一种探测推荐系统托攻击的无监督算法

李 聪 [1] 骆志刚 [1] 石金龙 [1]

摘 要 托攻击是当前推荐系统面临的重大安全性问题之一. 开发托攻击探测算法已成为保障推荐系统准确性与鲁棒性的关键. 针对现有托攻击探测算法无监督程度较低的局限, 在引入攻击概貌群体效应的定量度量及基于此的遗传优化目标函数的基础上, 将自适应参数的后验推断与攻击探测过程相融合, 提出了迭代贝叶斯推断遗传探测算法, 降低了算法探测性能对系统相关先验知识的依赖. 实验结果显示这种算法能够有效探测各种常见攻击.

关键词 推荐系统, 托攻击, 群体效应, 遗传算法, 贝叶斯推断

DOI 10.3724/SP.J.1004.2011.00160

An Unsupervised Algorithm for Detecting Shilling Attacks on Recommender Systems

LI Cong[1] LUO Zhi-Gang[1] SHI Jin-Long[1]

Abstract Shilling attack is one of the significant security problems involved in recommender systems. Developing detection algorithms against shilling attacks has become the key to guaranteeing both the preciseness and robustness of recommender systems. Considering the low degree of unsupervised features the existing algorithms suffer from, this paper proposes an iterative Bayesian inference genetic detection algorithm (IBIGDA) through the introduction of the quantitative metric for the group effect of attack profiles and the corresponding object function for genetic optimization. This algorithm combines the posterior inference for the adaptive parameters with the process of attack detection, thus relaxes the dependence of the detection performance on the relating prior knowledge of the systems. Experimental results show that this algorithm can effectively detect shilling attacks of typical types.

Key words Recommender system, shilling attack, group effect, genetic algorithm, Bayesian inference

推荐系统能主动推送终端用户感兴趣的信息, 显著降低信息检索的工作强度, 极大缓解了信息技术飞速发展导致的信息过载问题. 目前, 协同过滤 (Collaborative filtering) 推荐技术的发展最为完善, 应用最为广泛[1], Amazon[1], eBay[2], NetFlix[3] 等大型商业站点都应用了此项技术.

协同过滤的决策依据源于对评分矩阵 $R_{m\times n}$ 的分析与挖掘, $R_{m\times n}=[\boldsymbol{u}_1,\boldsymbol{u}_2,\cdots,\boldsymbol{u}_m]^{\mathrm{T}}$, 其中 $\boldsymbol{u}_i$ 称为用户概貌 (User profile), 它包含了用户 i 对系统中 n 个项 (书籍, 音乐, 电影等) 的评分[2]. 一般地, 协同过滤系统通过比对用户概貌, 找寻与目标用户兴趣关联度较大的用户, 称为最近邻, 之后向目标用户推荐其最近邻评价较高的项[3]. 这种工作模式有很大的安全隐患, 攻击者可以伪造用户概貌, 成为大量用户的最近邻, 致使系统产生虚假的推荐信息, 进而非法获益[4−5]. 这种攻击被称为 "托攻击 (Shilling attacks)" 或 "概貌注入攻击 (Profile injection attacks)"[6−7]. 托攻击的防御技术近年来引起了学界的广泛关注, 并已成为当前推荐系统领域的研究热点之一. 当前较为普遍的解决思路是利用有监督或无监督的探测技术, 剔除攻击者, 消除其不良影响.

应当注意, 有监督探测算法仅具理论价值, 因为实际应用中难以拟出合适的训练集来构造分类器. 无监督探测算法则兼具较少先验需求与较强泛化能力, 更符合托攻击防御的实际情境. 然而, 无监督只是相对概念, 少量的先验输入必不可少, 而且先验知识的准确与否通常是决定探测性能的关键.

为了进一步降低无监督探测算法对先验输入的依赖, 提高算法的无监督性, 本文提出了托攻击的迭代贝叶斯推断遗传探测算法 (Iterative Bayesian inference genetic detection algorithm, IBIGDA), 其主要思路包括: 1) 根据真实用户以及攻击用户在大尺度上的评分规律, 考察用户概貌之间相关性的分布特性, 导出基于广义方差 (Generalized variance) 的攻击者群体效应 (Group effect) 定量度量; 2) 在 1) 的基础上, 构建遗传优化目标函数 (即适应度函数), 此函数的全局最大值在理想状态下标志着探测效果达到最优; 3) 将贝叶斯推断思想融入遗传优化过程, 视目标函数的参数为自适应参数, 根据每步迭

收稿日期 2010-07-27 录用日期 2010-10-22
Manuscript received July 27, 2010; accepted October 22, 2010
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/
[3]http://www.netflix.com/

代优化的结果进行后验更新, 并以先验参数的身份参加下次迭代, 直至达到最佳探测效果.

实验结果表明 IBIGDA 算法在各种常见攻击配置下均可达到较好的探测水准, 且不需准确的先验输入.

1　推荐系统安全性问题及相关研究

协同过滤推荐的准确性有赖于大量用户的参与, 因而推荐系统无法过度限制用户的操作. 攻击者能以极低的代价向系统中恶意注入虚假概貌, 这些概貌的构建方式可使系统中大量用户的兴趣与攻击者相近, 少量的攻击概貌就能有效操纵推荐结果. 托攻击有两个目的: 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)[6]. 实际情况中推攻击更为普遍. 托攻击不仅涉及道德问题, 更能引起不可控的后果, 所以为推荐系统提供有效的防攻击手段变得愈加紧迫而必要.

1.1　推荐系统的托攻击

攻击概貌依照不同的攻击模型构建, 攻击模型是根据推荐系统中用户, 项以及评分等信息来构建攻击概貌的方式[8]. 攻击概貌是一个 n 维向量, n 是系统中项的个数. 图 1 是攻击概貌的形式结构.

图 1　攻击概貌的形式结构

Fig. 1　The general framework of attack profiles

目标项 (Target item) 作为攻击的目标, 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 ø, 即不予评分, 实际上未评分项广泛存在于推荐系统中, 因为用户一般只对少数项评分, 这就造成了评分矩阵 $R_{m\times n}$ 的极端稀疏性. 不同攻击模型的差异主要体现在对填充项 (Filler items) 与选择填充项 (Selected items) 的选取与评分策略上. 比例 $p^{\text{fill}} = |\{\text{filler items}\}|/n$ 为填充率 (Filler size), p^{fill} 不宜过大, 否则会降低攻击效果. 典型地, $p^{\text{fill}} \in (1\,\%, 15\,\%)$. 随机攻击 (Random attack), 均值攻击 (Average attack) 和流行攻击 (Bandwagon attack)[9−10] 是三种典型的攻击模型. 随机攻击易于部署, 只需攻击者粗略了解系统中的评分分布, 但其攻击效果不甚理想. 均值攻击的攻击效果较好, 但需攻击者细致掌握每个项的评分分布, 部署成本较高. 流行攻击是前两者的折中, 它在未显著加大部署成本的同时获得了能与均值攻击媲美的攻击效果.

比例 $p^{\text{att}} = |\{\text{攻击用户 }\}|/|\{\text{ 真实用户 }\}|$ 代表了攻击强度 (Attack size), p^{att} 不宜过大, 否则易于被探测. 典型地, $p^{\text{att}} \in (0\,\%, 20\,\%)$. 定义 $p^{\text{true}} = p^{\text{att}}/(1+p^{\text{att}})$, 代表攻击者在所有用户中所占比例.

1.2　无监督探测算法的相关研究

鉴于有监督探测算法的内在局限, 相关研究更多集中于无监督探测算法.

Zhang 等[11] 利用奇异值分解 (Singular value decomposition, SVD) 与期望最大化 (Expectation maximum, EM) 算法为评分矩阵构建低维线性模型, 用其计算每个用户概貌的产生概率, 认为概率越低, 成为攻击者的嫌疑越大. 本文暂将这种探测算法称为 EMSVD 算法. Mehta 等[12−13] 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 是一种生成模型 (Generative model), 可以对用户进行软聚类, 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. 鲁棒推荐算法 VarSelect SVD[14] 即将 PCA VarSelect 算法作为前端模块, 以标识可疑用户, 抑制其对推荐算法的干扰.

EMSVD 与 PLSA 探测算法需要输入用户类的个数, PCA VarSelect 算法需事先获知准确的攻击强度, 而这些对探测性能有显著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.

2　迭代贝叶斯推断遗传探测算法

这部分将详细介绍 IBIGDA 算法的构建过程, 包括攻击概貌群体效应定量度量的引入, 遗传优化目标函数的建立, 算法的具体流程及其他相关内容. 进一步讨论之前, 任一用户概貌 $\boldsymbol{u}_i$ 都需经过预处理. 预处理有两步: 首先, 用 0 代替 $\boldsymbol{u}_i$ 中缺失值 ø, 得到 $\boldsymbol{u}_i'$; 其次, 正规化 $\boldsymbol{u}_i'$, 得到 $\boldsymbol{u}_i''$, 使 $\boldsymbol{u}_i''^{\mathrm{T}}\boldsymbol{u}_i'' = 1$ 且 $\mathbf{1}^{\mathrm{T}}\boldsymbol{u}_i'' = 0$. 方便起见，仍用 $\boldsymbol{u}_i$ 表示 $\boldsymbol{u}_i''$.

2.1　存在攻击时用户概貌之间的统计特征

现为 MovieLens 100K 数据集 (实验部分将介绍) 注入随机攻击, 参数配置为: $p^{\text{att}} = 10\,\%$, $p^{\text{fill}} = 8\,\%$. 图 2 展示了此时用户概貌之间相关系数的分布状况. A, B, C 分别代表了真实概貌之间, 攻击概貌之间, 真实概貌与攻击概貌之间相关系数的分布状况.

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

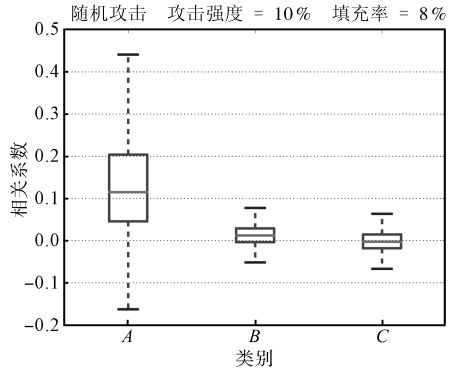


图 2 用户概貌之间相关系数的分布

Fig. 2 The distribution of correlation coefficients between profiles

预处理后, 任意两个用户概貌 $\boldsymbol{u}_i, \boldsymbol{u}_j$ 的相关系数等于它们在 n 维空间夹角的余弦值, 即 $\rho(\boldsymbol{u}_i, \boldsymbol{u}_j) = \cos(\boldsymbol{u}_i, \boldsymbol{u}_j)$. 分布 A 集中在 0.1 左右, 而分布 B, C 集中于 0 左右, 表明真实用户概貌之间的夹角为锐角, 而攻击用户概貌则近似互相垂直. 进一步实验显示, 这种概貌间的角度关系同样出现于典型参数配置下的均值攻击和流行攻击情形中. 直观上, 若 n 维空间的标准基 $[\boldsymbol{e}_1, \boldsymbol{e}_2, \cdots, \boldsymbol{e}_n]$ 表示 n 个不相关兴趣点, 则真实用户之间存在兴趣爱好的关联, 而攻击用户的兴趣则相对分散. 这种现象存在合理的解释[15]: 真实用户的评分对象一般是较为流行的项, 而非流行项获得的评分很少, 所以概貌间重合度高; 攻击用户的填充项来源于随机选取[9], 每个项以等概率获得评分, 所以重合度低.

为简化问题的讨论, 根据用户概貌之间的统计特征, 给出理想情况的定义:

定义 1. 设 K^f, K^t 分别为预处理后的攻击用户概貌与真实用户概貌的集合, 理想情况下, 有

$$\begin{cases} \cos(\boldsymbol{u}, \boldsymbol{v}) = 0, & \boldsymbol{u}, \boldsymbol{v} \in K^f,\ \boldsymbol{u} \neq \boldsymbol{v} \\ \cos(\boldsymbol{u}, \boldsymbol{v}) = 0, & \boldsymbol{u} \in K^f,\ \boldsymbol{v} \in K^t \\ \cos(\boldsymbol{u}, \boldsymbol{v}) = \alpha, & \boldsymbol{u}, \boldsymbol{v} \in K^t,\ \boldsymbol{u} \neq \boldsymbol{v},\ 0 < \alpha < 1 \end{cases}$$

下文中如无特别说明, 默认基于理想情况讨论问题.

2.2 广义方差诱导的攻击概貌群体效应度量

如前所述, 攻击模型决定了单个攻击概貌兴趣点的分散性, 这是攻击概貌的个体效应, 也是单个用户概貌的嫌疑性指标. 一方面, 仅利用概貌的个体效应进行攻击探测是不严谨的, 因为少量兴趣广泛的真实用户也具备这种效应, 而且这些用户的存在有利于增进推荐结果的新颖性[16]; 另一方面, 若具备这种个体效应的用户概貌多于一定数量, 且相互之间兴趣重合度低, 那么这种群体效应使我们有理由怀疑系统中出现了托攻击. 为将此思想转化为探测手段, 需对群体效应进行定量描述, 广义方差则提供了一个较好的解决途径. 广义方差定义为协方差矩阵的行列式. 设推荐系统中所有用户组成一个多元随机变量, 每个项所获评分是这个随机变量的观察值. 预处理后的评分矩阵与协方差矩阵有如下关系:

$$S_{m\times m} = \frac{1}{n-1} R_{m\times n} R_{m\times n}^{\mathrm{T}} \tag{1}$$

则广义方差为 $\det(S_{m\times m})$.

广义方差有优良的几何解释, 设 m 个用户概貌在 n 维空间围成的超体积为 V, 则 $\det(S_{m\times m})$ 与 V 满足关系[17]:

$$\det(S_{m\times m}) = (n-1)^{-m} V^2 \tag{2}$$

将式 (2) 代入式 (1), 得:

$$\begin{aligned} &\det\left(\frac{1}{n-1} R_{m\times n} R_{m\times n}^{\mathrm{T}}\right) = (n-1)^{-m} V^2 \Rightarrow \\ &(n-1)^{-m} \det(R_{m\times n} R_{m\times n}^{\mathrm{T}}) = (n-1)^{-m} V^2 \Rightarrow \\ &V = \sqrt{\det(R_{m\times n} R_{m\times n}^{\mathrm{T}})} \Rightarrow \\ &V = \sqrt{\det(X_{m\times m})} \quad (X_{m\times m} = R_{m\times n} R_{m\times n}^{\mathrm{T}}) \end{aligned}$$

可以证明: 预处理后, 当所有用户概貌位于同一超平面时, 广义方差为 0, 超体积最小; 当所有用户概貌互相垂直时, 广义方差最大, 超体积最大. 由于用户概貌长度为 1, 所以超体积最大值为 1.

例如:

$$R_{3\times 3} = [\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3]^{\mathrm{T}} = \begin{bmatrix} \frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{bmatrix}$$

易知 $\boldsymbol{u}_1, \boldsymbol{u}_2, \boldsymbol{u}_3$ 共面, 它们围成的超体积为

$$V = \sqrt{\det(X_{m\times m})} = \sqrt{\det\left(\begin{bmatrix} 1 & -\frac{1}{2} & -\frac{1}{2} \\ -\frac{1}{2} & 1 & -\frac{1}{2} \\ -\frac{1}{2} & -\frac{1}{2} & 1 \end{bmatrix}\right)} = 0$$

由于攻击概貌互相垂直, 显然它们围成的超体积 $V = 1$; 而真实概貌之间夹角为锐角, 故它们围成的超体积 $V \in [0, 1)$. 然而超体积作为有量纲的标量, 由不同个数的概貌围成的超体积具有不同级的量纲, 之间不存在可比性. 为解决此问题, 定义平

均超体积贡献度 (Average contribution degree for hyper-volume, Advo) 函数:

定义 2. 设 K 为推荐系统中用户概貌的一个子集, 其围成的超体积为 V 且 $|K|=m$, 则平均超体积贡献度函数 Advo$(\cdot)$ 定义为: $\mathrm{Advo}(K)=\sqrt[m]{V}=\sqrt[2m]{\det(X_{m\times m})}$.

为回避个体效应, 通常只考虑基数较大的概貌集合, 此时有如下定理:

定理 1. 设 K 为推荐系统中用户概貌的一个子集, 其中攻击概貌的比例 $\gamma\in[0,1]$. 当 $|K|=m\to\infty$ 时, Advo(K) 是关于 γ 的单增函数, 且最大值为 1.

证明. 设 K 中用户概貌构成评分矩阵 $R_{m\times n}$, 令 $X_{m\times m}=R_{m\times n}R_{m\times n}^{\mathrm{T}}$. $X_{m\times m}$ 经过有限次行列对称变换可得 $X'_{m\times m}$:

$$X'_{m\times m}=\begin{bmatrix}A_{m\gamma'\times m\gamma'} & \mathbf{0}\\ \mathbf{0} & I_{m\gamma\times m\gamma}\end{bmatrix},\ \gamma'=1-\gamma$$

其中,

$$A_{m\gamma'\times m\gamma'}=\begin{bmatrix}1 & \alpha & \alpha & \cdots & \alpha\\ \alpha & 1 & \alpha & \cdots & \alpha\\ \alpha & \alpha & 1 & \cdots & \alpha\\ \vdots & \vdots & \vdots & \ddots & \vdots\\ \alpha & \alpha & \alpha & \cdots & 1\end{bmatrix}$$

矩阵的行列对称变换不改变其行列式, 故有:

$$\det(X_{m\times m})=\det(X'_{m\times m})=\det(A_{m\gamma'\times m\gamma'})=(1-\alpha)^{m\gamma'-1}[1+(m\gamma'-1)\alpha]$$

将上式代入 Advo 函数的表达式:

$$\mathrm{Advo}(K)=\sqrt[2m]{\det(X_{m\times m})}=(1-\alpha)^{\frac{m\gamma'-1}{2m}}[1+(m\gamma'-1)\alpha]^{\frac{1}{2m}}$$

考虑极限情形:

$$\lim_{m\to\infty}(1-\alpha)^{\frac{m\gamma'-1}{2m}}=(1-\alpha)^{\lim\limits_{m\to\infty}\frac{m\gamma'-1}{2m}}=(1-\alpha)^{\frac{\gamma'}{2}}$$

$$\lim_{m\to\infty}[1+(m\gamma'-1)\alpha]^{\frac{1}{2m}}=1$$

综上可得:

$$\lim_{m\to\infty}\mathrm{Advo}(K)=(1-\alpha)^{\frac{\gamma'}{2}}=(1-\alpha)^{\frac{(1-\gamma)}{2}}$$

因为 $0<1-\alpha<1$, 所以 $(1-\alpha)^{(1-\gamma)/2}$ 是关于 γ 的单增函数, 且 $\gamma=1$ 时取得最大值 1. □

本文将 Advo 作为攻击概貌群体效应的定量度量. 根据上述讨论, 当 $|K|$ 高于一定水平, Advo(K) 值会随攻击概貌比例的提高而增加. 特别地, 当 K 中均为攻击概貌时, Advo(K) 取得最大值 1, 所以 Advo(K) 是 K 中攻击概貌富集程度的良好指标, 它将作为 IBIGDA 算法的构建基础.

2.3　遗传优化目标函数

首先定义指示向量 (Indication vector, IV) 及相关概念:

定义 3. 设推荐系统中有 m 个用户, 则指示向量 $\boldsymbol{IV}=[\mathrm{sgn}_1,\mathrm{sgn}_2,\cdots,\mathrm{sgn}_m]^{\mathrm{T}}$ 且 $\forall\,\mathrm{sgn}_i\in\{+,-\}$. 此外,

1) 若 $\boldsymbol{IV}$ 中第 j 个元素 $\boldsymbol{IV}[j]=+$, 表明用户 j 被视为攻击者. 显然用户 j 属于真正或假正.

2) $p_{\boldsymbol{IV}}^{t+}$ 为 $\boldsymbol{IV}$ 中真正的比例, $p_{\boldsymbol{IV}}^{f+}$ 为假正的比例, $p_{\boldsymbol{IV}}^{+}=p_{\boldsymbol{IV}}^{t+}+p_{\boldsymbol{IV}}^{f+}$.

3) 若 $\boldsymbol{IV}$ 中所有正用户的概貌构成集合 K^+, 则认为 $\mathrm{Advo}(\boldsymbol{IV})=\mathrm{Advo}(K^+)$.

利用 Advo 进行攻击探测是典型的组合优化问题. 最优化的目的是在保持 $\mathrm{Advo}(\boldsymbol{IV})=1$ 的前提下, 使关系 $p_{\boldsymbol{IV}}^{+}=p_{\boldsymbol{IV}}^{t+}=p^{\mathrm{true}}$ 成立. 本文选择遗传算法作为最优化手段, 需要引入以 $\boldsymbol{IV}$ 为自变量的目标函数 $g^{\mathrm{obj}}(\cdot)$, 使得当 $g^{\mathrm{obj}}(\cdot)$ 最大时, 探测效果达到最佳.

函数 $g^{\mathrm{obj}}(\cdot)$ 应具备两种能力: 1) 尽量维持 $\mathrm{Advo}(\boldsymbol{IV})$ 等于最大值 1; 2) 限制 $p_{\boldsymbol{IV}}^{+}$ 对 p^{true} 的过度偏离. 这表明 $g^{\mathrm{obj}}(\cdot)$ 包含 $\mathrm{Advo}(\boldsymbol{IV})$ 和 $p_{\boldsymbol{IV}}^{+}$ 偏离 p^{true} 时惩罚因子的相互作用. 本文选用中位数为 p^{true} 的柯西分布密度函数作为惩罚因子:

$$\mathrm{Penalty}(p_{\boldsymbol{IV}}^{+}|p^{\mathrm{true}},\sigma)=\frac{1}{\sigma\pi\left(1+\left(\dfrac{p_{\boldsymbol{IV}}^{+}-p^{\mathrm{true}}}{\sigma}\right)^2\right)}$$

其中, σ 是柯西分布的扩展因子.

为探索 $g^{\mathrm{obj}}(\boldsymbol{IV})$ 的具体表达式, 假定 $\boldsymbol{IV}$ 是随机变量, 从概率论角度, 有关系 $\mathrm{P}(\boldsymbol{IV}|X)\propto\mathrm{P}(X|\boldsymbol{IV})\mathrm{P}(\boldsymbol{IV})$. 若 $\mathrm{Advo}(\boldsymbol{IV})$ 和 $\mathrm{Penalty}(p_{\boldsymbol{IV}}^{+}|p^{\mathrm{true}},\sigma)$ 分别对应于似然 $\mathrm{P}(X|\boldsymbol{IV})$ 和先验 $\mathrm{P}(\boldsymbol{IV})$, 自然地, $g^{\mathrm{obj}}(\boldsymbol{IV})$ 为后验 $\mathrm{P}(\boldsymbol{IV}|X)$, 此时最优化的目的等价于求 $\boldsymbol{IV}$ 的最大后验估计 $\boldsymbol{IV}^{\mathrm{MAP}}$. 由上, 令:

$$g^{\mathrm{obj}}(\boldsymbol{IV}|p^{\mathrm{true}},\sigma)=\mathrm{Advo}(\boldsymbol{IV})\cdot\mathrm{Penalty}(p_{\boldsymbol{IV}}^{+}|p^{\mathrm{true}},\sigma)$$

当 $p_{\boldsymbol{IV}}^{+}=p_{\boldsymbol{IV}}^{t+}=p^{\mathrm{true}}$, 即探测效果达到最佳时, 根据定理 1 及惩罚因子的表达式, $\mathrm{Advo}(\boldsymbol{IV})$ 与 $\mathrm{Penalty}(p_{\boldsymbol{IV}}^{+}|p^{\mathrm{true}},\sigma)$ 分别取得最大值, 从而 $g^{\mathrm{obj}}(\boldsymbol{IV}|p^{\mathrm{true}},\sigma)$ 达到最大.

但是, 上述最优化暂不可行, 因为无法事先获知参数 p^{true} 的值. 为此, 本文借鉴了贝叶斯推断思想, 首先利用不小于 p^{true} 的先验比例 p' 启动遗传优化过程, 根据每次迭代稳定后的最优个体 $\boldsymbol{IV}$, 自适应地将 p' 后验更新为 $p^{+}_{\boldsymbol{IV}}$, 逐步使 p' 逼近 p^{true}, 同时保持 $p^{t+}_{\boldsymbol{IV}} = p^{\text{true}}$. 此过程由如下定理保证:

定理 2. 现有条件: a) 遗传算法能以一定概率对个体中至多 n ($\geq$ 2) 个元素进行变异操作; b) $p^{\text{true}} \leq p'$. 若 $\boldsymbol{IV} = \boldsymbol{IV}'$ 时遗传算法收敛, 那么:

1) 若 $p^{+}_{\boldsymbol{IV}'} < p^{\text{true}}$, 有 $p^{+}_{\boldsymbol{IV}'} = p^{t+}_{\boldsymbol{IV}'}$; 若 $p^{+}_{\boldsymbol{IV}'} \geq p^{\text{true}}$, 有 $p^{t+}_{\boldsymbol{IV}'} = p^{\text{true}}$.

2) $p^{\text{true}} \leq p^{+}_{\boldsymbol{IV}'} \leq p'$.

3) $p^{t+}_{\boldsymbol{IV}'} = p^{\text{true}}$.

证明. 在变异操作下, 设 $\boldsymbol{IV}$ 的 "邻居" 为 $\text{Neigh}(\boldsymbol{IV}) = \{\boldsymbol{IV}[-S] \mid S \subseteq \boldsymbol{IV}, |S| \leq n\}$, 其中 $\boldsymbol{IV}[-S]$ 表示对 S 中的元素实施变号. 根据题设, $g^{\text{obj}}(\boldsymbol{IV}|p', \sigma)$ 必在 $\boldsymbol{IV}'$ 处取得局部或全局最大值, 显然 $\nexists \boldsymbol{IV}'' \in \text{Neigh}(\boldsymbol{IV}')$, 使 $g^{\text{obj}}(\boldsymbol{IV}''|p', \sigma) > g^{\text{obj}}(\boldsymbol{IV}'|p', \sigma)$ 成立.

1) 若结论不成立, 则 $\boldsymbol{IV}'$ 中至少存在一个假正 sgn_{k^f+} 与一个假负 sgn_{k^f-}, 令 $\boldsymbol{IV}'' = \boldsymbol{IV}'[-\{\text{sgn}_{k^f+}, \text{sgn}_{k^f-}\}]$, $\boldsymbol{IV}''$ 的真正比例相对 $\boldsymbol{IV}'$ 得到提高, 根据定理 1 可知 $\text{Advo}(\boldsymbol{IV}'') > \text{Advo}(\boldsymbol{IV}')$, 又由 $\text{Penalty}(p^{+}_{\boldsymbol{IV}''}|p', \sigma) = \text{Penalty}(p^{+}_{\boldsymbol{IV}'}|p', \sigma)$. 综上可得, $g^{\text{obj}}(\boldsymbol{IV}''|p', \sigma) > g^{\text{obj}}(\boldsymbol{IV}'|p', \sigma)$, 矛盾.

2) 分两种情况讨论:

a) 若 $p^{+}_{\boldsymbol{IV}'} < p^{\text{true}}$, 由 1) 得 $p^{+}_{\boldsymbol{IV}'} = p^{t+}_{\boldsymbol{IV}'}$, 此时 $\boldsymbol{IV}'$ 中至少存在一个假负 sgn_{k^f-}, 令 $\boldsymbol{IV}'' = \boldsymbol{IV}'[-\{\text{sgn}_{k^f-}\}]$, 易知 $p^{t+}_{\boldsymbol{IV}'}/p^{+}_{\boldsymbol{IV}'} = p^{t+}_{\boldsymbol{IV}''}/p^{+}_{\boldsymbol{IV}''} = 1$, 所以 $\text{Advo}(\boldsymbol{IV}'') = \text{Advo}(\boldsymbol{IV}') = 1$. 又由 $p^{+}_{\boldsymbol{IV}'} < p^{+}_{\boldsymbol{IV}''} \leqq p'$ 知 $\text{Penalty}(p^{+}_{\boldsymbol{IV}''}|p', \sigma) > \text{Penalty}(p^{+}_{\boldsymbol{IV}'}|p', \sigma)$. 综上可得, $g^{\text{obj}}(\boldsymbol{IV}''|p', \sigma) > g^{\text{obj}}(\boldsymbol{IV}'|p', \sigma)$, 矛盾.

b) 若 $p^{+}_{\boldsymbol{IV}'} > p'$, 由 1) 得 $p^{t+}_{\boldsymbol{IV}'} = p^{\text{true}}$, 此时 $\boldsymbol{IV}'$ 中至少存在一个假正 sgn_{k^f+}, 令 $\boldsymbol{IV}'' = \boldsymbol{IV}'[-\{\text{sgn}_{k^f+}\}]$, 易知 $p^{t+}_{\boldsymbol{IV}''}/p^{+}_{\boldsymbol{IV}''} > p^{t+}_{\boldsymbol{IV}'}/p^{+}_{\boldsymbol{IV}'}$, 所以 $\text{Advo}(\boldsymbol{IV}'') > \text{Advo}(\boldsymbol{IV}')$. 又由 $p^{+}_{\boldsymbol{IV}'} > p^{+}_{\boldsymbol{IV}''} \geq p'$ 知 $\text{Penalty}(p^{+}_{\boldsymbol{IV}''}|p', \sigma) > \text{Penalty}(p^{+}_{\boldsymbol{IV}'}|p', \sigma)$. 综上可得, $g^{\text{obj}}(\boldsymbol{IV}''|p', \sigma) > g^{\text{obj}}(\boldsymbol{IV}'|p', \sigma)$, 矛盾.

故必有 $p^{\text{true}} \leq p^{+}_{\boldsymbol{IV}'} \leq p'$.

3) 由 1) 和 2) 立即可得. □

2.4 算法的描述与解释

下面是 IBIGDA 算法的描述.

输入. 遗传优化目标函数的参数 $p'_{(0)}$ 和 $\sigma_{(0)}$, 即攻击概貌的先验比例和惩罚函数的扩展因子, 且满足定理 2 的条件.

输出. 最优个体 $\boldsymbol{IV}^{\text{best}}$.

步骤 1. 预处理评分矩阵.

步骤 2. $n \leftarrow 0$, 初始化种群 $\boldsymbol{IV}^{[1]}_{(0)} \sim \boldsymbol{IV}^{[k]}_{(0)}$, k 为种群基数.

步骤 3. 开始第 n 次迭代.

步骤 3.1. 以 $\boldsymbol{IV}^{[1]}_{(n)} \sim \boldsymbol{IV}^{[k]}_{(n)}$ 为初始种群, 对 $g^{\text{obj}}(\boldsymbol{IV}|p'_{(n)}, \sigma_{(n)})$ 进行遗传优化, 直至收敛 (连续 z 次输出值不变), 得到种群 $\boldsymbol{IV}^{[1]}_{(n+1)} \sim \boldsymbol{IV}^{[k]}_{(n+1)}$.

步骤 3.2. 取其中最优个体 $\boldsymbol{IV}^{\text{best}}$, 后验更新 $p'_{(n+1)} \leftarrow p^{+}_{\boldsymbol{IV}^{\text{best}}}$, $\sigma_{(n+1)} \leftarrow \widehat{\sigma}(\boldsymbol{IV}^{\text{best}})$.

步骤 3.3. 若 $|p'_{(n+1)} - p'_{(n)}| < \varepsilon$, 返回 $\boldsymbol{IV}^{\text{best}}$, 执行结束; 否则, $n \leftarrow n + 1$, 转至步骤 3.

IBIGDA 算法中, 因为 p^{true} 一般较小, 则 $p'_{(0)}$ 任取一较大值即可, 不需任何先验知识. $p'_{(0)}$ 的取值越靠近 p^{true}, 算法所需迭代次数越少.

需强调的是, 算法每次迭代后不仅对 p', 同时也对 σ 进行更新, 这是针对非理想情况的对策.

理想情况下, 定理 2 决定了 IBIGDA 算法在固定 σ 值时就可获得最佳探测效果. 同时, σ 值的大小也会对 p' 的后验值产生影响: σ 越小, 则偏离 p' 时的惩罚越大, 后验值就越接近 p'; 反之就越接近 p^{true}. 因而取较大的 σ 可以加快算法的收敛.

然而真实系统都是运行在非理想情况下, 虽然两种情况的差别并不显著, 但此时两两垂直的攻击概貌只是所有攻击概貌的子集, 所以可能会在某次迭代后, 出现 $p^{+}_{\boldsymbol{IV}} < p^{\text{true}}$ 的越界情况, 影响算法的探测效果. 对此, IBIGDA 算法将 σ 也视为自适应参数, 在 $p^{+}_{\boldsymbol{IV}}$ 距 p^{true} 较远时, 使用较大的 σ 值, 加快 $p^{+}_{\boldsymbol{IV}}$ 向 p^{true} 的逼近速率; 而在 $p^{+}_{\boldsymbol{IV}}$ 接近 p^{true} 时, 为控制越界程度, 必须逐步减少 σ 值. 接近程度可以用 $\text{Advo}(\boldsymbol{IV})$ 衡量, $\text{Advo}(\boldsymbol{IV})$ 越大, 表明 $p^{+}_{\boldsymbol{IV}}$ 越接近 p^{true}, 所以 σ 是 $\text{Advo}(\boldsymbol{IV})$ 的函数. 非理想情况下, 攻击概貌的 Advo 值小于 1, 所以 $\text{Advo}(\boldsymbol{IV})$ 在接近 1 时, σ 要加速减小以尽量控制越界程度. IBIGDA 算法中, 令 $\widehat{\sigma}(\boldsymbol{IV}) = (1 - \text{Advo}(\boldsymbol{IV}))^{\frac{1}{2}}$.

3 实验与分析

实验使用了数据集 MovieLens 100K, 它包含了 943 个用户对 1 682 部电影的 10 万个评分, 评分范围为 1～5, 代表喜好程度从低到高, 每个用户至少评价了 20 部电影. 为了验证 IBIGDA 算法的探测能力, 假定这 943 个用户为真实用户, 在不同的参数配置下 (攻击强度 p^{att}, 填充率 p^{fill} 等), 分别向数据集注入三种攻击: 随机攻击, 均值攻击和流行攻击, 且均为推攻击.

IBIGDA 算法中相关参数取值为: $p'_{(0)} = 0.3$, $\sigma_{(0)} = 0.12$, $\varepsilon = 0.004$.

3.1　托攻击探测过程实例分析

图 3 展示了典型的攻击探测过程, 系统中存在 $p^{\text{att}} = 12\,\%$, $p^{\text{fill}} = 6\,\%$ 的随机攻击. 虚线处是更新自适应参数的时刻, 虚线之间是针对后验更新的目标函数的遗传优化过程. 算法在第 12 次迭代时返回结果. 在不同的攻击模型与参数配置下, 通过监测 $p^{+}_{\boldsymbol{IV}}$ 和 $p^{t+}_{\boldsymbol{IV}}$ 的变化, IBIGDA 算法得到的攻击探测过程均与图 3 类似.

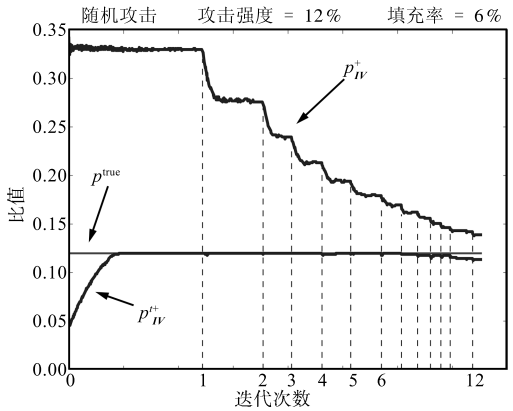


图 3　典型的攻击探测过程

Fig. 3　The typical procedure of attack detection

定理 2 表明, 理想情况下的探测过程应为每次迭代结束时, $p^{+}_{\boldsymbol{IV}}$ 进一步靠近 p^{true}, 且 $p^{t+}_{\boldsymbol{IV}} = p^{\text{true}}$, 最终达到 $p^{+}_{\boldsymbol{IV}} = p^{t+}_{\boldsymbol{IV}} = p^{\text{true}}$. 图 3 中, 尽管处于非理想情况, 但近似展现了理想的探测过程. 可看出, $p^{+}_{\boldsymbol{IV}}$ 在 σ 支配下, 逼近 p^{true} 的速率逐步放缓, 程序在此速率低于一定程度时退出, 以限制越界.

直观地, IBIGDA 算法包含两种过程的相互作用: 一种是对攻击概貌的吸纳过程, 目的是达到并维持 $p^{t+}_{\boldsymbol{IV}} \simeq p^{\text{true}}$; 另一种是对真实概貌的排空过程, 目的是降低假正率 $p^{f+}_{\boldsymbol{IV}}$ $(= p^{+}_{\boldsymbol{IV}} - p^{t+}_{\boldsymbol{IV}})$. 这两种过程贯穿了算法执行的始终. 图 3 中, 初始迭代时可以看出明显的吸纳过程, 而每次更新自适应参数后, 也表现了明显的排空过程. 最后几步迭代时, $p^{t+}_{\boldsymbol{IV}}$ 有略微下降的趋势, 这是由于真实系统中固有地存在少量兴趣奇特的真实用户, 他们的评分行为具有更强的攻击性, 导致排空过程错误地排除了某些具有弱攻击性的攻击概貌. 从而, IBIGDA 算法选择在 $p^{+}_{\boldsymbol{IV}}$ 的变化率低于一定程度时退出, 不仅限制了越界, 也恰好止住了 $p^{t+}_{\boldsymbol{IV}}$ 的下滑.

3.2　托攻击探测效果

托攻击探测效果的评价使用了准确率 f_{pre}, 召回率 f_{rec} 或两者的综合指标 F 值[18]. 设 IBIGDA 算法返回 $\boldsymbol{IV}^{\text{best}}$, 则:

$$\begin{cases} f_{\text{pre}} = \dfrac{p^{t+}_{\boldsymbol{IV}^{\text{best}}}}{p^{+}_{\boldsymbol{IV}^{\text{best}}}} \\ f_{\text{rec}} = \dfrac{p^{t+}_{\boldsymbol{IV}^{\text{best}}}}{p^{\text{true}}} \\ F = 2\dfrac{f_{\text{pre}} \times f_{\text{rec}}}{f_{\text{pre}} + f_{\text{rec}}} \end{cases}$$

实验采取 $3 \times 5 \times 6$ 的设计模式, 攻击模型 (随机攻击, 均值攻击, 流行攻击), 攻击强度 p^{att} $(5\,\%, 7\,\%, 10\,\%, 12\,\%, 15\,\%)$ 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. 每组实验配置下, 独立地向数据集注入 10 次攻击, 最终实验数据是这 10 次攻击探测的均值.

本文选用 PCA VarSelect, PLSA 与 EMSVD 算法作为 IBIGDA 算法的性能参照. 目前, PCA VarSelect 算法在 MovieLens 数据集上具备最佳的探测性能.

表 1～3 展示了 IBIGDA 算法的探测效果. 仅在面临填充率为 3 % 的流行攻击时, 算法的探测能力受限, 因为此时选择填充项所占比例较大, 所以攻击概貌与真实概貌相近, 攻击特征不明显. 其余情况下, 算法均取得了较好的探测性能, 尤其是召回率基本位于 90 % 以上, 显示了 IBIGDA 算法能够探测出绝大多数攻击概貌. 随着攻击强度和填充率的增大, 攻击概貌的嫌疑性愈加显著, 因而算法的探测性能有增强的趋势. 此外, 实验发现当不注入攻击时, $\boldsymbol{IV}^{\text{best}}$ 仅有 25 个正用户, 由于算法的探测准确率小于 1, 所以 "名义上" 的真正用户必然更少, 说明算法可以一定程度上显示出系统中不存在攻击.

一般地, 为了获得较高的召回率, 需要以牺牲一定的准确率为代价, IBIGDA 算法也是如此. 由于推荐系统中用户基数很大, 推荐过程中错误地屏蔽一些假正用户并不会对推荐结果产生显著影响.

表 1～3 中下划线数据表示在同等实验配置下, IBIGDA 算法要优于达到最佳探测性能的 PCA VarSelect 算法所得的对应数据. PCA VarSelect 算法达到最佳探测性能的前提是必须获知准确的攻击强度 (实验假定已知), 否则会严重降低准确率或召回率[13]. 同样, PLSA 与 EMSVD 探测算法的性能依赖于输入的用户类别数, 而此数一般由试探法确定. 图 4 展示了三种攻击模型在 $p^{\text{att}} = 10\,\%$, $p^{\text{fill}} = 6\,\%$ 时, IBIGDA, PLSA 与 EMSVD 算法以 F 值为指标的性能对比. 易看出, 在不同的用户类别数下, PLSA 算法的性能变化显著, 显示了其对先验知识较强的敏感性. EMSVD 算法的性能也随用户类别数的变化表现出一定波动, 敏感性虽不显著, 但探测能力不甚理想, 特别是面临均值攻击时, 算法近乎失效. 事实上, EMSVD 算法仅对高填充率的随机

攻击有效[11], 局限性较大, 因为攻击概貌极少采用高填充率, 这会加大攻击成本, 且可能降低攻击效果[9]. 实际应用中, 不论攻击强度还是用户类别数均是无法事先获知的, 而 IBIGDA 算法在上述实验中始终使用的是同一组输入参数, 无需准确的先验知识, 符合实际应用需求.

IBIGDA 算法的计算量主要消耗在对种群个体的 $g^{\mathrm{obj}}(\cdot)$ 的串行计算上, 但容易利用遗传算法内在的并行性, 每次迭代时单个个体映射至单个处理机, 使 IBIGDA 算法获得并行加速.

表 1 随机攻击探测的准确率与召回率

Table 1 Detection precision and recall for random attack

p^{fill}	3 %		6 %		9 %		12 %		15 %		20 %	
p^{att}	f_{pre}	f_{rec}	f_{pre}	f_{rec}	f_{pre}	f_{rec}	f_{pre}	f_{rec}	f_{pre}	f_{rec}	f_{pre}	f_{rec}
5 %	0.58	0.91	0.62	1.00	0.60	1.00	0.64	1.00	0.62	1.00	0.64	1.00
7 %	0.70	1.00	0.67	1.00	0.67	1.00	0.65	1.00	0.71	1.00	0.71	1.00
10 %	0.75	0.97	0.78	1.00	0.78	1.00	0.74	1.00	0.74	1.00	0.76	1.00
12 %	0.80	0.95	0.82	0.95	0.81	0.99	0.78	1.00	0.81	0.99	0.80	1.00
15 %	0.84	0.90	0.85	0.96	0.83	0.99	0.86	0.99	0.85	0.99	0.86	0.99

表 2 均值攻击探测的准确率与召回率

Table 2 Detection precision and recall for average attack

p^{fill}	3 %		6 %		9 %		12 %		15 %		20 %	
p^{att}	f_{pre}	f_{rec}	f_{pre}	f_{rec}	f_{pre}	f_{rec}	f_{pre}	f_{rec}	f_{pre}	f_{rec}	f_{pre}	f_{rec}
5 %	0.63	0.85	0.65	0.98	0.65	1.00	0.64	1.00	0.69	1.00	0.65	1.00
7 %	0.69	0.86	0.68	0.98	0.73	0.98	0.70	1.00	0.71	0.97	0.74	1.00
10 %	0.78	0.85	0.78	0.91	0.80	0.97	0.81	0.98	0.82	0.99	0.81	0.97
12 %	0.79	0.82	0.84	0.90	0.85	0.96	0.85	0.98	0.85	0.97	0.84	0.97
15 %	0.80	0.69	0.85	0.84	0.86	0.89	0.89	0.94	0.86	0.89	0.88	0.96

表 3 流行攻击探测的准确率与召回率

Table 3 Detection precision and recall for bandwagon attack

p^{fill}	3 %		6 %		9 %		12 %		15 %		20 %	
p^{att}	f_{pre}	f_{rec}	f_{pre}	f_{rec}	f_{pre}	f_{rec}	f_{pre}	f_{rec}	f_{pre}	f_{rec}	f_{pre}	f_{rec}
5 %	0.57	0.68	0.59	1.00	0.59	1.00	0.63	1.00	0.59	1.00	0.62	1.00
7 %	0.64	0.77	0.70	0.98	0.72	1.00	0.70	1.00	0.68	1.00	0.69	1.00
10 %	0.68	0.61	0.79	0.95	0.78	0.99	0.78	1.00	0.78	1.00	0.78	1.00
12 %	0.67	0.57	0.81	0.94	0.82	0.99	0.81	0.99	0.82	1.00	0.81	1.00
15 %	0.73	0.59	0.83	0.87	0.84	0.97	0.84	0.96	0.84	0.97	0.87	0.99

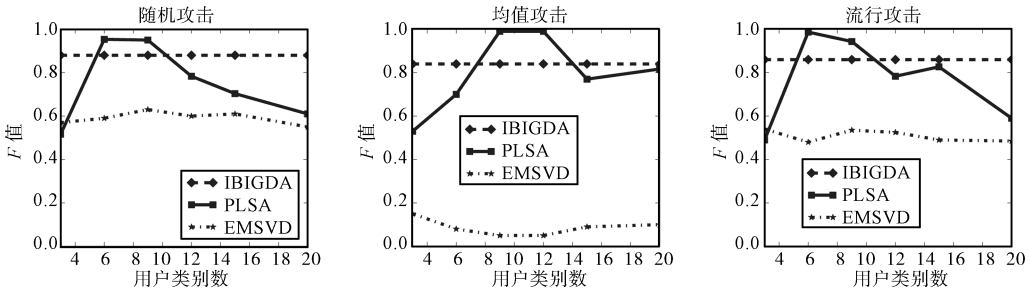


图 4 探测随机攻击, 均值攻击和流行攻击 (均满足 $p^{\mathrm{att}} = 10\,\%$, $p^{\mathrm{fill}} = 6\,\%$) 的 F 值

Fig. 4 F score for detecting random attack, average attack, and bandwagon attack (all under $p^{\mathrm{att}} = 10\,\%$, $p^{\mathrm{fill}} = 6\,\%$)

4 结论

本文给出了攻击概貌群体效应的定量度量和遗传优化目标函数的详细构建过程与可行性论证, 在此基础上提出了迭代贝叶斯推断遗传探测算法. 较之现有攻击探测技术, 本文算法具备更高的无监督程度, 即使在缺少攻击强度, 攻击模型等先验知识的情况下, 仍可对托攻击进行可靠、准确的探测, 为推荐系统管理者与研究者提供了较实用的攻击探测手段和新的研究思路.

References

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

2 Li Q, Kim B M. Constructing user profiles for collaborative recommender system. In: Proceedings of the 6th Asia Pacific Web Conference. Hangzhou, China: Springer, 2004. 100−110

3 Herlocker J L, Konstan J A, Borchers A, Riedl J. An algorithmic framework for performing collaborative filtering. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New York, USA: ACM, 1999. 230−237

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

5 Burke R, Mobasher B, Zabicki R, Bhaumik R. Identifying attack models for secure recommendation. In: Proceedings of the Beyond Personalization Workshop on the International Conference on Intelligent User Interfaces. San Diego, USA: ACM Press, 2005. 347−361

6 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

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

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. New York, USA: ACM, 2006. 542−547

9 Mobasher B, Burke R, Bhaumik R, Williams C. Toward trustworthy recommender systems: an analysis of attack models and algorithm robustness. *ACM Transactions on Internet Technology*, 2007, **7**(4): 1−38

10 Burke R, Mobasher B, Bhaumik R, Williams C. Segment-based injection attacks against collaborative filtering recommender systems. In: Proceedings of the 5th IEEE International Conference on Data Mining. Washington D. C., USA: IEEE, 2005. 577−580

11 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 Conference on Research and Development in Information Retrieval. New York, USA: ACM, 2006. 517−524

12 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

13 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. New York, USA: ACM, 2007. 14−21

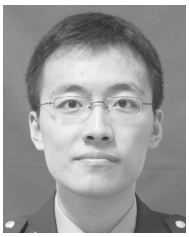
14 Mehta B, Nejdl W. Attack resistant collaborative filtering. In: Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. New York, USA: ACM, 2008. 75−82

15 Hurley N, Cheng Z P, Zhang M. Statistical attack detection. In: Proceedings of the ACM Conference on Recommender Systems. New York, USA: ACM, 2009. 149−156

16 Herlocker J L, Konstan J A, Terveen L G, Reidl J T. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 2004, **22**(1): 5−53

17 Anderson T W. *An Introduction to Multivariate Statistical Analysis (Third Edition)*. New York: Wiley-Interscience, 2003. 266−270

18 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. New York, USA: Springer-Verlag, 1994. 3−12



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.)

石金龙 国防科学技术大学计算机学院博士研究生. 主要研究方向为数据挖掘与生物信息学.
E-mail: jlshi@nudt.edu.cn
(**SHI Jin-Long** Ph. D. candidate at the School of Computer, National University of Defense Technology. His research interest covers data mining and bioinformatics.)