filtering/recommendation models. We assume a highly motivated attacker with knowledge of both the learning algorithms and parameters of the learner following the Kerckhoffs principle to ensure reliable vulnerability analysis in the worst case.

An attacker would want to conceal his attacks by doing minimal manipulations to the training data that produce the best results as mentioned in this paper. The main aim of the attacker would be to make as few changes as possible to the data and disrupt the system to his favor as much as the attacker could. This would be a bi-level optimization problem. Our main contributions in this paper are

- Formalizing this bi-level optimization problem using user similarity as a metric for attack.

- Our second contribution is to demonstrate our attack framework on Linked graph data.

The rest of the paper is defined as follows. We formally define the problem of data poisoning attacks on linked data using user similarity in section 3; introduce our new framework for data poisoning in section 3, with KKT conditions and user-similarity; present empirical evaluation with discussion in Section 4 and the related work in Section 5; and conclude this work in Section 6.

Chapter 2

PROBLEM STATEMENT

We discuss our problem statement here and formulate the optimization to solve. In this paper, scalars are denoted by lower-case letters (a, b, . . .), vectors are written as lower-case bolded letters (**a**, **b**, . . .), and matrices correspond to boldfaced upper-case letters (**A**, **B**, . . .). We also assume that attacker is fully aware of the system's learning algorithm. There are two types of Machine learner problems for social media. We formulate the both of them using following equations.

$$\hat{\boldsymbol{\theta}}_D \, \epsilon \, argmin_{\theta \epsilon D} \qquad O_L(D, \theta),$$

$$s.t. \qquad g_i(\theta) \leq 0, i = 1...m,$$

$$h_i(\theta) = 0, i = 1...p$$

$$\min_{\mathbf{X} \epsilon R^{mxn}} \quad ||R(\mathbf{M\text{-}X})||_F^2, \;\; s.t. \, rank(X) \leq k,$$

Where D is training data. In classic machine learning, D is an iid sample from the underlying task distribution but in the case of social media, D cannot be iid as discussed above in the introduction because of the linkage between the data. $O_L(D, \theta)$ is the learner's objective: For example, regularized risk minimization can be formulated as

$$O_L(D, \theta) = R_L(D, \theta) + \lambda \delta \theta$$

where, for some learner's empirical risk function RL and regularizer δ. The g and h functions are potentially nonlinear; together with the hypothesis space θ they deter-

mine the feasible region. $\hat{\boldsymbol{\theta}}_D$ is the learned model (recall argmin returns the set of minimizers).

In equation 4, we consider machine learners that can be posed as a matrix completion problem which are also optimization problems. Let $M \epsilon R_{mn}$ be a data matrix consisting of m rows and n columns. $M_{ij}\, for\, i \epsilon [m]\, and\, j \epsilon [n]$ would then correspond to the rating the ith user gives for the jth item. We use δ = (i, j) : M_{ij} is observed to denote all observable entries in M and assume that $|\delta|||$ ¡¡ mn. We also use i [n] and j [m] for columns (rows) that are observable at the ith row (jth column). The goal of collaborative filtering (also referred to as matrix completion in the statistical learning literature [2]) is then to recover the complete matrix M from few observations M. One standard assumption is that **M** is a low ranked matrix which can be obtained by solving equation 4.

Chapter 3

PROPOSED METHOD

The increase in online user usage has led to a variety of information which could be used in multiple ways. If one wants to watch a movie, it would be a painful experience to go through all the movies and pick one. Recommender systems help users to ease their process of selection by suggesting items to users that users might find beneficial. Recommender systems are the essential parts of most software companies ranging from Google search to Amazon Ecommerce to Netflix video recommendations etc. They also have become a key part of people's social life via Facebook, Twitter, Youtube, and Netflix etc. Online recommender systems root back to several disciplines such as cognitive science, information retrieval, and etc. Precise recommender systems help both the spectrum of industry i.e. Users and Vendors. Users would find their targets easily and Vendors would in turn make profits in less time than needed and also keep customers happy. Netflix through its contest awarded 1 million dollars to the team that improved their recommender system's accuracy.

Due to their dominance, they became an independent area of research from mid 1990s. Social Networks in online platforms increase the social life of people. Social recommender systems help people find their potential friends or inspirational figures etc. There are many approaches in solving social recommender systems. One of the most frequent way used is collaborative filtering. In a typical collaborative filtering systems an n*m user matrix is created, where n users preferences about m products are represented as ratings. The collaborative filtering systems maps similar users and similar movies and tries to predict unseen ratings. With the increase in prominence of social recommender systems using collaborative filtering, the threats from people

who want to misuse them increase exponentially. Malicious users or rival companies try to insert fake users or manipulate the data matrix available for two main purposes. Malicious users want to change the recommender systems in such a way that it recommends the items that are valuable for malicious users more frequently. While the rival companies tend to inject fake users so as to decrease the accuracy of the recommender system.

These type of attacks are called shilling attacks. In the case of social networks like Facebook, malicious users might want to increase their popularity and hence inject the recommendation system in such a way that it suggests them. The same case follows for Youtube, Twitter and etc. In the case of Netflix, rival companies might introduce injections to reduce its accuracy. While the e-commerce systems like Amazon, Zappos are vulnerable for both kinds of attacks.

3.0.1 Identifying K.K.T. Conditions

In this section, we talk about how to identify the KKT conditions of our bi-level optimization problem and formulate them. Bi-level optimization problems are NP hard in general. We present an efficient solution for a broad class of training- set attacks. Specifically, we require the attack space D to be differentiable (e.g. the attacker can change the continuous features in D for classification, or the real-valued target in D for regression). Attacks on a discrete D, such as changing the labels in D for classification, are left as future work. We also require the learner to have a convex and regular objective OL.

Under these conditions, the bi-level problem Eq (5) can be reduced to a single-level constrained optimization problem via the Karush-Kuhn-Tucker (KKT) conditions of the lower-level problem (Burges 1998). We first introduce KKT multipliers i,i = 1...m and i,i = 1...p for the lower-level constraints g and h, respectively. Since the lower-

level problem is regular, we replace the lower-level problem with its KKT conditions (the constraints are stationarity, complementary slackness, primal and dual feasibility, respectively):

3.0.2 Attack Against Matrix Factorization Systems

Most of the recommender systems use Matrix Factorization systems, here we formulate our attacks on those systems. In the attack against collaborative filtering, the data matrix consists of m users and n items. Since, every user wouldn't have rated every field in the matrix is not filled in and collaborative filtering algorithms is used to fill in the matrix. An attacker can add αm users. Since, we would like to avoid being detected each user can give his preference only up to N items and in the range of -l, l. The main reason behind it is to go undetected from the agent.

These are the notations used in the paper. M as the original matrix. M with a cap to represent the matrix that consists of all the malicious users. The dimensonality of original Matrix is m * n and the dimensionality of malicious matrix M cap is alpha m * n. Alpham represents number of malicious users. Since, we want a risk averse model, we assume the maximum value of alpha to be 0.3. Since, this is a bi-level optimization problem, equation 4 in the problem statement can now be formulated as

$$\boldsymbol{\theta}_\lambda(\hat{M} : M) = argmin_{U,\hat{U},V^T} ||R_\omega(M - UV^T)||_F^2 +$$
$$||R_{\hat{\omega}}(\hat{M} - \hat{U}V^T)||_F^2 + 2\lambda_U(||U||_F^2 + ||\hat{U}||_F^2) + 2\lambda_V(||V||_F^2)$$

where the resulting output consists of low rank latent factors U, U cap without

or with malicious users respectively while

The sign from the equation is a variable consisting of low-rank matrix factors U, U cap, and V repressing normal users, malicious users and items respectively. The Graph Regularization term used in the equation helps finding optimized U, U cap, and V values. We have M cap = UVT. The goal of the attacker is to find optimal malicious users, M cap such that, Equation 6 from the paper.

Here M = new M is the malicious data which we use to attack the collaborative filtering system. And S(Mcap, M) denotes the utility score that describes how good the attack is.

There could be two kinds of attacks based on the attacker utility.

3.0.3 Availability Attack

The main aim of this attack is to disrupt the collaborative systems so that it gives completely different predictions. Lets say that M dash is the systems prediction without data poisoning and M cap is the systems prediction. Then the utility function is defined as follows.

$$R^{av}(\hat{M}, M) = R(\hat{M} - \bar{M})^2$$

. The effectiveness of the attack is defined by the value of R, the higher it is the more severe the attack is.

3.0.4 Integrity attack

The main aim of this attack is to make few items in the set more popular. Let J is that subset and w is the weightage given to each item in set J by the attacker. Then the utility function is defined as follows.

$$R^{in}(\hat{M}, M) = \Sigma_{i=1}^{m} \Sigma_{j \epsilon J_0} w(j) M.$$

Where function R is the loss function for integrity function and and M hat is the prediction of system with data poisoning attack and M is predictions without data poisoning attack

3.0.5 Optimal Attack Strategy

We use the projected gradient agent (PGA) method for solving the optimization problem in Eq. (6) with respect to the alternating minimization (12) formulation in Eq. (4). In iteration t we update M cap as follows.

$$\hat{\boldsymbol{M}}^{*} \epsilon \, argmin_{\hat{M} \epsilon M} R(\hat{M}(\theta_{\lambda}(\hat{M}; M)), M)$$

Here, projection gradient is used so we keep all the malicious users preferences in a range of (-v, v) and st is the step size. Note that the estimated matrix M depends on the model (M ; M) learnt on the joint data matrix, which further depends on the malicious users M . Since the constraint set M is highly non-convex, we generate B items uniformly at random for each malicious user to rate. The ProjM() operator then reduces to projecting each malicious users rating vector onto an l ball of diameter , which can be easily evaluated by truncating all entries in M at the level of .

We next show how to (approximately) compute M R(M ,M). This is challenging because one of the arguments in the loss function involves an implicit optimization problem. We first apply chain rule to arrive at

$$\nabla_{\hat{M}} R(\hat{M}, M) = \nabla_{\hat{M}} \theta_{\lambda}(\hat{M}; M) \nabla_{\theta} R(\hat{M}, M)$$

Algorithm 1 Optimizing $\widetilde{\mathbf{M}}$ via PGA

1: **Input**: Original partially observed $m \times n$ data matrix $\mathbf{M}$, algorithm regularization parameter λ, attack budget parameters α, B and Λ, attacker's utility function R, step size $\{s_t\}_{t=1}^{\infty}$.
2: **Initialization**: random $\widetilde{\mathbf{M}}^{(0)} \in \mathbb{M}$ with both ratings and rated items uniformly sampled at random; $t = 0$.
3: **while** $\widetilde{\mathbf{M}}^{(t)}$ does not converge **do**
4: Compute the optimal solution $\boldsymbol{\Theta}_\lambda(\widetilde{\mathbf{M}}^{(t)}; \mathbf{M})$.
5: Compute gradient $\nabla_{\widetilde{\mathbf{M}}} R(\widehat{\mathbf{M}}, \mathbf{M})$ using Eq. (10).
6: Update: $\widetilde{\mathbf{M}}^{(t+1)} = \text{Proj}_{\mathbb{M}}(\widetilde{\mathbf{M}}^{(t)} + s_t \nabla_{\widetilde{\mathbf{M}}} R)$.
7: $t \leftarrow t + 1$.
8: **end while**
9: **Output**: $m' \times n$ malicious matrix $\widetilde{\mathbf{M}}^{(t)}$.

Figure 3.1: Algorithm Used

Chapter 4

EXPERIMENTS

4.0.1 Data

In this section we explain the datasets that we used, and pre-processing that we made to the datasets to fit to this problem. To evaluate the effectiveness of our proposed poisoning attack strategy, we use the publicly available MovieLens dataset(cite) for testing attacks on Collaborative-Filtering. The dataset contains 20 millions ratings and 465,000 tag applications applied to 27,000 movies by 138,000 users [23]. Each user who has watched a movie rates the movie from 1 to 5. We shift the rating range to [-2, 2] for computation convenience and setting neutrality to zero. To avoid the cold-start problem, we consider users who have rated at least 20 movies. The second dataset that we used is Amazon Instant Video dataset which has ratings about amazon videos, we removed users that have less than 10 ratings. Statistics of the dataset can be found on Table2. Precisely, even though several users are removed by preprocessing, Amazon dataset is still extremely sparse compared to the others. Two metrics are employed to measure the relative performance of the systems before and after data poisoning attacks: root mean square error (RMSE) for the predicted unseen entries and average rating for specific items.

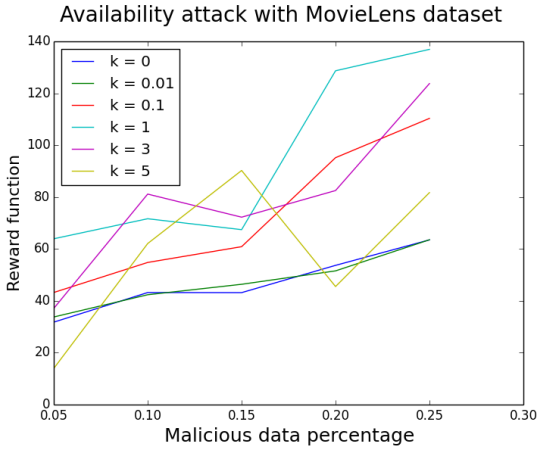


Figure 4.1: Availabality Attack on MovieLens Dataset

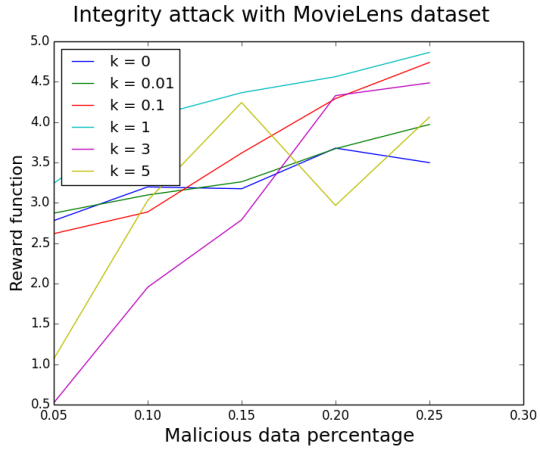


Figure 4.2: Integrity Attack on MovieLens Dataset

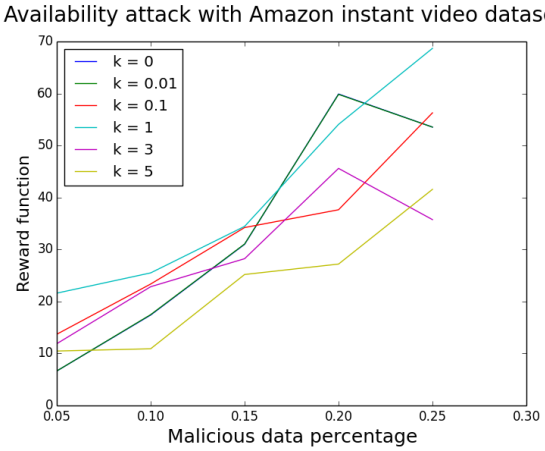


Figure 4.3: Availability Attack on Amazon Instant Dataset

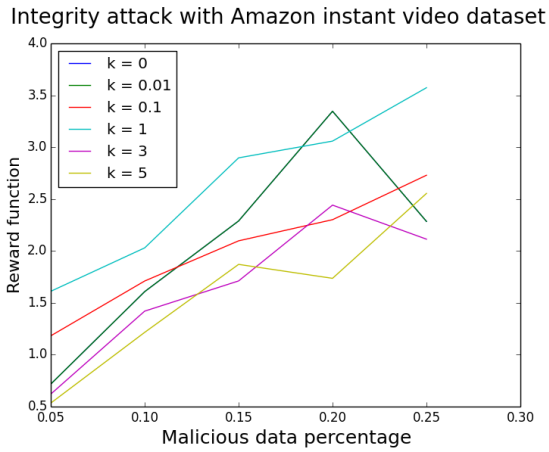


Figure 4.4: Integrity Attack on Amazon Instant Dataset

Table 4.1: Movielens Dataset Statistics

	Statistics
Users	943
Movies	1682
Ratings	100,000

Table 4.2: Amazon Instant Video Dataset Statistics

	Statistics
Users	5000
Movies	10843
Ratings	50,000

The first step of our experiment was to check the best value for the hyper-parameter Regularization constant as this was crucial in determining how significant

Graph Regularization is for Linked data, we have tried values ranging from 0.001 to 100 for the same. The results can be seen in the figure below when experimented with availability attack along with integrity attack. Figure1, and Figure2 clearly depicts for availability attacks that Utility value linearly increases with reguralization constant peaking at the value of 1 and then starts decreasing from there. The same can be said for integrity attacks that the graph follows the same trend while utility peaks when reguralization constant is 0.84 by looking at figures 3, 4.

Our first experiment with malicious data was to test the bi-level optimization problem with graph regularization on MovieLens data with collaborative filtering on availability attacks.

Table 4.3: Malicious Percent with Utility for Movielens Dataset on Availability Attack

Malicious percent	utility	utility with Graph regularization
0.05	6.63	21.603
0.1	17.43	25.501
0.15	31.000	34.444
0.2	53.541	54.058
0.25	67.309	68.681

As you can see from the figures from 1 to 4, the movielens dataset being more dense than Amazon Instant Video dataset produces better results.

The above table depicts how much the error rises with the rise in Malicious percentage of users. As you can see, the increase is semi-linear with the peak slope occurring at 0.1 percent of malicious users. The calculation of utility value is measured as per equation 6. The third column explains the utility value when bi-level optimization is done with graph regularization. As you can see, the utility value is

Table 4.4: Malicious Percent with Utility for Amazon Instant Video Dataset for Availability Attack

Malicious percent	utility	utility with Graph regularization
0.05	31.778	63.921
0.1	43.194	71.642
0.15	43.160	67.434
0.2	53.671	128.675
0.25	63.497	136.933

almost double with graph regularization.

Table 4.5: Regularization Constant at 1

Malicious percent	utility with Graph regularization for MovieLens dataset on AA
0.05	21.603
0.1	25.501
0.15	34.444
0.2	54.058
0.25	68.681

As we discussed above, integrity attacks are widely popular as the key aim in most attacks is to increase the popularity of few items which might have been sponsored by the attackers. To see how data poisoning attacks fare with integrity attacks we have repeated similar experiments as above for Collaborative filtering with Graph Regularization on integrity attacks.

The above table depicts how much the error rises with the rise in Malicious percentage of users. As you can see, the increase is semi-linear with the peak slope

Table 4.6: Malicious Percent with Utility for Movielens Dataset on Integrity Attack

Malicious percent	utility	utility with Graph regularization
0.05	2.779	3.242
0.1	3.197	4.053
0.15	3.174	4.364
0.2	3.677	4.562
0.25	3.497	4.863

Table 4.7: Malicious Percent with Utility for Amazon Instant Video Dataset on Integrity Attack

Malicious percent	utility	utility with Graph regularization
0.05	0.716	1.611
0.1	1.607	2.029
0.15	2.288	2.897
0.2	3.346	3.058
0.25	2.284	3.572

occurring at 0.1 percent of malicious users. The calculation of utility value is measured as per equation 6. The third column explains the utility value when bi-level optimization is done with graph regularization. As you can see, the utility value is almost double with graph regularization.

After hyper-parameter tuning for regularization constant, we got the best results at the regularization constant as 1. The utility value at different percentage of malicious users is given in the above table.

Table 4.8: Regularization Constant at 1

Malicious percent	utility with Graph regularization on Movielens dataset with IA
0.05	3.242
0.1	4.053
0.15	4.364
0.2	4.562
0.25	4.863

Chapter 5

FUTURE WORK

Although we focused on formulating the optimal training-set attack in this paper, our ultimate goal for the poisoning attack analysis is to develop possible defensive strategies based on the careful analysis of adversarial behaviors. Our optimal training-set attack formulation opens the door for an alternative defense: flagging the parts of training data likely to be attacked and focus human analysts attention on those parts. And also, since the poisoning data is optimized based on the attackers malicious objectives, the correlations among features within a feature vector may change to appear different from normal instances. Therefore, tracking and detecting deviations in the feature correlations and other accuracy metrics can be one potential defense. Additionally, defender can also apply the combinational models or sampling strategies, such as bagging, to reduce the influence of poisoning attacks. We would also like to extend our work in other Machine Learning algorithms like SVM for classification tasks to prove that Graph Regularization is ubiquitous in getting the best attacks when used in social media platforms as the data present over there is not i.i.d.

Chapter 6

CONCLUSIONS

As you can see that from the above results, a simple addition of graph regularization in the bi-level optimization sky-rockets the utility values which show the effectiveness of the solution. The increase in the utility values you can see from the results is near linear in case of both integrity attacks and availability attacks.

We would like to conclude by saying that graph regularization is a less explored feature, which goes well with linked data for example social media data as the data is not i.i.d.

REFERENCES

[1] D. Blackwell, C. Leaman, R. Tramposch, C. Osborne, M. Liss *Extraversion, neuroticism, attachment style and fear of missing out as predictors of social media use and addiction.* Personality and Individual Differences, 116 (2017), pp. 69-72

[2] Ekstrand, M.D., Riedl, J.T., Konstan, J.A.: *Collaborative filtering recommender systems.* Found. Trends Hum.-Comput. Interact 4 (2011)

[3] J. Tang and H. Liu. *Feature selection with linked data in social media*. In SDM, 2012.

[4] Li, B., Wang, Y., Singh, A., and Vorobeychik, Y. *Data poisoning attacks on factorization-based collaborative filtering*. In Advances in Neural Information Processing Systems (NIPS), 2016a.

[5] Liang Wu, Diane Hu, Liangjie Hong, Huan Liu, *Turning Clicks into Purchases: Revenue Optimization for Product Search in E-Commerce.* ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2018), July 8-12, 2018. Ann Arbor, Michigan

[6] Jun Wang, Arjen de Vires, and Marcel Reinders. *Unifying user-based and item-based collaborative filtering approaches by similarity fusion.* In SIGIR, 2006.

[7] Emmanuel Cands and Ben Recht. *Exact matrix completion via convex optimization.* Foundations of Computational Mathematics, 9(6):717772, 2007.

[8] Shike Mei and Xiaojin Zhu. *Using machine teaching to identify optimal training-set attacks on machine learners*. In AAAI, 2015.

[9] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. *Item-based collaborative filtering recommendation algorithms.* In Proceedings of the 10th international conference on World Wide Web. ACM, 285/-295.

[10] Bamshad Mobasher, Robin Burke, Runa Bhaumik, and Chad Williams. *Effective attack models for shilling item-based collaborative filtering systems.* In Proceedings of the 2005 WebKDD Workshop, held in conjuction with ACM SIGKDD2005, 2005.

[11] Michael P OMahony, Neil J Hurley, and Guenole CM Silvestre. *Promoting recommendations: An attack on collaborative filtering.* In Database and Expert Systems Applications, pages 494503. Springer, 2002.

[12] Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. *Low-rank matrix completion using alternating minimization.* In STOC, 2013.

[13] *Amazon Instant Video Dataset. 2018.* http://jmcauley.ucsd.edu/data/amazon/

[14] Joseph A. Calandrino, Ann Kilzer, Arvind Narayanan, Edward W. Felten, and Vitaly Shmatikov. 2011. *You Might Also Like: Privacy Risks of Collaborative Filtering.* In IEEE Symposium on Security and Privacy

[15] Y Koren, R Bell, and C Volinsky. 2009. *Matrix factorization techniques for recommender systems.* Computer 8 (2009), 3037

[16] Xinyu Xing, Wei Meng, Dan Doozan, Alex C Snoeren, Nick Feamster, and Wenke Lee. 2013. *Take is Personally: Pollution Attacks on Personalized Services.* In USENIX Security. 671686.

[17] *MovieLens Dataset. 2018.* https://grouplens.org/datasets/movielens/

[18] M. OMahony, N. Hurley, N. Kushmerick, and G. Silvestre. 2004. *Collaborative Recommendation: A Robustness Analysis.* ACM Transactions on Internet Technology 4, 4 (2004), 344377.

APPENDIX

1 SLIDES FOR PRESENTATION

Master's Thesis Defense

Data Poisoning Attacks on Linked Data with Graph Regularization

by
Venkatesh Magham

Graduate Supervisory Committee:
Dr. Huan Liu, Chair
Dr. Liang Wu
Dr. Hani Ben Amor

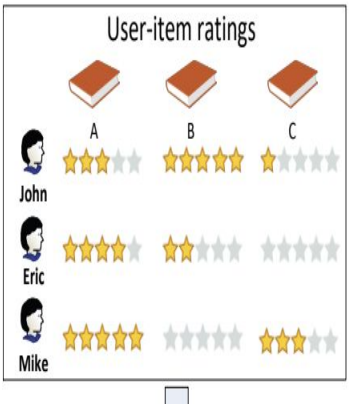
Arizona State University

Agenda

- **What is this thesis about?**
- **What are our contributions in this thesis ?**
- **What are the results ?**
- **Any future work plans ?**

Collaborative Filtering for Recommender Systems

- Matrix factorization for collaborative filtering
- It is widely used in various recommender systems
- Ex: Finding affinity relations, recommendations

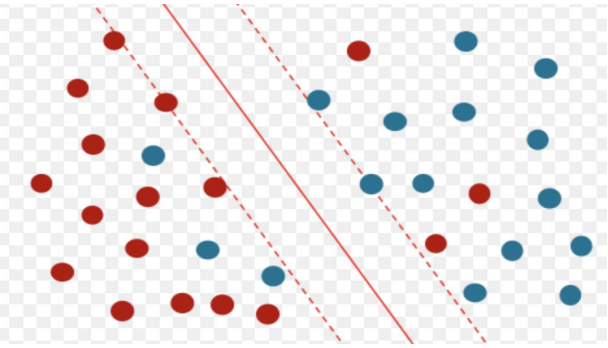


Data poisoning for collaborative filtering

- Poisoning the training data to break recommender systems
- Loss of trust
- Availability attack
  - Recommender system gives wrong recommendations as other fruits should be recommended first
- Integrity attack
  - In this scenario, attacker might be promoting items like milk, and hence it is recommended.

Related work on Classification based SVM systems

- Data poisoning has been applied to classification based SVM systems
- The experiments were done on wine dataset
- **Reference**: Shike Mei and Xiaojin Zhu. “Using Machine Teaching to Identify Optimal Training-Set Attacks on Machine Learners.” In: AAAI. 2015, pp. 2871–2877



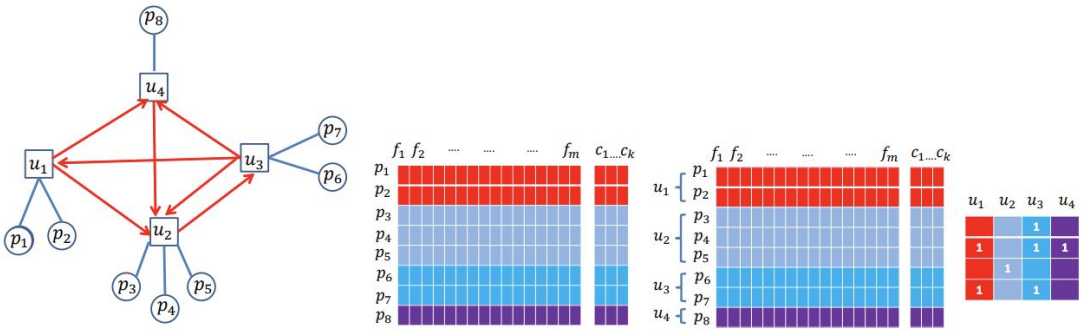
Related work

- Data poisoning for recommender systems, proposed in 2016 NIPS
- The work was done on MovieLens dataset
- The optimization function for this thesis was built based off of this paper

Reference: Bo Li and Yining Wang. “Data poisoning attacks on factorization-based collaborative filtering”. In Advances in Neural Information Processing Systems (NIPS*), 2016a.

*NIPS is now called NeurIPS

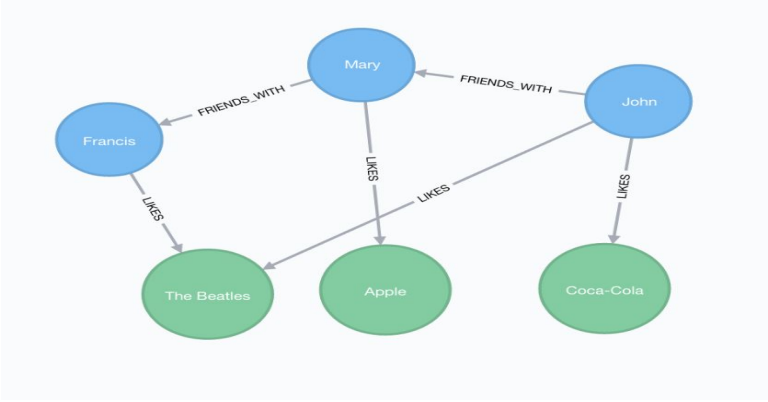
Structure of Linked Data



(a) Social Media Data Example (b) Attribute-Value Data (c) Representation with Social Context

Invalid IID assumption

- IID: Each datapoint is mutually independent and follows the same distribution
- In the context of social media, the data does not hold iid assumption as each user/product is linked to one another thus creating linked data



Our Contributions in this thesis

- Formulating the bi-level optimization function by using graph regularization
- Pre-processing the datasets and experimenting to find the best hyper parameters
- Experimentally proving that using graph regularization produces better results for data poisoning attacks

Attacks with Graph Regularization

- Optimization function without Graph Regularization

$$\boldsymbol{\theta}\lambda(\hat{M} : M) = argminU, V^T ||R_\omega(M - UV^T)||_F^2 + 2\lambda U(||U||^2 F) + 2\lambda V(||V||^2 F)$$

- Optimization function with Graph Regularization

$$\boldsymbol{\theta}\lambda(\hat{M} : M) = argminU, V^T ||R_\omega(M - UV^T)||_F^2 + 2\lambda U(||U||^2 F) + 2\lambda V(||V||^2 F) + kU^T LU$$

Loss/Reward function for availability attack and integrity attack

- Availability attack:

$$R^{av}(\hat{M}, M) = R(\hat{M} - \bar{M})^2$$

- Integrity attack:

$$R^{in}(\hat{M}, M) = \Sigma_{i=1}^{m} \Sigma_{j \epsilon J_0} w(j) M$$

MovieLens dataset

- MovieLens 100k dataset is formulated by GroupLens Research
- The dataset that we used consists of 943 users rating 1682 movies
- A total of 100,000 ratings were recorded
- Each rating is in a range from 0 to 4

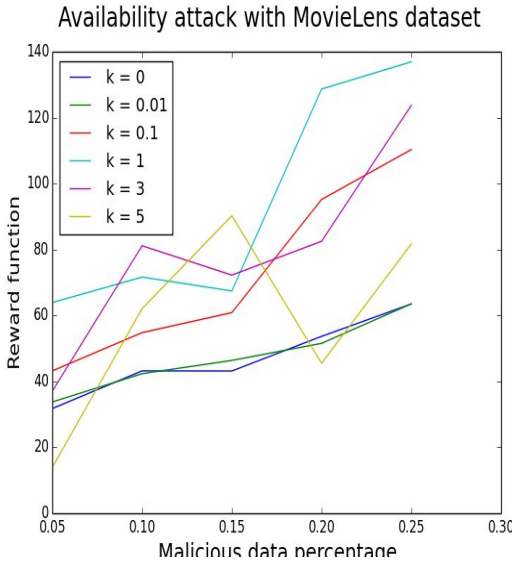
Amazon Instant Video dataset

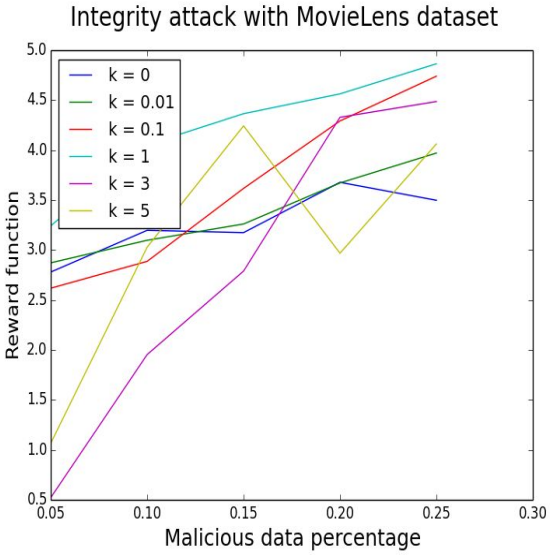
- This dataset has 5,073 users who have rated 10,843 items
- There were around 50,000 ratings in total

Hyper-parameters used for experiments

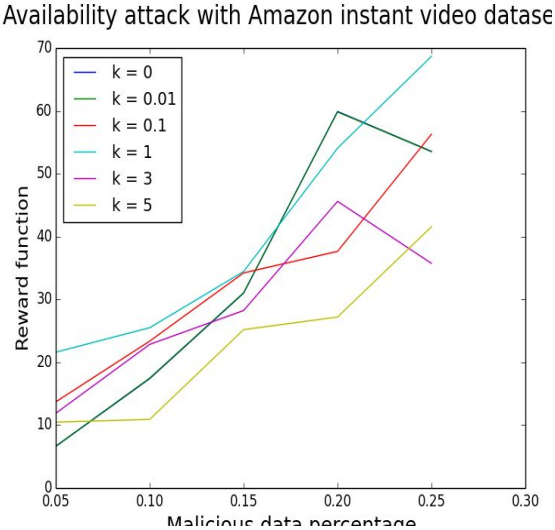
- Malicious user percentages to be injected
- Regularization Constant
- Dimensionality of Latent Vectors

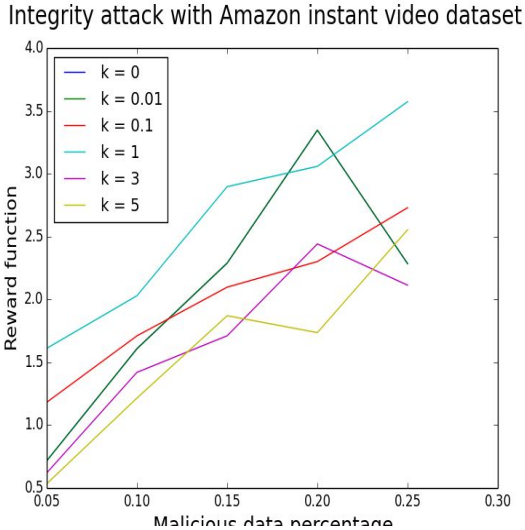
Results on MovieLens dataset





Results on Amazon instant video dataset





Conclusions

- From the experimental results from both the datasets we can see that using graph regularization easily outperforms the ones without using it.
- Graph regularization is a less explored feature

Future Work

- Expansion to Classification based models
- Building Defense systems against data poisoning attacks

Major references

- Bo Li and Yining Wang. “Data poisoning attacks on factorization-based collaborative filtering”. In Advances in Neural Information Processing Systems (NIPS*), 2016a.
- Shike Mei and Xiaojin Zhu. “Using Machine Teaching to Identify Optimal Training-Set Attacks on Machine Learners.” In: AAAI. 2015, pp. 2871–2877
- J. Tang and H. Liu. “Feature selection with linked data in social media”. In SDM, 2012.
- Amazon Instant Video Dataset
- MovieLens Dataset