協調フィルタリングに対するModel Inversion攻撃の提案

披田野 清良[1]　村上 隆夫[2]　勝又 秀一[2,3]　清本 晋作[1]　花岡 悟一郎[2]

概要：協調フィルタリングに基づき，ユーザの評価データ（あるいは購買履歴）からおすすめのアイテムを提示する推薦システムが利用されている．推薦システムの安全性に関する研究として，特定のアイテムを評価する（あるいは購入する）ことで，推薦精度の低下や特定アイテムの人気向上を引き起こすポイゾニング攻撃が提案されているが，それらはユーザのプライバシーの暴露を意図していない．そこで，本稿では，推薦システムに潜在するプライバシーリスクを明らかにすることを目的とし，ポイゾニングによりユーザのプライバシーを暴露する新たな攻撃として協調フィルタリングに対する Model Inversion 攻撃を提案する．本攻撃は，推薦システムがユーザに提示したアイテムから当該ユーザが過去に高く評価した（あるいは購入した）センシティブなアイテムを暴露する．

キーワード：推薦システム，協調フィルタリング，ポイゾニング攻撃，Model Inversion 攻撃

A Framework for Model Inversion Attacks on Collaborative Filtering

Seira Hidano[1]　Takao Murakami[2]　Shuichi Katsumata[2,3]　Shinsaku Kiyomoto[1]
Goichiro Hanaoka[2]

Abstract: Recommender systems using collaborative filtering have become increasingly popular in recent years. Such system predicts items that users would prefer based on their ratings (or purchase history). Data poisoning attack on collaborative filtering has been introduced by Li et al. in 2016, for the purpose of degrading the performance of a recommender system, or boosting/reducing the popularity of specific items. In this paper, we use data poisoning to expose user privacy and propose a model inversion attack on collaborative filtering. Our proposed attack enables to infer the sensitive items that a user rated highly (or purchased) from a recommended item.

Keywords: Recommender system, Collaborative Filtering, Poisoning Attack, Model Inversion Attack

1. はじめに

AIが注目を集める昨今，様々なITサービスにおいて機械学習が重要な役割を担っている．それらの中でもとりわけ，ユーザの過去のアクティビティに基づきおすすめのアイテムを提示する推薦システムは，eコマースなどのオンラインサービスにおいて欠かせない存在である．しかしながら，推薦システムはユーザのアイテムに対する評価データや購入履歴といったパーソナルデータに基づくものであり，ユーザのプライバシーに関する問題も懸念される．そこで，本稿では，推薦システムに潜在するプライバシーリスクを明らかにすることを目的とし，ユーザのプライバシーを暴露する新たな攻撃を提案する．

推薦システムの多くは協調フィルタリングと呼ばれる手法を用いている [1–3]．協調フィルタリングでは，ユーザのアイテムに対する評価データ（もしくは，購入履歴）を行列で表現し（以下，評価行列），評価行列に基づきユーザの未評価（もしくは，未購入）のアイテムに対する評価を予測する．以下，評価データを例として述べる．協調フィ

[1] KDDI 総合研究所
KDDI Research, Inc.
[2] 産業技術総合研究所
National Institute of Advanced Industrial Science and Technology (AIST)
[3] 東京大学
The University of Tokyo

ルタリングに対する攻撃としては，ポイゾニング攻撃と呼ばれる手法が提案されている [4–6]．ポイゾニング攻撃では，推薦精度の低下や，特定アイテムの人気向上/低下を目的とし，攻撃対象の推薦システムにおいて不正にアイテムを評価する．以下，区別のため，不正にアイテムを評価するユーザを悪性ユーザと呼ぶ．ポイゾニング攻撃の初期の研究 [4,5] では，悪性ユーザの評価データを作成するにあたり，評価するアイテムをランダムに選択して評価値もランダムに決定する，もしくは人気を操作したいアイテムを高く/低く評価する，といった直観的な方法が提案されていた．これに対して，2016 年の NIPS で提案された Li らの手法 [6] では，行列分解に基づく協調フィルタリングを想定し，悪性ユーザの評価データの選択を最適化問題として定義することで，評価データを攻撃の目的に応じて理論的に最適化している．しかしながら，いずれの検討もユーザのプライバシーの暴露を目的としていない．

本稿では，まず，協調フィルタリングに対するプライバシーの暴露を目的とした新たな攻撃として，ポイゾニングを用いた Model Inversion 攻撃を提案する．Model Inversion 攻撃は 2014 年と 2015 年に Fredrikson らにより提案された学習モデルの入出力を利用してプライバシーを暴露する攻撃であり，線形回帰や決定木，ニューラルネットワークへの適用が検討されている [7,8]．攻撃者は，攻撃対象のユーザに対する学習モデルの出力情報を事前に取得し，学習モデルを不正に利用することで入力値の候補の絞り込みを行い，入力情報に含まれていた当該ユーザのセンシティブな情報を復元する．本稿では，この概念を協調フィルタリングに応用し，ポイゾニング攻撃と組み合わせることで，推薦されたアイテムからユーザが過去に高く評価したアイテムを暴露する攻撃を提案する．そして，Model Inversion 攻撃を実現するにあたり，Li らの行列分解に基づく協調フィルタリングに対するポイゾニング攻撃 [6] を応用し，少数の悪性ユーザで最も効率的に攻撃を達成する評価データの導出方法を明らかにする．

1.1 本研究の貢献

本研究の貢献は以下の通りである．

- 協調フィルタリングに基づく推薦システムに対する新たな攻撃として，推薦されたアイテムからユーザが過去に高く評価したアイテムを暴露する Model Inversion 攻撃を提案する．本攻撃では，推薦システムにおいて悪性ユーザが意図的にアイテムを評価するポイゾニング攻撃を用いて，ユーザがセンシティブなアイテムを高く評価した際に，攻撃用の “おとり” アイテムを推薦されやすくすることで，センシティブなアイテムの推測を可能とする．さらに，本稿では，行列分解に基づく協調フィルタリングに対して上記のポイゾニングを少数の悪性ユーザで効率的に行うことを目的とし，Li らの手法 [6] に基づき悪性ユーザの評価データの選択を最適化問題として定義する．
- 上記の検討で定義した最適化問題の計算方法を明らかにし，ポイゾニング攻撃の具体的な攻撃アルゴリズムを提案する．本稿では，Li らの攻撃アルゴリズム [6] と同様に，射影付き勾配法を用いて悪性ユーザの評価データを導出するための最適化問題を解く．ただし，評価データを更新するための勾配については，Biggio らの手法 [9,10] を応用し，1 次の KKT 条件に基づき最適化問題の解の陰勾配を近似的に計算して算出する．

本稿の構成は以下の通りである．まず，2 章において，本稿で着目する協調フィルタリングに基づく推薦システムと，Li らにより提案されたポイゾニング攻撃の定義について述べる．次いで，3 章において，協調フィルタリングに対するポイゾニングを用いた Model Inversion 攻撃を提案し，4 章において，ポイゾニング攻撃の具体的な攻撃アルゴリズムについて述べる．5 章において，関連研究について概説する．6 章は本稿の結論である．

2. 準備

本稿で着目する協調フィルタリングに基づく推薦システムを定義するとともに，2016 年に Li らにより提案された行列分解に基づく協調フィルタリングに対するポイゾニング攻撃 [6] について述べる．

協調フィルタリングに基づく推薦システム． 推薦システムは，ユーザの評価データ（もしくは，購入履歴）に基づき，ユーザが高く評価する（もしくは，購入する）可能性の高い未評価（もしくは，未購入）のアイテムをユーザに推薦する．以下，同様の議論となるため，評価データを例として述べる．$\mathbf{M} \in \mathbb{R}^{m \times n}$ を m 人のユーザの n 個のアイテムに対する評価データからなる行列（評価行列）とする．$i \in [m]$，$j \in [n]$ に対して，$\mathbf{M}_{ij}$ は，i 番目のユーザによる j 番目のアイテムの評価を示す．ただし，一般的に，アイテム数 n はきわめて大きな値であることが多いため，いずれのユーザも少数のアイテムだけを評価することが想定される．このとき，評価行列 $\mathbf{M}$ は評価が未観測の要素（すなわち，欠損値）を多く含む行列となる．協調フィルタリングを用いた推薦システムでは，評価行列 $\mathbf{M}$ の欠損値を何らかの方法で補完し，復元した値を評価の予測値として使いアイテムを推薦する（たとえば，予測値が大きいアイテムから順に K 個のアイテムを提示する）．本稿では，Li らの議論 [6] と同様に，評価行列 $\mathbf{M}$ を低ランクの行列で近似できると仮定し，行列分解に基づき $\mathbf{M}$ の欠損値を補完する方法を考える．

$\Omega = \{(i,j) : \mathbf{M}_{ij} \text{ is observed}\}$ を評価行列 $\mathbf{M}$ において評価が観測された（すなわち，欠損値ではない）要素のインデックス集合とする．$k \ll \min(m,n)$ とし，評価行列 $\mathbf{M}$ は k ランクの行列 $\mathbf{X} \in \mathbb{R}^{m \times n}$ で近似できると仮定する．

このとき，評価行列 $\mathbf{M}$ の補完問題は以下の最適化問題として定義できる．

$$\min_{X\in\mathbb{R}^{m\times n}} ||\mathcal{R}_\Omega(\mathbf{M}-\mathbf{X})||_F^2,\ s.t.\ \mathrm{rank}(\mathbf{X})\le k \qquad (1)$$

$||\mathbf{A}||_F^2=\sum_{i,j}\mathbf{A}_{ij}^2$ は行列 $\mathbf{A}$ の 2 次のフロベニスノルムである．$[\mathcal{R}_\Omega(\mathbf{A})]_{ij}$ は，$(i,j)\in\Omega$ のときに $\mathbf{A}_{ij}$ をとり，そうでないときに 0 をとる関数である．しかしながら，k ランク以下の行列は凸集合でないため，式 (1) は凸最適化問題として解くことができない．このため，評価行列 $\mathbf{M}$ の補完問題は，Alternating Minimization [11] と呼ばれる以下の最適化問題に帰着させることで，近似的に解く．

$$\min_{\mathbf{U}\in\mathbb{R}^{m\times k},\mathbf{V}\in\mathbb{R}^{n\times k}} \{||\mathcal{R}_\Omega(\mathbf{M}-\mathbf{U}\mathbf{V}^\top)||_F^2 + 2\lambda_{\mathbf{U}}||\mathbf{U}||_F^2+2\lambda_{\mathbf{V}}||\mathbf{V}||_F^2\} \qquad (2)$$

$\mathbf{U}\in\mathbb{R}^{m\times k}$ と $\mathbf{V}\in\mathbb{R}^{n\times k}$ はそれぞれユーザとアイテムの因子行列である．$\lambda_{\mathbf{U}}>0$，$\lambda_{\mathbf{V}}>0$ は正則化パラメータである．式 (2) は $\mathbf{U}$ と $\mathbf{V}$ についての両凸最適化問題（Biconvex Optimization Problem）であり，Alternating Minimization アルゴリズム [11] を適用して解くことができる．$\mathbf{\Theta}_\lambda(\mathbf{M})=(\mathbf{U},\mathbf{V})$ を評価行列 $\mathbf{M}$ と正則化パラメータ $\lambda=(\lambda_{\mathbf{U}},\lambda_{\mathbf{V}})$ が与えられたときの式 (2) の最適解とすると，$\mathbf{M}$ の欠損値が補完された行列 $\widehat{\mathbf{M}}\in\mathbb{R}^{m\times n}$（以下，補完行列）は次式により与えられる．

$$\widehat{\mathbf{M}}(\mathbf{\Theta}_\lambda(\mathbf{M}))=\mathbf{U}\mathbf{V}^\top \qquad (3)$$

協調フィルタリングに対するポイゾニング攻撃． Li らにより提案された式 (2) の Alternating Minimization に対するポイゾニング攻撃 [6] について述べる．本攻撃では，既存ユーザの評価行列 $\mathbf{M}\in\mathbb{R}^{m\times n}$ が与えられたときに，$\mathbf{M}$ を利用して $m'=\alpha m$ 人の悪性ユーザの評価データからなる評価行列 $\widetilde{\mathbf{M}}\in\mathbb{R}^{m'\times n}$ を作成する．ただし，悪性ユーザはそれぞれ最大で B 個のアイテムを評価するとし，評価値の範囲は $[-\Lambda,\Lambda]$ とする．$(\widetilde{\mathbf{M}};\mathbf{M})$ を既存ユーザと悪性ユーザの評価データが混在した評価行列とすると，式 (2) の Alternating Minimization の最適解 $\mathbf{\Theta}'_\lambda$ は次式で表せる．

$$\begin{aligned}\mathbf{\Theta}'_\lambda(\widetilde{\mathbf{M}};\mathbf{M}) = \mathop{\arg\min}_{\mathbf{U},\widetilde{\mathbf{U}},\mathbf{V}} &||\mathcal{R}_\Omega(\mathbf{M}-\mathbf{U}\mathbf{V}^\top)||_F^2\\ &+ ||\mathcal{R}_{\widetilde{\Omega}}(\widetilde{\mathbf{M}}-\widetilde{\mathbf{U}}\mathbf{V}^\top)||_F^2\\ &+ 2\lambda_{\mathbf{U}}(||\widetilde{\mathbf{U}}||_F^2+||\mathbf{U}||_F^2)+2\lambda_{\mathbf{V}}||\mathbf{V}||_F^2 \qquad (4)\end{aligned}$$

$\widetilde{\Omega}$ は悪性ユーザの評価行列 $\widetilde{\mathbf{M}}$ において評価が観測された要素のインデックス集合とする．最適解 $\mathbf{\Theta}'_\lambda=(\mathbf{U},\widetilde{\mathbf{U}},\mathbf{V})$ は，既存ユーザと悪性ユーザの因子行列 $\mathbf{U}\in\mathbb{R}^{m\times k}$, $\widetilde{\mathbf{U}}\in\mathbb{R}^{m'\times k}$ と，アイテムの潜在因子行列 $\mathbf{V}\in\mathbb{R}^{n\times k}$ からなる．このとき，既存ユーザの評価行列 $\mathbf{M}$ の補完行列 $\widehat{\mathbf{M}}\in\mathbb{R}^{m\times n}$ は最適解 $\mathbf{\Theta}'_\lambda$ を用いて $\widehat{\mathbf{M}}(\mathbf{\Theta}'_\lambda(\widetilde{\mathbf{M}};\mathbf{M}))=\mathbf{U}\mathbf{V}^\top$ と表せる．そして，Li らのポイゾニング攻撃では，$R(\widehat{\mathbf{M}},\mathbf{M})$ をポイゾニング攻撃の効用を示す Utility 関数とし，以下の最適化問題を解くことで，攻撃目的に応じて最適な悪性ユーザの評価行列 $\widetilde{\mathbf{M}}^*$ を得る．

$$\widetilde{\mathbf{M}}^*=\mathop{\arg\max}_{\widetilde{\mathbf{M}}\in\mathbb{M}} R(\widehat{\mathbf{M}}(\mathbf{\Theta}'_\lambda(\widetilde{\mathbf{M}};\mathbf{M})),\mathbf{M}) \qquad (5)$$

ただし，$\mathbb{M}=\{\widetilde{\mathbf{M}}\in\mathbb{R}^{m'\times n}:|\widetilde{\Omega}_i|\le B,\max|\widetilde{\mathbf{M}}_{ij}|\le\Lambda\}$ は $\widetilde{\mathbf{M}}$ の実行可能領域である．また，$\widetilde{\Omega}_i$ は悪性ユーザの評価行列 $\widetilde{\mathbf{M}}$ の i 番目のユーザの評価データにおいて評価が観測された要素のインデックス集合である．文献 [6] において，Li らは，ポイゾニングにより推薦システムの推薦精度を著しく低下させる Availability Attack と，特定のアイテムの人気を向上（もしくは，低下）させる Integrity Attack を想定して，それぞれの Utility 関数を定義し，式 (5) の具体的な計算方法を提示している．しかしながら，どちらの攻撃もユーザのプライバシーの暴露を意図していない．そこで，本稿では，まず，ポイゾニングによりユーザのプライバシーを暴露する新たな攻撃を提案するとともに，この攻撃に適切な Utility 関数を定義する（3 章）．次いで，Li らの議論と同様に，新しく定義した Utility 関数を用いて式 (5) から悪性ユーザの評価行列の最適解 $\widetilde{\mathbf{M}}^*$ を算出する方法を明らかにする（4 章）．

3. 協調フィルタリングに対する Model Inversion 攻撃

本章では，協調フィルタリングに基づく推薦システムに対するユーザのプライバシーを暴露する新たな攻撃として，ポイゾニングを用いた Model Inversion 攻撃を提案する．また，ポイゾニングに用いる悪性ユーザの評価行列を 2 章で示した Li らの手法を応用して導出することを想定し，式 (5) の中で用いられているポイゾニング攻撃の効用を示す Utility 関数を定義する．

3.1 ポイゾニングを用いた Model Inversion 攻撃

本稿では，協調フィルタリングに対する Model Inversion 攻撃として，協調フィルタリングを用いて推薦されたアイテムから，ユーザが過去に高く評価したセンシティブなアイテムを暴露する攻撃を提案する．ただし，センシティブなアイテムとは成人向けのコンテンツ等，ユーザが一般的に高く評価したことを隠したいものとする．本攻撃では，推薦システムに対してポイゾニングを行うことで，攻撃者が用意した “おとり” アイテムとセンシティブなアイテムを意図的に関連付ける（すなわち，センシティブなアイテムを高く評価したときにおとりアイテムが推薦されるようにする）．これにより，攻撃者はユーザにおとりアイテムが推薦された際に，当該ユーザが関連付けしたセンシティブなアイテムを過去に高く評価したことを知ることができる．以下，攻撃のフレームワークについて述べる．

m 人の既存ユーザと n 個のアイテムからなる推薦システムを考える．$\mathcal{X} \subset [n]$ をセンシティブなアイテム集合，$\mathcal{Y} \subset [n] \backslash \mathcal{X}$ をユーザがセンシティブなアイテムを高く評価した際に意図的に推薦される “おとり” アイテムの集合とする．既存ユーザの評価行列 $\mathbf{M} \in \mathbb{R}^{m \times n}$ が与えられたとき，攻撃者は以下の手順により攻撃を実行する．

(1) **悪性ユーザの評価行列の作成**．推薦システムが評価行列 $\mathbf{M}$ の現在の補完行列 $\widehat{\mathbf{M}} \in \mathbb{R}^{m \times n}$ を更新した際に，センシティブなアイテム $x \in \mathcal{X}$ を高く評価したユーザに対しておとりアイテム $y \in \mathcal{Y}$ が推薦される補完行列に変更可能な悪性ユーザの評価行列 $\widetilde{\mathbf{M}}^* \in \mathbb{R}^{m' \times n}$ を作成する．

(2) **ポイゾニング**．攻撃対象の推薦システムにおいて，通常のユーザを装い，作成した評価行列 $\widetilde{\mathbf{M}}^*$ に基づきアイテムを評価する．この操作は $\widetilde{\mathbf{M}}^*$ の行毎にユーザを変えて実行し，m' 人分について同様の操作を繰り返す．上記の操作後に，推薦システムは内部で通常ユーザと悪性ユーザの評価データが混在する評価行列 $(\widetilde{\mathbf{M}}^*; \mathbf{M}) \in \mathbb{R}^{(m+m') \times n}$ に対して協調フィルタリングを適用し，$\mathbf{M}$ の補完行列 $\widehat{\mathbf{M}} \in \mathbb{R}^{m \times n}$ を更新する．

(3) **Model Inversion**．推薦システムが更新した補完行列 $\widehat{\mathbf{M}}$ に基づき攻撃対象のユーザにおとりアイテム $y \in \mathcal{Y}$ を推薦した際に，攻撃者はその事実を何らかの方法で観測し，ユーザがセンシティブなアイテム集合 $\mathcal{X}$ のいずれかのアイテムを高く評価したと推測する．

攻撃手順 (1) における悪性ユーザの評価行列 $\widetilde{\mathbf{M}}^*$ の具体的な作成方法としては，2 章で示した Li らのポイゾニング攻撃 [6] が協調フィルタリングに対するものの中で少数の悪性ユーザで最も効率的に攻撃を行うことができる手法であることから，本稿では Li らの手法を応用して実現する方法について考える．Li らの手法を応用するためには，式 (5) の Utility 関数を上記の攻撃手順 (1) で示した条件（センシティブなアイテム $x \in \mathcal{X}$ を高く評価したユーザに対しておとりアイテム $y \in \mathcal{Y}$ が推薦される）にしたがって定義する必要がある．次節において Utility 関数の具体的な定義について述べる．また，攻撃手順 (3) で述べた攻撃者がユーザに推薦されたアイテムを知る方法としては，ユーザのブログや SNS サイトがあげられる．近年の推薦システムには推薦された商品をブログや SNS サイトなどで紹介する機能がついているため，攻撃者はそれらの機能を利用して投稿された記事からユーザに推薦されたアイテムを知ることができる．ただし，推薦されたアイテムがセンシティブなものの場合，ユーザがその情報を自らブログや SNS サイトで暴露する可能性は低い．このため，本攻撃ではセンシティブでないおとりアイテム $y \in \mathcal{Y}$ を用意してセンシティブなアイテムとの関連付けを行う．

3.2 Utility 関数の定義

3.1 節で提案した Model Inversion 攻撃を Li らのポイゾニング攻撃 [6] を用いて実現するにあたり，2 章で示した式 (5) の Utility 関数 $R(\widehat{\mathbf{M}}, \mathbf{M})$ を定義する．協調フィルタリングにおいて，センシティブなアイテム $x \in \mathcal{X}$ を高く評価した際に，おとりアイテム $y \in \mathcal{Y}$ が推薦されるようにするためには，x の評価が高いときに，y の評価の予測値が高くなるようにしなければならない．また，センシティブなアイテム x の評価が低いときに，おとりアイテム y の評価の予測値が高いと，x を高く評価していないにもかかわらず，y が推薦される．この場合，攻撃者は推薦されたおとりアイテム y から，ユーザが x を高く評価したと推測するが，実際はそうでないため攻撃は失敗する．したがって，Utility 関数は，補完行列 $\widehat{\mathbf{M}}$ において，センシティブなアイテム $x \in \mathcal{X}$ とおとりアイテム $y \in \mathcal{Y}$ の評価が同様の値となるように定義する．ただし，ポイゾニングの前後で補完行列 $\widehat{\mathbf{M}}$ が著しく変化した場合，推薦精度が低下し，攻撃が検出される可能性がある．このため，Utility 関数 $R(\widehat{\mathbf{M}}, \mathbf{M})$ は，上記の条件に加えて，ポイゾニングの前後で補完行列 $\widehat{\mathbf{M}}$ が大きく変わらないように定義する．以下，具体的な定義について述べる．

$\overline{\mathbf{M}}(\mathbf{\Theta}_\lambda(\mathbf{M})) \in \mathbb{R}^{m \times n}$ をポイゾニング攻撃を行わなかった場合（すなわち，悪性ユーザの評価データを追加しなかった場合）の既存ユーザの評価行列 $\mathbf{M}$ の補完行列とする．ただし，$\mathbf{\Theta}_\lambda$ は式 (2) の最適解である．また，Ω^C を評価行列 $\mathbf{M}$ において評価が観測されなかった要素のインデックス集合とする．このとき，Model Inversion 攻撃用の Utility 関数 $R^{\mathsf{mi}}_{\mathcal{Y}}(\widehat{\mathbf{M}}, \mathbf{M})$ を次のように定義する．

$$R^{\mathsf{mi}}_{\mathcal{Y}}(\widehat{\mathbf{M}}, \mathbf{M}) = ||\mathcal{R}_{\Omega^C}(\widehat{\mathbf{M}} - \overline{\mathbf{M}})||_F^2 + \mu \sum_{i=1}^{m} \sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} ||\widehat{\mathbf{M}}_{ix} - \widehat{\mathbf{M}}_{iy}||_F^2 \quad (6)$$

$\mu > 0$ は重み係数である．また，Li らのポイゾニング攻撃では，悪性ユーザの評価行列 $\widetilde{\mathbf{M}}^*$ を Utility 関数 $R(\widehat{\mathbf{M}}, \mathbf{M})$ の最大解として定義していたが，Model Inversion 攻撃では，$\widetilde{\mathbf{M}}^*$ を次のように Utility 関数 $R^{\mathsf{mi}}_{\mathcal{Y}}(\widehat{\mathbf{M}}, \mathbf{M})$ の最小解として定義する．

$$\widetilde{\mathbf{M}}^* = \mathop{\arg\min}_{\widetilde{\mathbf{M}} \in \mathbb{M}} R^{\mathsf{mi}}_{\mathcal{Y}}(\widehat{\mathbf{M}}, \mathbf{M}) \quad (7)$$

式 (6) の第二項を最小化することで，ポイゾニング後の補完行列 $\widehat{\mathbf{M}}$ において，センシティブなアイテム $x \in \mathcal{X}$ とおとりアイテム $y \in \mathcal{Y}$ の評価値の差が小さくなり，センシティブなアイテムを高く評価した場合のみおとりアイテムが推薦される（低く評価した場合は推薦されない）．また，式 (6) の第一項を最小化することで，ポイゾニング前後の補完行列 $\overline{\mathbf{M}}$，$\widehat{\mathbf{M}}$ の差が小さくなり，推薦精度の低下を防ぐとともに，攻撃検知を回避する．

Algorithm 1 悪性ユーザの評価行列 $\widetilde{\mathbf{M}}$ の最適化

Input: Observed rating matrix $\mathbf{M} \in \mathbb{R}^{m \times n}$, parameters $\lambda = (\lambda_{\mathbf{U}}, \lambda_{\mathbf{V}})$, α, B, Λ, and $\{s_t\}_{t=1}^{\infty}$.
Initialize: Generate initial malicious rating matrix $\widetilde{\mathbf{M}}^{(0)} \in \mathbb{M}$ where ratings and rated items are uniformly sampled at random; t = 0.
while $\widetilde{\mathbf{M}}^{(t)}$ does not converge **do**
  Compute the optimal solution $\boldsymbol{\Theta}'_{\lambda}(\widetilde{\mathbf{M}}^{(t)}; \mathbf{M})$ using the alternative minimization algorithm.
  Compute gradient $\nabla_{\widetilde{\mathbf{M}}} R(\widehat{\mathbf{M}}, \mathbf{M})$.
  $\widetilde{\mathbf{M}}^{(t+1)} \leftarrow \mathsf{Proj}_{\mathbb{M}}\left(\widetilde{\mathbf{M}}^{(t)} - s_t \cdot \nabla_{\widetilde{\mathbf{M}}} R^{\mathsf{mi}}_{\mathcal{Y}}(\widehat{\mathbf{M}}, \mathbf{M})\right)$
  $t \leftarrow t + 1$
end while
Output: tth poison rating matrix $\widetilde{\mathbf{M}}^{(t)} \in \mathbb{M}$.

4. 攻撃アルゴリズム

本章では，3.2 節で定義した悪性ユーザの評価行列の最適解 $\widehat{\mathbf{M}}^*$ を実際に計算するにあたり，式 (7) の最適化問題を解くための最適化アルゴリズムのフレームワークを示すとともに，その具体的な計算方法について述べる．

4.1 悪性ユーザの評価行列の最適化

悪性ユーザの評価行列の最適解 $\widetilde{\mathbf{M}}^* \in \mathbb{R}^{m' \times n}$ は，Li らの手法 [6] と同様に，射影付き勾配法を式 (7) に適用して算出する．t 回目の繰り返しにおける $\widetilde{\mathbf{M}}^{(t)}$ の更新式を次のように定義する．

$$\widetilde{\mathbf{M}}^{(t+1)} = \mathsf{Proj}_{\mathbb{M}}\left(\widetilde{\mathbf{M}}^{(t)} - s_t \cdot \nabla_{\widetilde{\mathbf{M}}} R^{\mathsf{mi}}_{\mathcal{Y}}(\widehat{\mathbf{M}}, \mathbf{M})\right) \quad (8)$$

s_t は t 回目の更新におけるステップサイズである．$\mathbb{M}$ は $\widetilde{\mathbf{M}}$ の実行可能領域であり，2 章で示した定義と同様とする．$\mathsf{Proj}_{\mathbb{M}}(\cdot)$ は，実行可能領域 $\mathbb{M}$ への射影である．ただし，$\mathbb{M}$ は凸集合ではないため，悪性ユーザ毎に評価する B 個のアイテムをランダムに選択することで対応する．また，$\mathsf{Proj}_{\mathbb{M}}(\cdot)$ は，評価値が $\pm\Lambda$ を超えた場合に $\pm\Lambda$ と置き換える操作により容易に実現できる [6]．式 (8) を用いた悪性ユーザの評価行列 $\widetilde{\mathbf{M}}$ の最適化アルゴリズムのフレームワークを Algorithm 1 に示す．$\nabla_{\widetilde{\mathbf{M}}} R(\widehat{\mathbf{M}}, \mathbf{M})$ の具体的な計算方法については次節で述べる．

4.2 $\nabla_{\widetilde{\mathbf{M}}} R^{\mathsf{mi}}_{\mathcal{Y}}(\widehat{\mathbf{M}}, \mathbf{M})$ の計算

$\nabla_{\widetilde{\mathbf{M}}} R^{\mathsf{mi}}_{\mathcal{Y}}(\widehat{\mathbf{M}}, \mathbf{M})$ は，Chain Rule により，次のように変形して計算する．

$$\nabla_{\widetilde{\mathbf{M}}} R^{\mathsf{mi}}_{\mathcal{Y}}(\widehat{\mathbf{M}}, \mathbf{M}) = \nabla_{\widetilde{\mathbf{M}}} \boldsymbol{\Theta}'_{\lambda}(\widetilde{\mathbf{M}}, \mathbf{M}) \nabla_{\boldsymbol{\Theta}'} R^{\mathsf{mi}}_{\mathcal{Y}}(\widehat{\mathbf{M}}, \mathbf{M}) \quad (9)$$

4.2.1 節，4.2.2 節において，式 (9) の第二項と第一項の計算方法についてそれぞれ述べる．

4.2.1 $\nabla_{\boldsymbol{\Theta}'} R^{\mathsf{mi}}_{\mathcal{Y}}(\widehat{\mathbf{M}}, \mathbf{M})$ の計算

$\nabla_{\boldsymbol{\Theta}'} R^{\mathsf{mi}}_{\mathcal{Y}}(\widehat{\mathbf{M}}, \mathbf{M}) \in \mathbb{R}^{|\boldsymbol{\Theta}'|}$ は，Chain Rule により，次のように変形して計算する．

$$\nabla_{\boldsymbol{\Theta}'} R^{\mathsf{mi}}_{\mathcal{Y}}(\widehat{\mathbf{M}}, \mathbf{M}) = \left(\nabla_{\boldsymbol{\Theta}'} \widehat{\mathbf{M}}\right)\left(\nabla_{\widehat{\mathbf{M}}} R^{\mathsf{mi}}_{\mathcal{Y}}(\widehat{\mathbf{M}}, \mathbf{M})\right) \quad (10)$$

$\nabla_{\widehat{\mathbf{M}}} R^{\mathsf{mi}}_{\mathcal{Y}}(\widehat{\mathbf{M}}, \mathbf{M})$ は，式 (6) より，次式で表せる．

$$\frac{\partial R^{\mathsf{mi}}_{\mathcal{Y}}}{\partial \widehat{\mathbf{M}}_{ij}} = \begin{cases} \Pi + \mu \sum_{y \in \mathcal{Y}} 2(\widehat{\mathbf{M}}_{ij} - \widehat{\mathbf{M}}iy) & (\text{if } j \in \mathcal{X}) \\ \Pi + \mu \sum_{x \in \mathcal{X}} 2(\widehat{\mathbf{M}}_{ij} - \widehat{\mathbf{M}}ix) & (\text{if } j \in \mathcal{Y}) \\ \Pi & (\text{otherwise}) \end{cases} \quad (11)$$

ただし，$\Pi = 2(\widehat{\mathbf{M}}_{ij} - \overline{\mathbf{M}}_{ij}) \cdot I[(i,j) \notin \Omega]$ と定義する．$I[(i,j) \in \Omega]$ は，$(i,j) \notin \Omega$ のときに 1 をとり，そうでないときに 0 をとる指示関数である．次に，$\nabla_{\boldsymbol{\Theta}'} \widehat{\mathbf{M}} \in \mathbb{R}^{|\boldsymbol{\Theta}'| \times m \times n}$ は，次式で表せる [6]．

$$\frac{\partial \widehat{\mathbf{M}}_{ij}}{\partial \mathbf{U}_{lt}} = \mathbf{V}_{jt} \cdot I[i = l], \frac{\partial \widehat{\mathbf{M}}_{ij}}{\partial \mathbf{V}_{lt}} = \mathbf{U}_{it} \cdot I[j = l] \quad (12)$$

ただし，$I[i = l]$ は，$i = l$ のときに 1 をとり，そうでないときに 0 をとる指示関数である．

以上より，$\nabla_{\boldsymbol{\Theta}'} R^{\mathsf{mi}}_{\mathcal{Y}}(\widehat{\mathbf{M}}, \mathbf{M})$ は，$\nabla_{\widehat{\mathbf{M}}} R^{\mathsf{mi}}_{\mathcal{Y}}(\widehat{\mathbf{M}}, \mathbf{M})$ と $\nabla_{\boldsymbol{\Theta}'} \widehat{\mathbf{M}}$ を式 (11)，式 (12) を用いてそれぞれ計算し，最後にそれらの積をとることにより算出する．

4.2.2 $\nabla_{\widetilde{\mathbf{M}}} \boldsymbol{\Theta}'_{\boldsymbol{\lambda}}(\widetilde{\mathbf{M}}, \mathbf{M})$ の計算

$\nabla_{\widetilde{\mathbf{M}}} \boldsymbol{\Theta}'_{\lambda}(\widetilde{\mathbf{M}}, \mathbf{M})$ は，Li らの手法 [6] と同様に，式 (4) で表された $\boldsymbol{\Theta}'_{\lambda}(\widetilde{\mathbf{M}}; \mathbf{M})$ の最適化問題の KKT 条件を用いて算出する．当該最適化問題の KKT 条件を以下に示す．

$$\lambda_{\mathbf{U}} \mathbf{u}_i = \sum_{j \in \Omega_i} (\mathbf{M}_{ij} - \mathbf{u}_i^\top \mathbf{v}_j) \mathbf{v}_j \quad (13)$$

$$\lambda_{\mathbf{U}} \widetilde{\mathbf{u}}_i = \sum_{j \in \widetilde{\Omega}_i} (\widetilde{\mathbf{M}}_{ij} - \widetilde{\mathbf{u}}_i^\top \mathbf{v}_j) \mathbf{v}_j \quad (14)$$

$$\lambda_{\mathbf{V}} \mathbf{u}_j = \sum_{i \in \Omega'_j} (\mathbf{M}_{ij} - \mathbf{u}_i^\top \mathbf{v}_j) \mathbf{u}_j + \sum_{j \in \widetilde{\Omega}'_j} (\widetilde{\mathbf{M}}_{ij} - \widetilde{\mathbf{u}}_i^\top \mathbf{v}_j) \widetilde{\mathbf{u}}_i \quad (15)$$

ただし，$\mathbf{u}_i$ と $\widetilde{\mathbf{u}}_i$ はそれぞれ既存ユーザと悪性ユーザの因子行列 $\mathbf{U}$，$\widetilde{\mathbf{U}}$ の i 行目の k 次元ベクトル，$\mathbf{v}_j$ はアイテムの因子行列 $\mathbf{V}$ の j 行目の k 次元ベクトルである．$(\mathbf{x}^\top \mathbf{a})\mathbf{a} = (\mathbf{a}^\top \mathbf{x})\mathbf{a} = (\mathbf{a}\mathbf{a}^\top)\mathbf{x}$ より，式 (13)，式 (14)，ならびに式 (15) はそれぞれ次のように表せる．

$$\left(\lambda_{\mathbf{U}} + \sum_{j \in \Omega_i} \mathbf{v}_j \mathbf{v}_j^\top\right) \mathbf{u}_i = \sum_{j \in \Omega_i} \mathbf{M}_{ij} \mathbf{v}_j \quad (16)$$

$$\left(\lambda_{\mathbf{U}} + \sum_{j \in \widetilde{\Omega}_i} \mathbf{v}_j \mathbf{v}_j^\top\right) \widetilde{\mathbf{u}}_i = \sum_{j \in \widetilde{\Omega}_i} \widetilde{\mathbf{M}}_{ij} \mathbf{v}_j \quad (17)$$

$$\left(\lambda_{\mathbf{U}} + \sum_{j \in \Omega'_j} \mathbf{u}_i \mathbf{u}_i^\top + \widetilde{\mathbf{u}}_i \widetilde{\mathbf{u}}_j^\top\right) \mathbf{v}_j = \sum_{i \in \Omega'_j} \mathbf{M}_{ij} \mathbf{u}_i + \sum_{i \in \Omega'_j} \widetilde{\mathbf{M}}_{ij} \widetilde{\mathbf{u}}_i \quad (18)$$

したがって，$\frac{\partial \mathbf{u}_i(\widetilde{\mathbf{M}})}{\partial \widetilde{\mathbf{M}}_{ij}}, \frac{\partial \widetilde{\mathbf{u}}_i(\widetilde{\mathbf{M}})}{\partial \widetilde{\mathbf{M}}_{ij}}, \frac{\partial \mathbf{v}_j(\widetilde{\mathbf{M}})}{\partial \widetilde{\mathbf{M}}_{ij}} \in \mathbb{R}^k$ はそれぞれ次式を用いて計算する．

$$\frac{\partial \mathbf{u}_i(\widetilde{\mathbf{M}})}{\partial \widetilde{\mathbf{M}}_{ij}} = \mathbf{0} \tag{19}$$

$$\frac{\partial \widetilde{\mathbf{u}}_i(\widetilde{\mathbf{M}})}{\partial \widetilde{\mathbf{M}}_{ij}} = \left(\lambda_{\mathbf{U}}\mathbf{M}_k + \Sigma_{\mathbf{U}}^{(i)}\right)^{-1}\mathbf{v}_j \tag{20}$$

$$\frac{\partial \mathbf{v}_j(\widetilde{\mathbf{M}})}{\partial \widetilde{\mathbf{M}}_{ij}} = \left(\lambda_{\mathbf{V}}\mathbf{M}_k + \Sigma_{\mathbf{V}}^{(j)}\right)^{-1}\widetilde{\mathbf{u}}_i \tag{21}$$

ただし，$\Sigma_{\mathbf{U}}^{(i)}$ と $\Sigma_{\mathbf{V}}^{(j)}$ は，それぞれ次のように定義する．

$$\Sigma_{\mathbf{U}}^{(i)} = \sum_{j\in\widetilde{\Omega}_i}\mathbf{v}_j\mathbf{v}_j^\top \tag{22}$$

$$\Sigma_{\mathbf{V}}^{(j)} = \sum_{i\in\Omega'_j}(\mathbf{u}_i\mathbf{u}_i^\top + \widetilde{\mathbf{u}}_i\widetilde{\mathbf{u}}_i^\top) \tag{23}$$

5. 関連研究

ポイゾニング攻撃と Model Inverstion 攻撃は機械学習のセキュリティに関する代表的な研究事例である．以下，それぞれの攻撃ついて概説する．

ポイゾニング攻撃は，学習モデルの性能を著しく低下することを目的とし，訓練データに悪性データを混入する [6,9,10,12–24]．初期の研究では，PAC（Probably Approximately Correct）学習の観点から理論解析が行われた [12–14]．それらの研究では，攻撃者が訓練データの一部を任意のデータに置き換えることを想定し，操作されるデータの割合に基づき PAC 学習可能な条件を導出している．また，具体的な学習アルゴリズムを対象とした攻撃としては，2012 年と 2014 年に提案された Biggio らによるサポートベクタマシンに対する攻撃があげられる [9,10]．Biggio らは 1 次の KKT 条件に基づき最適化問題の解の陰勾配を近似的に計算することで悪性データを最適化し，少量の悪性データで学習モデルの性能を著しく低下させる攻撃アルゴリズムを提案している．Biggio らの手法はその後，Lasso 回帰 [16]，トピックモデル [17]，自己回帰モデル [18] に応用されている．さらに，文献 [19] において攻撃アルゴリズムの一般化フレームワークが提案されている．協調フィルタリングに対するポイゾニング攻撃としては，初期の研究において，特定アイテムの人気向上/低下を目的とし，特定アイテムを高く/低く評価する悪性ユーザを追加する攻撃が提案されている [4,5]．しかしながら，それらの研究は，協調フィルタリングの特定のアルゴリズムを想定したものではなく，学習アルゴリズムに応じて効率的に目的を達成する最適な悪性ユーザの作成方法等については十分に検討されていなかった．これに対して，2016 年に Li らにより，行列分解に基づく協調フィルタリングを想定し，上記の Biggio らの手法を応用して攻撃目的に応じて悪性ユーザを最適化する攻撃アルゴリズムが提案されている [6]．ただし，いずれの研究も本稿で提案した攻撃とは異なり，ユーザのプライバシーの暴露を意図としたものではない．

一方，Model Inversion 攻撃は，2014 年と 2015 年に，Fredrikson らにより提案された学習モデルの入出力を利用してユーザのプライバシーを暴露する攻撃である [7,8,25]．攻撃者は，攻撃対象のユーザに対する学習モデルの出力情報を事前に取得し，学習モデルを不正に利用することで入力値の絞り込みを行い，入力情報に含まれていた当該ユーザのセンシティブな情報を復元する．Fredrikson らは，2014 年に文献 [7] において本攻撃を線形回帰モデルに適用し，2015 年に文献 [8] において決定木やニューラルネットワーク等の非線形モデルに適用している．いずれの検討においても，実データによる評価実験を通して，従来の事前分布だけを用いてセンシティブな情報を復元する方法よりも，攻撃性能が向上することが示されている．また，2016 年に，Wu らにより，Model Inversion 攻撃の定式化が行われている [25]．本稿で提案した攻撃は Model Inversion 攻撃の概念を協調フィルタリグに応用したものであるが，従来の攻撃 [7,8,25] とは異なりポイゾニング攻撃と組み合わせることにより実現する．このため，本攻撃は Model Inversion 攻撃の中でも異なるラインの新しい攻撃といえる．

6. おわりに

本稿では，協調フィルタリングに基づく推薦システムに潜在するプライバシーリスクの顕在化を目的として，推薦システムにおいて（悪性）ユーザが不正にアイテムを評価するポイゾニングを用いてユーザのプライバシーを暴露する新たな攻撃を提案した．本攻撃は，Fredrikson らにより提案された Model Inversion 攻撃の概念を応用したものであり，推薦システムがユーザに提示したアイテムから当該ユーザが過去に高く評価したアイテムを暴露する．また，本稿では，Li らにより提案された行列分解に基づく協調フィルタリングに対するポイゾニング攻撃を応用し，悪性ユーザの評価データを最適化することで，上記の攻撃を少数の悪性ユーザだけで効率的に実現するための攻撃アルゴリズムを明らかにした．

今後の課題としては，まず，実データを用いた提案手法の評価があげられる．また，本攻撃では，攻撃者が協調フィルタリングで用いる既存ユーザの評価データをすべて入手できることを想定しているが，実際にすべての評価データを入手できるケースは稀である．このため，今後は，一部の評価データのみ入手できる，もしくは，推薦システムをブラックボックスとして利用する状況を想定した新たな攻撃についても検討する．

参考文献

[1] Su, X. and Khoshgoftaar, T. M.: A Survey of Collaborative Filtering Techniques, *Advances in Artificial Intelligence*, Vol. 2009, No. 4, pp. 1–19 (2009).

[2] Linden, G., Smith, B. and York, J.: Amazon.com Recommendations: Item-to-Item Collaborative Filtering,

IEEE Internet Computing, Vol. 7, No. 1, pp. 76–80 (2003).

[3] Koren, Y., Bell, R. and Volinsky, C.: Matrix Factorization Techniques for Recommender Systems, *Computer*, Vol. 42, No. 8, pp. 30–37 (2009).

[4] O'Mahony, M. P., Hurley, N., Burke, R. and Hurley, N.: Promoting Recommendations: An Attack on Collaborative Filtering, *Proceedings of the 45th annual ACM symposium on Theory of computing (STOC 2013)*, pp. 494–503 (2002).

[5] Mobasher, B., Burke, R., Bhaumik, R. and Williams, C.: Effective attack models for shilling item-based collaborative filtering systems year = 2005,, *Proceedings of the 2005 WebKDD Workshop in conjuction with ACM SIGKDD 2015*, pp. 1–8.

[6] Li, B., Wang, Y., Singh, A. and Vorobeychik, Y.: Data Poisoning Attacks on Factorization-Based Collaborative Filtering, *Proceedings of the 3rd Neural Information Processing Systems (NIPS 2016)*, pp. 1–13 (2016).

[7] Fredrikson, M., Lantz, E. and Jha, S.: Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures, *the Proceedings of the 23rd USENIX Security Symposium (USENIX 2014)*, pp. 17–32 (2014).

[8] Fredrikson, M., Jha, S. and Ristenpart, T.: Model Inversion Attacks that Exploit Confidence Information and Basic Countermeasures, *Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security (CCS 2015)*, pp. 1322–1333 (2015).

[9] Biggio, B., Nelson, B. and Laskov, P.: Poisoning Attacks against Support Vector Machines, *Proceedings of the 29th International Conference on Machine Learning (ICML 2012)* (2012).

[10] Biggio, B., Fumera, G. and Roli, F.: Security Evaluation of Pattern Classifiers under Attack, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 26, No. 4, pp. 984–996 (2014).

[11] Jain, P., Netrapalli, P. and Sanghavi, S.: Low-rank Matrix Completion using Alternating Minimization, *Proceedings of the 45th annual ACM symposium on Theory of computing (STOC 2013)*, pp. 665–674 (2013).

[12] Kearnsy, M. and Liz, M.: Learning in the Presence of Malicious Errors, No. 4, pp. 807–837 (2012).

[13] Auer, P. and Cesa-Bianchi, N.: On-line learning with malicious noise and the closure algorithm, Vol. 23, No. 1, pp. 83–99 (1998).

[14] Bshouty, N. H., Eiron, N. and Kushilevitz, E.: PAC learning with nasty noise, *Proceedings of the 10th International Conference on Algorithmic Learning Theory (AAL 1999)*, pp. 206–218 (1999).

[15] Dalvi, N., Domingos, P., Mausam, Sanghai, S. and Verma, D.: Adversarial classification, *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2004)*, pp. 99–108 (2004).

[16] Xiao, H., Biggio, B., Brown, G., Fumera, G., Eckert, C. and Roli, F.: Is Feature Selection Secure against Training Data Poisoning?, *Proceedings of the 32nd International Conference on Machine Learning (ICML 2015)*, pp. 1689–1698 (2015).

[17] Mei, S. and Zhu, X.: The Security of Latent Dirichlet Allocation, *Proceedings of the 18th International Conference on Artificial Intelligence and Statistics (AISTATS 2015)*, pp. 681–689 (2015).

[18] Alfeld, S., Zhu, X. and Barford, P.: Data poisoning attacks against autoregressive models, *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI 2016)*, pp. 1452–1458 (2016).

[19] Mei, S. and Zhu, X.: Using machine teaching to identify optimal training-set attacks on machine learners, *Proceedings of the 29th AAAI Conference on Artificial Intelligence (AAAI 2015)*, pp. 2871–2877 (2015).

[20] Lowd, D. and Meek, C.: Adversarial Learning, *Proceedings of the 11th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2005)*, pp. 641–647 (2005).

[21] Barreno, M., Nelson, B., Sears, R., Joseph, A. D. and Tygar, J. D.: Can machine learning be secure?, *Proceedings of the 2006 ACM Symposium on Information, computer and communications security (ASIACCS 2006)*, pp. 16–25 (2006).

[22] Huang, L., Joseph, A. D., Nelson, B., Rubinstein, B. I. P. and Tygar, J. D.: Adversarial Machine Learning, *In Proceedings of 4th ACM Workshop on Artificial Intelligence and Security (AISec 2011)* (2011).

[23] Li, B. and Vorobeychik, Y.: Feature cross-substitution in adversarial classification, *Proceedings of the 27th International Conference on Neural Information Processing Systems*, pp. 2087–2095 (2014).

[24] Li, B. and Vorobeychik, Y.: Scalable optimization of randomized operational decisions in adversarial classification settings, *Proceedings of the 18th International Conference on Artificial Intelligenceand Statistics*, pp. 599–607 (2015).

[25] Wu, X., Jha, M. F. S. and Naughton, J. F.: A Methodology for Formalizing Model-Inversion Attacks, *Proceedings of the 29th IEEE Computer Security Foundations Symposium (CSF 2016)*, pp. 355–370 (2016).