攻撃者の条件を緩和した推薦システムに対するModel Inversion攻撃

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：ユーザの行動履歴に基づきお薦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．推薦システムに対するプライバシー暴露の攻撃としては，CSS 2017 において，著者らにより提案された推薦アイテムから過去にセンシティブなアイテムを高く評価したことを暴露する Model Inversion 攻撃がある．しかしながら，本攻撃は，計算量や攻撃者の前提知識の観点から非現実的な手法であった．そこで，本稿では，それらの条件を緩和した，より実用的な Model Inversion 攻撃を提案するとともに，実データを用いた評価実験を通して，提案手法の実現性を明らかにする．

キーワード：Model Inversion 攻撃，推薦システム，協調フィルタリング，データポイゾニング，プライバシー暴露

Practical Model Inversion Attacks for Recommender Systems

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

Abstract: Recommender systems using collaborative filtering (CF) have become increasingly popular in recent years. In CSS 2017, we proposed a model inversion attack on factorization-based CF, which enables us to disclose the past sensitive items highly rated by a user. However, that attack was not unfeasible due to its high computational cost and an unrealistic assumption about adversary's knowledge. In this paper, we further investigate a more practical model inversion attack, and validate its effectiveness through experiments using an actual dataset.

Keywords: model inversion attacks, recommender systems, collaborative filtering, data poisoning, privacy exposure

1. はじめに

ユーザの行動履歴に基づき選好性の高いアイテムを提示する推薦システムは，Amazon や Netflix などの昨今の EC サービスにおいて欠かせない存在となっている [1–3]．それらのシステムの多くは，ユーザらによる各種アイテムに対する評価データを行列で表現し（以下，評価行列），協調フィルタリングと呼ばれる手法を用いて評価行列において評価が未観測のアイテムの評価値を予測することで推薦アイテムを決定する．

また，近年，ソーシャルネットワークの普及が推薦システムのさらなる発展に重要な役割を担っている [4]．たとえば，ユーザが Twitter や Facebook などのソーシャルネットワークを通して，商品に対する評価や推薦されたアイテムなどを共有しながら，共通の関心を持つ他のユーザと交流を深めることで，EC サービスの宣伝効果が向上する．他にも，過去に訪問したスポットからユーザが関心を持つ可能性の高い未訪問のスポットを推薦する POI（Point Of Interest）推薦システム [5] と，ソーシャルネットワークが連携することで，レストランやカフェなどの実店舗への集客拡大が期待できる．

推薦システムは今後もより一層の発展が期待されるが，成人向けコンテンツや医薬品などのセンシティブなアイテムに関する情報を取り扱うことや，上述したように第三者が得られる情報も様々であることから，プライバシー

[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

への関心が非常に高まっている．推薦システムのプライバシーリスク，特に外部の攻撃者を想定した研究事例としては，Ramakrishnan らと Calandrino らによる研究事例がある [6,7]．Ramakrishnan らは，文献 [6] において，推薦システムが他のユーザの評価データを利用して推薦するアイテムを決定していることから，攻撃者に対して推薦されたアイテムから他のユーザが過去に評価したアイテムを暴露できることを明らかにした．一方，Calandrino らは，文献 [7] において，アイテムベースの協調フィルタリングを用いた推薦システムを対象として，ユーザが過去に評価した情報を不正に入手し，その情報と推薦システムの内部情報の変化から，当該ユーザが新たに評価したアイテムを推測できることを明らかにした．

上述したように，攻撃者が得られる情報が拡大しつつあるが，外部の攻撃者を想定した研究事例は非常に少ない．推薦システムの潜在的なプライバシーリスクを明らかにするためには，さらなる検討が必要不可欠であると考えられる．そこで，本稿では，行列分解に基づく協調フィルタリングを用いた推薦システムを対象とし，CSS 2017 において著者らにより提案された，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 Model Inversion 攻撃 [8] に着目する．本攻撃では，Li らにより提案されたデータポイゾニングと呼ばれる手法 [9] を応用し，推薦システムに悪性ユーザの評価行列を追加することで，センシティブなアイテムと “おとり” アイテムを意図的に関連付ける．しかしながら，従来の Model Inversion 攻撃では，悪性ユーザの評価行列を作成する際の計算量は $O((m+m')^2n^2k)$ であった．m, m', n, k はそれぞれ推薦システムの既存ユーザ数，悪性ユーザ数，アイテム数，評価行列を行列分解した際に得られる因子行列の列数を表す．推薦システムのユーザ数やアイテム数は一般的に膨大であり（たとえば，Amazon の場合，ユーザ数は数千万台，アイテム数は数百万台に及ぶ [1]），m や n はきわめて大きな値を取るため，上記の計算量の場合，攻撃を実現するためには多大な計算リソースが必要となる．また，従来手法では，攻撃者が事前に既存ユーザの評価行列を入手できることを仮定していたが，推薦システムが一般的に既存ユーザの評価行列（$m \times n$ 行列）を公開することはない．したがって，著者らにより提案された従来手法は，非現実的な攻撃であったと考えられる．

貢献． 本稿では，行列分解に基づく協調フィルタリングを用いた推薦システムに着目し，より実用的な Model Inversion 攻撃を提案する．以下，本稿の具体的な貢献を示す．

- 計算量および攻撃者の知識に関する条件を緩和するデータポイゾニング用の評価行列 $\mathbf{M}'$ の作成方法を明らかにする（4 章）．提案手法の計算量は $O(m'n^2k)$ であり，従来手法の計算量 $O((m+m')^2n^2k)$ と比較すると，計算コストが大幅に改善されることが分かる．また，攻撃者は事前に既存ユーザの評価行列 $\mathbf{M}$ を入手する必要はなく，$\mathbf{M}$ を行列分解した際に得られるアイテム因子行列 $\mathbf{V}$ のみから $\mathbf{M}'$ を計算する．$\mathbf{V}$ はプライバシーに配慮する必要のない情報であるため推薦システムが公開する可能性があり [10]，また他の類似のデータセットからも疑似的に作成できるため [11]，$\mathbf{M}$ を利用するよりもはるかに現実的である．
- 提案手法を POI（Point Of Interest）推薦システムに適用し，実データを用いた評価実験を通して，Model Inversion 攻撃が現実的に実現できることを明らかにする（5 章）．

2. 準備

本稿で着目する行列分解に基づく協調フィルタリング [3] と，Li らにより提案されたデータポイゾニング攻撃 [9] について述べる．

2.1 共通記号

協調フィルタリングを用いた推薦システムでは，複数のユーザの評価データからなる評価行列を利用して未評価のアイテムに対するユーザの関心度を予測し，予測結果に基づきアイテムを推薦する．m を推薦システムのユーザ数，n を評価対象のアイテム数とする．$\mathbf{M} \in \mathbb{R}^{m \times n}$ を評価行列とする．$\mathbf{M}$ の (i,j) 番目の要素 $\mathbf{M}_{i,j}$ は，i 番目のユーザによる j 番目のアイテムに対する評価値を示す．

1 章で述べたように，アイテム数 n は一般的に膨大であり，ユーザは少数のアイテムのみを評価するため，評価行列 $\mathbf{M}$ は評価が未観測（評価値のない）の要素を多く含む．そこで，$\mathbf{M}$ において評価が観測された（評価値のある）要素の集合を $\Omega = \{(i,j) : \mathbf{M}_{i,j}$ は評価が観測された要素 $\}$ と表し，Ω によるマスク関数 $\mathcal{R}_\Omega : \mathbb{R}^{m \times n} \to \mathbb{R}^{m \times n}$ を定義する．行列 $\mathbf{A} \in \mathbb{R}^{m \times n}$ が与えられたとき，$\mathcal{R}_\Omega(\mathbf{A})$ の (i,j) 番目の要素 $[\mathcal{R}_\Omega(\mathbf{A})]_{i,j}$ は次式で表せる．

$$[\mathcal{R}_\Omega(\mathbf{A})]_{i,j} = \begin{cases} \mathbf{A}_{i,j} & \text{if } (i,j) \in \Omega \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

$N\ (\leq n)$ を推薦システムが各ユーザに推薦するアイテムの個数とする．

2.2 行列分解に基づく協調フィルタリング

行列分解に基づく協調フィルタリングでは，評価行列 $\mathbf{M}$ の未観測の要素を行列分解を用いて補完し，補完行列 $\widehat{\mathbf{M}}$ に基づきユーザにアイテムを推薦する．具体的には，$k \ll \min(m,n)$ とし，以下に示す k ランク行列分解により，評価行列 $\mathbf{M}$ を補完する．

$$\widehat{\mathbf{M}} = \mathbf{U}\mathbf{V}^\top \tag{2}$$

ただし，$\mathbf{U} \in \mathbb{R}^{m \times k}$ および $\mathbf{V} \in \mathbb{R}^{n \times k}$ は，それぞれユー

ザの潜在的な特徴を表すユーザ因子行列およびアイテムの潜在的な特徴を表すアイテム因子行列を表す．また，本稿では，ユーザ因子行列 $\mathbf{U}$ とアイテム因子行列 $\mathbf{V}$ の集合を $\Theta = \{\mathbf{U}, \mathbf{V}\}$ と表す．$\Theta = \{\mathbf{U}, \mathbf{V}\}$ を評価行列 $\mathbf{M}$ の潜在モデルと呼ぶ．

評価行列 $\mathbf{M}$ から潜在モデル Θ を導出する方法としては，様々な手法が提案されているが，本稿では幅広く利用されている交互最小化手法（Alternating Minimization）[3] に着目する．交互最小化手法では，以下に示す最適化問題の近似解を計算することで，潜在モデル Θ を導出する．

$$\min_{\mathbf{U}\in\mathbb{R}^{m\times k},\mathbf{V}\in\mathbb{R}^{n\times k}} \{||\mathcal{R}_{\Omega}(\mathbf{M} - \mathbf{U}\mathbf{V}^{\top})||_F^2 + 2\lambda||\Omega||_F^2\} \quad (3)$$

ただし，$||\cdot||$ は行列の 2 次のフロベニウスノルムを表す．すなわち，行列 $\mathbf{A}$ が与えられたとき，$||\mathbf{A}||_F^2 = \sum_{i,j}\mathbf{A}_{i,j}^2$ と表せる．$\lambda \geq 0$ は正則化パラメータを表す．

式 (3) は非凸最適化問題であるため，厳密解を求めることは一般的に困難である．しかしながら，$\mathbf{U}$ もしくは $\mathbf{V}$ のいずれかを固定し，もう一方のみを変数とみなすと，式 (3) は凸最適化問題となる．このため，交互最小化手法では，まず，$\mathbf{V}$ を固定し，$\mathbf{U}$ について式 (3) を解くことで，$\mathbf{U}$ を更新する．次いで，$\mathbf{U}$ を固定し， $\mathbf{V}$ について式 (3) を解くことで， $\mathbf{V}$ を更新する．そして，上記の更新処理を収束するまで繰り返し行うことで，潜在モデル Θ を近似的に計算する．補完行列 $\widehat{\mathbf{M}}$ は，得られた潜在モデル Θ の近似解と式 (2) より計算する．

推薦システムは，評価行列 $\mathbf{M}$ において評価が未観測かつ補完行列 $\widehat{\mathbf{M}}$ において評価の予測値が高いアイテムをユーザに推薦する．たとえば，N 個のアイテムを推薦する場合，ユーザ毎に $\mathbf{M}$ において評価が未観測のアイテムを抽出し，その中で $\widehat{\mathbf{M}}$ における評価の予測値がトップ N のアイテムを推薦する．

2.3 データポイゾニング攻撃

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データポイゾニング攻撃 [9] について述べる．$m' \in \mathbb{N}$ を悪性ユーザ数とし，$\mathbf{M}' \in \mathbb{R}^{m'\times n}$ を悪性ユーザの評価行列とする．また，$\mathbf{M}'$ において評価が観測された要素の集合を $\Omega' = \{(i,j) : \mathbf{M}'_{i,j}$は評価が観測された要素 $\}$ と表す．攻撃者は，対象の推薦システムにおいて m' 人分のユーザアカウントを作成し，事前に作成した $\mathbf{M}'$ にしたがってアイテムを評価することで，既存ユーザの評価行列 $\mathbf{M}$ に $\mathbf{M}'$ を追加する．

データポイゾニング後，推薦システムは潜在モデル Θ を更新し，$\mathbf{M}$ および $\mathbf{M}'$ における未観測の要素を次のように補完する．

$$\begin{bmatrix} \widehat{\mathbf{M}}^* \\ \widehat{\mathbf{M}}' \end{bmatrix} = \begin{bmatrix} \mathbf{U}^* \\ \mathbf{U}' \end{bmatrix} \mathbf{V}^{*\top} \quad (4)$$

ただし，$\widehat{\mathbf{M}}^* \in \mathbb{R}^{m\times n}$ および $\widehat{\mathbf{M}}' \in \mathbb{R}^{m'\times n}$ はそれぞれデータポイゾニング後の $\mathbf{M}$ および $\mathbf{M}'$ の補完行列を表す．$\mathbf{U}^* \in \mathbb{R}^{m\times k}$ および $\mathbf{U}' \in \mathbb{R}^{m'\times k}$ はそれぞれデータポイゾニング後の既存ユーザに対するユーザ因子行列および悪性ユーザに対するユーザ因子行列を表す．$\mathbf{V}^* \in \mathbb{R}^{n\times k}$ はデータポイゾニング後のアイテム因子行列を表す．データポイゾニング後の潜在モデルを $\Omega^* = \{\mathbf{U}^*, \mathbf{U}', \mathbf{V}^*\}$ と表す．Ω^* は 2.2 節と同様の方法を用いて $\mathbf{M}$ および $\mathbf{M}'$ から導出する．すなわち，以下に示す最適化問題を解くことで近似的に計算する．

$$\min_{\Theta^*}\{||\mathcal{R}_{\Omega}(\mathbf{M} - \mathbf{U}^*\mathbf{V}^{*\top})||_F^2 + ||\mathcal{R}_{\Omega'}(\mathbf{M}' - \mathbf{U}'\mathbf{V}^{*\top})||_F^2 + 2\lambda(||\Theta^*||_F^2\} \quad (5)$$

Li らのデータポイゾニング攻撃では，$\widehat{\mathbf{M}}^*$ と $\widehat{\mathbf{M}}$ を入力として，攻撃の効用度を出力する効用関数 $R : \mathbb{R}^{m\times n} \times \mathbb{R}^{m\times n} \to \mathbb{R}$ を用いて，悪性ユーザの評価行列 $\mathbf{M}'$ を作成する．たとえば，Li らは推薦システムの精度を劣化させるための効用関数や，特定のアイテムが推薦されやすくなるための効用関数を定義している（詳細は文献 [9] を参照）．具体的には，攻撃者が既存ユーザの評価行列 $\mathbf{M}$ を入手できると仮定し，勾配法を用いて効用関数 $R(\widehat{\mathbf{M}}^*, \widehat{\mathbf{M}})$ を最大化する $\mathbf{M}'$ を計算することで，$\mathbf{M}'$ の最適化を図る．以下，$\mathbf{M}'$ の計算方法について概説する．

各悪性ユーザが評価するアイテムの最大数を $b_{\max} \in \mathbb{N}$ とし， 評価値の範囲を $[-\Lambda, \Lambda]$ とする．ただし，Λ は正の実数を表す．$\mathbb{M}_1$ を以下に示す $\mathbf{M}'$ の実行可能領域とする．

$$\mathbb{M}_1 = \{\mathbf{M}' | \forall i \in [m'], \forall j \in [n], \mathbf{M}'_{i,j} \in [-\Lambda, \Lambda]\} \quad (6)$$

まず，悪性ユーザ毎に評価する b_{max} 個のアイテムをランダムに選択する．そして，射影付き勾配法（Projected Gradient Ascent）を用いて効用関数 $R(\widehat{\mathbf{M}}^*, \widehat{\mathbf{M}})$ を最大化する $\mathbf{M}'$ を近似的に計算する．勾配法の t 回目の繰り返し処理における $\mathbf{M}'$ の更新式を以下に示す．

$$\mathbf{M}'^{(t+1)} = \mathrm{Proj}_{\mathbb{M}_1}\left(\mathbf{M}'^{(t)} + s_t \cdot \nabla_{\mathbf{M}'} R(\widehat{\mathbf{M}}^{*(t)}, \widehat{\mathbf{M}})\right) \quad (7)$$

ただし，s_t は t 番目のステップサイズを表す．$\nabla_{\mathbf{M}'} R(\widehat{\mathbf{M}}^*, \widehat{\mathbf{M}}) \in \mathbb{R}^{m'\times n}$ は $\mathbf{M}'$ における $R(\widehat{\mathbf{M}}^*, \widehat{\mathbf{M}})$ の勾配を表し，$\mathrm{Proj}_{\mathbb{M}_1}$ は $\mathbb{M}_1$ への射影を表す．

連鎖律を用いることで，$\nabla_{\mathbf{M}'} R(\widehat{\mathbf{M}}^*, \widehat{\mathbf{M}})$ は次式で表せる．

$$\nabla_{\mathbf{M}'} R(\widehat{\mathbf{M}}^*, \widehat{\mathbf{M}}) = (\nabla_{\mathbf{M}'}\Theta^*)(\nabla_{\Theta^*}\widehat{\mathbf{M}}^*)(\nabla_{\widehat{\mathbf{M}}^*} R(\widehat{\mathbf{M}}^*, \widehat{\mathbf{M}})) \quad (8)$$

ただし，$\nabla_{\mathbf{M}'}\Theta^* \in \mathbb{R}^{(m'\times n)\times|\Theta^*|}$，$\nabla_{\Theta^*}\widehat{\mathbf{M}}^* \in \mathbb{R}^{|\Theta^*|\times(m\times n)}$，$\nabla_{\widehat{\mathbf{M}}^*} R(\widehat{\mathbf{M}}^*, \widehat{\mathbf{M}}) \in \mathbb{R}^{m\times n}$ である．$\nabla_{\mathbf{M}'}\Theta^*$，$\nabla_{\Theta^*}\widehat{\mathbf{M}}^*$，$\nabla_{\widehat{\mathbf{M}}^*} R(\widehat{\mathbf{M}}^*, \widehat{\mathbf{M}})$ は $\mathbf{M}$ の知識を利用して計算できる．また，式 (8) の計算量は $O(|\Theta^*| \cdot mn + m'n \cdot |\Theta^*|) = O((m+m')^2 n^2 k)$ である．

3. 推薦システムに対する Model Inversion 攻撃

本章では，CSS 2017 において著者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を用いた推薦シス

テムに対する Model Inversion 攻撃 [8] について述べる．3.1 節において，攻撃手法の概要を示し，3.2 節において，Model Inversion 攻撃のための従来の効用関数を示す．3.3 節において従来手法の問題点を示す．

3.1 概要

推薦システムに対する Model Inversion 攻撃 [8] では，2.3 節で示した Li らのデータポイゾニング攻撃を応用し，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．攻撃者は，悪性ユーザの評価行列 $\mathbf{M}'$ を既存ユーザの評価行列 $\mathbf{M}$ に追加することで，センシティブなアイテムと事前に用意しておいた “おとり” アイテムを意図的に関連付ける．これにより，センシティブなアイテムを高く評価したユーザに対しておとりアイテムが推薦される可能性が高まるため，推薦されたおとりアイテムから過去に高く評価したセンシティブなアイテムを推測できる．以下，本攻撃のフレームワークを示す．

おとりアイテムの選定． 成人向けコンテンツや医薬品など，暴露したいセンシティブなアイテムを選択するとともに，関連付けるおとりアイテムを選択する．おとりアイテムは，文芸小説や日常品など，一般的に推薦されたことを隠す必要のないものとする．本稿では，センシティブなアイテムの集合を $\mathcal{X} \subseteq [n]$，“おとり” アイテムの集合を $\mathcal{Y} \subseteq [n] \setminus \mathcal{X}$ と表す．おとりアイテムの選び方としては，たとえば，非センシティブなアイテム $[n] \setminus \mathcal{X}$ からランダムに選択する方法がある．

データポイゾニング． 既存ユーザの評価行列 $\mathbf{M}$ において，いずれかのユーザがセンシティブなアイテム $\mathcal{X}$ を高く評価したと仮定する．攻撃対象の推薦システムにおいて悪性ユーザのアカウントを作成し，評価行列 $\mathbf{M}'$ にしたがってアイテムを評価する．ただし，悪性ユーザの評価行列 $\mathbf{M}'$ は，$\mathbf{M}$ に $\mathbf{M}'$ を追加した際に，センシティブなアイテムとおとりアイテムが関連付くように設計する．$\mathbf{M}'$ の具体的な作成方法については 3.2 節で述べる．

Model Inversion． データポイゾニング後，推薦システムは潜在モデルを Θ から Θ^* に更新し，補完行列 $\widehat{\mathbf{M}}^*$ に基づきユーザに N 個のアイテムを推薦する．データポイゾニングにより，センシティブなアイテム $\mathcal{X}$ とおとりアイテム $\mathcal{Y}$ の間の類似度が高まっているため，センシティブなアイテムを高く評価したユーザに対して高い確率で 1 つ以上のおとりアイテムが推薦される．1 章で述べたように，ユーザはソーシャルネットワークなどで推薦されたアイテムを公開する可能性がある．攻撃者は，ユーザが公開した推薦アイテムを観測し，それがおとりアイテムであれば，当該ユーザをセンシティブなアイテム $\mathcal{X}$ を高く評価した可能性のあるユーザの候補として記録する．そして，事前に閾値 $l \in [|\mathcal{Y}|]$ を設定しておき，候補ユーザに対しておとりアイテムが l 個以上推薦されたことを観測したとき，当該ユーザが $\mathcal{X}$ の中のいずれかのアイテムを高く評価したと判定する．

3.2 悪性ユーザの評価行列の作成方法

Model Inversion 攻撃を実現するための悪性ユーザの評価行列 $\mathbf{M}'$ の作成方法について述べる．

3.1 節で述べたように，$\mathbf{M}'$ はセンシティブなアイテム $\mathcal{X}$ とおとりアイテム $\mathcal{Y}$ を関連付けるように設計する必要がある．また，推薦精度の低下により，攻撃を検知される恐れがあるため，$\mathbf{M}'$ を追加することで推薦精度が著しく低下しないことが望まれる．

上記の条件を考慮し，著者らは，$\mathbf{M}'$ を導出するための効用関数 $R_1^{\mathsf{mi}} : \mathbb{R}^{m\times n} \times \mathbb{R}^{m\times n} \to \mathbb{R}$ を次のように定義した [8]．

$$R_1^{\mathsf{mi}}(\widehat{\mathbf{M}}^*, \widehat{\mathbf{M}}) = \sum_{i=1}^{m} \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} |\widehat{\mathbf{M}}^*_{i,x} - \widehat{\mathbf{M}}^*_{i,y}|^2 + \mu ||\mathcal{R}_{\Omega^C}(\widehat{\mathbf{M}}^* - \widehat{\mathbf{M}})||_F^2 \qquad (9)$$

ただし，Ω^C は評価行列 $\mathbf{M}$ における評価が未観測の要素の集合を表す．μ は非負値のパラメータを表す．式 (9) の第 1 項の値を小さくすることで，センシティブなアイテム $\mathcal{X}$ とおとりアイテム $\mathcal{Y}$ の間の類似度が高まり，その結果，Model Inversiou 攻撃の精度が向上する．第 2 項の値が小さいとき，データポイゾニングにより推薦精度が大きく低下しない．したがって，式 (9) の効用関数 $R_1^{\mathsf{mi}}(\widehat{\mathbf{M}}^*, \widehat{\mathbf{M}})$ を最小化することで，推薦システムの精度の低下を抑えつつ，Model Inversion 攻撃の精度を最大化できると考えられる．さらに，μ (≥ 0) の値を調整することで，攻撃精度と推薦精度の間のトレードオフを制御できる．

3.3 従来手法の問題点

式 (9) の効用関数 $R_1^{\mathsf{mi}}(\widehat{\mathbf{M}}^*, \widehat{\mathbf{M}})$ の問題点について考える．効用関数 $R_1^{\mathsf{mi}}(\widehat{\mathbf{M}}^*, \widehat{\mathbf{M}})$ 自体は Model Inversin 攻撃を実現しうるものであるが，実用的な観点から 2 つの問題をはらんでいる．

第一に，計算量の問題がある．2.3 節で示したように，連鎖律を用いた勾配計算により $R_1^{\mathsf{mi}}(\widehat{\mathbf{M}}^*, \widehat{\mathbf{M}})$ を最小化するときの計算量は $O((m+m')^2 n^2 k)$ である．1 章で述べたように，ユーザ数 m やアイテム数 n は一般的に非常に大きな値であるため，$R_1^{\mathsf{mi}}(\widehat{\mathbf{M}}^*, \widehat{\mathbf{M}})$ を用いて $\mathbf{M}'$ を計算することは計算コストの観点から容易ではない．

第二に，攻撃者の前提知識に関する仮定に問題がある．$R_1^{\mathsf{mi}}(\widehat{\mathbf{M}}^*, \widehat{\mathbf{M}})$ を最小化するためには，既存ユーザの評価行列 $\mathbf{M}$ が既知でなければならない．これは，Li らの手法 [9] と同様の条件ではあるが，1 章で述べたように，推薦システムが一般的に $\mathbf{M}$ を公開することはない．このため，外部の攻撃者が $\mathbf{M}$ を入手できる可能性は低く，現実的に $R_1^{\mathsf{mi}}(\widehat{\mathbf{M}}^*, \widehat{\mathbf{M}})$ を用いて $\mathbf{M}'$ を計算することは困難である．

4. 提案手法

本章では，攻撃者の条件を緩和したより実用的な Model Inversion 攻撃を提案する．4.1 節において，3.3 節で示した問題点を改善する Model Inversion 攻撃のための新たな効用関数を定義する．4.2 節において，再定義した効用関数を最小化する悪性ユーザの評価行列 $\mathbf{M}'$ の具体的な計算方法を示す．

4.1 効用関数の再定義

3.3 章で示した 2 つの問題を解決するために，本稿では，アイテム因子行列 $\mathbf{V}^*$ および $\mathbf{V}$ を入力とする新たな効用関数 $R_2^{\mathsf{mi}} : \mathbb{R}^{n\times k} \times \mathbb{R}^{n\times k} \to \mathbb{R}$ を定義する．$R_2^{\mathsf{mi}}(\mathbf{V}^*, \mathbf{V})$ は次式で表せる．

$$\begin{aligned} R_2^{\mathsf{mi}}(\mathbf{V}^*, \mathbf{V}) = &\sum_{i=1}^{k} \sum_{x\in\mathcal{X}} \sum_{y\in\mathcal{Y}} |\mathbf{V}_{i,y}^{*\top} - \mathbf{V}_{i,x}^{\top}|^2 \\ &+ \sum_{i=1}^{k} \sum_{x\in\mathcal{X}} |\mathbf{V}_{i,x}^{*\top} - \mathbf{V}_{i,x}^{\top}|^2 \\ &+ \mu||\mathbf{V}^* - \mathbf{V}||_F^2 \end{aligned} \tag{10}$$

アイテム因子行列 $\mathbf{V}^{*\top}$ (もしくは，$\mathbf{V}^{\top}$）の k 個の列は評価行列 $\widehat{\mathbf{M}}^*$（もしくは，$\widehat{\mathbf{M}}$）におけるユーザの振る舞いを表す基底ベクトルである．したがって，式 (10) の効用関数 $R_2^{\mathsf{mi}}(\mathbf{V}^*, \mathbf{V})$ を最小化する $\mathbf{M}'$ は，式 (9) の効用関数 $R_1^{\mathsf{mi}}(\widehat{\mathbf{M}}^*, \widehat{\mathbf{M}})$ を最小化する $\mathbf{M}'$ をよく近似すると考えられる．ただし，$R_2^{\mathsf{mi}}(\mathbf{V}^*, \mathbf{V})$ では，$R_1^{\mathsf{mi}}(\widehat{\mathbf{M}}^*, \widehat{\mathbf{M}})$ の第 1 項に該当する部分を $\sum_{i=1}^{k}\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}}|\mathbf{V}_{i,x}^{*\top} - \mathbf{V}_{i,y}^{*\top}|^2$ とせずに，第 1 項と第 2 項に分けている．これは，最適化の結果，$\mathbf{V}_{i,x}^* = \mathbf{V}_{i,y}^* = \mathbf{0}$ となることを回避するためである．

連鎖律を用いた勾配計算により効用関数 $R_2^{\mathsf{mi}}(\mathbf{V}^*, \mathbf{V})$ を最小化する際の計算量は $O(m'n^2k)$ である（詳細は 4.2.1 節を参照）．これは，効用関数 $R_1^{\mathsf{mi}}(\widehat{\mathbf{M}}^*, \widehat{\mathbf{M}})$ を最小化する際の計算量 $O((m+m')^2n^2k)$ よりもはるかに小さい．したがって，効用関数 $R_2^{\mathsf{mi}}(\mathbf{V}^*, \mathbf{V})$ を用いることで，$\mathbf{M}'$ の計算コストは大幅に改善される．

3.3 章で示した 2 つ目の攻撃者の前提知識に関する問題は，連鎖律を用いた勾配計算により式 (10) の効用関数 $R_2^{\mathsf{mi}}(\mathbf{V}^*, \mathbf{V})$ を最小化する際に，ユーザ因子行列 $\mathbf{U}$ の初期値をランダムに生成することで解決する（詳細は 4.2.2 節を参照）．これにより，攻撃者の前提知識は，既存ユーザの評価行列 $\mathbf{M}$ からアイテム因子行列 $\mathbf{V}$ のみに削減できる．$\mathbf{V}$ はアイテムの特徴を表すものであり，ユーザのセンシティブな情報を一切含まない．このため，推薦システムが $\mathbf{V}$ をユーザに共有することや一般に公開することはシステムの構成やサービスの性質上ありうることで，$\mathbf{M}$ よりも容易に入手できる可能性が高い [10]．また，攻撃者は，公開された類似のデータセットから $\mathbf{V}$ を推定できる場合もある [11]．したがって，$\mathbf{M}$ に比べて $\mathbf{V}$ の知識を持つ攻撃者を仮定することはより現実的な条件といえる．

4.2 $\mathbf{M}'$ の計算

本節では，式 (10) の効用関数 $R_2^{\mathsf{mi}}(\mathbf{V}^*, \mathbf{V})$ を最小化する悪性ユーザの評価行列 $\mathbf{M}'$ の計算方法について述べる．ただし，4.1 節で述べたように，攻撃者はアイテム因子行列 $\mathbf{V}$ の知識を持つと仮定する．

まず，2.3 節で示した Li らの手法と同様に，各悪性ユーザが評価するアイテムの最大数を $b_{\max} \in \mathbb{N}$ とする．悪性ユーザが著しく多くのアイテムを評価した場合，他のユーザと振る舞いが明らかに異なるため，検知が容易である．$b_{\max} \in \mathbb{N}$ を設けることで，悪性ユーザとして多くのアイテムを評価するコストを抑えるとともに，攻撃の検知を回避する．また，評価値の範囲を $[-\Lambda_{\min}, \Lambda_{\max}]$ とする．ただし，$\Lambda_{\min}$ および $\Lambda_{\max}$ はそれぞれの正の実数とする．2.3 節では，評価値の範囲を $[-\Lambda, \Lambda]$ としていたが，5 章で示すように，評価値の範囲は必ずしも正負対称ではないため，本章以降では上記の設定を用いる．$\mathbb{M}_2$ を以下に示す $\mathbf{M}'$ の実行可能領域とする．

$$\mathbb{M}_2 = \{\mathbf{M}' | \forall i \in [m'], \forall j \in [n], \mathbf{M}'_{i,j} \in [-\Lambda_{\min}, \Lambda_{\max}]\} \tag{11}$$

上記の条件設定の下，悪性ユーザの評価行列 $\mathbf{M}'$ は以下の最適化問題を解くことにより計算する．

$$\min_{\mathbf{M}'\in\mathbb{M}_2} R_2^{\mathsf{mi}}(\mathbf{V}^*, \mathbf{V}) \ \text{ s.t. } \ \forall i \in [m'], \ ||\mathbf{M}'_i||_0 \le b_{\max} \tag{12}$$

ただし，$\mathbf{M}'_i$ は $\mathbf{M}'$ の i 番目の行を表し，$||\cdot||_0$ は L_0 ノルムを表す．すなわち，$||\mathbf{M}'_i||_0$ は $\mathbf{M}'_i$ における非ゼロ要素の個数を示す．

しかしながら，式 (12) は非凸最適化問題であるため，計算するのが非常に困難である．このため，2.3 節で示した Li らの手法と同様に，悪性ユーザ毎に $b_{\max} \in \mathbb{N}$ 個のアイテムを選択し，射影付き勾配法を用いて近似的に $\mathbf{M}'$ の最適解を計算する．勾配法の t 回目の繰り返し処理における $\mathbf{M}'$ の更新式を以下に示す．

$$\mathbf{M}'^{(t+1)} = \mathrm{Proj}_{\mathbb{M}_2}\left(\mathbf{M}'^{(t)} + s_t \cdot \nabla_{\mathbf{M}'} R_2^{\mathsf{mi}}(\mathbf{V}^{*(t)}, \mathbf{V})\right) \tag{13}$$

ただし，s_t は t 番目のステップサイズを表す．また，$\nabla_{\mathbf{M}'} R_2^{\mathsf{mi}}(\mathbf{V}^*, \mathbf{V}) \in \mathbb{R}^{m'\times n}$ とし，$\mathrm{Proj}_{\mathbb{M}_2}$ は $\mathbb{M}_2$ への射影を表す．すなわち，$\mathrm{Proj}_{\mathbb{M}_2}$ は，$\mathbf{M}'_{i,j}$ が $[-\Lambda_{\min}, \Lambda_{\max}]$ の範囲を超えた場合に，値を $\Lambda_{\min}$ もしくは $\Lambda_{\max}$ の近い方に置き換える．

$\mathbf{M}'$ の計算方法の要約を Algorithm 1 に示す．また，$\nabla_{\mathbf{M}'} R_2^{\mathsf{mi}}(\mathbf{V}^*, \mathbf{V})$ の具体的な計算方法を 4.2.1 節に示す．

4.2.1 $\nabla_{\mathbf{M}'} R_2^{\mathsf{mi}}(\mathbf{V}^*, \mathbf{V})$ の計算

式 (13) における $\nabla_{\mathbf{M}'} R_2^{\mathsf{mi}}(\mathbf{V}^*, \mathbf{V}) \in \mathbb{R}^{m'\times n}$ の計算方法について述べる．連鎖律により，$\nabla_{\mathbf{M}'} R_2^{\mathsf{mi}}(\mathbf{V}^*, \mathbf{V})$ は次式で表せる．

Algorithm 1 $\mathbf{M}'$ の最適化

入力: アイテム因子行列 $\mathbf{V}$, パラメータ $k, \lambda, m', b_{\max}, \Lambda_{\min}, \Lambda_{\max}, \mu, \{s_t\}_{t=1}^{\infty}$.
初期化: ユーザ因子行列の初期値 $\mathbf{U}^{(0)}$ および悪性ユーザの評価行列の初期値 $\mathbf{M}'^{(0)}$ をランダムに生成する.
$\mathbf{M}^{(0)} \leftarrow \mathbf{U}^{(0)}\mathbf{V}^\top$
$t \leftarrow 0$
repeat
  $\Theta^{*(t)} \leftarrow \arg\min_{\Theta^*}\{||\mathcal{R}_\Omega(\mathbf{M}^{(t)} - \mathbf{U}^*\mathbf{V}^{*\top})||_F^2 + ||\mathcal{R}_{\Omega'}(\mathbf{M}'^{(t)} - \mathbf{U}'\mathbf{V}^{*\top})||_F^2 + 2\lambda(||\Theta^*||_F^2\}$
  式 (14) より, $\nabla_{\mathbf{M}} R(\mathbf{V}^{*(t)}, \mathbf{V})$ を計算する.
  $\mathbf{M}'^{(t+1)} \leftarrow \mathrm{Proj}_{\mathbb{M}_2}\left(\mathbf{M}'^{(t)} + s_t \cdot \nabla_{\mathbf{M}'} R_2^{\mathsf{mi}}(\mathbf{V}^{*(t)}, \mathbf{V})\right)$
  $t \leftarrow t+1$
until $\mathbf{M}'^{(t)}$ が収束条件を満たす.
出力: 評価行列 $\mathbf{M}^{(t)}$.

$$\nabla_{\mathbf{M}'} R_2^{\mathsf{mi}}(\mathbf{V}^*, \mathbf{V}) = (\nabla_{\mathbf{M}'}\mathbf{V}^*)(\nabla_{\mathbf{V}^*} R_2^{\mathsf{mi}}(\mathbf{V}^*, \mathbf{V})) \quad (14)$$

ただし, $\nabla_{\mathbf{M}'}\mathbf{V}^* \in \mathbb{R}^{(m'\times n)\times(n\times k)}$, $\nabla_{\mathbf{V}^*} R_2^{\mathsf{mi}}(\mathbf{V}^*, \mathbf{V}) \in \mathbb{R}^{n\times k}$ である. $\nabla_{\mathbf{M}'}\mathbf{V}^*$ および $\nabla_{\mathbf{V}^*} R_2^{\mathsf{mi}}(\mathbf{V}^*, \mathbf{V})$ の計算方法は, それぞれ 4.2.2 節および 4.2.3 節に示す. また, 式 (14) の計算量は $O((m'\times n)\times(n\times k)) = O(m'n^2k)$ である.

4.2.2 $\nabla_{\mathbf{M}'}\mathbf{V}^*$ の計算

$\nabla_{\mathbf{M}'}\mathbf{V}^*$ は, Li らの手法と [9] と同様に, 式 (5) の最適解についての KKT（Karush-Kuhn-Tucker）条件を利用して計算する.

式 (5) の最適解 Θ^* の KKT 条件は次式で表せる.

$$\lambda\mathbf{v}_j^* = \sum_{i\in\Omega_j}(\mathbf{M}_{i,j} - \mathbf{u}_i^{*\top}\mathbf{v}_j^*)\mathbf{u}_i^* + \sum_{i\in\Omega'_j}(\mathbf{M}'_{i,j} - \mathbf{u}_i'^{\top}\mathbf{v}_j^*)\mathbf{u}_i' \quad (\text{for } j = 1, \cdots, n) \quad (15)$$

ただし, $\mathbf{u}_i^* \in \mathbb{R}^k$ および $\mathbf{u}_i' \in \mathbb{R}^k$ はそれぞれ $\mathbf{U}^*$ および $\mathbf{U}'$ の i 番目の行を表す. $\mathbf{v}_j^* \in \mathbb{R}^k$ は $\mathbf{V}^*$ の j 番目の行を表す. $\Omega_j \subseteq [m]$ および $\Omega'_j \subseteq [m']$ はそれぞれ $\mathbf{M}$ および $\mathbf{M}'$ の j 番目の列において評価が観測された行の集合を表す. すなわち, $\Omega_j = \{i : \mathbf{M}_{i,j}$は評価が観測された要素 $\}$ および $\Omega'_j = \{i : \mathbf{M}'_{i,j}$は評価が観測された要素 $\}$ と表せる. 任意のベクトル $\mathbf{a}, \mathbf{b} \in \mathbb{R}^k$ に対して, $(\mathbf{a}^\top\mathbf{b})\mathbf{a} = (\mathbf{a}\mathbf{a}^\top)\mathbf{b}$ が成立するため, 式 (15) は次式で表せる.

$$\left(\lambda\mathbf{M}_k + \sum_{i\in\Omega_j}\mathbf{u}_i^*\mathbf{u}_i^{*\top} + \sum_{i\in\Omega'_j}\mathbf{u}'_i\mathbf{u}_i'^{\top}\right)\mathbf{v}_j = \sum_{i\in\Omega_j}\mathbf{M}_{i,j}\mathbf{u}_i^* + \sum_{i\in\Omega'_j}\mathbf{M}'_{i,j}\mathbf{u}'_i \quad (\text{for } j = 1, \cdots, n) \quad (16)$$

ただし, $\mathbf{M}_k$ は $k\times k$ の単位行列を表す. 式 (16) の両側を $\mathbf{M}'_{i,j}$ について微分することで, 次式が得られる.

$$\frac{\partial\mathbf{v}_j}{\partial\mathbf{M}'_{i,j}} = \left(\lambda\mathbf{M}_k + \sum_{i\in\Omega_j}\mathbf{u}_i^*\mathbf{u}_i^{*\top} + \sum_{i\in\Omega'_j}\mathbf{u}'_i\mathbf{u}_i'^{\top}\right)^{-1}\mathbf{u}'_i \quad (17)$$

以上より, $\nabla_{\mathbf{M}'}\mathbf{V}^*$ は, 式 (17) を用いて, すべての $i \in [m']$ と $j \in [n]$ に対して $\partial\mathbf{v}_j/\partial\mathbf{M}'_{i,j}$ を計算して導出する. ただし, 攻撃者は, 既存ユーザの評価行列 $\mathbf{M}$ に関する知識を持たないため, 勾配法の最初のステップで用いる $\mathbf{U}^{*(0)}$ は, ユーザ因子行列の初期値 $\mathbf{U}^{(0)}$ をランダムに生成して $\mathbf{M}^{(0)} = \mathbf{U}^{(0)}\mathbf{V}^\top$ を計算し, $\mathbf{M}^{(0)}$ に $\mathbf{M}'^{(0)}$ を追加して行列分解を適用することで導出する.

4.2.3 $\nabla_{\mathbf{V}^*} R_2^{\mathsf{mi}}(\mathbf{V}^*, \mathbf{V})$ の計算

式 (10) の両側を $\mathbf{V}_{i,j}^{*\top}$ について微分すると, 次式が得られる.

$$\frac{\partial R_2^{\mathsf{mi}}(\mathbf{V}^*, \mathbf{V})}{\partial\mathbf{V}_{i,j}^{*\top}} = \begin{cases} 2(1+\mu)(\mathbf{V}_{i,j}^{*\top} - \mathbf{V}_{i,j}^\top) & \text{if } j \in \mathcal{X} \\ \sum_{x\in\mathcal{X}} 2(\mathbf{V}_{i,j}^{*\top} - \mathbf{V}_{i,x}^\top) + 2\mu(\mathbf{V}_{i,j}^{*\top} - \mathbf{V}_{i,j}^\top) & \text{if } j \in \mathcal{Y} \\ 2\mu(\mathbf{V}_{i,j}^{*\top} - \mathbf{V}_{i,j}^\top) & \text{otherwise} \end{cases} \quad (18)$$

以上より, $\nabla_{\mathbf{V}^*} R_2^{\mathsf{mi}}(\mathbf{V}^*, \mathbf{V})$ は, 式 (18) を用いて, すべての $i \in [k]$ および $j \in [n]$ に対して $\partial R_2^{\mathsf{mi}}(\mathbf{V}^*, \mathbf{V})/\partial\mathbf{V}_{i,j}^*$ を計算して導出する.

5. 評価実験

行列分解に基づく協調フィルタリングを用いた推薦システムとして, POI（Point Of Interest）推薦システム [5] を想定し, 4 章で提案した新たな Model Inversion 攻撃について評価実験を行った. 5.1 節において, 利用したデータセットや本実験の設定を示し, 5.2 節において, 実験結果を示す.

5.1 実験諸元

POI 推薦システムは, 過去に訪れたスポットからユーザが関心を持つ可能性の高い未訪問のスポットを推薦する [5]. 本稿では, POI 推薦システムを構築するにあたり, 世界中のユーザの各種スポットへのチェックイン情報が収録された Foursquare の Global-scale Check-in [12] をデータセットとして利用した. ただし, 本実験では一例として, ニューヨーク州のマンハッタン地区（緯度 : -74.04〜-73.92, 経度 : 40.7〜40.8）のスポットのみを抽出して利用した. また, 15 箇所以上のスポットにチェックインしたユーザのみを対象とした. このとき, ユーザ数 $m = 2,061$, アイテム数 $n = 18,176$, 総チェックイン数 111,288 であった. ユーザ毎にチェックインしたスポットのうち 70%を訓練データとして抽出し, 残りの 30%をテストデータとして利用した. 評価行列 $\mathbf{M}$ は訓練データを用いて作成した. ただし, 評価値 $\mathbf{M}_{i,j}$ は, ユーザがそのスポットに 1 回以上チェックインしていれば, 値 “1” を取る. また, $\mathbf{M}_{i,j}$ は負の値を取らないため, 潜在モデル Θ の導出は非負値行列因子分解 [2] を用いて行った. 本実験では, 推薦精度の評価指標として, 以下に示す Precision および Recall を用いた.

$$\mathsf{Precision}_N^{\mathsf{rs}} = \frac{\sum_i |\mathcal{S}_i \bigcap \mathcal{T}_i|}{\sum_i |\mathcal{S}_i|} \quad (19)$$

$$\mathsf{Recall}_N^{\mathsf{rs}} = \frac{\sum_i |\mathcal{S}_i \bigcap \mathcal{T}_i|}{\sum_i |\mathcal{T}_i|} \quad (20)$$

ただし, $\mathcal{S}_i$ は評価行列 $\mathbf{M}$ の i 番目のユーザに推薦された

アイテムの集合を表す．$\mathcal{T}_i$ はテストデータの中で $\mathbf{M}$ の i 番目のユーザが値 “1” と評価したアイテムの集合を表す．

まず，予備実験として，ユーザ因子行列 $\mathbf{U}$ およびアイテム因子行列 $\mathbf{V}$ の列数を $k = 10$，正則化パラメータを $\lambda = 0.05$，推薦アイテム数を $N = 5$ とし，式 (19) および式 (20) を用いて推薦システムの Precision および Recall を測定した．本実験を 20 回行い，それぞれの指標について平均を算出した結果，Precision および Recall の平均はそれぞれ 0.0446 および 0.0438 であった．以下，パラメータ k, λ, N については，すべて上記の値を用いる．

次に，上記の POI 推薦システムに対して Model Inversion 攻撃を適用し，提案手法の攻撃性能を評価した．本実験では，推薦されたスポットからユーザが過去に訪問した病院を暴露するシナリオを想定し，上記データセットの “Hospital” カテゴリの中で最も多くのユーザ（23 人）がチェックインしたスポットをセンシティブなアイテムとして利用した．また，Hospital カテゴリ以外からランダムに選択した 5 つのスポットをおとりアイテムとして利用した．評価値の範囲 $[-\Lambda_{\min}, \Lambda_{\max}]$ は，$\mathbf{M}_{i,j}$ が非負であることから，$[0, 1]$ と設定した．ただし，上記の推薦システムでは値 “1” 以外の評価は入力できないため，実際に作成した悪性ユーザの評価行列 $\mathbf{M}'$ に基づきデータポイゾニングを行う際は，$\mathbf{M}'_{i,j} \geq 0.5$ に該当するアイテムに対してのみ値 “1” と評価した．各悪性ユーザが評価するアイテムの個数 $b_{\max}$ は，評価行列 $\mathbf{M}$ において，各ユーザが評価したアイテムの個数の平均が 27 であったことから，$b_{\max} = 27$ と設定した．悪性ユーザ数 m' およびパラメータ μ は，それぞれ $m' = 0.05m, 0.1m, 0.15m, 0.2m = 103, 206, 309, 412$，$\mu = 0, 10^5$ と変化させた．3.1 節で示した Model Inversion 攻撃のための閾値 l は $l = 1, 2, 3, 4, 5$ と変化させた．本実験では，閾値 l のときの攻撃精度の評価指標として，以下に示す Precision および Recall を用いた．

$$\mathsf{Precision}_l^{\mathsf{mi}} = \frac{|\mathcal{U}_{\mathcal{X}} \bigcap \mathcal{U}_{\mathcal{Y},l}|}{|\mathcal{U}_{\mathcal{Y},l}|} \tag{21}$$

$$\mathsf{Recall}_l^{\mathsf{mi}} = \frac{|\mathcal{U}_{\mathcal{X}} \bigcap \mathcal{U}_{\mathcal{Y},l}|}{|\mathcal{U}_{\mathcal{X}}|} \tag{22}$$

ただし，$\mathcal{U}_{\mathcal{X}}$ は評価行列 $\mathbf{M}$ においてセンシティブなアイテム $\mathcal{X}$ のいずれかのアイテムを値 “1” と評価したユーザの集合を表す．$\mathcal{U}_{\mathcal{Y},l}$ はおとりアイテム $\mathcal{Y}$ のうち l 個以上のアイテムが推薦されたユーザの集合を表す．

上記の設定の下，実際に Model Inversion 攻撃を実行し，式 (19) および式 (20) を用いてデータポイゾニング後の推薦システムの Precision および Recall を測定するとともに，式 (21) および式 (22) を用いて Model Inversion 攻撃の Precision および Recall を測定した．そして，悪性ユーザの評価行列 $\mathbf{M}'$ の初期値と，ユーザ因子行列 $\mathbf{U}$ の初期値を変えながら，本実験を 20 回行い，それぞれの指標について平均を算出し，提案手法の攻撃性能を評価した．

表 1 実験結果（$\mu = 0$）

m'	103	206	309	412
$\mathsf{Precision}^{\mathsf{rs}}_{N=5}$	0.0443	0.0437	0.0425	0.0423
$\mathsf{Recall}^{\mathsf{rs}}_{N=5}$	0.0435	0.0429	0.0427	0.0425
$\mathsf{Precision}^{\mathsf{mi}}_{l=1}$	0.5461/16	0.5660/18	0.5779/20	0.5416/20
$\mathsf{Precision}^{\mathsf{mi}}_{l=2}$	0.5706/16	0.5892/18	0.5940/19	0.5910/20
$\mathsf{Precision}^{\mathsf{mi}}_{l=3}$	0.6004/16	0.6112/18	0.6251/19	0.6180/20
$\mathsf{Precision}^{\mathsf{mi}}_{l=4}$	0.6909/16	0.7512/18	0.7660/19	0.7776/19
$\mathsf{Precision}^{\mathsf{mi}}_{l=5}$	1.0/16	0.9875/15	0.9911/15	0.9853/18
$\mathsf{Recall}^{\mathsf{mi}}_{l=1}$	0.7196	0.7826	0.8196	0.8783
$\mathsf{Recall}^{\mathsf{mi}}_{l=2}$	0.6826	0.7413	0.7826	0.8065
$\mathsf{Recall}^{\mathsf{mi}}_{l=3}$	0.6521	0.7	0.7283	0.7261
$\mathsf{Recall}^{\mathsf{mi}}_{l=4}$	0.5848	0.6304	0.6152	0.6043
$\mathsf{Recall}^{\mathsf{mi}}_{l=5}$	0.4935	0.3609	0.3217	0.3282

表 2 実験結果（$\mu = 10^5$）

m'	103	206	309	412
$\mathsf{Precision}^{\mathsf{rs}}_{N=5}$	0.0444	0.0446	0.0445	0.0446
$\mathsf{Recall}^{\mathsf{rs}}_{N=5}$	0.0436	0.0437	0.0437	0.0438
$\mathsf{Precision}^{\mathsf{mi}}_{l=1}$	0.4884/14	0.4322/10	0.3516/12	0.2602/6
$\mathsf{Precision}^{\mathsf{mi}}_{l=2}$	0.5367/14	0.5081/9	0.3550/12	0.2845/6
$\mathsf{Precision}^{\mathsf{mi}}_{l=3}$	0.5526/13	0.4942/8	0.3900/7	0.3904/6
$\mathsf{Precision}^{\mathsf{mi}}_{l=4}$	0.7323/13	0.5852/7	0.4408/7	0.5359/4
$\mathsf{Precision}^{\mathsf{mi}}_{l=5}$	1.0/9	1.0/6	1.0/4	1.0/1
$\mathsf{Recall}^{\mathsf{mi}}_{l=1}$	0.5543	0.3043	0.2913	0.1261
$\mathsf{Recall}^{\mathsf{mi}}_{l=2}$	0.5283	0.2283	0.2587	0.1087
$\mathsf{Recall}^{\mathsf{mi}}_{l=3}$	0.4370	0.2	0.1348	0.0978
$\mathsf{Recall}^{\mathsf{mi}}_{l=4}$	0.3348	0.1543	0.0804	0.0413
$\mathsf{Recall}^{\mathsf{mi}}_{l=5}$	0.2108	0.0761	0.0348	0.0304

5.2 評価結果

表 1 および表 2 に $\mu = 0$ および $\mu = 10^5$ のときの実験結果を示す．どちらの表も悪性ユーザ数を $m' = 103, 206, 309, 412$ としたときの推薦システムの Precision および Recall と，Model Inversion 攻撃の Precision および Recall を示す．ただし，Model Inversion 攻撃の Precision および Recall については閾値を $l = 1, 2, 3, 4, 5$ としたときの結果を示す．5.1 節で述べたように，表の値は初期条件を変えて攻撃を 20 回試行したときの平均値である．また，Model Inversion 攻撃の Precision については，式 (21) の分母が 0 のときの結果は平均を求める際に除外している．このため，20 回の試行のうち式 (21) の分母が非ゼロだった場合の数を表の該当箇所において “/” の後に併記している．

表 1 の結果から，まず，推薦システムに対する Model Inversion 攻撃が実際に実現可能であることが確認できる．また，悪性ユーザ数を増やすことで，攻撃精度，特に，Recall が大きく向上する．Precision については，上述したようにパラメータによって平均を求める際の母数が変動するため一部の結果に誤差が生じている可能性があるが，全体的に悪性データを増やすことで精度が向上する傾向がみられる．さらに，閾値 l を調整することで，Model Inversion 攻撃の Precision と Recall の間のトレードオフを制御できることが分かる．推薦精度については，想定していた通り，データポイゾニングを行うことで Precision および Recall の両方が減少する結果となった．しかしながら，表 1 と表 2 を比較すると，μ の値を大きくすることで，攻撃精度が減少する一方，推薦精度の低下を抑えられることが分かる．したがって，パラメータ μ を調整することで，攻撃の検知を回避しつつ，攻撃を実施できると考えられる．

以上より，本稿で提案した式 (10) の効用関数 $R_2^{\mathrm{mi}}(\mathbf{V}^*, \mathbf{V})$ は，推薦システムに対する Model Inversion 攻撃を実現する上で非常に有用であるといえる．

6. 関連研究

データポイゾニング攻撃および Model Inversion 攻撃に関する研究事例について概説する．

データポイゾニング攻撃では，学習モデルの性能劣化を目的とし，訓練データに悪性データを追加する [9,13–15]．2012 年および 2014 年に Biggio らにより提案された SVM に対する攻撃では，悪性データに関する最適化問題を設定し，勾配法と KKT 条件を組み合わせて近似的にその最適解を計算することで，悪性データを導出する [13,14]．Biggio らは，評価実験を通して，本手法により，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られることを明らかにした．本手法はその後，協調フィルタリング [9] や深層学習 [15] などに応用されている．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ていない．

一方，Model Inversion 攻撃では，学習システムの入出力を利用してユーザのセンシティブな情報を暴露する [16–18]．本攻撃は，2014 年および 2015 年に Fredrikson らにより提案され，文献 [16] において線形回帰モデル，文献 [17] 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された．さらに，2017 年に Hidano らにより，一部の補助情報を用いない，より実用的な線形回帰モデルに対する Model Inversion 攻撃が提案された [18]．Hidano らの手法 [18] では，補助情報を用いない代わりに，データポイゾニングを併用することで，Model Inversion 攻撃を実現する．本稿で提案した攻撃は文献 [18] の概念を推薦システムに応用したものである．

7. おわりに

本稿では，行列分解に基づく協調フィルタリングを用いた推薦システムに着目して，著者らが CSS 2017 において提案したプライバシー暴露の攻撃である Model Inversion 攻撃を改善し，攻撃者の条件を緩和したより実用的な攻撃を提案した．そして，本攻撃を POI（Point Of Interest）推薦システムに適用し，実データを用いた評価実験を通して，本攻撃が実現可能であることを明らかにした．

今後は，本攻撃を他の推薦システムへ適用し，妥当性のさらなる検証を行うとともに，対策について検討する．

参考文献

[1] 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).

[2] Lin, C.: Projected Gradient Methods for Nonnegative Matrix Factorization, *Neural Computation*, Vol. 19, No. 10, pp. 2756–2779 (2007).

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

[4] Tang, J., Hu, X. and Liu, H.: Social Recommendation: A Review, *Social Network Analysis and Mining*, Vol. 3, No. 4, pp. 1113–1133 (2013).

[5] Liu, Y., Pham, T. N., Cong, G. and Yuan, Q.: An Experimental Evaluation of Point-of-interest Recommendation in Location-based Social Networksk, *Proceedings of the VLDB Endowment*, Vol. 10, No. 10, pp. 1010–1021 (2017).

[6] Ramakrishnan, N., Keller, B. J., Mirza, B. J., Grama, A. Y. and Karypis, G.: Privacy Risks in Recommender Systems, *IEEE Internet Computing*, Vol. 5, No. 6, p. 54 (2001).

[7] Calandrino, J. A., Kilzer, A., Narayanan, A., Felten, E. W. and Shmatikov, V.: "You Might Also Like:" Privacy Risks of Collaborative Filtering, *Security and Privacy (SP), 2011 IEEE Symposium on*, IEEE, pp. 231–246 (2011).

[8] 披田野清良，村上隆夫，勝又秀一，清本晋作，花岡悟一郎：協調フィルタリングに対する Model Inversion 攻撃の提案，コンピュータセキュリティシンポジウム 2017（CSS2017）論文集，pp. 312–318 (2017).

[9] Li, B., Wang, Y., Singh, A. and Vorobeychik, Y.: Data Poisoning Attacks on Factorization-Based Collaborative Filtering, *Proceedings of the 30th Annual Conference on Neural Information Processing Systems (NIPS 2016)*, pp. 1885–1893 (2016).

[10] Meng, X., Wang, S., Shu, K., Li, J., Chen, B., Liu, H. and Zhang, Y.: Personalized Privacy-Preserving Social Recommendation, *Proceedings of 32nd AAAI Conference on Artificial Intelligence (AAAI 2018)*, pp. 1–8 (2018).

[11] Shani, G., Chickering, M. and Meek, C.: Mining Recommendations From the Web, *Proceedings of the 2008 ACM conference on Recommender systems (RecSys 2008)*, pp. 35–42 (2008).

[12] Yang, D., Zhang, D. and Qu, B.: Participatory Cultural Mapping Based on Collective Behavior Data in Location Based Social Networks, *ACM Transction on Intelligent Systems and Technology (TIST)*, Vol. 7, No. 30, pp. 1–23 (2016).

[13] 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).

[14] 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).

[15] Muñoz González, L., Biggio, B., Demontis, A., Paudice, A., Wongrassamee, V., Lupu, E. C. and Roli, F.: Towards Poisoning of Deep Learning Algorithms with Back-gradient Optimization, *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security (AISec 2017)*, pp. 27–38 (2017).

[16] Fredrikson, M., Lantz, E. and Jha, S.: Privacy in Pharmacogenetics: An End-to-End Case Study of Personalized Warfarin Dosing, *the Proceedings of the 23rd USENIX Security Symposium (USENIX 2014)*, pp. 17–32 (2014).

[17] 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).

[18] Hidano, S., Murakami, T., Katsumata, S., Kiyomoto, S. and Hanaoka, G.: Model Inversion Attacks for Online Prediction Systems: Without Knowledge of Non-Sensitive Attributes, *Proceedings of the 15th International Conference on Privacy, Security, and Trust (PST 2017)*, pp. 1–10 (2017).