基于数据非随机缺失机制的推荐系统托攻击探测

李 聪 [1] 骆志刚 [1]

摘 要 协同过滤推荐系统极易受到托攻击的侵害. 开发托攻击探测技术已成为保障推荐系统可靠性与鲁棒性的关键. 本文以数据非随机缺失机制为依托, 对导致评分缺失的潜在因素进行解析, 并在概率产生模型框架内将这些潜在因素与 Dirichlet 过程相融合, 提出了用于托攻击探测的缺失评分潜在因素分析 (Latent factor analysis for missing ratings, LFAMR) 模型. 实验表明, 与现有探测技术相比, LFAMR 具备更强的普适性和无监督性, 即使缺乏系统相关先验知识, 仍可有效探测各种常见托攻击.

关键词 协同过滤, 托攻击, 缺失数据, Dirichlet 过程, 变分推断

引用格式 李聪, 骆志刚. 基于数据非随机缺失机制的推荐系统托攻击探测. 自动化学报, 2013, **39**(10): 1681−1690

DOI 10.3724/SP.J.1004.2013.01681

Detecting Shilling Attacks in Recommender Systems Based on Non-random-missing Mechanism

LI Cong[1] LUO Zhi-Gang[1]

Abstract Collaborative filtering recommender systems are highly vulnerable to shilling attacks. Developing detection techniques against shilling attacks has become the key to guaranteeing both the reliability and robustness of recommender systems. Through revealing the latent factors invoking missing ratings under the non-random-missing mechanism, and further combining these latent factors with Dirichlet process in the framework of probabilistic generative model, this paper proposes a latent factor analysis for missing ratings (LFAMR) model for attack detection. Experimental results show that comparing with the existing detection techniques, LFAMR is more universal and unsupervised, and that it can effectively detect shilling attacks of typical types even in lack of system-related prior knowledge.

Key words Collaborative filtering, shilling attacks, missing data, Dirichlet process, variational inference

Citation Li Cong, Luo Zhi-Gang. Detecting shilling attacks in recommender systems based on non-random-missing mechanism. *Acta Automatica Sinica*, 2013, **39**(10): 1681−1690

协同过滤 (Collaborative filtering) 推荐[1] 是一项新兴的信息检索技术, 已在 Web 领域取得了成功应用, 典型的如 Amazon[1]、eBay[2] 等商业站点. 协同过滤推荐系统 (以下简称推荐系统) 能向终端用户提供个性化的信息服务, 有效缓解了信息过载 (Information overload) 问题.

推荐系统包含两个实体集: 用户集 $\{\text{user}_i|i = 1\sim I\}$ 与项集 $\{\text{item}_j|j = 1\sim J\}$. 项通指书籍、音乐、电影等检索对象. 用户对系统的操作历史存储为评分矩阵 $R_{I\times J}$, 其元素 R_{ij} 是 user_i 对 item_j 的评分, 代表偏好程度. 一般地, 推荐系统通过挖掘评分矩阵, 找出目标用户的最近邻, 即与目标用户兴趣偏好相近的群体, 之后向目标用户推荐其最近邻评价较高的项[2]. 这种工作模式虽行之有效, 却有严重的安全隐患. 恶意用户可向系统中注入虚假评分, 成为大量用户的最近邻, 致使系统产生虚假的推荐信息, 从而牟取不当利益[3]. 这种可能引发严重后果的恶意行为称为 "托攻击 (Shilling attacks)"[4−5]. 如何防御托攻击已成为当前推荐系统研究领域的热点之一.

当前主要的解决思路是研究托攻击探测技术, 特别是无监督探测技术. 然而托攻击的多样性与隐蔽性使现有的探测技术面临两种局限: 1) 普适性差, 仅对某些托攻击有效, 适用范围有限; 2) 无监督程度低, 需输入未知的关键参数, 如攻击者数目等. 克服这些局限的最有效途径就是充分挖掘推荐系统中一切已知的, 甚至是隐含信息的利用价值. 现有的探测技术均忽视了一个具有潜在价值的信息 — 评分矩阵的稀疏性[6]. 稀疏性是指评分矩阵中含有大量缺失评分 (典型缺失率在 90 % 左右), 这种特性的形成是由于用户通常只评价系统中少数感兴趣的项. 显然, 缺失评分与用户偏好等因素存在特定关联. 挖掘缺

收稿日期 2011-02-28 录用日期 2012-06-29
Manuscript received February 28, 2011; accepted June 29, 2012
本文责任编委 田捷
Recommended by Associate Editor TIAN Jie
1. 国防科学技术大学计算机学院 长沙 410073
1. School of Computer, National University of Defense Technology, Changsha 410073
[1]http://www.amazon.com/
[2]http://www.ebay.com/

失评分背后的潜在信息必将有助于更高效的攻击探测.

为此, 本文提出了缺失评分潜在因素分析 (Latent factor analysis for missing ratings, LFAMR) 模型. 主要思路包括: 1) 依据数据非随机缺失机制, 解析导致评分缺失的潜在因素, 建立评分缺失事件与对应潜在因素间的 Logistic 回归模型; 2) 基于 1) 的 Logistic 回归模型与 Dirichlet 过程 (Dirichlet process), 建立用于用户聚类的概率产生模型 (Probabilistic generative model), 并利用变分期望最大化 (Variational expectation maximization, Variational EM) 方法学习模型参数; 3) 从理论上揭示攻击者富集类在理想情况下的识别特征, 实现托攻击的探测. 最后, 将通过实验检验 LFAMR 的托攻击探测能力.

1 推荐系统安全性问题及研究现状

由于推荐系统内在的开放性及交互式特点, 攻击者能以极低代价向系统中注入虚假评分, 轻易改变推荐结果. 依攻击目的, 托攻击可划为两类[4]: 提高目标项的评价, 称为推攻击 (Push attack); 降低目标项的评价, 称为核攻击 (Nuke attack). 实际情况中推攻击更为普遍.

1.1 推荐系统的托攻击

攻击者的所有评分构成攻击概貌 (Attack profile), 它是一个 n 维向量, n 是系统中项的个数. 图 1 为攻击概貌的形式结构.

目标项	填充项	选择填充项	未评分项

图 1 攻击概貌的形式结构

Fig. 1 General framework of attack profile

目标项 (Target item) 在推攻击时设为最高评分 $r_{\max}$, 在核攻击时设为最低评分 $r_{\min}$. 未评分项 (Unrated items) 的评分设为 $\emptyset$, 即不予评分. 攻击概貌构建自不同的攻击模型, 随机攻击 (Random attack)、均值攻击 (Average attack)、流行攻击 (Bandwagon attack) 与段攻击 (Segment attack) 是 4 种典型的攻击模型[7], 四者的差异主要体现在项的评分策略与选择填充项 (Selected items) 的选取策略上, 其次, 还有攻击对象、攻击能力与部署成本等方面的不同, 唯一相同之处是填充项 (Filler items) 的随机性选取, 这点的利用价值将在下文体现. 定义填充率 (Filler size) $p^{\text{fill}} = |\{\text{填充项}\}|/n$, 攻击强度 (Attack size) $p^{\text{att}} = |\{\text{攻击者}\}|/|\{\text{真实用户}\}|$. p^{att} 不宜过大, 否则, 会加大攻击成本, 且不利于隐蔽.

1.2 托攻击探测技术的相关研究

总体上, 现有攻击探测技术可归为有监督及无监督两类. 有监督探测技术通过提取攻击概貌有别于真实用户的一些属性, 形成特征向量, 依此构建决策树, 支持向量机等分类器[8]. 然而实际应用中合适的训练集是难以拟出的, 所以有监督探测技术仅具理论价值. 目前的研究更多集中于无监督探测技术, 出现了一些具有较好理论基础与实验效果的算法.

Zhang[9] 等利用奇异值分解 (Singular value decomposition, SVD) 与 EM 算法为评分矩阵构建低维线性模型, 用其计算每个用户的产生概率, 视概率偏低的用户为攻击者 (本文暂称这种算法为 EMSVD). Mehta 等[10−11] 提出了概率潜在语义分析 (Probabilistic latent semantic analysis, PLSA) 探测算法与主成分分析变量选择 (Variable-selection using principal component analysis, PCA VarSelect) 探测算法. PLSA 是一种概率产生模型, 可对用户进行软聚类, 由于攻击概貌的极端相似性, 认为平均统计距离最小的类是攻击者的富集类. PCA VarSelect 算法对评分矩阵做主成分分析, 以每个用户对应的前 1～3 个主成分系数的大小为指标进行攻击探测. Bryan 等[12] 提出的无监督攻击概貌探查 (Unsupervised retrieval of attack profiles, UnRAP) 算法借鉴并改进了识别基因表达矩阵中双聚类时所用的目标度量 H_v 值 (H_v-score), 算法在判别目标项的基础上, 利用攻击者普遍具有高 H_v 值的特性, 设计了特定的滑动窗口技术以探测攻击者.

然而, 无监督只是相对概念, 少量的先验输入必不可少. EMSVD、PLSA 与 PCA VarSelect 算法均需输入系统相关参数, 而这些对探测性能有潜在影响的参数通常是无法准确获得的, 削弱了这些算法的无监督性. UnRAP 算法虽有较强的无监督性, 但普适性并不理想, 适用范围有限. 类似的普适性问题也同样存在于 EMSVD 与 PCA VarSelect 等算法中. 普适性及无监督性上的局限均降低了这些算法的实用化程度.

鉴于此, 本文以托攻击探测技术的实用性为设计原则, 致力于开发兼具较强普适性和无监督性的探测技术.

2 预备知识

2.1 数据缺失模式

与评分矩阵类似, 现实中大部分数据集都不可避免地含有缺失数据. 如何处理缺失数据是算法设计者普遍面临的问题.

数据的缺失模式可分为三类[13]: 完全随机缺失

(Missing completely at random, MCAR)、随机缺失 (Missing at random, MAR) 和非随机缺失 (Not missing at random, NMAR). 形式化地, 考虑随机向量 $\boldsymbol{R}$ (推荐系统中为评分矩阵 $R_{I\times J}$) 与指示向量 $\boldsymbol{M}$, 其中, $\boldsymbol{R}=R^o\cup R^m$, R^o 与 R^m 分别为观测数据与缺失数据. $\boldsymbol{M}$ 的元素 $M_i\in\{0,1\}$, 当 $M_i=0$ 时, $R_i\in R^m$, 否则 $R_i\in R^o$. 令 $\boldsymbol{R},\boldsymbol{M}$ 与隐含变量 T 的联合分布 $p(\boldsymbol{R},T,\boldsymbol{M})=p(\boldsymbol{R},T)p(\boldsymbol{M}|\boldsymbol{R},T)$, $p(\boldsymbol{R},T)$ 称为数据模型 (Data model), $p(\boldsymbol{M}|\boldsymbol{R},T)$ 为选择模型 (Selection model). MCAR 与 MAR 的选择模型分别满足 $p(\boldsymbol{M}|\boldsymbol{R},T)=p(\boldsymbol{M})$ 与 $p(\boldsymbol{M}|\boldsymbol{R},T)=p(\boldsymbol{M}|\boldsymbol{R}^o)$, 而 NMAR 的选择模型则要依赖于缺失数据 R^m 或隐含变量 T. 在 MAR 或 MCAR 的条件下, 可推得式 (1) 成立:

$$p(T|R^o,\boldsymbol{M})=p(T|R^o) \tag{1}$$

上式表明缺失模式为 MAR 或 MCAR 时, 数据缺失不会影响变量的贝叶斯推断. 而缺失模式为 NMAR 时, 则不具备这种良好性质. 推荐系统中, 用户一般不会评价不喜好的项, 这表明评分缺失并非独立的随机事件, 而是依赖于用户偏好等潜在因素, 因此, 推荐系统的评分缺失模式应为 NMAR. 更直观地, 例如考察 MovieLens100K 数据集[3]中的用户评分分布, 从图 2 中易见其分布并不均匀, 而是明显偏向高评价区域, 显然用户偏好与评分缺失事件存在关联. 在 NMAR 时, 式 (1) 不再成立, 必须考虑评分缺失对模型推断的影响.

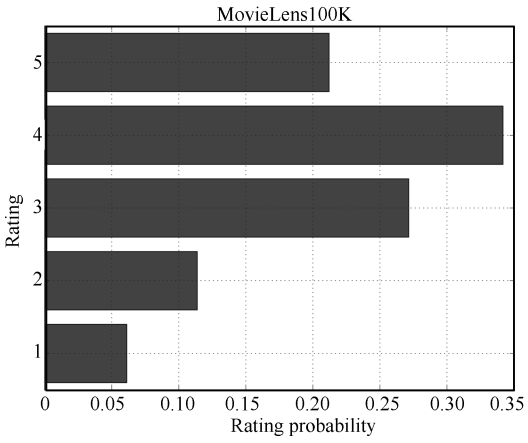


图 2　评分分布

Fig. 2　Distribution of rating values

2.2　Dirichlet 过程

本文的核心思路是依特定相似性对用户进行聚类, 并从中识别攻击者富集类. LFAMR 承担了用户聚类功能, 为避免事先设定未知聚类个数而可能引起的错误拟合, LFAMR 采用了一种非参数化贝叶斯方法 — Dirichlet 过程[14]. 这种方法一方面允许聚类个数按需增减, 赋予模型充分的可伸缩性; 另一方面借助贝叶斯方法内在的 “Occam 剃刀”[15] 效应, 决定合理的聚类个数.

本质上, Dirichlet 过程是一种分布的分布, 意味着其每个抽样本身也是概率分布. Dirichlet 过程可记做 $GP(\alpha,G_0)$, α 与 G_0 分别为缩放因子与基分布. Dirichlet 过程的典型应用包含两个抽样层次:

$$\begin{aligned} G|\alpha,G_0 &\sim GP(\alpha,G_0)\\ \theta_k &\sim G \end{aligned}$$

其中, θ_k 的抽样符合中国餐馆过程 (Chinese restaurant process)[16]. 简言之, 下一抽样值倾向于重复之前出现次数较多的抽样值, 此即 Dirichlet 过程的聚类效应. $GP(\alpha,G_0)$ 的每个抽样 G 是离散分布, 准确地说, G 是无穷混合单点质量分布 (Point mass distribution), 这可由 Dirichlet 过程的断棒 (Stick-breaking) 表示法[17] 明确揭示. 考虑随机变量 $\beta_k|\alpha\sim\mathrm{Beta}(1,\alpha)$ 与 $\theta_k\sim G_0$, 则 G 的断棒表示为

$$\begin{aligned} \chi_k &= \beta_k\prod_{n=1}^{k-1}(1-\beta_n)\\ G &= \sum_{k=1}^{\infty}\chi_k\delta_{\theta_k} \end{aligned} \tag{2}$$

断棒表示法为基于 Dirichlet 过程的贝叶斯模型构建与参数推断提供了极大便利.

确切地讲, LFAMR 进行用户聚类时利用的是 Dirichlet 过程的一种重要应用形式 — Dirichlet 过程混合 (Dirichlet process mixture, DPM) 模型[18]. DPM 定义了可用于聚类分析的无穷混合模型, 由三个抽样层次组成:

$$\begin{aligned} G|\alpha,G_0 &\sim GP(\alpha,G_0)\\ \theta_k &\sim G\\ \eta_n|\theta_k &\sim F(\theta_k) \end{aligned}$$

此时 θ_k 具有双重身份, 在概率意义上, 它是混合成分分布 F 的参数, 而在实际意义上, 它是第 k 个数据类的 “类型中心”. DPM 利用聚类效应提升或抑制某些混合成分, 从而确定出适合当前数据集的数据类.

3　缺失评分潜在因素分析模型

为体现评分非随机缺失性, 同时不引入过多复杂因素, LFAMR 采用形如 $p(\boldsymbol{M}|\boldsymbol{R},T)=p(\boldsymbol{M}|T)$ 的选择模型, 即 $\boldsymbol{M}$ 仅依赖于隐含量 T, 忽略评分的

[3]http://www.grouplens.org/node/73

影响. LFAMR 认为此隐含量代表用户对项的兴趣值, 因为 "感兴趣" 一般是触发评分事件的首要因素, 评分只是用户对感兴趣项的事后评价. 兴趣值作为缺失评分的潜在因素, 是构建 LFAMR 模型的关键.

3.1 缺失评分的潜在因素

设 user_i 对 item_j 的兴趣值为 T_{ij}, 它可视作三种因素的叠加:

1) 用户因素 U_i: user_i 可能会给出偏离平均水平的异常 (极多或极少) 评分量, 这种个性化的评分行为由用户因素解释;

2) 项因素 H_j: 若 item_j 非常流行, 即获得了大量评分, 则项因素将提升用户对此项的兴趣; 反之, 将抑制用户对此项的兴趣;

3) 用户 – 项因素 UH_{ij}: 这是 user_i 对 item_j 的纯粹兴趣值, 完全由用户内在偏好与项本质属性的契合程度决定, 不夹杂任何外界因素.

至此, 可建立 T_{ij} 对 M_{ij} 的 Logistic 回归模型:

$$\begin{aligned} T_{ij} &= U_i + H_j + UH_{ij} \\ p(M_{ij}) &= \text{Bern}(M_{ij}|\sigma(T_{ij})) \end{aligned}$$

其中, Logistic sigmoid 函数 $\sigma(x) = (1 + \exp(-x))^{-1}$. 显然, 任一因素值的增加都会提升兴趣值 T_{ij}, 从而降低 R_{ij} 的缺失概率; 反之, 则会增加缺失概率.

3.2 LFAMR 的形式化描述

LFAMR 是一种融合了上述 Logistic 回归模型与 DPM 的概率产生模型. 图 3 为 LFAMR 的概率图模型 (贝叶斯网络), 各随机变量均被赋予特定的概率分布. LFAMR 的抽样过程可归纳为 4 个层次:

1) 顶层: Dirichlet 过程的断棒表示

$$\begin{aligned} &\alpha \sim \text{Gam}(w_1, w_2) \\ &\text{对}\ \forall k, k = 1 \sim \infty: \\ &\quad V_k \sim \text{Beta}(1, \alpha) \\ &\quad \text{类型中心}\ \boldsymbol{\theta}_k \sim \text{N}(\mathbf{0}, \Lambda_0^{-1}) \\ &\quad \text{混合系数}\ \chi_k = V_k \prod_{n=1}^{k-1}(1 - V_n) \end{aligned}$$

2) 项层: 抽样与项相关的变量

$$\begin{aligned} &\text{对}\ \forall \text{item}_j,\ j = 1 \sim J: \\ &\quad \text{项因素}\ H_j \sim \text{N}(0, \tau_1^2) \\ &\quad \text{项属性特征}\ \boldsymbol{X}_j \sim \text{N}(\mathbf{0}, \Lambda_1^{-1}) \end{aligned}$$

3) 用户层: 抽样与用户相关的变量

$$\begin{aligned} &\gamma \sim \text{Gam}(\nu_1, \nu_2) \\ &\text{对}\ \forall \text{user}_i,\ i = 1 \sim I: \\ &\quad \text{用户类属}\ Z_i \sim \text{Mult}(\{\chi_k\}) \\ &\quad \text{用户因素}\ U_i \sim \text{N}(0, \tau_2^2) \\ &\quad \text{用户偏好特征}\ \boldsymbol{G}_i \sim \prod_{k=1}^{\infty} \text{N}(\boldsymbol{\theta}_k, (\gamma\Lambda_2)^{-1})^{Z_{ik}} \end{aligned} \tag{3}$$

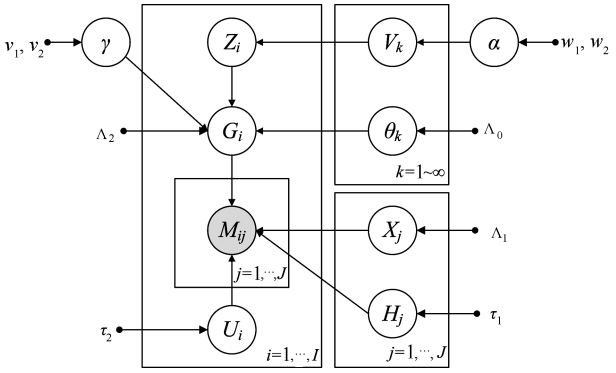


图 3 缺失评分潜在因素分析的图模型

Fig. 3 Graphical model for latent factor analysis for missing ratings (LFAMR)

4) 评价层: 决定评分缺失与否

$$\begin{aligned} &\text{对}\ \forall \text{user}_i, \text{item}_j: \\ &\quad \text{缺失指示}\ M_{ij} \sim \text{Bern}(\sigma(T_{ij})) \\ &\quad \text{兴趣值}\ T_{ij} = U_i + H_j + UH_{ij} \\ &\quad \text{用户 – 项因素}\ UH_{ij} = \boldsymbol{G}_i^{\text{T}} \boldsymbol{X}_j \end{aligned} \tag{4}$$

其中, $\boldsymbol{X}_j$, $\boldsymbol{G}_i$ 与 $\boldsymbol{\theta}_k$ 均为 d 元随机变量. 若将 d 视为推荐系统中项的抽象类型数, 则 $\boldsymbol{X}_j$ 与 $\boldsymbol{G}_i$ 分别代表 item_j 在每个类型上的权重与 user_i 在每个类型上的亲和度, 因此, 用户 – 项因素可用两者内积表示. LFAMR 依偏好特征的相似性将用户聚类, Z_i ("1-of-K" 形式) 指出 user_i 的类属, $\boldsymbol{\theta}_k$ 代表类 k 中用户的总体偏好. 式 (3) 中 $\text{N}(\boldsymbol{\theta}_k, (\gamma\Lambda_2)^{-1})$ 是 DPM 的混合成分分布, 同类中用户偏好的特异性来自 $\text{N}(\boldsymbol{\theta}_k, (\gamma\Lambda_2)^{-1})$ 对类型中心 $\boldsymbol{\theta}_k$ 的正态扰动. 扰动幅度取决于变量 γ, 它决定着用户类型中心的影响范围.

3.3 变分推断

令 Ω^o 与 Ω^h 分别代表 LFAMR 中的可见随机变量 $\{\boldsymbol{M}\}$ 与隐含随机变量 $\{\alpha, \gamma, \boldsymbol{\theta}, \boldsymbol{V}, \boldsymbol{Z}, \boldsymbol{G}, \boldsymbol{X}, \boldsymbol{H}, \boldsymbol{U}\}$. 由变量间的条件独立性, 可得 Ω^o 与 Ω^h 的联合分布:

$$\begin{aligned} &p(\Omega^o \cup \Omega^h) = \\ &\quad p(\alpha|w_1, w_2)p(\gamma|\nu_1, \nu_2)p(\boldsymbol{V}|\alpha)p(\boldsymbol{\theta}|\Lambda_0) \times \\ &\quad p(\boldsymbol{X}|\Lambda_1)p(\boldsymbol{H}|\tau_1)p(\boldsymbol{U}|\tau_2)p(\boldsymbol{Z}|\boldsymbol{V}) \times \end{aligned}$$

$$p(\boldsymbol{G}|\boldsymbol{\theta},\boldsymbol{Z},\gamma,\Lambda_2)p(\boldsymbol{M}|\boldsymbol{U},\boldsymbol{H},\boldsymbol{G},\boldsymbol{X}) \tag{5}$$

在模型推断时, 会发现难以求出后验分布 $p(\Omega^h|\Omega^o)$ 的规则形式. 针对此问题, 本文采取变分贝叶斯方法[19] 寻求 $p(\Omega^h|\Omega^o)$ 的近似替代 $q(\Omega^h)$.

随机变量 Ω^o 的对数边缘似然满足关系:

$$\ln p(\Omega^o) = \mathcal{L}(q) + \mathrm{KL}(q||p)$$

其中

$$\begin{aligned}\mathcal{L}(q) &= \int q(\Omega^h)\ln\frac{p(\Omega^o\cup\Omega^h)}{q(\Omega^h)}\,\mathrm{d}\Omega^h \\ \mathrm{KL}(q||p) &= -\int q(\Omega^h)\ln\frac{p(\Omega^h|\Omega^o)}{q(\Omega^h)}\,\mathrm{d}\Omega^h\end{aligned} \tag{6}$$

$\mathrm{KL}(q||p)$ 代表变分后验 $q(\Omega^h)$ 与真实后验 $p(\Omega^h|\Omega^o)$ 间的 Kullback-Leibler (KL) 离散度, 其值非负, 所以 $\mathcal{L}(q)$ 是定值 $\ln p(\Omega^o)$ 的下界. 为使 $q(\Omega^h)\to p(\Omega^h|\Omega^o)$, 只需 $\mathrm{KL}(q||p)\to 0$, 这等价于最大化 $\mathcal{L}(q)$. 为简化优化过程, 通常对 $q(\Omega^h)$ 的形式作以下假设:

$$q(\Omega^h) = \prod_{\varphi\in\Omega^h} q(\varphi)$$

此时, 任取随机变量 $\varphi\in\Omega^h$, 固定其余分布 $\{q(\varphi')|\varphi'\in\Omega^h\wedge\varphi'\neq\varphi\}$. $q(\varphi)$ 满足如下关系时 $\mathcal{L}(q)$ 达到最大[19]:

$$q(\varphi)\propto\exp\mathrm{E}_{-\varphi}[\ln p(\Omega^o\cup\Omega^h)] \tag{7}$$

其中, $\mathrm{E}_{-\varphi}[\cdot]$ 是关于分布 $\prod_{\varphi'\neq\varphi}q(\varphi')$ 的期望算子. 可用式 (7) 迭代更新 Ω^h 中每个变量的变分后验, 每步迭代均使下界 $\mathcal{L}(q)$ 得到提升. 迭代收敛时, $q(\Omega^h)$ 则为 $p(\Omega^h|\Omega^o)$ 的良好近似. 以上是变分贝叶斯法的概要过程.

对 LFAMR 进行变分推断时, 需解决两个问题: 节点 $\boldsymbol{M}$ 与其父节点 $\boldsymbol{H},\boldsymbol{X},\boldsymbol{U},\boldsymbol{G}$ 间的非共轭性以及变量 k 的无界性.

首先, 可采取局部变分法克服上述节点间的非共轭性. 现有不等式[20]:

$$\sigma(a)\geqslant\sigma(\xi)\exp\left(\frac{a-\xi}{2}-\lambda(\xi)(a^2-\xi^2)\right)=\sigma'(a,\xi)$$

其中, $\lambda(\xi)=\left(\frac{1}{4\xi}\right)\tanh\left(\frac{\xi}{2}\right)$. 此时有:

$$\begin{aligned}&p(\boldsymbol{M}|\boldsymbol{U},\boldsymbol{H},\boldsymbol{G},\boldsymbol{X}) = \\ &\quad\prod_{ij}\mathrm{Bern}(M_{ij}|\sigma(T_{ij})) = \\ &\quad\prod_{ij}\sigma(T_{ij})^{M_{ij}}(1-\sigma(T_{ij}))^{1-M_{ij}} = \\ &\quad\prod_{ij}\exp(T_{ij}M_{ij})\sigma(-T_{ij})\geq \\ &\quad\prod_{ij}\exp(T_{ij}M_{ij})\sigma'(-T_{ij},\xi_{ij})\end{aligned}$$

上式代入式 (5) 得:

$$p(\Omega^o\cup\Omega^h)\geq p^v(\Omega^o\cup\Omega^h|\Phi) \tag{8}$$

其中, 变分参数 $\Phi=\{\xi_{ij}|i=1\sim I\wedge j=1\sim J\}$. 式 (8) 代入式 (6) 得 $\mathcal{L}(q)$ 的下界:

$$\mathcal{L}(q)\geq\int q(\Omega^h)\ln\frac{p^v(\Omega^o\cup\Omega^h|\Phi)}{q(\Omega^h)}\,\mathrm{d}\Omega^h=\mathcal{L}^v(q,\Phi)$$

此后, $\mathcal{L}^v(q,\Phi)$ 将代替 $\mathcal{L}(q)$ 成为优化目标.

其次, 变量 k 的无界性问题可用 Dirichlet 过程的截尾断棒 (Truncated stick-breaking) 表示法[21] 解决, 基本思想是将式 (2) 中的 G 近似表示为有限混合分布, 即 $G\approx\sum_{k=1}^{T}\chi_k\delta_{\theta_k}$, 正整数 T 为截断值, 变分推断时, 只需固定 $q(V_T=1)=1$[22].

至此, 可用变分 EM 算法[20] 最优化 $\mathcal{L}^v(q,\Phi)$: 初始化分布 $q^{(0)}$ 和参数 $\Phi^{(0)}$, 之后交替固定一个因子, 以另一因子为变元最大化 $\mathcal{L}^v(q,\Phi)$, 直至迭代收敛. 形式化地:

$$\begin{aligned}&\text{E-step}: \quad q^{(k+1)}=\arg\max_q\mathcal{L}^v(q,\Phi^{(k)}) \\ &\text{M-step}: \quad \Phi^{(k+1)}=\arg\max_\Phi\mathcal{L}^v(q^{(k)},\Phi)\end{aligned}$$

在 M-step, 对 $\forall i,j$ 有:

$$\begin{aligned}&\frac{\partial\mathcal{L}^v(q,\Phi)}{\partial\xi_{ij}}=\frac{\partial\mathrm{E}[\ln p^v(\Omega^o\cup\Omega^h|\Phi)]}{\partial\xi_{ij}}=0 \\ \Rightarrow\quad &\xi_{ij}^2=\mathrm{E}[T_{ij}^2]=\mathrm{E}[(U_i+H_j+UH_{ij})^2]\end{aligned}$$

在 E-step, 仍用式 (7) 推导变分后验 q, 只需将 $p(\Omega^o\cup\Omega^h)$ 替换为 $p^v(\Omega^o\cup\Omega^h|\Phi)$. 以下为最终结果, 其中各分布参数见附录 A:

$$\begin{aligned}&q(\alpha)=\mathrm{Gam}(w_1',w_2'), \quad q(\gamma)=\mathrm{Gam}(\nu_1',\nu_2') \\ &q(V_k)=\mathrm{Beta}(\alpha_1^k,\alpha_2^k), \quad q(Z_i)\propto\prod_{k=1}^{T}\exp(s_k^i)^{Z_{ik}} \\ &q(\boldsymbol{\theta}_k)=\mathrm{N}(\boldsymbol{\mu}_\theta^k,\Lambda_\theta^k), \qquad q(\boldsymbol{X}_j)=\mathrm{N}(\boldsymbol{\mu}_x^j,\Lambda_x^j) \\ &q(H_j)=\mathrm{N}(\mu_h^j,\tau_h^j), \qquad q(\boldsymbol{G}_i)=\mathrm{N}(\boldsymbol{\mu}_g^i,\Lambda_g^i) \\ &q(U_i)=\mathrm{N}(\mu_u^i,\tau_u^i)\end{aligned}$$

上述变分 EM 算法的时间复杂度为 #iter $\times$ $\mathrm{O}((Td^3+Jd^3+T^2)I)$, 其中 #iter 代表迭代次数. 大量的矩阵乘法与求逆操作导致算法较为耗时. 不过, 算法具有良好的并行加速潜力. 在 M-step 中, 可以同步更新变分参数 $\{\xi_{ij}|i=1\sim I\wedge j=1\sim J\}$,

因为在此过程中不同的 ξ_{ij} 之间不存在相关性. 此外, 在 E-step 中, 也可对除 α,γ 以外每组随机变量的分布参数采取这种并行更新策略. 因此, 若将算法部署于多处理机平台, 会显著提升其执行效率. 算法的空间复杂度为 $\mathrm{O}((J+T)I+(I+J+T)d^2)$, 存储空间主要用于存放 M-step 中的变分参数以及 E-step 中每个后验分布的参数.

迭代收敛后, 近似认为 LFAMR 中各变量的真实值等于此变量关于其变分后验的期望.

3.4 攻击类的识别

为便于讨论, 首先给出下列定义:

定义 1. 设用户类集 $C=\{cls_k|k=1,\cdots,T\}$, 变分 EM 算法收敛后, 则:

1) user_i 属于类 cls_{z^i}, $z^i=\mathrm{argmax}_k\,\mathrm{E}[Z_{ik}]$;

2) cls_k 的成员集 $CM_k=\{\mathrm{user}_i|i=1,\cdots,I\wedge z^i=k\}$;

3) 有效用户类集 $EC=\{cls_k|cls_k\in C\wedge|CM_k|>\varepsilon\}$, 其中, ε 为一正整数.

定义 2. 设 Q 为项的属性特征集合, Q 中任一元素 $\boldsymbol{e}\in R^d$. 则集合 $\widetilde{Q}=\{\boldsymbol{e}_n|n=1,\cdots,d{+}1\wedge\boldsymbol{e}_n\in Q\}$ 称为 Q 的活跃子集当且仅当:

1) $\{\boldsymbol{e}_n|n=1,\cdots,d\}$ 为 R^d 的一组基;

2) $\boldsymbol{e}_{d+1}=\sum_{n=1}^{d}\kappa_n\boldsymbol{e}_n$ 且 $\sum_{n=1}^{d}\kappa_n\neq 1$.

活跃子集并非一定存在, 但在真实推荐系统中, 项基数极其庞大 ($|Q|\gg d$), $\widetilde{Q}$ 的存在性几乎是毋庸置疑的. 此外, 真实系统中除了极少数流行项外, 所有项的流行程度并无显著差别, 而且攻击概貌中目标项和选择填充项的数目大大少于填充项. 据此抽象出理想情况:

定义 3. 理想情况下, 活跃子集 $\widetilde{Q}$ 一定存在, 项因素均为 $\overline{H}$ 且攻击者仅评价填充项.

基于理想情况, 可以论证攻击者必然富集于某类, 且此类有明确的识别特征. 具体见引理 1 及定理 1:

引理 1. 设方程组:

$$\begin{cases}\boldsymbol{X}^{\mathrm{T}}\boldsymbol{e}_n=y, & n=1,\cdots,d\\ \boldsymbol{X}^{\mathrm{T}}\boldsymbol{e}_{d+1}=y\end{cases}$$

其中, $\{\boldsymbol{e}_n|n=1,\cdots,d{+}1\}$ 满足定义 2 中条件 (1) 和 (2), 则上述方程组有唯一解 $\boldsymbol{X}=\boldsymbol{0},y=0$.

证明. 存在性: $\boldsymbol{X}=\boldsymbol{0},y=0$ 显然是方程组的解; 唯一性: 若方程组有解 $\boldsymbol{X},y$, 由已知条件得:

$$\boldsymbol{e}_{d+1}=\sum_{n=1}^{d}\kappa_n\boldsymbol{e}_n\Rightarrow\boldsymbol{X}^{\mathrm{T}}\boldsymbol{e}_{d+1}=\sum_{n=1}^{d}\kappa_n\boldsymbol{X}^{\mathrm{T}}\boldsymbol{e}_n\Rightarrow$$

$$y\sum_{n=1}^{d}\kappa_n=y\Rightarrow y=0$$

$$\boldsymbol{X}=([\boldsymbol{e}_1,\boldsymbol{e}_2,\cdots,\boldsymbol{e}_d]^{\mathrm{T}})^{-1}\boldsymbol{1}y\Rightarrow\boldsymbol{X}=\boldsymbol{0}$$

从而方程组有唯一解 $\boldsymbol{X}=\boldsymbol{0}$, $y=0$. □

定理 1. 理想情况下, 攻击者的偏好特征必为 $\boldsymbol{0}$.

证明. 设 U^* 和 $\boldsymbol{G}^*$ 分别为攻击者 user^* 的用户因素与偏好特征. 填充项的随机选取表明每个填充项以等概率获得评分, 则由式 (4), 对 $\forall\boldsymbol{e}\in Q$ 有:

$$U^*+\overline{H}+\boldsymbol{G}^{*\mathrm{T}}\boldsymbol{e}=c$$

其中 c 为一常数. 显然, $\boldsymbol{G}^*=\boldsymbol{0},U^*=c-\overline{H}$ 满足上式; 此外, 限定 $\boldsymbol{e}\in\widetilde{Q}$, 由引理 1 得 $\boldsymbol{G}^*=\boldsymbol{0},U^*=c-\overline{H}$. 综上, 攻击者的偏好特征必为 $\boldsymbol{0}$. □

用户偏好的相似性是 LFAMR 的聚类依据, 定理 1 表明攻击者具有相同的偏好特征 ($\boldsymbol{0}$ 表示无兴趣偏好), 所以攻击者必然富集于某类. 进一步地, 由于真实用户通常具有特定的兴趣倾向, 其偏好特征不为 $\boldsymbol{0}$, 故较之真实用户类, 攻击类的类型中心应最接近原点. 现已知 cls_k 类型中心的近似值 $\mathrm{E}[\boldsymbol{\theta}_k]$, 根据上述讨论, 令 $k^*=\arg\min_k||\mathrm{E}[\boldsymbol{\theta}_k]||_2$, 其中, $||\cdot||_2$ 为 Frobenius 范数且 $k\in\{k'|cls_{k'}\in EC\}$, 则攻击类为 cls_{k^*}, 并认为 CM_{k^*} 中均为攻击者. 实验部分将证实此结论的有效性.

需注意, 这里仅在定义 1 规定的有效用户类集中识别攻击类, 阈值 ε 一般取为较小的正整数. 用户聚类时, 难免会出现一些基数较小的用户类, 这些类产生的原因通常是由于算法优化过程陷入局部极值或存在极少数兴趣偏好异常的用户. 之前提到, 用户类的代表性偏好特征反映于类型中心, 它是类中大量用户的统计规律. 如果某用户类基数过小, 则其类型中心仅由类中的极少数用户决定, 此时, 类型中心的代表性将大打折扣. 这些类的存在会对正确识别攻击类造成一定干扰. 所以, 认为类基数不超过阈值 ε 的用户类为噪声类, 在识别攻击类时不予考虑.

4 实验结果及分析

4.1 实验设置及探测性能评价指标

实验选用了由美国 Minnesota 大学 GroupLens 研究组发布的并获得广泛应用的两个数据集: MovieLens100K 与 MovieLens1M. 两者的概况见表 1. 假定数据集中原有用户为真实用户. 在不同的攻击强度 p^{att} 与填充率 p^{fill} 下, 分别向数据集注入 4 种典型托攻击 (如无特别说明, 默认注入推攻击).

表 1　数据集概况

Table 1　Data set statistics

	MovieLens100K	MovieLens1M
用户数	943	6 040
项 (电影) 数	1 682	3 900
总评分数	100 000	1 000 209

LFAMR 模型的先验参数设定为: $w_1 = w_2 = \nu_1 = 1, \nu_2 = 10^{-3}, \Lambda_0 = \Lambda_1 = \Lambda_2 = \boldsymbol{I}$ (单位矩阵), $\tau_1^2 = \tau_2^2 = 0.1$, 并且取元数 $d = 5$, 截断值 $T = 20$, 阈值 $\varepsilon = 2$. 所有实验均在一台 Intel Pentium Dual CPU 1.60 GHz 的 PC 机上进行, 程序采用 Python + MySQL 实现.

托攻击探测效果的评价使用了准确率 f_p 与召回率 f_r 的综合指标 F 值[23]. 设 N, N^a, N^t 分别为攻击类中的用户数, 攻击类中的攻击者数以及系统中的攻击者总数, 则:

$$\begin{cases} f_p = \dfrac{N^a}{N} \\ f_r = \dfrac{N^a}{N^t} \\ F = \dfrac{2 f_p f_r}{f_p + f_r} \end{cases}$$

4.2　托攻击探测示例

下面通过实例展示 LFAMR 的用户聚类过程以及攻击类的识别. 向 MovieLens100K 数据集注入参数为 $p^{\text{att}} = 10\,\%$, $p^{\text{fill}} = 20\,\%$ 的随机攻击. 为便于图示, 在每次迭代后, 利用 Fisher 判别法[24] 将用户的期望偏好特征 ($\mathrm{E}[\boldsymbol{G}_i]$) 投影至二维空间. 图 4 上半部显示迭代次数为 $0, 3, 7, 23$ 时的聚类状况, 下半部显示各用户类在对应迭代次数时的混合系数, cls_k 的混合系数为 $\mathrm{E}[V_k]\prod_{n=1}^{k-1}(1-\mathrm{E}[V_n])$. Dirichlet 过程通过调整用户类的混合系数, 保留能兼顾数据拟合与泛化性能的用户类. 由于采用了低维投影技术, 所以多数用户类在图中重叠于一起, 界限模糊. 然而, 可以明显观察到, 攻击者逐渐从真实用户的背景中分离出来, 向原点附近聚拢, 并于第 23 次迭代时在 cls_{10} 类高度富集, 此类即为要识别的攻击类. 经计算得知 cls_{10} 的类型中心的确最接近原点, 印证了攻击类识别方法的有效性. 特别地, 从图 4 可看出 cls_{10} 也最接近这个平面坐标系的原点. 同时, cls_{10} 的混合系数, 即类中用户占所有用户的比重为 0.09, 这也吻合攻击者的真实比重 ($p^{\text{att}}/(1+p^{\text{att}}) \approx 0.09$).

4.3　探测结果分析

本文首先在 MovieLens100K 数据集上检验 LFAMR 的攻击探测能力. 实验采取 $4 \times 4 \times 6$ 的设计模式, 攻击模型 (随机攻击、均值攻击、流行攻击、段攻击), 攻击强度 p^{att} ($5\,\%, 7\,\%, 10\,\%, 12\,\%$) 和填充率 p^{fill} ($3\,\%, 6\,\%, 9\,\%, 12\,\%, 15\,\%, 20\,\%$) 的不同组合对应一组实验配置. 每组配置下的实验结果取自十次独立实验的均值. 这里选用 PCA VarSelect, PLSA、EMSVD 与 UnRAP 算法作为 LFAMR 的性能参照. 目前, PCA VarSelect 在此数据集上具备最佳的探测性能.

表 2　探测随机攻击的 F 值

Table 2　F score for detecting random attack

p^{att} \ p^{fill}	3 %	6 %	9 %	12 %	15 %	20 %
5 %	0.69 (0.98)	0.98 (0.99)	0.99 (0.98)	0.99 (0.99)	0.98 (0.99)	0.99 (0.99)
7 %	0.87 (0.98)	0.99 (0.99)	0.99 (0.99)	0.99 (0.99)	0.99 (0.99)	0.99 (0.99)
10 %	0.80 (0.99)	0.99 (0.99)	0.99 (0.99)	0.99 (0.99)	0.99 (0.99)	0.99 (0.99)
12 %	0.62 (0.99)	1.00 (0.99)	0.99 (0.99)	0.99 (0.99)	0.99 (0.99)	0.99 (0.99)

表 3　探测均值攻击的 F 值

Table 3　F score for detecting average attack

p^{att} \ p^{fill}	3 %	6 %	9 %	12 %	15 %	20 %
5 %	0.75 (0.98)	0.98 (0.98)	0.99 (0.97)	0.99 (0.97)	0.99 (0.97)	0.98 (0.96)
7 %	0.88 (0.97)	1.00 (0.98)	0.99 (0.98)	0.99 (0.98)	0.99 (0.97)	0.99 (0.97)
10 %	0.79 (0.98)	0.99 (0.98)	1.00 (0.98)	1.00 (0.98)	0.99 (0.98)	0.99 (0.97)
12 %	0.71 (0.98)	0.99 (0.98)	1.00 (0.98)	1.00 (0.98)	0.99 (0.98)	0.99 (0.97)

表 4　探测流行攻击的 F 值

Table 4　F score for detecting bandwagon attack

p^{att} \ p^{fill}	3 %	6 %	9 %	12 %	15 %	20 %
5 %	0.60 (0.96)	0.99 (0.97)	0.98 (0.98)	0.99 (0.98)	0.99 (0.98)	0.98 (0.99)
7 %	0.61 (0.97)	0.99 (0.98)	0.99 (0.98)	0.99 (0.99)	0.99 (0.99)	0.99 (0.99)
10 %	0.70 (0.96)	0.99 (0.98)	0.99 (0.99)	0.99 (0.99)	0.99 (0.99)	0.99 (0.99)
12 %	0.71 (0.97)	0.99 (0.98)	0.99 (0.99)	0.99 (0.99)	1.00 (0.99)	0.99 (0.99)

表 5 探测段攻击的 F 值

Table 5 F score for detecting segment attack

p^{att}	p^{fill}					
	3 %	6 %	9 %	12 %	15 %	20 %
5 %	0.51 (0.00)	0.98 (0.00)	1.00 (0.28)	0.99 (0.58)	0.99 (0.68)	0.98 (0.78)
7 %	0.54 (0.00)	0.99 (0.00)	0.99 (0.00)	0.99 (0.00)	0.99 (0.29)	0.99 (0.64)
10 %	0.63 (0.00)	0.99 (0.00)	0.99 (0.00)	0.99 (0.00)	0.99 (0.00)	0.99 (0.00)
12 %	0.63 (0.00)	0.99 (0.00)	1.00 (0.00)	0.99 (0.00)	0.99 (0.00)	0.99 (0.00)

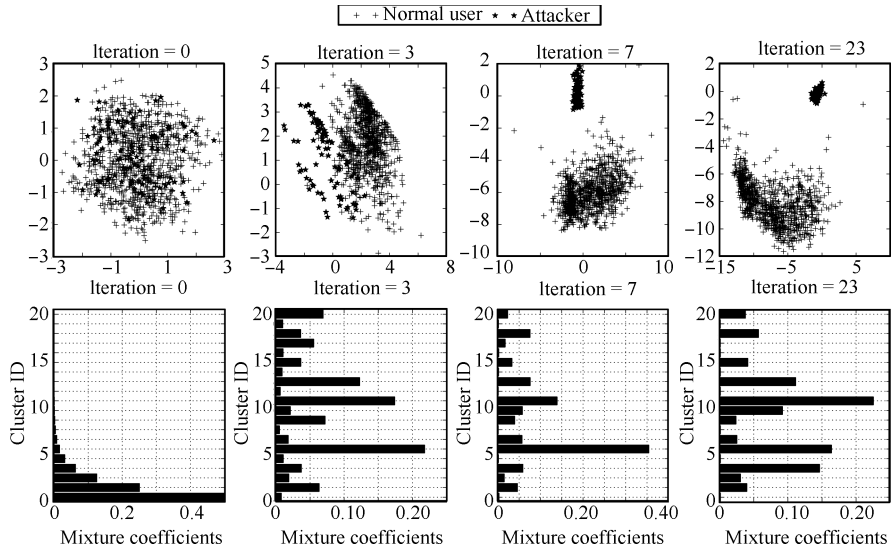


图 4 LFAMR 的应用示例

Fig. 4 Illustrative application of LFAMR

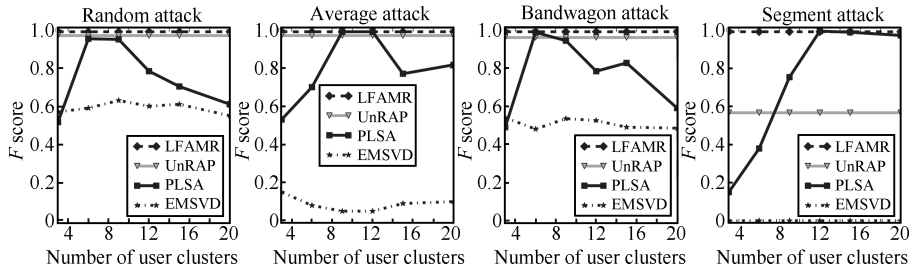


图 5 探测性能对输入参数的依赖

Fig. 5 Dependence of the detection performance on input parameter

表 2～5 括号外的数据展示了 LFAMR 的探测效果. 仅当填充率为 3 % 时, LFAMR 的探测能力受限, 因为此时填充率较低, 目标项和选择填充项的数目与填充项相差并不悬殊, 不能简单假定攻击者仅评价填充项, 这与定义 3 的理想情况存在一定差距, 所以定理 1 不再适用. 而其余情况下, 算法均取得了良好的探测性能, F 值基本位于 98 % 以上, 显示了 LFAMR 能够准确探测出近乎所有攻击者.

表 2～5 括号内的数据为 PCA VarSelect 在对应实验配置下达到最佳探测性能时的 F 值. 可看出, 在大多数情况下, LFAMR 的探测能力要优于 PCA VarSelect. 特别地, PCA VarSelect 面临段攻击时几乎失效, 而 LFAMR 的有效探测范围则涵盖了这 4 种攻击模型. 需强调的是, PCA VarSelect 达到最佳探测性能的前提是必须获知准确的攻击强度, 否则会严重降低 F 值[11]. 类似地, PLSA、EMSVD 与 UnRAP 也不同程度地存在普适性或无监督性方面的局限. 图 5 展示了 4 种攻击模型在 $p^{\text{att}} = 10\,\%$, $p^{\text{fill}} = 6\,\%$ 时, LFAMR、PLSA、EMSVD 与 UnRAP 算法的探测性能对比. PLSA 与 EMSVD 都需要用户类别数作为输入参数, 而此参数一般只能通过试探法获得. 易看出, 在不同的用户类别数下, PLSA 的探测性能对于输入的变化非常敏感, 不能保证用同样的用户类别数在所有攻击下都取得最佳探测效果. EMSVD 的探测性能对输入参数的敏感性虽弱于 PLSA, 但

它无法有效探测均值攻击, 而面临段攻击时则完全失效. 相比之下, LFAMR 能有效探测 4 种托攻击, 且无需根据攻击情境的变化调整输入参数, 无监督程度较高. UnRAP 的无监督程度与 LFAMR 相当, 但普适性稍弱, 无法有效探测段攻击, 不过在其余三种攻击模型下, 其探测能力与 LFAMR 相差无几.

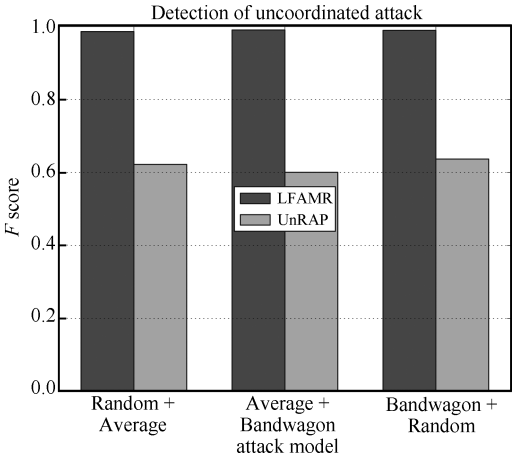


图 6　LFAMR 与 UnRAP 对非协调攻击的探测性能比较

Fig. 6　Comparison between LFAMR and UnRAP under uncoordinated attack

为更细粒度展示 LFAMR 与 UnRAP 的差异, 实验从 MovieLens1M 数据集中随机选取四分之一 (约 1 510 个) 的用户, 在非协调攻击 (Uncoordinated attack)[25] 情形下进一步评测两者的探测能力. 非协调攻击是指推荐系统中同时存在多种托攻击, 且每种攻击的目标项一般互不相同. 真实环境中的托攻击有相当一部分是非协调攻击. 图 6 展示了这两种算法对非协调攻击的探测能力, 图中每种攻击的参数配置为 $p^{\mathrm{att}}=5\,\%$, $p^{\mathrm{fill}}=6\,\%$. 易看出, 在所有攻击情形下, LFAMR 算法表现均衡, F 值基本位于 98 % 以上, 探测性能明显优于 UnRAP 算法, 后者的 F 值仅达到 60 % 左右. 这是因为 UnRAP 算法每次探测仅能关注单个目标项, 故无法有效应对多目标项的情况, 而 LFAMR 算法则无需考虑目标项. 所以 LFAMR 算法能有效探测非协调攻击, 进一步显示出其普适性方面的优势.

5　结论

本文提出的缺失评分潜在因素分析 (LFAMR) 模型, 以数据非随机缺失机制为构建基础, 充分利用了 Dirichlet 过程的聚类效应, 通过进行用户聚类并揭示攻击类的识别特征, 达到了探测托攻击的目的. 较之现有探测技术, LFAMR 具备更强的普适性及无监督性, 能够准确可靠地探测各种常见攻击, 且无需用户类数或攻击强度等先验输入, 显示出较高的实用价值.

附录 A

$$w_1' = w_1 + T - 1$$

$$w_2' = w_2 - \sum_{k=1}^{T-1} \mathrm{E}[\ln(1-V_k)]$$

$$\nu_1' = \nu_1 + \frac{1}{2}dI$$

$$\nu_2' = \nu_2 + \frac{1}{2}\sum_{i=1}^{I}\sum_{k=1}^{T} \mathrm{E}[Z_{ik}]\mathrm{E}[(\boldsymbol{G}_i - \boldsymbol{\theta}_k)^{\mathrm{T}}\Lambda_2(\boldsymbol{G}_i - \boldsymbol{\theta}_k)]$$

$$\alpha_1^k = 1 + \sum_{i=1}^{I}\mathrm{E}[Z_{ik}]$$

$$\alpha_2^k = \mathrm{E}[\alpha] + \sum_{i=1}^{I}\sum_{m=k+1}^{T}\mathrm{E}[Z_{im}]$$

$$s_k^i = \mathrm{E}[\ln(V_k)] + \sum_{n=1}^{k-1}\mathrm{E}[\ln(1-V_n)] - \frac{d}{2}\ln(2\pi) + \frac{1}{2}\ln(|\Lambda_2|) + \frac{d}{2}\mathrm{E}[\ln(\gamma)] - \frac{1}{2}\mathrm{E}[(\boldsymbol{G}_i - \boldsymbol{\theta}_k)^{\mathrm{T}}\gamma\Lambda_2(\boldsymbol{G}_i - \boldsymbol{\theta}_k)]$$

$$\Lambda_\theta^{k\,-1} = \Lambda_0 + \mathrm{E}[\gamma]\Lambda_2\sum_{i=1}^{I}\mathrm{E}[Z_{ik}]$$

$$\boldsymbol{\mu}_\theta^k = \Lambda_\theta^k\mathrm{E}[\gamma]\Lambda_2\sum_{i=1}^{I}\mathrm{E}[\boldsymbol{G}_i]\mathrm{E}[Z_{ik}]$$

$$\Lambda_x^{j\,-1} = \Lambda_1 + 2\sum_{i=1}^{I}\lambda(\xi_{ij})\mathrm{E}[\boldsymbol{G}_i\boldsymbol{G}_i^{\mathrm{T}}]$$

$$\boldsymbol{\mu}_x^j = \Lambda_x^j\sum_{i=1}^{I}\mathrm{E}[\boldsymbol{G}_i](M_{ij} - \frac{1}{2} - 2\lambda(\xi_{ij})(\mathrm{E}[H_j] + \mathrm{E}[U_i]))$$

$$\tau_h^{j\,-1} = \tau_1^{-2} + 2\sum_{i=1}^{I}\lambda(\xi_{ij})$$

$$\mu_h^j = \tau_h^j\sum_{i=1}^{I}(M_{ij} - \frac{1}{2} - 2\lambda(\xi_{ij})(\mathrm{E}[\boldsymbol{X}_j^{\mathrm{T}}]\mathrm{E}[\boldsymbol{G}_i] + \mathrm{E}[U_i]))$$

$$\Lambda_g^{i\,-1} = \mathrm{E}[\gamma]\Lambda_2\sum_{k=1}^{T}\mathrm{E}[Z_{ik}] + 2\sum_{j=1}^{J}\lambda(\xi_{ij})\mathrm{E}[\boldsymbol{X}_j\boldsymbol{X}_j^{\mathrm{T}}]$$

$$\boldsymbol{\mu}_g^i = \Lambda_g^i(\mathrm{E}[\gamma]\Lambda_2\sum_{k=1}^{T}\mathrm{E}[Z_{ik}]\mathrm{E}[\boldsymbol{\theta}_k] + \sum_{j=1}^{J}\mathrm{E}[\boldsymbol{X}_j]\times (M_{ij} - \frac{1}{2} - 2\lambda(\xi_{ij})(\mathrm{E}[H_j] + \mathrm{E}[U_i])))$$

$$\tau_u^{i\,-1} = \tau_2^{-2} + 2\sum_{j=1}^{J}\lambda(\xi_{ij})$$

$$\mu_u^i = \tau_u^i\sum_{j=1}^{J}(M_{ij} - \frac{1}{2} - 2\lambda(\xi_{ij})(\mathrm{E}[\boldsymbol{X}_j^{\mathrm{T}}]\mathrm{E}[\boldsymbol{G}_i] + \mathrm{E}[H_j]))$$

References

1 Adomavicius G, Tuzhilin A. Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, 2005, **17**(6): 734−749

2 Su X Y, Khoshgoftaar T M. A survey of collaborative filtering techniques. *Advances in Artificial Intelligence*, 2009, **2009**: 1−20

3 Mobasher B, Burke R, Bhaumik R, Sandvig J J. Attacks and remedies in collaborative recommendation. *IEEE Intelligent Systems*, 2007, **22**(3): 56−63

4 Lam S K, Riedl J. Shilling recommender systems for fun and profit. In: Proceedings of the 13th International Conference on World Wide Web. New York, USA: ACM, 2004. 393−402

5 O'Mahony M P, Hurley N J, Kushmerick N, Silvestre G C M. Collaborative recommendation: a robustness analysis. *ACM Transactions on Internet Technology*, 2004, **4**(4): 344−377

6 Huang Z, Chen H, Zeng D. Applying associative retrieval techniques to alleviate the sparsity problem in collaborative filtering. *ACM Transactions on Information Systems*, 2004, **22**(1): 116−142

7 Mobasher B, Burke R D, Bhaumik R, Williams C. Toward trustworthy recommender systems: an analysis of attack models and algorithm robustness. *ACM Transactions on Internet Technology*, 2007, **7**(4): 1−40

8 Burke R, Mobasher B, Williams C, Bhaumik R. Classification features for attack detection in collaborative recommender systems. In: Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Philadelphia, Pennsylvania, USA: ACM, 2006. 542−547

9 Zhang S, Ouyang Y, Ford J, Makedon F. Analysis of a low-dimensional linear model under recommendation attacks. In: Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Seattle, Washington, USA: ACM, 2006. 517−524

10 Mehta B, Nejdl W. Unsupervised strategies for shilling detection and robust collaborative filtering. *User Modeling and User-Adapted Interaction*, 2009, **19**(1−2): 65−97

11 Mehta B, Hofmann T, Fankhauser P. Lies and propaganda: detecting spam users in collaborative filtering. In: Proceedings of the 12th International Conference on Intelligent User Interfaces. Honolulu, Hawaii: ACM, 2007. 14−21

12 Bryan K, O′Mahony M P, Cunningham P. Unsupervised retrieval of attack profiles in collaborative recommender systems. In: Proceedings of the 2008 ACM Conference on Recommender Systems. Lausanne, Switzerland: ACM, 2008. 155−162

13 Little R J A, Rubin D B. *Statistical Analysis with Missing Data*. New York: John Wiley, 1987. 13−17

14 Ferguson T S. A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1973, **1**(2): 209−230

15 MacKay D J C. Bayesian interpolation. *Neural Computation*, 1992, **4**(3): 415−447

16 Frigyik B A, Kapila A, Gupta M R. Introduction to the Dirichlet Distribution and Related Processes, Technical Report, Department of Electrical Engineering, University of Washington, 2010

17 Sethuraman J. A constructive definition of Dirichlet priors. *Statistica Sinica*, 1994, **4**: 639−650

18 Antoniak C E. Mixtures of Dirichlet processes with applications to Bayesian nonparametric problems. *The Annals of Statistics*, 1974, **2**(6): 1152−1174

19 Attias H. Inferring parameters and structure of latent variable models by variational bayes. In: Proceedings of the 15th Conference on Uncertainty in Artificial Intelligence. San Francisco, CA, USA: Morgan Kaufmann, 1999. 21−30

20 Jaakkola T S, Jordan M I. Bayesian parameter estimation via variational methods. *Statistics and Computing*, 2000, **10**(1): 25−37

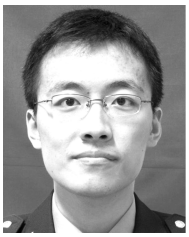
21 Ishwaran J, James L F. Gibbs sampling methods for stick-breaking priors. *Journal of the American Statistical Association*, 2001, **96**(453): 161−173

22 Blei D M, Jordan M I. Variational inference for Dirichlet process mixtures. *Bayesian Analysis*, 2006, **1**(1): 121−144

23 Lewis D D, Gale W A. A sequential algorithm for training text classifiers. In: Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. Dublin, Ireland: Springer, 1994. 3−12

24 Fisher R A. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 1936, **7**(2): 179−188

25 Mehta B, Hofmann T. A survey of attack-resistant collaborative filtering algorithms. *IEEE Data Engineering Bulletin*, 2008, **31**(2): 14−22



李 聪 国防科学技术大学计算机学院博士研究生. 主要研究方向为机器学习, 人工智能与信息检索. 本文通信作者.
E-mail: licongwhy@gmail.com
(**LI Cong** Ph. D. candidate at the School of Computer, National University of Defense Technology. His research interest covers machine learning, artificial intelligence, and information retrieval. Corresponding author of this paper.)

骆志刚 国防科学技术大学计算机学院教授. 主要研究方向为高性能计算, 数据挖掘与生物信息学.
E-mail: zgluo@nudt.edu.cn
(**LUO Zhi-Gang** Professor at the School of Computer, National University of Defense Technology. His research interest covers high performance computing, data mining, and bioinformatics.)