中国科学 : 信息科学　2022 年　第 52 卷　第 8 期: 1481–1499
SCIENTIA SINICA Informationis
论文

满足本地化差分隐私的推荐系统中隐私预算的优化设置

暴婷[1], 徐蕾[2*], 祝烈煌[2], 王丽宏[3]

1. 北京理工大学计算机学院, 北京 100081
2. 北京理工大学网络空间安全学院, 北京 100081
3. 国家互联网应急技术处理协调中心, 北京 100029
* 通信作者. E-mail: 6120180029@bit.edu.cn

收稿日期: 2021–09–04; 修回日期: 2021–11–08; 接受日期: 2021–12–17; 网络出版日期: 2022–08–03

国家自然科学基金 (批准号: 61871037) 和北京市自然科学基金 (批准号: M21035) 资助项目

摘要 推荐系统可帮助用户从众多的数据中发现用户所需数据, 与此同时, 上传用户原始数据给服务器也可能泄露用户隐私. 本文使用本地化差分隐私技术为推荐系统中的用户数据提供隐私保护. 在本地化差分隐私模型中, 隐私预算控制用户数据的隐私保护程度, 较高的隐私预算通常意味着较高的分析准确性. 为在最小化隐私损失的同时最大化推荐准确性, 我们将隐私预算设置问题建模为多臂赌博机问题, 并提出基于置信度上界的学习策略帮助用户选择最优的隐私预算. 考虑到用户对不同数据的敏感程度不同, 我们对学习策略进行了改进. 真实数据集上的实验结果表明, 所提策略可以帮助用户选出合适的隐私预算, 可有效提高用户的累计收益.

关键词 推荐系统, 本地化差分隐私, 隐私预算, 强化学习, 多臂赌博机

1 引言

近年来, 推荐系统被广泛应用于我们的日常生活中, 帮助用户从海量的项目 (例如, 电影、书籍和兴趣点等) 中找到自己可能感兴趣的项目 [1~3]. 用户为了获得准确的推荐, 需要将自己的一些数据发送至服务器, 以供服务器了解该用户的喜好. 然而, 这些数据中通常包含了用户的敏感信息, 例如, 家庭住址、健康状况、政治观点和性别取向等. 直接将用户数据发送给服务器将会给用户的隐私带来威胁. 如何在保证用户隐私不被泄露的前提下为用户提供满意的推荐结果, 成为了推荐系统研究中的重要关注点.

为了解决推荐系统中的隐私问题, 研究者们相继提出了很多隐私保护技术. 典型的技术有密码学技术 [4,5]、哈希技术 [6,7] 和差分隐私技术 [8,9]. 这些隐私保护技术都需要可信服务器的支持. 然而在

引用格式: 暴婷, 徐蕾, 祝烈煌, 等. 满足本地化差分隐私的推荐系统中隐私预算的优化设置. 中国科学: 信息科学, 2022, 52: 1481–1499, doi: 10.1360/SSI-2021-0304
Bao T, Xu L, Zhu L H, et al. Optimized setting of privacy budget in a recommendation system with local differential privacy (in Chinese). Sci Sin Inform, 2022, 52: 1481–1499, doi: 10.1360/SSI-2021-0304

实际中, 很难找到一个完全可信的服务器, 服务器可能被攻击者攻击, 也可能因为某些商业目的而故意泄露用户隐私. 考虑到不可信服务器的存在, 用户在将数据发送给服务器之前, 需要先在本地对这些数据进行扰动以实现数据的隐私化. 本地化差分隐私技术 [10,11] 作为差分隐私的扩展, 要求每位用户在本地独立扰动自己的数据, 服务器利用用户扰动的数据来完成后续统计任务, 这使得用户对数据的保护不再需要可信服务器的支持. 相比差分隐私, 本地化差分隐私可为用户提供更加强大的隐私保护, 但同时也为用户数据引入了更多的噪声, 进而导致服务器推荐准确性下降 [12,13]. 无论是用户隐私遭受到泄露还是得到的推荐准确性较差, 都会导致用户对推荐系统的满意度降低, 从而影响推荐系统在实际中的应用. 如何平衡用户的隐私保护需求和推荐准确性要求, 成为了当前亟需解决的问题.

在满足本地化差分隐私的推荐系统中, 隐私预算作为一项重要参数, 一方面决定着用户数据的隐私化程度, 另一方面也决定着推荐结果的准确性 [13~15]. 选择较小的隐私预算可以为用户提供较高的隐私保护程度, 但同时也会导致推荐准确性较差. 因此, 为用户选择合适的隐私预算, 对于平衡用户的隐私保护需求和推荐准确性要求起着至关重要的作用. 前人对于本地化差分隐私的研究默认由服务器决定用户的隐私预算, 且为每位用户分配相同的隐私预算. 考虑到本地化差分隐私中用户数据的扰动在用户本地端完成, 原则上每位用户都可以根据自己对隐私保护的需求在本地独立设置其隐私预算. 然而, 隐私预算对于用户来说是一个抽象的参数, 用户难以直观地设置其隐私预算的取值, 因此, 亟需找到一种方法帮助用户在满足本地化差分隐私的推荐系统中选择出最优的隐私预算的取值.

本文考虑在用户与服务器不断交互的本地化差分隐私推荐系统中, 如何帮助用户选择最优隐私预算. 考虑到用户在使用推荐系统的过程中, 希望以最小的隐私损失获得最满意的推荐结果, 我们将用户遭受的隐私损失和获得的推荐准确性量化为用户收益, 并分析隐私预算取值对用户收益的影响. 用户每次请求服务器推荐时, 需将完成推荐所需数据发送给服务器. 为了保护隐私, 用户首先设置其隐私预算的取值, 而后对原始数据进行相应程度的扰动. 为了最大化累计收益, 用户需根据以往交互中获得的收益估计每个隐私预算取值的期望收益, 以此为依据动态调整本轮隐私预算的取值. 我们引入多臂赌博机模型 [16] 对用户隐私预算的优化设置问题进行建模, 并提出相应的学习策略帮助用户选择隐私预算的取值 [17~19]. 本文的主要贡献总结如下:

(1) 本文首次在满足本地化差分隐私的推荐系统中, 应用多臂赌博机模型对用户的隐私预算优化设置问题进行建模;

(2) 提出基于置信度上界 (upper confidence bound, UCB) [20~22] 的学习策略帮助用户在与服务器交互的过程中选择最优的隐私预算, 所提策略考虑到了用户对不同数据敏感程度不同的情况;

(3) 以两种典型的推荐系统应用场景为例, 分别给出了所提策略在电影推荐系统和位置推荐系统中的具体应用办法. 在真实世界数据集上进行了一系列仿真实验, 验证了所提隐私预算设置策略可以有效提升用户的累计收益.

2 相关工作

2.1 满足本地化差分隐私的推荐系统

近年来, 随着推荐系统的兴起, 研究者们提出了许多在推荐系统中为用户数据提供本地化差分隐私保护的方法. Shen 等 [23] 设计了一个 EpicRec 框架, 该框架利用本地化差分隐私技术保护用户对项目的浏览记录. 在他们的框架中, 所有项目被划分为不同的类别, EpicRec 通过为不同的类别设置不同的敏感程度来为用户提供基于类别的隐私控制. 这种方法的缺点是可能会泄露用户的类别偏好. Hua

等 [24] 提出了一种可支持不可信服务器的隐私保护矩阵分解 (matrix factorization, MF) 推荐模型, 他们的模型可实现对用户评分值数据的本地化差分隐私保护. 但是, 用户的评分行为 (即用户是否对该项目进行过评分) 可能会遭到泄露. 考虑到在实际中, 用户对项目的评分值和评分行为都是用户的敏感信息 [25], Shin 等 [26] 将本地化差分隐私技术应用于基于矩阵分解的推荐系统中, 提出了可同时为用户的评分值和评分行为提供隐私保护的方法. 研究者们也提出了一些方法为位置推荐系统中的用户位置提供本地化差分隐私保护. Asada 等 [27] 提出了一种位置推荐模型, 允许用户决定是否向服务器发布位置签到数据, 并使用随机响应机制保护用户的隐私. Chen 等 [28] 使用秘密共享机制和随机响应机制扰动用户数据, 服务器根据扰动后数据生成动态的位置特征, 提出的方法可为用户的位置签到行为提供本地化差分隐私保护, 同时服务器也可计算出每个位置点的流行度. Kim 等 [29] 将本地化差分隐私应用于基于矩阵分解的兴趣点推荐系统中, 为用户上传给服务器的梯度数据提供本地化差分隐私保护. 上述文献提出了将本地化差分隐私应用于推荐系统中的方法, 都强调了隐私预算对于隐私保护的重要性, 分析了隐私预算与隐私保护程度和推荐准确性的关系. 然而, 在他们的文献中并没有给出如何设置隐私预算取值的具体办法.

2.2 差分隐私中隐私预算的优化设置

隐私预算是差分隐私模型中的一项重要参数, 它一方面控制着用户数据的隐私保护程度, 另一方面也决定着扰动后数据的可用性. 对推荐系统而言, 用户数据的可用性与推荐的准确性密切相关. 设置合适的隐私预算是平衡隐私保护与推荐准确性的关键. 研究者们已经提出了一些优化设置隐私预算的方法 [30~32]. Lee 等 [30] 针对经典的梯度下降算法提出了为梯度值提供差分隐私保护的方法. 所提方法考虑到梯度下降算法在迭代过程中对梯度值准确度的要求随迭代次数的增加而增加, 因此在为每轮的梯度值分配隐私预算时, 隐私预算也随迭代次数的增加而增加. Xia 等 [32] 针对兴趣点查询应用中存在的位置隐私泄露问题, 提出了基于拉普拉斯 (Laplace) 机制的位置隐私保护方法, 并提出了基于路径前缀树的隐私预算分配方法, 可降低添加的噪声对用户查询结果准确性的影响. Jin 等 [33] 也针对位置查询应用中的隐私问题提出了一种满足地理不可区分性 (地理差分隐私) 的隐私保护机制, 并提出了一种根据用户访问次数设置隐私预算的方法. Ye 等 [34] 提出了一种满足差分隐私的用户轨迹数据保护机制, 并提出了一种基于滑动窗口的隐私预算设置方法, 该方法允许用户根据自己对位置的敏感程度为不同的位置数据设置不同的隐私预算.

现有研究大多考虑的是中心化差分隐私模型中的隐私预算设置问题, 所提设置方法由中心服务器实施. 而在本地化差分隐私模型中, 数据的隐私化处理在用户端完成, 这就意味着可以让每位用户根据实际情况独立设置其隐私预算的取值. 因此, 本文研究在满足本地化差分隐私的推荐系统中帮助用户选择隐私预算取值的方法.

3 预备知识

3.1 本地化差分隐私

假设存在 N 个用户, 每个用户持有一个敏感数据 x. 传统的差分隐私将用户敏感数据收集到服务器, 服务器对数据进行处理后, 发布满足差分隐私的相关统计信息. 传统的差分隐私对于用户敏感数据的保护是基于可信服务器假设的, 即保证服务器不会窃取或泄露用户的敏感数据. 然而, 完全可信的服务器在实际中几乎是不存在的. 因此, 在服务器不可信的场景下, 需引入本地化差分隐私 (local differential privacy, LDP) 来保护用户的隐私. LDP 的正式定义 [11] 如下.

定义1 (本地化差分隐私 (LDP)) 给定一个随机算法 $\mathcal{A}$ 及其定义域 $\mathrm{Dom}(\mathcal{A})$ 和值域 $\mathrm{Ran}(\mathcal{A})$, 若算法 $\mathcal{A}$ 在任意两组数据 X^i 和 X^j $(X^i, X^j \in \mathrm{Dom}(\mathcal{A}))$ 上得到相同的输出结果 $X^*(X^* \in \mathrm{Ran}(\mathcal{A}))$ 的概率满足下列不等式, 则算法 $\mathcal{A}$ 满足 ϵ-本地化差分隐私.

$$\Pr[\mathcal{A}(X^i) = X^*] \leqslant e^{\epsilon} \times \Pr[\mathcal{A}(X^j) = X^*]. \tag{1}$$

根据定义 1 得知, 满足 ϵ-本地化差分隐私的算法可保证任意两组数据的输出结果具有一定程度相似性, 从而保证了其他人无法根据算法 $\mathcal{A}$ 的输出结果判断出输入数据是什么. 本地化差分隐私要求每位用户独立在本地对其敏感数据进行扰动, 然后再将扰动后的数据发送给服务器, 服务器基于用户扰动后的数据完成相关任务, 因此避免了不可信服务器的隐私泄露.

前文描述的是满足本地化差分隐私的随机算法需要满足的条件, 而实现对用户敏感数据的保护需要选取合适的数据扰动机制. 前人的研究中已经提出了多种数据扰动机制来处理不同类型的用户敏感数据. 随机响应机制 (randomized response, RR) 是经典的数据扰动机制, 其主要用于对包含两种可能取值的数据进行扰动, 具体而言, 假设用户敏感数据 $x \in \{0, 1\}$, 用户以概率 p 得到扰动后的数据为 x, 以 $1-p$ 的概率得到 $1-x$. 为了保证随机响应机制满足本地化差分隐私, 隐私预算 ϵ 可设置为 $\ln(p/(1-p))$.

3.2 强化学习

强化学习 (reinforcement learning, RL) 是机器学习的方法之一, 它模拟了人类与环境交互的过程中根据环境反馈学习知识的过程 [35]. 具体来说, 当学习者需解决某个特定问题时, 不会提前知道自己应该采取的动作, 而需根据当前环境做出判断, 并根据强化学习策略选择动作, 之后学习者会得到环境的反馈, 即获得相应的收益 (奖励) 值, 将该收益值反馈给强化学习策略, 以完成 "学习" 过程 [36]. 学习者在后续遇到类似问题时, 根据以往经验选择可能令其获得较高收益的动作即可.

强化学习的目标是在有限的时间内最大化累计收益值. 在强化学习任务中, 学习者选择一个动作所获得的收益一般是在一段时间之后才能被观测到. 本文研究一种较为简单的情况: 最大化单步收益, 也就是说, 在当前时刻学习者选择一个动作就会带来相应的即时收益. 单步强化学习任务中最经典的模型为多臂赌博机模型 (multiarmed bandit, MAB) [16]. 具体而言, 假设在多臂赌博机模型中存在 K 个摇臂 $E = \{e_1, e_2, \ldots, e_K\}$, 学习者每选择一个摇臂都会带来相应的收益. 学习者的目标是通过学习策略选择摇臂, 在有限摇臂次数下, 最大化自己的累计收益. 为了达到目标, 学习者需在每次选择摇臂时选择出令其获得收益最高的摇臂. 如果学习者知道每个摇臂可获得的收益, 则在每次都选择可获得收益最高的摇臂即可. 然而, 学习者在选择摇臂通常之前不能确切知道每个摇臂的收益, 但可以对其进行估计, 选择出期望收益或者平均收益最高的摇臂. 我们将学习者在时刻 t 选择的摇臂记为 Y_t, 对应的收益记作 μ_t. 学习者选择期望收益最高的摇臂 Y_t 的方式为

$$Y_t = \underset{e_i \in E}{\mathrm{argmax}}(\mathbb{E}[\mu_t | Y_t = e_i]). \tag{2}$$

通过上述分析可知, 学习者想要最大化累计收益, 需在每一次选择摇臂时, 选择期望收益最高的摇臂. 如果每个摇臂带来的收益是确定的, 那么学习者通过遍历所有摇臂就可以得到期望收益最高的摇臂, 在之后一直选择这一摇臂即可. 而在实际中, 每个摇臂的收益通常是以符合某种概率分布的形式随机产生的. 因此, 遍历一次所有摇臂无法找到真实具有最大期望收益的摇臂. 这就引出了多臂赌博机模型中需面临的一个挑战: "开发" 与 "试探" 的权衡 [37, 38], 即在选择摇臂时, "开发" 已有经验, 选

表 1 符号与含义

Table 1 Notations

Notation	Description
U, N	The user set; the number of users
I, M	The item set; the number of items
u_n	The nth user in the set U
v_n^t, p_n^t	The original data of the user u_n; the perturbed data
ϵ_n^t	The privacy budget selected by the user u_n in the t round of interaction
L_n^t	The privacy loss of the user u_n in the t round of interaction
S_n^t	The item set recommended to the user u_n
C_n^t	The items selected by the user u_n from the set S_n^t
R_n^t	The reward of the user u_n in the t round of interaction
μ_n^t	The payoff of the user u_n in the t round of interaction
$\tilde{\mu}_{n,i}^t$	The expected payoff of the arm e_i for the user u_n in the t round of interaction
E, e_i	The set of arms; the ith arm in the set

择在过去为其带来较高收益的摇臂, 还是 "试探" 其他摇臂以发现可能会带来更大收益的摇臂. 在实际应用中, 需提出相应的学习策略解决这一权衡问题.

4 系统模型设计

本节分别从隐私保护推荐模型和多臂赌博机模型两个方面来对本文提出的系统模型进行详细介绍. 在此之前, 首先列出本节所用到的主要符号及其含义, 如表 1 所示.

4.1 隐私保护推荐模型

考虑一个包含 N 个用户和 M 个项目的推荐系统, N 个用户的集合记为 $U=\{u_1,u_2,\ldots,u_N\}$, M 个项目的集合记为 $I=\{i_1,i_2,\ldots,i_M\}$. 每当用户 u_n 需要从服务器处获得推荐时, 需提供数据给服务器. 例如, 在电影推荐系统中, 用户需将观影记录发送给服务器, 以供服务器了解用户的喜好; 位置推荐中, 用户需上传自己的当前位置, 以便服务器返回用户附近的位置点. 本文将用户发送数据给服务器而后得到推荐结果的过程称为用户与服务器的一轮 "交互". 用户 u_n 在第 t 轮交互时发送给服务器的数据记为 v_n^t. 由于 v_n^t 中可能包含用户的敏感信息, 直接将 v_n^t 发送给推荐服务器可能会泄露用户的隐私. 因此, u_n 在将 v_n^t 发送给服务器之前, 需使用满足 ϵ_n^t-LDP 的随机算法 $\mathcal{A}$ 对 v_n^t 进行扰动, 其中 ϵ_n^t 是用户设置的隐私预算. 扰动结果 $\mathcal{A}(v_n^t)$ 是一个随机变量, 服从一个与 ϵ_n^t 有关的概率分布 $Q_{\epsilon_n^t}$. 用户发送给服务器的数据, 记为 p_n^t, 可以理解为对随机变量 $\mathcal{A}(v_n^t)$ 进行一次抽样得到的样本.

根据本地化差分隐私的定义, 扰动后的数据仍有可能透露关于原始数据的信息. 因此, 用户 u_n 将 p_n^t 提供给服务器还是会遭受隐私损失的. 隐私损失 L_n^t 的大小取决于数据的扰动程度, 数据扰动程度越大, 扰动后数据 p_n^t 与原始数据 v_n^t 的差距也就越大, p_n^t 中包含的敏感信息也就越少. 因此, 用户 u_n 可以用 p_n^t 与 v_n^t 之间的差距来衡量隐私损失 L_n^t. 我们定义

$$L_n^t = f_{\text{loss}}(d_{\text{loss}}(v_n^t, p_n^t)), \tag{3}$$

其中, $d_{\text{loss}}(v_n^t,p_n^t)$ 是度量 p_n^t 与 v_n^t 之间差距的函数, 差距越大, $d_{\text{loss}}(v_n^t,p_n^t)$ 值也就越大, 函数 f_{loss} 随

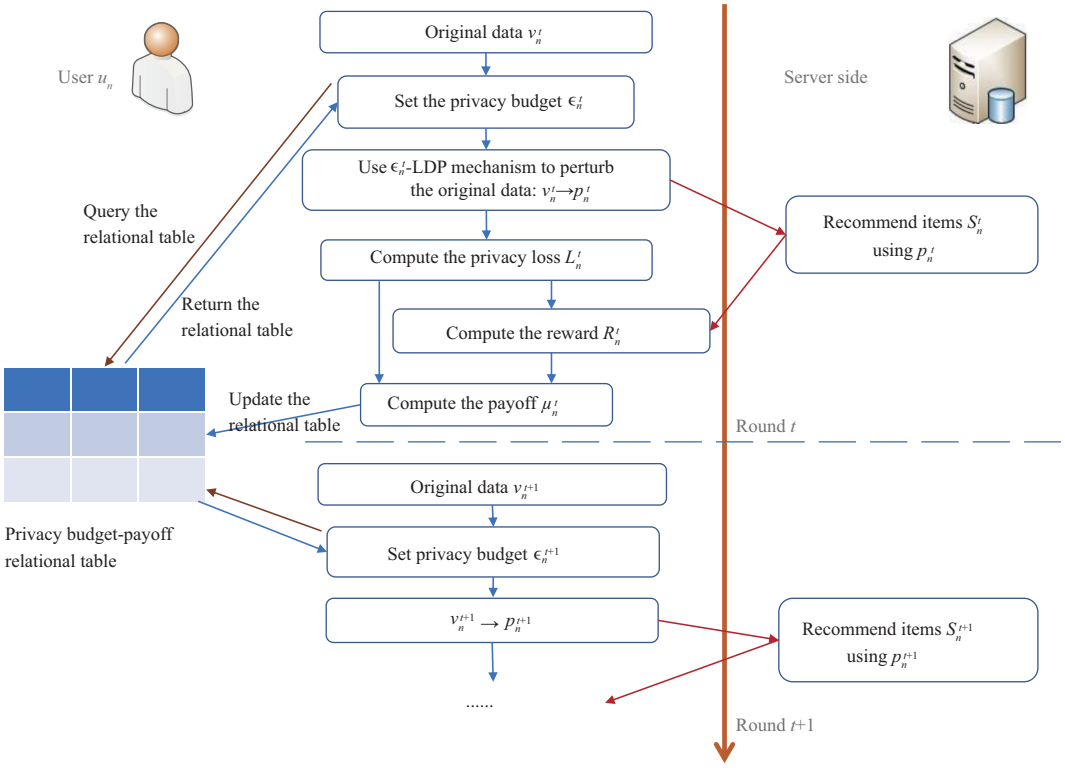

图 1　(网络版彩图) 隐私保护推荐模型

Figure 1　(Color online) Privacy preserving recommendation model

$d_{\text{loss}}(v_n^t, p_n^t)$ 单调递减. 在第 6 节, 我们会结合特定的应用场景给出 $d_{\text{loss}}(v_n^t, p_n^t)$ 和 f_{loss} 的具体形式. 由于 $p_n^t \sim Q_{\epsilon_n^t}$, 所以 L_n^t 服从一个与 ϵ_n^t 有关的未知概率分布.

服务器收到 p_n^t 后, 利用某种推荐算法为用户 u_n 推荐其可能感兴趣的项目集 S_n^t. 用户 u_n 收到 S_n^t 后, 从中选择出真正感兴趣的项目集合 C_n^t. 我们称 C_n^t 为用户收到的有效推荐集. 可以用集合 S_n^t 中有效推荐的占比 $d_{\text{acc}}(S_n^t, C_n^t)$ 来衡量服务器的推荐准确率, 其中 $d_{\text{acc}}(S_n^t, C_n^t) = |S_n^t \cap C_n^t|/|S_n^t|$. 这里需强调的是, 用户提供的扰动数据 p_n^t 越接近真实数据 v_n^t, 服务器基于 p_n^t 分析出的用户喜好越准确, 为用户提供推荐的准确率也就越高. 我们用服务器给出的推荐准确性来衡量用户 u_n 在一轮交互中获得的报酬. 推荐准确性越高, 用户获得的报酬也就越高. 用 R_n^t 表示用户 u_n 在第 t 轮交互中获得的报酬, R_n^t 的计算方式如下:

$$R_n^t = f_{\text{reward}}(d_{\text{acc}}(S_n^t, C_n^t)), \tag{4}$$

其中, 函数 f_{reward} 随 $d_{\text{acc}}(S_n^t, C_n^t)$ 单调递增. 由于服务器基于 p_n^t 为用户 u_n 提供推荐, 且 $p_n^t \sim Q_{\epsilon_n^t}$, 因此 R_n^t 服从一个与 ϵ_n^t 有关的未知概率分布. 至此, 用户 u_n 与服务器完成了第 t 轮交互, 用户计算在此轮交互中获得的收益 μ_n^t. μ_n^t 的计算方式如下:

$$\mu_n^t = f_{\text{payoff}}(L_n^t, R_n^t). \tag{5}$$

这里的函数 f_{payoff} 需要满足以下两个条件: (1) 给定 R_n^t, 收益 μ_n^t 随着 L_n^t 的增加而减少; (2) 给定 L_n^t, 收益 μ_n^t 随着 R_n^t 的增加而增加. 由于 L_n^t 和 R_n^t 分别服从与 ϵ_n^t 有关的未知概率分布, 因此收益 μ_n^t 也服从一个与 ϵ_n^t 有关的未知概率分布. 上述模型的流程图如图 1 所示.

用户每次请求服务器推荐时, 其目标是最大程度的提高收益. 前面我们讲到, 用户 u_n 在第 t 轮与服务器交互中, 提出的模型为数据 v_n^t 提供满足 ϵ_n^t-LDP 的隐私保护, 这里隐私预算 ϵ_n^t 如何选取, 直接影响用户的收益. 众所周知, ϵ_n^t 越小, 数据的隐私保护程度越高. 所以如果 ϵ_n^t 设置过小, 会使得数据的扰动程度过高, p_n^t 与 v_n^t 的差距过大, 从而导致服务器无法做出准确的推荐. ϵ_n^t 设置过大, 又会导致用户产生的隐私损失过大. 本文假设用户 u_n 不清楚服务器使用的推荐算法, 也不了解每个项目的特征与属性, 这种情况下, 用户无法直接确定隐私预算 ϵ_n^t 的取值, 而必须通过与服务器的多次交互逐步学习出最佳的 ϵ_n^t.

4.2 多臂赌博机模型

4.1 小节介绍的隐私保护推荐模型中, 用户每次请求服务器推荐时, 需要设置隐私预算的取值来对用户原始数据进行扰动, 此时, 用户会面临一个 “开发” 与 “试探” 的选择问题. 所谓 “开发”, 是指利用现有知识设置隐私预算的取值, 通常选择迄今为止带来最高收益的隐私预算; 所谓 “试探”, 是指尝试可能会带来更高收益的其他取值. 设计学习策略去解决 “开发” 与 “试探” 的权衡问题通常被形式化为赌博机问题. 接下来, 我们详细介绍如何将本文所提模型形式化为赌博机问题. 用户 u_n 在第 t 轮与服务器交互时, 从集合 $E \triangleq \{e_i | e_i = i/K, i = 1, \ldots, K\}$ 中选择一个值 e_i 作为其隐私预算 ϵ_n^t. 根据赌博机术语, 这里的每个取值 $e_i \in E$ 都被称作为一个摇臂. 如果用户 u_n 选择 e_i 作为其在第 t 轮的隐私预算, 则使用满足 e_i-LDP 的数据扰动机制对数据 v_n^t 进行扰动, 得到扰动后的数据 p_n^t 上传至服务器. 此时用户 u_n 利用式 (3) 计算隐私损失. 待收到服务器的推荐项目集 S_n^t 后, 用户 u_n 利用集合 S_n^t 和式 (4) 计算此轮推荐的报酬, 即可利用式 (5) 计算在第 t 轮交互中获得的收益 μ_n^t. 从 4.1 节的讨论可知, 用户 u_n 选择任意臂 $e_i \in E$ 后可获得的收益 μ_n^t 是一个随机变量, 服从一个与 e_i 有关的未知概率分布 P_{e_i}, 用户 u_n 选择摇臂 e_i 获得的期望收益为

$$\tilde{\mu}_{n,i}^t = \int \mu_n^t P_{e_i} \mathrm{d}\mu_n^t. \tag{6}$$

由于用户 u_n 缺少对概率分布 $P_{e_i}(\cdot)$ 的先验知识, 需要应用一个学习策略去帮助用户选择摇臂赋值给 ϵ_n^t. 前面我们提到, 每一轮用户 u_n 在选择摇臂时, 都面临着开发和试探的选择: 选择开发, 用户利用当前知识选择期望收益最大的摇臂 e_{I^*}, 其中 $I^* = \mathrm{argmax}_i \tilde{\mu}_{n,i}^{(t)}$; 选择试探, 从其他摇臂中随机选取一个摇臂赋值给 ϵ_n^t, 用以提升对其他摇臂收益估计的准确性. 本文提出两种策略去解决不同场景下的开发 – 试探的权衡问题.

在上述模型中, 每位用户与服务器交互的目标是在有限的交互次数内最大化累计收益. 令 $\{e_{I_1}, e_{I_2}, \ldots, e_{I_T}\}$ 为用户 u_n 与服务器在 T 轮交互中利用学习策略选择出的摇臂序列, $U_n(T)$ 为 T 轮交互后获得的累计收益, 其计算方式为

$$U_n(T) = \sum_{t=1}^{T} \mu_{n,I_t}^t, \tag{7}$$

其中, μ_{n,I_t}^t 为用户 u_n 在第 t 轮与服务器交互中选择摇臂 e_{I_t} 所获得的真实收益值. 用户在与服务器的 T 轮交互中, 使用学习策略选择每轮隐私预算的优化目标, 即为 $\max U_n(T)$.

5 学习策略

第 4 节提出的模型假设每位用户不了解服务器所使用的推荐算法和每个项目的特征属性, 因而直接找出最优隐私预算设置是很困难的. 一种可行的方法是设计某种行为策略, 使每位用户在与服务器

迭代交互的过程中, 依据行为策略不断调整隐私预算的设置, 使该用户累计收益达到最大化. 本节引入了基于置信度上界 (UCB) [20~22] 的学习策略来帮助用户解决隐私预算的优化设置问题. 同时, 我们改进了基础的 UCB 策略以适应用户数据敏感程度不同的情况.

5.1 基于置信度上界的学习策略

第 4 节提出的模型中, 用户 u_n 在第 t 轮与服务器交互时, 需要从集合 E 中选择一个摇臂赋值给 ϵ_n^t. 对于任意 $e_i \in E$, 用户 u_n 利用前 $t-1$ 轮交互中获得的收益情况来估计 e_i 在第 t 轮交互中的期望收益, 并选择期望收益最高的摇臂 $e_{I_t^*}$ 赋值给 ϵ_n^t. 在赌博机术语中, 上述动作称为 “开发”. 然而, 基于现有观察得到的期望收益可能会不准确, 从而导致对于每个摇臂的期望收益估计存在不确定性, 所以 “试探” 是必须的.

基于置信度上界 (UCB) 的学习策略被广泛应用于解决赌博机问题中开发与试探的权衡问题. UCB 策略的基本思想是在每次选择摇臂时, 同时考虑每个摇臂的期望收益和期望收益估计结果的不确定性. 具体而言, 在用户 u_n 与服务器的 T 轮交互中, UCB 策略为每个摇臂 e_i 记录两个值: $N_{n,i}$ 和 $\tilde{\mu}_{n,i}$. $N_{n,i}$ 表示 T 轮交互中 e_i 被选择的次数, $\tilde{\mu}_{n,i}$ 表示选择 e_i 产生收益的平均值. 我们可以将 $\tilde{\mu}_{n,i}$ 看作对摇臂 e_i 的期望收益的估计, 再用 $N_{n,i}$ 衡量期望收益估计的不确定性. 在用户 u_n 第 t 轮与服务器的交互中, 利用 UCB 策略选择摇臂下标的具体方式如下:

$$I_t = \underset{i=1,2,\ldots,K}{\operatorname{argmax}} \left(\tilde{\mu}_{n,i} + \alpha \sqrt{\frac{\ln t}{N_{n,i}}} \right), \tag{8}$$

其中, I_t 是选择出来的摇臂下标, $\tilde{\mu}_{n,i} + \alpha\sqrt{\frac{\ln t}{N_{n,i}}}$ 参数 α 控制置信度的宽度. UCB 策略利用平方根项度量每个摇臂期望收益估计的不确定性, 每次摇臂 e_i 被选择时, $N_{n,i}$ 值增大, 不确定性可能降低; 而每次选择 e_i 以外的其他摇臂时, 分子上的 t 增大, 而 $N_{n,i}$ 没有变化, 所以不确定性增加了. 随着交互次数的增加, 具有较低估计收益的摇臂和已经被选择了多次的摇臂被选择的频率较低. 将 UCB 策略应用于所提模型的详细过程如算法 1 所示.

Algorithm 1 UCB learning policy

Input: $\alpha \in \mathbb{R}^+$.

for $t = 1$ to K **do**
  Choose arm $I_t = t$;
  $N_{n,t} = 1$;
  Use Eqs. (3)~(5) to compute L_n^t, R_n^t, and μ_n^t;
  $\tilde{\mu}_{n,t} = \mu_n^t$;
end for
for $t = K+1$ to T **do**
  Choose arm $I_t = \operatorname{argmax}_{i=1,2,\ldots,K} \left[\tilde{\mu}_{n,i} + \alpha\sqrt{\frac{\ln t}{N_{n,i}}}\right]$;
  $N_{n,I_t} = N_{n,I_t} + 1$;
  Use Eqs. (3)~(5) to compute L_n^t, R_n^t, and μ_n^t;
  $\tilde{\mu}_{n,t} = \frac{\tilde{\mu}_{n,t} \times (N_{n,I_t} - 1) + \mu_n^t}{N_{n,I_t}}$;
end for

5.2 考虑数据敏感度的学习策略

5.1 小节中, 用户 u_n 利用 UCB 策略选择第 t 轮交互中隐私预算 ϵ_n^t 的取值, ϵ_n^t 的大小决定了用户数据 v_n^t 的扰动程度. 然而, UCB 策略选择 ϵ_n^t 时, 仅考虑了每个摇臂的预期收益和收益估计的不确

定性, 未考虑原始数据 v_n^t 的敏感程度. 实际中, 用户对于不同数据的敏感程度往往不同, 且不同用户对于隐私的重视程度也不同. 例如, 用户对于自己的姓名和性别信息通常敏感程度较低, 而对于患病信息则十分敏感; 位置数据对于名人来说是高度敏感的, 因为非理性的粉丝可能会通过位置数据来跟踪他们, 但对于公交车司机来说并不是那么敏感. 用户更希望为敏感程度更高的数据提供更强的隐私保护, 也就是为其选择更小的 ϵ_n^t 的取值. 本小节考虑了用户对数据的敏感程度, 改进了 UCB 策略. 下面详细介绍改进后的 UCB 策略.

当用户 u_n 请求服务器推荐时, 需发送数据 v_n^t 给服务器, 此时, 用户 u_n 赋予数据 v_n^t 敏感程度 $\varphi_n^t \in [0,1]$, 表示 u_n 对数据 v_n^t 的隐私重视程度. φ_n^t 值越大, 表示 v_n^t 的敏感程度越高. 而后, 用户 u_n 利用改进后的 UCB 策略从集合 E 中选择一个摇臂 e_{I_t} 赋值给 ϵ_n^t, e_{I_t} 的选择方法如下:

$$I_t = \underset{i=1,2,\ldots,K}{\operatorname{argmax}} \left(\tilde{\mu}_{n,i} + \alpha \sqrt{\frac{\ln t}{N_{n,i}}} + \beta w_i \right), \tag{9}$$

其中, w_i 为考虑用户对数据的敏感程度后, 为不同的摇臂所赋予的权重. 权重 w_i 的计算方式为

$$w_i = 1 - \frac{|\lceil (1-\varphi_n^t)K \rceil - i|}{K}. \tag{10}$$

由上式可以看出, 在摇臂集合 $E \triangleq \{e_i | e_i = i/K, i = 1, \ldots, K\}$ 中, 随着下标 i 的增大, 摇臂 e_i 的值增大, 对应的权重 w_i 呈现先增后减的趋势, 当 $i = \lceil (1-\varphi_n^t)K \rceil$ 时, 权重 w_i 最大. 由此可以看出, 数据敏感度 φ_n^t 越大, 集合 E 中具有最大敏感度权重的摇臂值 $e_{\lceil (1-\varphi_n^t)K \rceil} = \frac{\lceil (1-\varphi_n^t)K \rceil}{K}$ 越小, 此时就越易选择具有较小值的摇臂为隐私预算 ϵ_n^t 赋值. 另外, 由于我们的模型中加入了对数据敏感程度的考虑, 在计算隐私损失时, 也应将数据敏感程度考虑进来, 即用户认为更加敏感的数据, 在遭受到泄露时, 隐私损失程度应该更高. 所以, 我们将隐私损失计算 (3) 式改为

$$L_n^t = \varphi_n^t \times f_{\text{loss}}(d_{\text{loss}}(v_n^t, p_n^t)). \tag{11}$$

改进后的 UCB 策略的详细过程如算法 2 所示, 我们将改进后的 UCB 策略叫做 SM-UCB.

Algorithm 2 SM-UCB learning policy

Input: $\alpha \in \mathbb{R}^+$, $\theta \in [0,1]$.

for $t = 1$ to K **do**
  Choose arm $I_t = t$;
  $N_{n,t} = 1$;
  Use original data v_n^t to compute data sensitivity φ_n^t;
  Use Eqs. (11), (4), and (5) to compute L_n^t, R_n^t, and μ_n^t;
  $\tilde{\mu}_{n,t} = \mu_n^t$;
end for
for $t = K+1$ to T **do**
  Choose arm $I_t = \operatorname{argmax}_{i=1,2,\ldots,K} \left[\tilde{\mu}_{n,i} + \alpha \sqrt{\frac{\ln t}{N_{n,i}}} + \beta w_i \right]$;
  $N_{n,I_t} = N_{n,I_t} + 1$;
  Use Eqs. (11), (4), and (5) to compute L_n^t, R_n^t, and μ_n^t;
  $\tilde{\mu}_{n,t} = \frac{\tilde{\mu}_{n,t} \times (N_{n,I_t} - 1) + \mu_n^t}{N_{n,I_t}}$;
end for

6 应用场景举例

本节给出两个具体的推荐场景的实例: 电影推荐和位置推荐, 来具体说明前文所提模型, 并分别在两个推荐场景中给出用户数据的扰动机制和用户收益的计算方法.

6.1 电影推荐场景

由于本文研究的重点在于如何利用多臂赌博机模型帮助用户选择出可最大化累计收益的隐私预算取值, 所以为了简便, 本小节用于举例的电影推荐模型选择最基本的基于内容的推荐模型. 具体而言, 用户发送电影历史观看记录给推荐服务器, 服务器根据电影的属性特征 (例如: 电影类型、演职人员、上映时间和内容简介等) 计算出任意电影之间的相似度, 为用户推荐与该用户感兴趣的电影相似的电影.

假设在一个电影推荐系统中, 对于任意用户 u_n, 在收到服务器为其推荐的电影 S_n^t 后, u_n 选择是否观看该电影, 如果选择观看, 则认为用户 u_n 对电影 S_n^t 感兴趣. 我们引入变量 y_n^t 表示用户 u_n 是否观看电影 S_n^t. 观看: $y_n^t = 1$; 否则: $y_n^t = 0$. 如果用户 u_n 观看了电影 S_n^t, 则为这部电影赋予敏感程度 φ_n^t.

当用户 u_n 想要再次获得电影推荐时, 为了得到更加准确的推荐, 用户 u_n 发送电影 S_n^t 的观看情况 y_n^t 给服务器, 以便服务器了解该用户的喜好. 为了保护用户 u_n 的隐私, u_n 首先在本地利用随机扰动机制对 y_n^t 进行满足 ϵ_n^t-LDP 的数据扰动, 得到扰动后的观看行为 $p_n^t \in \{0, 1\}$, 其中, 隐私预算 ϵ_n^t 的取值利用第 5.2 小节提出的学习策略得到. 用户 u_n 将 p_n^t 发送至服务器, 而后计算发送 p_n^t 给服务器所带来的隐私损失 L_n^t, 计算方式如下:

$$L_n^t = y_n^t \times \varphi_n^t. \tag{12}$$

在服务器端, 我们首先将服务器预先得到的任意两部电影 i_m 和 i_z 的相似度记为 $\mathrm{Sim}(m, z)$. 对于用户 u_n, 服务器根据 u_n 发送过来的 p_n^t, 为 u_n 推荐其可能感兴趣的电影, 记作 S_n^{t+1}. 若 $p_n^t = 1$, 服务器根据如下公式得到 S_n^{t+1}:

$$S_n^{t+1} = \underset{i_m \in I}{\mathrm{argmax}}(\mathrm{Sim}(m, \pi(S_n^t))), \tag{13}$$

其中, $\pi(S_n^t)$ 表示电影 S_n^t 在电影集合 I 中的下标. 若 $p_n^t = 0$, 服务器随机从电影集合 I 中选择一部电影作为 S_n^{t+1}. 然后, 服务器将推荐结果 S_n^{t+1} 发送给用户 u_n. 用户 u_n 在收到 S_n^{t+1} 后, 选择是否观看 S_n^{t+1}, 若观看, $y_n^{t+1} = 1$, 否则 $y_n^{t+1} = 0$. 此时 u_n 计算获得的报酬 R_n^t:

$$R_n^t = y_n^{t+1}. \tag{14}$$

至此, 用户 u_n 完成了与服务器的第 t 轮交互, 而后计算此轮交互中获得的收益 μ_n^t, 计算方式如下:

$$\mu_n^t = \frac{\gamma \times R_n^t - L_n^t}{\gamma + 1} + 1, \tag{15}$$

其中, $\gamma \in [0, +\infty)$ 为平衡参数, 用于调节获得的收益对于隐私损失和推荐报酬的敏感度, γ 越小, 表示收益对于隐私损失的敏感程度更高, 模型更偏向于保护用户隐私. 公式的后半部分加 1 是为了保证计算得到的收益不小于 0, 由于每一次计算收益时都会加 1, 所以不影响比较不同摇臂产生收益大小.

6.2 位置推荐场景

位置推荐是推荐领域的一项重要应用, 我们选择一个简单的位置推荐场景说明所提模型如何应用于位置推荐中. 在此之前, 我们先介绍本小节使用的位置数据扰动方法: PL 方法, 该方法可为用户位置数据提供满足地理不可区分性的隐私保护. Andrés 等 [39] 首次提出了地理不可区分性的概念, 这一概念也被称作为地理差分隐私. 地理不可区分性的定义如下.

定义2 (地理不可区分性 (geo-indistinguishability)) 给定随机算法 $\mathcal{A}$ 和某个地理区域 X, 若算法 $\mathcal{A}$ 在任意两个位置点 X^i 和 X^j $(X^i, X^j \in X)$ 上得到相同的输出结果 $z \in Z$ 的概率满足以下不等式, 则称算法 $\mathcal{A}$ 满足 ϵ-地理不可区分性.

$$\Pr[\mathcal{A}(X^i) = z] \leqslant e^{\epsilon d(X^i, X^j)} \times \Pr[\mathcal{A}(X^j) = z]. \tag{16}$$

上述定义给出了随机算法 $\mathcal{A}$ 满足地理不可区分性的条件, 将该算法 $\mathcal{A}$ 从服务器端迁移到用户端执行, 则算法 $\mathcal{A}$ 满足本地地理不可区分性 (本地地理差分隐私). 本地地理不可区分性与本地化差分隐私的定义类似, 都是通过控制输出结果的相似性来实现对用户数据的隐私保护. 不同点在于, 在本地化差分隐私中, $d(X^i, X^j) = 1$ 表示敏感数据 X^i 和 X^j 之间的汉明 (Hamming) 距离为 1 (即, 相邻数据库), 而在本地地理不可区分性中, $d(X^i, X^j)$ 为位置点 X^i 和 X^j 之间的欧式距离.

Andrés 等 [39] 同时提出了一种满足地理不可区分性的位置数据扰动方法 (PL 方法), 该方法将极坐标拉普拉斯噪声注入到用户的真实位置 x 中, 得到扰动后的位置 z. 该方法将真实位置 x 表示为极坐标原点. 以如下方式计算出扰动后的位置与真实位置 x 之间的距离 r:

$$r = -\frac{1}{\epsilon}\left(W_{-1}\left(\frac{p-1}{e}\right) + 1\right), \tag{17}$$

其中, W_{-1} 是 Lambert W 函数 (-1 分支). 然后, 该方法再从区间 $[0, 2\pi)$ 中随机选取出夹角 θ. 最后, 得到扰动后的位置点 z 的坐标为

$$z = x + \langle r\cos\theta, r\sin\theta\rangle. \tag{18}$$

接下来, 我们介绍将前文所提模型应用于位置推荐中的方法. 假设在位置推荐中, 用户 u_n 想要服务器推荐距离其所在位置 1 km 以内的地点, u_n 需上传当前位置给服务器. 记用户 u_n 当前位置为 v_n^t, 为了保护位置隐私, 用户 u_n 在本地利用 PL 方法对位置 v_n^t 进行满足 ϵ_n^t-地理不可区分性的扰动, 得到扰动后的位置点 p_n^t, 隐私预算 ϵ_n^t 的取值使用第 5.2 小节提出的学习策略得到. 同时, u_n 为当前位置点 v_n^t 赋予敏感程度 φ_n^t. 用户 u_n 将扰动后的位置 p_n^t 发送至服务器, 而后计算发送 p_n^t 给服务器所带来的隐私损失 L_n^t, 我们用扰动后得到的位置点 p_n^t 与用户真实位置点 v_n^t 的距离衡量用户的隐私损失, L_n^t 的计算方式如下:

$$L_n^t = \frac{\varphi_n^t}{e^{d(v_n^t, p_n^t)}}, \tag{19}$$

其中, $d(v_n^t, p_n^t)$ 表示 v_n^t 和 p_n^t 之间的欧式距离. 在服务器端, 服务器根据扰动后的位置点 p_n^t, 为用户 u_n 推荐与位置点 p_n^t 距离 3 km 以内的地点集, 记为 S_n^t. 用户 u_n 收到 S_n^t 后, 根据集合 S_n^t 中的地点与用户真实位置点 v_n^t 之间的距离, 选择出符合要求的集合 C_n^t 作为此轮推荐中的有效推荐集. 此时用户 u_n 计算本轮推荐获得的报酬 R_n^t:

$$R_n^t = \frac{|S_n^t \bigcap C_n^t|}{|S_n^t|}. \tag{20}$$

至此, 用户 u_n 完成了与服务器的第 t 轮交互, 而后计算第 t 轮交互中获得的收益 μ_n^t:

$$\mu_n^t = \frac{\gamma \times R_n^t - L_n^t}{\gamma + 1} + 1, \tag{21}$$

其中, $\gamma \in [0, +\infty)$, 用于调节获得的收益对于隐私损失和推荐报酬的敏感度.

7 仿真实验

前面的章节中, 将 UCB 策略应用于满足本地化差分隐私的推荐系统中去解决隐私预算的设置问题, 同时提出了 SM-UCB 策略以适应用户数据敏感程度不同的情况. 为了评估 UCB 和 SM-UCB 策略的有效性, 我们使用真实世界数据集做了一系列仿真实验. 本节首先介绍实验所用的数据集, 然后介绍实验所需参数, 并对不同参数设置下的实验结果进行对比, 最后引入两个简单的策略分别与 UCB 和 SM-UCB 策略进行对比, 以验证 UCB 和 SM-UCB 策略的可行性.

7.1 实验设置

7.1.1 数据集

本小节分别使用来自真实世界的两个数据集: MovieLens[1)]和 Gowalla[2)]进行仿真实验.

MovieLens 数据集包含来自 71567 位用户对 10681 部电影的评分记录. 为了减少数据稀疏性带来的负面影响, 我们筛选出至少观看过 20 部电影的用户, 和至少被 50 位用户观看过的电影, 筛选后的数据集包含 69977 位用户对 2033 部电影的评分记录. 考虑到数据集中缺乏足够的电影描述信息, 且实验的重点在于验证学习策略的有效性, 而非电影相似度的计算方式. 所以为了简便, 我们首先选取至少观看过 500 部电影, 且观看电影不超过 1200 部的用户集, 再从选取的用户集合中随机抽取 20 位用户作为测试集来评估学习策略的效果, 而后用其余 69957 位用户对 2033 部电影和评分记录计算电影之间的余弦相似度.

Gowalla 数据集包含来自 196586 位用户对 1280969 个 POI 产生的 6442892 个打卡数据. 我们抽取了 10 位打卡记录大于 1800 次的用户, 将抽取的 10 位用户的打卡记录作为我们的测试集. 抽取之后的打卡记录包含 10 位用户对于 16608 个 POI 的打卡记录.

7.1.2 实验步骤

本小节使用 MovieLens 数据集验证 5.1 小节所提模型的效果. 在用户与服务器的第 1 轮交互中, 用户随机从电影集合中选择一部电影 S_n^0, 扰动用户对该电影的真实观看行为 y_n^0. 在获得服务器推荐的电影 S_n^1 后, 根据该用户的真实观看行为 y_n^1 计算推荐报酬, 用户在与服务器的下一轮交互时, 再对 y_n^1 进行扰动, 以此循环. 实验中, 我们设置每位用户与服务器的交互轮数为 20000 轮.

本小节使用 Gowalla 数据集验证 5.2 小节所提模型的效果. 实验设置用户每轮与服务器交互时, 从该用户的打卡记录中随机选取一条记录, 扰动记录中的位置并发送给服务器, 而后得到服务器推荐的一组位置点. 实验中, 我们同样设置每位用户与服务器的交互轮数为 20000 轮.

1) http://grouplens.org/datasets/movielens.
2) https://snap.stanford.edu/data/loc-gowalla.html.

7.1.3 参数设置

(1) 交互轮数 T. 第 4.2 小节中提到, 用户与服务器交互的目标是在有限的交互次数内最大化累计收益 $U_n(T)$, 本小节利用累计收益评估所提策略的有效性, 其中 $T = 20000$.

(2) 数据敏感程度 φ_n^t. 为了模拟 SM-UCB 学习策略, 首先, 需要为用户数据指定敏感程度. 考虑到不同用户的隐私重视程度不同, 为了模拟实际环境, 对于任意用户 u_n, 为其生成一个服从均匀分布的随机数 $\varphi_n \in [0.3, 0.8]$, φ_n 表示用户 u_n 的隐私重视程度. 考虑到同一用户对不同数据的敏感程度不同, 用户 u_n 在为 v_n^t 指定敏感程度 φ_n^t 时, 随机产生一个服从正态分布 $N(\mu, \sigma^2)$ 的随机数 $\tilde{\varphi}_n^t$, 其中 $\mu = \varphi_n$, $\sigma = 1/20$. 由于 φ_n^t 的取值应该在 0 到 1 之间, 我们用如下方式选择 φ_n^t 的取值:

$$\varphi_n^t = \begin{cases} 0, & \text{if } \tilde{\varphi}_n^t < 0, \\ 1, & \text{if } \tilde{\varphi}_n^t > 1, \\ \tilde{\varphi}_n^t, & \text{others.} \end{cases} \tag{22}$$

(3) 参数 β 的设置. SM-UCB 策略中, 参数 β 控制数据敏感程度对于摇臂选择的影响力, β 越大, 表示在选择摇臂时, 更多考虑数据敏感程度的影响. 我们分别在 MovieLens 和 Gollowa 数据集上进行了一组不同 β 取值的对比实验, 实验结果得出, $\beta = 10^{-2}$ 时, SM-UCB 策略的表现最好.

(4) 参数 γ 的设置. 在第 6 节的实例场景中, 参数 γ 调节收益对于隐私损失和推荐准确性的敏感程度, γ 值越小, 隐私损失对收益的影响越高. 为了找到最佳的 γ, 我们分别在 MovieLens 和 Gollowa 数据集上对 UCB 和 SM-UCB 策略进行了一组不同 γ 取值的对比实验, 结果表明: MovieLens 数据集上, $\gamma = 1$ 效果最佳; 在 Gollowa 数据集上, 对于 UCB 策略 $\gamma = 1$ 效果最佳, 对于 SM-UCB 策略 $\gamma = 0.5$ 效果最佳.

(5) 摇臂个数设置. 前文提到每位用户从集合 $E \triangleq \{e_i | e_i = i/K, i = 1, \ldots, K\}$ 中选取隐私预算, 在下面的所有仿真实验中, 设置 $K = 10$, 则 $E = \{0.1, 0.2, 0.3, \ldots, 1\}$.

7.2 实验结果

7.2.1 参数 α 的影响

在 UCB 和 SM-UCB 策略中, 参数 α 控制期望收益估计的置信区间宽度, 显著影响着学习策略的性能. 为此, 我们分别在 MovieLens 和 Gollowa 数据集上进行了一组实验, 以测试参数 α 如何影响 UCB 和 SM-UCB 策略的性能. 对于 UCB 策略, 在 MovieLens 数据集上, 我们选取 $\alpha \in \{0.01, 0.02, 0.04, 0.08, 0.1, 0.15, 0.2, 0.25, 0.3, 0.4, \ldots, 2\}$, 在 Gollowa 数据集上, 选取 $\alpha \in \{0.01, 0.02, 0.04, 0.06, 0.08, 0.1, 0.2, \ldots, 2\}$, 分别计算在不同的 α 的取值下, 所有用户在与服务器进行 T 轮交互后产生的累计收益的均值 $U(T) = \frac{1}{N}\sum_{n=1}^{N} U_n(T)$. 实验结果如图 2 所示, 图中可以看出, 在 MovieLens 数据集上, α 取 0.3 时效果最佳; 在 Gollowa 数据集上, α 取 0.05 时效果最佳. 为了分析两个数据集上 α 最佳取值差异大的原因, 我们另外做了一组实验, 实验中用户与服务器进行了 20 轮交互, 每位用户在每轮交互中固定选择一个摇臂 $e_i \in 0.1, 0.5, 1$, 计算所有用户在每轮交互中产生的单步收益的平均值 $\mu^t = \frac{1}{N}\mu_n^t$, 实验结果如图 3 所示. 通过实验结果可以发现, 在 MovieLens 数据集上, 固定选择摇臂 e_i, 每轮所产生的收益的随机性较高, 而在 Gollowa 数据集上, 得到的收益相对稳定, 这是由于扰动机制和推荐算法的差异所导致的. 因此在 Gollowa 数据集上, 用户只需要较少次的 "试探" 就能得出相对准确的期望收益的估计. 类似的, 对于 SM-UCB 策略, 在 MovieLens 数据集上, 选取 $\alpha \in \{0.01, 0.05, 0.1, 0.15, 0.2, 0.25, \ldots, 0.8, 1, 1.2\}$, 在 Gollowa 数据集上, 选取

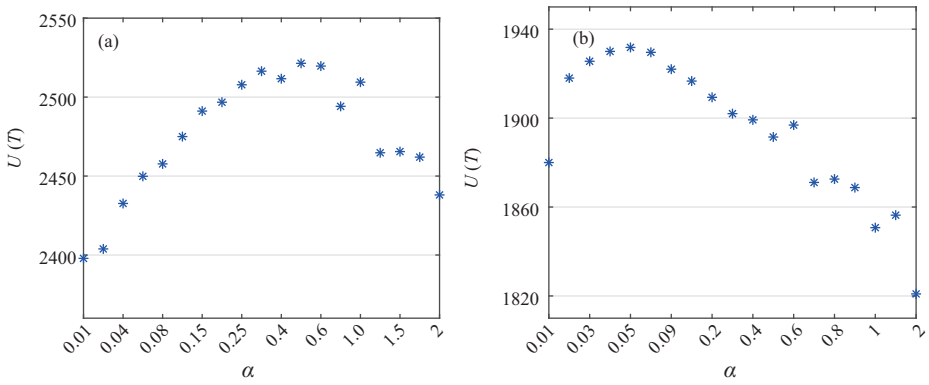

图 2　(网络版彩图) UCB 策略中参数 α 对用户累计收益的影响

Figure 2　(Color online) The influence of the parameter α in the UCB policy on total user payoffs. (a) MovieLens; (b) Gollowa

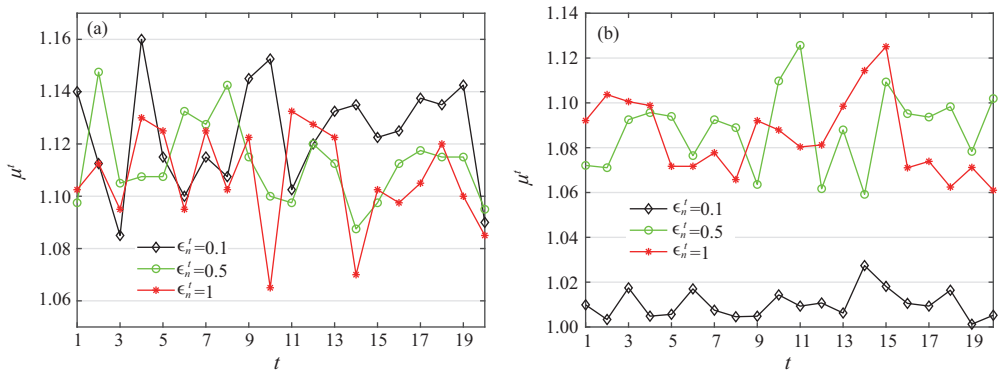

图 3　(网络版彩图) MovieLens 数据集与 Gollowa 数据集在用户单步收益稳定性上的比较

Figure 3　(Color online) A comparison of MovieLens and Goldowa on the stability of user single-step payoff. (a) MovieLens; (b) Gollowa

$\alpha \in \{0.02, 0.04, 0.06, \ldots, 0.28, 0.3, 0.35, 0.4, 0.45, 0.5\}$, 分别计算在不同 α 的取值下的 $U(T)$. 实验结果如图 4 所示, 图中可以看出, 对于 SM-UCB 策略, α 的取值对累计收益的影响不明显, 但基本也可以看出在 MovieLens 数据集上, α 取 0.3 时效果最佳; 在 Gollowa 数据集上, α 取 0.1 时效果最佳. 从图 2 和 4 中可以看出, 4 种曲线基本呈现先升后降的趋势, 且都拥有一个峰值, 我们认为累计收益取到峰值时, 即为 α 的最佳取值, 此时用户 “开发” 与 “试探” 动作达到平衡点.

7.2.2　不同策略的比较

在第 5 节中, 我们给出了两种学习策略: UCB 和 SM-UCB 策略. 为了验证提出策略的有效性, 我们另外给出了两种简单的策略: 第 1 种策略称为 FixHalf, 即用户始终将隐私预算设置为 0.5; 第 2 种策略称为 Random, 即用户在每轮从集合 E 中随机选择一个摇臂作为隐私预算的取值. 由于 UCB 和 SM-UCB 策略考虑的场景不同, 计算收益的方式也不同, 所以我们分别将 UCB 和 SM-UCB 与 FixHalf 和 Random 进行对比. 在 FixHalf 和 Random 策略与 UCB 策略的对比实验中, FixHalf 和 Random 在参数设置与收益计算方式上都与 UCB 策略相同. 为了观察 3 种机制的差异, 我们随机选取了 3 个用户, 对于每个用户, 计算用户与服务器交互 20000 轮后的累计收益 $U_n(T)$. 图 5 分别展示了 MovieLens

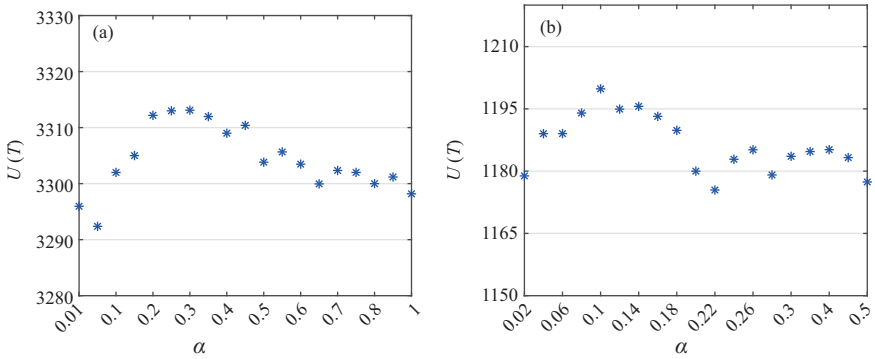

图 4 (网络版彩图) SM-UCB 策略中参数 α 对用户累计收益的影响

Figure 4 (Color online) The influence of the parameter α in the SM-UCB policy on the total user payoffs. (a) MovieLens; (b) Gollowa

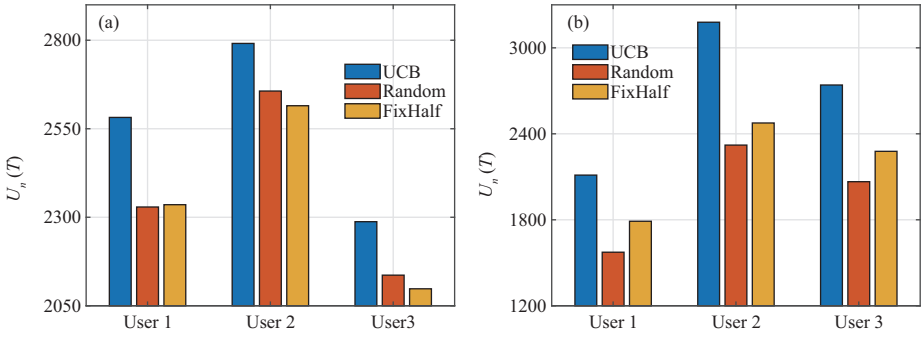

图 5 (网络版彩图) UCB, Random 和 FixHalf 策略的用户累计收益对比

Figure 5 (Color online) A comparison of UCB policy, Random policy, and FixHalf policy in terms of the total user payoffs. (a) MovieLens; (b) Gollowa

和 Gollowa 数据集下的实验结果, 结果显示 UCB 策略的累计收益明显高于 FixHalf 和 Random 策略, 证明了 UCB 策略的有效性. 类似的, 在 FixHalf 和 Random 策略与 SM-UCB 策略的对比实验中, FixHalf 和 Random 在参数设置、数据的敏感程度的设置与收益计算方式上与 SM-UCB 策略相同. 我们随机选择 3 个用户计算用户与服务器交互 20000 轮后的累计收益 $U_n(T)$. 如图 6 所示, 实验结果同样证明了 SM-UCB 策略的有效性.

7.2.3 UCB 策略与 SM-UCB 策略的比较

第 5 节中介绍的 SM-UCB 策略在选择隐私预算时考虑了用户数据的敏感程度, 为敏感程度高的用户选择较小的隐私预算, 从而更好地保护敏感数据的隐私. 为了验证 SM-UCB 策略是否可以帮助用户提高收益, 我们将传统 UCB 与 SM-UCB 策略进行对比. 实验中, 对于 UCB 与 SM-UCB 策略, 我们都假设用户数据的敏感程度不同, 在计算隐私损失时也都将数据敏感程度考虑进来. 不同点在于 UCB 策略选择摇臂时, 不考虑用户数据敏感程度. 为了观察两种机制的差异, 我们随机选择了 3 个用户, 对于每个用户, 我们计算用户与服务器交互 20000 轮后的累计收益 $U_n(T)$, 实验结果如图 7 所示. 根据实验结果可以看出, 对于 MovieLens 和 Gollowa 数据集, 使用 SM-UCB 策略的累计收益大于 UCB 策略. 但是 MovieLens 的差异不太明显, 分析原因是: 在 MovieLens 数据集上, 用户数据扰动程度的变化对于推荐准确性的影响较大, 所以虽然 SM-UCB 策略可以帮助用户更好地保护隐私, 但是由于数据扰

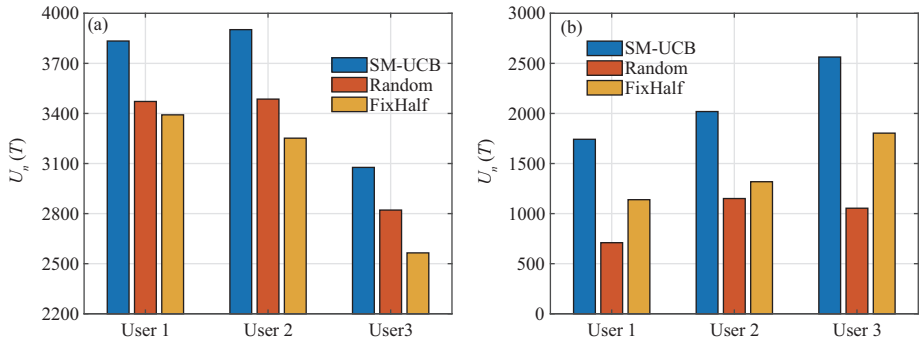

图 6　(网络版彩图) SM-UCB, Random 和 FixHalf 策略的用户累计收益对比

Figure 6　(Color online) A comparison of SM-UCB policy, Random policy, and FixHalf policy in terms of the total user payoffs. (a) MovieLens; (b) Gollowa

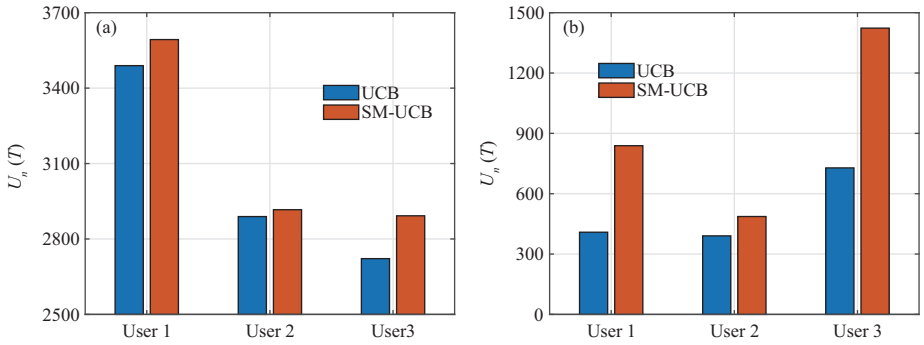

图 7　(网络版彩图) UCB 与 SM-UCB 策略的用户累计收益对比

Figure 7　(Color online) A comparison of UCB and SM-UCB policy in terms of the total user payoffs. (a) MovieLens; (b) Gollowa

动程度的增加导致用户推荐报酬的显著下降, 从而使收益的变化不明显.

8　结论

本文针对满足本地化差分隐私的推荐系统, 提出了帮助用户选择隐私预算的方法. 首先, 提出了一种满足本地化差分隐私的推荐模型, 模型中用户与服务器不断交互. 在每轮交互中, 用户选择隐私预算扰动数据, 服务器利用扰动后的数据为用户做出推荐. 在选择隐私预算时, 用户目标是以尽可能小的隐私损失获得尽可能高的推荐准确性. 我们将用户一轮交互中遭受的隐私损失和获得的推荐准确性量化为用户收益. 为了最大化累计收益, 用户根据以往收益动态调整隐私预算的取值. 为解决隐私预算选择所面临的 “开发” 与 “试探” 的权衡问题, 我们引入了多臂赌博机模型为用户隐私预算设置问题进行建模, 并引入了 UCB 策略帮助用户选择最优的隐私预算. 我们改进了基础的 UCB 策略以适应用户对数据敏感程度不同的情况. 通过在真实世界数据集上进行仿真实验, 证明了使用所提策略选择隐私预算可以有效提升用户的累计收益.

本文提出的方法假设用户本轮的收益 (遭受的隐私损失和获得的推荐准确性) 仅与用户在本轮的隐私预算选择有关. 然而, 随着用户与服务器不断交互, 服务器获取的用户数据越来越多. 众所周知, 在本地化差分隐私模型中, 服务器收集到的用户数据越多, 计算出的统计结果越准确, 同时用户数据

隐私泄露的风险也会加大. 因此, 下一步拟针对上述问题, 分析用户数据量对隐私损失和推荐准确性的影响, 研究可随时间变化的隐私预算选择方法.

参考文献

1 Zhang S, Yao L N, Sun A X, et al. Deep learning based recommender system: a survey and new perspectives. ACM Comput Surv, 2020, 52: 5

2 Li P F, Lu H, Zheng G, et al. Exploiting ratings, reviews and relationships for item recommendations in topic based social networks. In: Proceedings of World Wide Web Conference, 2019. 995–1005

3 Chen X S, Liu D, Xiong Z W, et al. Learning and fusing multiple user interest representations for micro-video and movie recommendations. IEEE Trans Multimed, 2021, 23: 484–496

4 Canny J F. Collaborative filtering with privacy via factor analysis. In: Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, 2002. 238–245

5 Erkin Z, Veugen T, Toft T, et al. Generating private recommendations efficiently using homomorphic encryption and data packing. IEEE Trans Inform Forensic Secur, 2012, 7: 1053–1066

6 Qi L Y, Zhang X Y, Dou W C, et al. A two-stage locality-sensitive hashing based approach for privacy-preserving mobile service recommendation in cross-platform edge environment. Future Gener Comput Syst, 2018, 88: 636–643

7 Qi L, Zhang X, Li S, et al. Spatial-temporal data-driven service recommendation with privacy-preservation. Inf Sci, 2020, 515: 91–102

8 McSherry F, Mironov I. Differentially private recommender systems: building privacy into the Netflix Prize contenders. In: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Paris, 2009. 627–636

9 Abowd J M. The U.S. census bureau adopts differential privacy. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, 2018. 19–23

10 Dwork C. Differential privacy. In: Proceedings of International Colloquium on Automata, Languages, and Programming, 2006. 1–12

11 Ye Q Q, Meng X F, Zhu M J, et al. Survey on local differential privacy. J Soft, 2018, 29: 159–183 [叶青青, 孟小峰, 朱敏杰, 等. 本地化差分隐私研究综述. 软件学报, 2018, 29: 159–183]

12 Wu Y J, Ge C, Zhang L Q, et al. An algorithm for differential privacy streaming data publication based on matrix mechanism under exponential decay mode. Sci Sin Inform, 2017, 47: 1493–1509 [吴英杰, 葛晨, 张立群, 等. 指数衰减模式下基于矩阵机制的差分隐私流数据发布算法. 中国科学: 信息科学, 2017, 47: 1493–1509]

13 Ren X B, Xu J Y, Yang X Y, et al. Bayesian network-based high-dimensional crowdsourced data publication with local differential privacy. Sci Sin Inform, 2019, 49: 1586–1605 [任雪斌, 徐静怡, 杨新宇, 等. 基于 Bayes 网络的高维感知数据本地隐私保护发布. 中国科学: 信息科学, 2019, 49: 1586–1605]

14 Meiser S, Mohammadi E. Tight on budget?: Tight bounds for r-fold approximate differential privacy. In: Proceedings of ACM SIGSAC Conference on Computer and Communications Security, Toronto, 2018. 247–264

15 Pujol D, Wu Y K, Fain B, et al. Budget sharing for multi-analyst differential privacy. 2021. ArXiv:2011.01192

16 Bubeck S. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. FNT Mach Learn, 2012, 5: 1–122

17 Xu L, Jiang C X, Qian Y, et al. Dynamic privacy pricing: a multi-armed bandit approach with time-variant rewards. IEEE Trans Inform Forensic Secur, 2017, 12: 271–285

18 Basu D, Dimitrakakis C, Tossou A C Y. Differential privacy for multi-armed bandits: what is it and what is its cost? 2019. ArXiv:1905.12298

19 Han Y X, Liang Z P, Wang Y, et al. Generalized linear bandits with local differential privacy. 2021. ArXiv:2106.03365

20 Jouini W, Ernst D, Moy C, et al. Upper confidence bound based decision making strategies and dynamic spectrum access. In: Proceedings of IEEE International Conference on Communications, 2010

21 Liang Y, Huang C L, Bao X G, et al. Sequential dynamic event recommendation in event-based social networks: an upper confidence bound approach. Inf Sci, 2021, 542: 1–23

22 Liu X C, Derakhshani M, Lambotharan S, et al. Risk-aware multi-armed bandits with refined upper confidence bounds. IEEE Signal Process Lett, 2021, 28: 269–273

23 Shen Y L, Jin H X. Epicrec: towards practical differentially private framework for personalized recommendation. In: Proceedings of ACM SIGSAC Conference on Computer and Communications Security, 2016. 180–191
24 Hua J Y, Xia C, Zhong S, Differentially private matrix factorization. In: Proceedings of the 24th International Joint Conference on Artificial Intelligence, 2015. 1763–1770
25 Jiang J Y, Li C T, Lin S D. Towards a more reliable privacy-preserving recommender system. Inf Sci, 2019, 482: 248–265
26 Shin H, Kim S, Shin J, et al. Privacy enhanced matrix factorization for recommendation with local differential privacy. IEEE Trans Knowl Data Eng, 2018, 30: 1770–1782
27 Asada M, Yoshikawa M, Cao Y. When and where do you want to hide? Recommendation of location privacy preferences with local differential privacy. In: Proceedigns of IFIP Annual Conference on Data and Applications Security and Privacy, 2019. 164–176
28 Chen C C, Zhou J, Wu B Z, et al. Practical privacy preserving POI recommendation. ACM Trans Intell Syst Technol, 2020, 11: 1–20
29 Kim J S, Kim J W, Chung Y D. Successive pointof-interest recommendation with local differential privacy. 2019. ArXiv:1908.09485
30 Lee J, Kifer D. Concentrated differentially private gradient descent with adaptive per-iteration privacy budget. In: Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, 2018. 1656–1665
31 Li H T, Ren X Y, Wang J, et al. Continuous location privacy protection mechanism based on differential privacy. J Commun, 2021, 42: 164–175 [李洪涛, 任晓宇, 王洁, 等. 基于差分隐私的连续位置隐私保护机制. 通讯学报, 2021, 42: 164–175]
32 Xia Y, Mao H R, Zhang X, et al. Differential privacy protection method for location recommendation. Comput Sci, 2017, 44: 38–41 [夏英, 毛鸿睿, 张旭, 等. 面向位置推荐的差分隐私保护方法. 计算机科学, 2017, 44: 38–41]
33 Jin B, Zhang Z Y, Zhao T. Location nearest neighbor query method for social netword based on differential privacy. J Comput Appl, 2020, 40: 2340–2344 [金波, 张志勇, 赵婷. 基于差分隐私的社交网络位置近邻查询方法. 计算机应用, 2020, 40: 2340–2344]
34 Ye A Y, Meng L Y, Zhao Z W, et al. Trajectory differential privacy protection mechanism based on prediction and sliding window. J Commun, 2020, 41: 123–133 [叶阿勇, 孟玲玉, 赵子文, 等. 基于预测和滑动窗口的轨迹差分隐私保护机制. 通信学报, 2020, 41: 123–133]
35 Yu K, Jia L, Chen Y Q, et al. Deep learning: yesterday, today, and tomorrow. J Comput Res Dev, 2013, 50: 1799–1804 [余凯, 贾磊, 陈雨强, 等. 深度学习的昨天、今天和明天. 计算机研究与发展, 2013, 50: 1799–1804]
36 Sutton R S, Barto A G. Reinforcement Learning: An Introduction. Cambridge: MIT Press, 1998
37 March J G. Exploration and exploitation in organizational learning. Organiz Sci, 1991, 2: 71–87
38 Wang H Z, Zhao Q, Wu Q Y, et al. Global and local differential privacy for collaborative bandits. In: Proceedings of the 14th ACM Conference on Recommender Systems, 2020. 150–159
39 Andrés M E, Emilio B, Konstantinos C, et al. Geo-indistinguishability: differential privacy for location-based systems. In: Proceedings of ACM SIGSAC Conference on Computer & Communications Security, 2013. 901–914

Optimized setting of privacy budget in a recommendation system with local differential privacy

Ting BAO[1], Lei XU[2*], Liehuang ZHU[2] & Lihong WANG[3]

1. *School of Computer Science and Technology, Beijing Institute of Technology, Beijing* 100081*, China*;
2. *School of Cyberspace Science and Technology, Beijing Institute of Technology, Beijing* 100081*, China*;
3. *National Computer Network Emergency Response Technical Team/Coordination Center of China, Beijing* 100029*, China*
* Corresponding author. E-mail: 6120180029@bit.edu.cn

Abstract Recommendation system can help users find the data they need from the massive amounts of data. At the same time, uploading original user data to the server may reveal user privacy. We utilize local differential privacy techniques to provide privacy protection for users in the recommendation system. In the local differential privacy model, the degree of privacy protection is measured by the privacy budget, and a high privacy budget usually means high analysis accuracy. To help users minimize privacy loss and maximize recommendation accuracy, we model the privacy budget setting problem as a multiarmed bandit problem and propose the upper confidence bound learning policy to help each user choose the privacy budget. Considering that users have different sensitivity levels to different data, we modify the above policy. Experimental results reveal that the proposed policy can help users choose an appropriate privacy budget, which can effectively increase the total user payoff.

Keywords recommendation system, local differential privacy, privacy budget, reinforcement learning, multi-armed bandit