Shilling Attack Detection Combined with Time Dimensions

融合时间维度的托攻击检测研究

熊庆宇　田仁丽　袁　泉　黄海魂　著

科学出版社

融合时间维度的托攻击检测研究

Shilling Attack Detection Combined with Time Dimensions

熊庆宇　田仁丽　袁　泉　黄海魂　著

科学出版社
北　京

内 容 简 介

随着网络信息资源的爆炸式增长，个性化推荐技术作为有效缓解信息过载的个性化推荐技术得到了广泛应用，然而推荐系统开放性的特点使其易受托攻击的影响。托攻击者通过注入虚假信息操纵推荐结果，影响推荐系统的公正性。托攻击研究中较少考虑攻击时间方面的特性，针对此问题，本书完成了三方面工作：一是提出推荐系统中融合时间维度的托攻击形式，并分析融合时间和评分两个维度的托攻击模型对推荐系统的影响；二是从项目评分序列的角度，提出基于项目类型及时间序列动态划分的异常项目检测方法和基于狄利克雷假设检验的异常项目检测方法；三是结合项目评分时间序列进行异常用户检测，提出结合异常项目的检测方法、基于单类分类器的检测方法和半监督的检测方法，以应对不同的推荐系统的应用场景。

本书适合作为计算机科学与技术和软件工程相关专业研究生、本科生及业界人员的参考书。

图书在版编目(CIP)数据

融合时间维度的托攻击检测研究 /熊庆宇等著. —北京：科学出版社，2018.11

ISBN 978-7-03-059116-6

Ⅰ. ①融… Ⅱ. ①熊… Ⅲ. ①软件工程-研究 Ⅳ. ①TP311

中国版本图书馆 CIP 数据核字(2018)第 240127 号

责任编辑：阚 瑞 / 责任校对：郭瑞芝

责任印制：师艳茹 / 封面设计：迷底书装

科学出版社出版

北京东黄城根北街 16 号

邮政编码：100717

http://www.sciencep.com

文林印务有限公司印刷

科学出版社发行 各地新华书店经销

*

2018 年 11 月第 一 版 开本：720×1000 1/16

2018 年 11 月第一次印刷 印张：7 3/4

字数：152 000

定价：58.00 元

(如有印装质量问题，我社负责调换)

前　言

随着信息资源的爆炸式增长，个性化推荐技术作为有效缓解信息过载的工具得到了广泛的应用。但是，由于推荐系统采用用户数据协同产生推荐结果，托攻击者可以向推荐系统中注入虚假用户评分改变目标项目的推荐排名，以此达到攻击的目的。虚假评分的注入将严重影响推荐系统的有效性，因此托攻击检测的研究意义重大。

目前托攻击检测主要通过计算用户概貌属性特征区分真实用户与虚假用户，该类方法对具体攻击模型有良好的检测效果，但难以检测混合及新的攻击模型。这些方法主要从用户评分角度出发，未考虑评分时间方面的特征，也缺少对系统中异常项目检测问题的考虑。

本书深入分析推荐系统和托攻击的研究背景、意义及现状，在介绍主流推荐算法、托攻击模型及托攻击检测相关方法的基础上，完成如下研究工作。

(1)针对目前托攻击模型中没有考虑注入时间这一重要攻击特征的问题，同时考虑时间和评分两个方面，提出融合时间维度的托攻击模型，并分析模型对推荐系统的影响。

(2)进行融合时间维度的异常项目检测研究，提出基于项目类型及时间序列动态划分的异常项目检测方法和基于狄利克雷假设检验的异常项目检测方法。

(3)研究融合时间维度的异常用户检测，首先结合异常项目检测，提出一种平均评分偏移-时间序列的检测方法；然后针对系统中虚假用户标签难以获取的问题，提出基于单类分类器的检测方法；最后提出半监督的检测方法以充分利用系统中大量无标签和少量有标签数据。

由于本书作者的学识水平有限，书中难免存在不足之处，恳请读者批评指正。

目　录

第1章 绪 论

1.1 推荐系统概述

随着互联网的发展，用户对信息的需求日益强烈。用户对信息的满意度以及信息的多样性已成为互联网产业迅速扩张的方向。互联网企业都在极力搭建各种电商网站，社交网络应用、分类网站和线上到线下(online to offline，O2O)交易服务等信息服务平台，这使得信息的交互与流通能够覆盖更多的领域。例如，国内自营式电商企业京东在线销售13大类3150万种商品；线上到线下交易服务网站大众点评网月活跃用户数超过7000万，收录的商户数量超过400万家，点评数量超过2600万条。大量信息扑面而来，信息资源呈现爆炸性增长，人们逐渐从信息匮乏的时代步入信息过载的时代[1]。在海量的资源中找到感兴趣的部分如同大海捞针，其难度和成本都增加了，这就是信息过载(information overload)。信息过载对信息生产者和信息消费者都是巨大的挑战：信息消费者从大量信息中找到自己感兴趣的信息非常困难，信息生产者如何使自己的信息展现给更多用户也同样非常困难[2]。

目前已有很多解决信息过载问题的方案，其中最具代表性的包括分类目录、搜索引擎和推荐系统。

(1)分类目录通过对著名网站进行分类方便用户根据类别快速查找网站。雅虎、hao123等都是知名的分类目录网站。随着互联网的不断发展，各类网站大量增多，分类目录网站也只能覆盖少量热门网站，因此难以满足用户的需求。

(2)随着互联网信息的迅速增长，分类目录帮助用户查找信息的局限性越来越明显，因此产生了搜索引擎。搜索引擎是一个对互联网的信息进行搜集整理并供用户输入关键词查询的系统[3]。搜索引擎根据用户查询关键词快速返回相应的结果[4]。虽然搜索引擎能根据主题关键字过滤信息，但它仍然存在几个不足之处。首先，搜索引擎不会响应用户的历史记录，无法主动分析用户的信息需求并为用户提供个性化的信息资源以实现决策支持。其次，许多用户的需求包含结构上的知识，无法直接使用关键字匹配进行描述。最后，搜索引擎并不主动提供服务，

而是需要用户主动请求，在与其交互之中，用户只是服务的调用者。在这样的交互模式下，用户必须通过主动对信息服务提出请求，才能提取出符合自己兴趣的信息。而追求更高体验的用户要求有趣的信息和新颖的事物能够主动找上门，而不是自己长时间地在网上搜索。创建这种信息自动推送服务，需要系统根据用户的长期历史兴趣分析其需求和兴趣并对信息进行个性化定制，根据用户当前的情景适时地提供给目标用户。但传统信息服务提供方在服务的过程中，并未对用户的历史记录进行搜集与挖掘，形成用户感兴趣的描述模型，也就无法主动地为用户推送个性化的定制信息，难以提高用户对信息服务的满意度。

搜索引擎由于自身的局限性不能很好地应对上述问题，而作为另一类信息过滤技术的推荐系统(recommender system)通过分析用户历史数据猜测用户的兴趣，并主动推送给用户信息，进一步提升了用户体验。如今推荐系统已经成为新一代 Web 应用中不可缺少的个性化信息服务模式。协同过滤作为推荐系统的一种早期的基本思想，已经被广泛地研究和应用。

(3)和搜索引擎相似，推荐系统也是帮助用户快速发现有用信息的工具。但推荐系统不需要用户提供明确的需求，其通过分析用户历史行为信息给用户兴趣建模，从而主动给用户推荐能够满足其需求的信息。1992 年，Glodberg 等首次提出协同过滤(collaborative filtering)的概念，而这一概念的提出推动了信息过滤技术的急速发展与进步。尤其是近年来，电子商务的崛起使得对推荐系统的需求更加强劲，也为推荐系统的发展带来了新的动力。精准的项目推荐能获得用户关注，产生用户黏性，提高用户的忠诚度。推荐系统提供的个性化信息服务除了电子商务领域之外，在电影和视频、音乐、社交网络、阅读、个性化邮件、广告和基于位置的服务等领域也得到了广泛的应用。自 2006 年以来，亚马逊有 1/3 的销售来自推荐系统[5]。2012 年团购网 GROUPON 收购了个性化服务公司 Adku，以增强其捕捉用户个性化购物目标的能力。Foursquare 2013 年推出了个性化推荐引擎，帮助用户快捷地找到喜欢的商家。在中国，豆瓣、百分点、淘宝网、当当网、腾讯等知名企业和网站近年来也开始采用个性化推荐技术来提升用户体验。豆瓣通过用户的收藏和评价向用户推荐一系列书籍或音乐，为每个用户提供不一样的个性化界面，2012 年豆瓣月度覆盖用户超过 1 亿。2012 年百分点推荐引擎在 IWOWCase 上线，它是一款提升电子商务零售网站整体营销性能的个性化推荐工具，其用户数已达 1.4 亿，超过 200 亿个消费偏好标签，合作伙伴近 300 家，涵盖各个行业。近两年来，在新兴的移动应用中，社交网络(如微信、Google Plus 等)的发展尤为迅速。社交网络为用户提供了一个内容发布、分享的平台。在此类应用中，用户会制造大量的信息，如文本内容、朋友关系网和社交活动等。这些信息中，有价值的信息被无趣的信息吞没。如何利用推荐系统来为用户找到有趣

的信息也是目前非常热门的一个研究议题。总之，推荐系统不仅在工业界得到了无比重视，在学术界也掀起了一股研究的热潮。

1.2 推荐系统的托攻击问题

目前，虽然个性化推荐技术在如冷启动问题、宏观与微观的悖论和推荐系统与用户行为的相互影响以及预测精度和基于情境的推荐等方面取得了一定的进展[6-8]，但仍有一些问题没有得到有效的解决，其中托攻击问题对电子商务的危害最为严重[9,10]。研究表明，只需1%的虚假用户即可将项目推荐置顶[11]。

推荐系统的数据源最主要是由用户概貌组成的评分矩阵。评分矩阵通常很稀疏，推荐系统需要尽可能多的评分项来产生更准确的结果。所以网站对评分的收集是很开放的。用户可以随意地创建和修改用户概貌，这给了攻击者可乘之机。托攻击利用推荐系统的开放性注入大量有偏见的用户概貌以干扰系统的推算过程，这使得系统对被攻击项的预测有极大的偏移，进而影响对用户推送的信息。在商业领域，一些厂商会设法利用无保护的推荐系统向用户推荐自己的产品，以此打压竞争对手并获取不正当利益。

系统被恶意注入虚假评分后将严重影响项目推荐排名，用户无法找到自己喜欢的商品，致使用户满意度下降，进而导致系统用户流失、商家利润降低等后果。目前，虽然有一些关于托攻击模型及检测的研究，但托攻击模型的研究主要从项目选择策略及相应的评分策略着手，对于托攻击的注入时间很少提及，而托攻击的攻击时间周期将直接影响攻击的效果及躲避系统检测的可能性：攻击在短时间内注入，攻击成本低，能在短时间内达到预期的攻击效果，但被系统检测的概率增大；攻击在长时间内注入，虽然被系统检测的概率降低，但攻击成本高且需要较长时间才能达到攻击效果，同时随着时间的推移，系统内的真实评分增加需要注入更多的虚假概貌来达到预期的攻击效果。因此，攻击注入时间是托攻击模型需要考虑的重要因素之一。本书从时间维度着手，提出新的攻击模型及检测算法。

1.3 托攻击检测的国内外研究现状

托攻击就是攻击者通过注入虚假用户概貌信息，试图改变系统推荐结果的攻击，其中用户概貌即用户对所有项目的评分集合。托攻击对推荐系统具有严重的威胁，虚

假评分一旦注入系统，将严重影响推荐排名，进而导致用户体验变差。因此，关于托攻击及检测算法的研究已经成为研究者关注的重点。KDD、SIGIR、ICDM、AAAI、WWW、RecSys 等发表了相关的研究成果。

托攻击的相关研究分为攻击模型和托攻击检测算法的研究。Lam 等[12]最先提出了两种基本的攻击模型，随机攻击(random attack)和均值攻击(average attack)。Burke 等[13]又相继提出了流行攻击(bandwagon attack)和段攻击(segment attack)。后续又有研究者提出好恶攻击(love/hate attack)、逆流行攻击(reverse bandwagon attack)、探测攻击(probe attack)等攻击模型[14]。

托攻击检测可以分为虚假用户和基于时间序列的目标项目检测。最早的托攻击检测算法是由 Chirita 等[15]提出的基于用户概貌属性特征的虚假用户检测方法，此方法主要采用平均评分偏移(rating deviation from mean agreement，RDMA)和最近邻相似度(degree of similarity with top neighbors，DegSim)区分虚假用户与真实用户。Mobasher 等[16]提出填充目标平均差异(filler mean target difference，FMTD)属性可以解决小规模段攻击检测问题。针对虚假用户检测的研究有很多，大致可以分为有监督、无监督和半监督三类检测算法。基于监督的检测算法使用 k 最近邻(k-nearest neighbor，kNN)、C4.5 和支持向量机(support vector machine，SVM)等分类器对用户概貌特征(如 RDMA、DegSim、FMTD)进行分类以检测虚假用户。已经定义的所有用户概貌特征都可以用于半监督训练无标签数据并建立分类模型检测虚假用户。Wu 等[17]提出了一种启发式的方法“MC-Relief”(multi class-relief)。利用有标记的数据集训练一个分类器并用于分类无标记的概貌。无监督的托攻击检测通常使用聚类方法对用户进行聚类从而检测虚假用户，Zhang 等[18,19]提出的大部件(large component，LC)搜索和谱聚类(spectral clustering，SC)算法是一种基于相似度矩阵的无监督算法，这两种方法的基本假设是虚假用户之间的相似度高于正常用户，通过聚类方法可以有效地找出虚假用户群体。尽管基于用户概貌特征的虚假用户检测的研究很多，但仍然存在如下一些问题：随着用户量的增大，计算量剧增；某些算法只对特定的攻击模型有效，算法普适性较差；半监督算法需要一个有标签的数据集来训练分类模型，如果训练集中虚假用户和真实用户太相似就不能有效地检测托攻击。

托攻击检测中基于时间序列的目标项目检测方法最初由 Zhang 等[20]提出，其通过启发式的方法设定时间窗口值，并通过计算每个区间的样本熵和样本均值查找异常区间，包含异常区间的项目为异常项目。Gao 等[21]提出项目类型各不相同，对于不同的项目类型应该设置不同的时间窗口值，从而提高检测结果。袁泉[22]提出基于重要点的时间序列动态划分方法，避免因时间窗口设置不当而影响检测结果。

1.4 本章小结

本章首先介绍了推荐系统的起源和意义，然后分析了推荐系统的特点和恶意用户注入虚假概貌形成的托攻击问题，最后总结了国内外托攻击问题的研究现状。

第 2 章　推荐算法与托攻击检测关键技术

推荐系统作为解决信息过载的有效工具在学术界和工业界都得到了广泛的关注，不断有学者提出新的推荐算法。但一些恶意用户通过向系统中注入虚假评分来获取利润。为了提高推荐系统的有效性，研究者对虚假评分的注入方式进行了研究，并提出了不同的攻击模型，针对不同攻击模型的托攻击检测算法也不断被提出。因此，本章首先介绍不同类型的推荐算法；其次简单介绍不同攻击模型的攻击目的、选择项目和填充项目的选择策略及相应的评分策略等；最后对托攻击的检测算法及本书实验所用数据集进行简单的介绍。

2.1　推 荐 算 法

推荐系统作为一种帮助用户快速发现有用信息的工具，并不需要用户提供明确的需求，而是通过分析用户的历史行为数据给用户的兴趣建模，从而主动给用户推荐能够满足他们兴趣和需求的信息。此外，由于推荐系统通过发掘用户的行为找到用户的个性化需求，所以能够帮助用户发现他们感兴趣但很难发现的商品，也可以更好地发掘长尾商品。Burke[23]将推荐算法划分为六类：协同过滤的推荐算法、基于内容的推荐算法、基于人口统计学的推荐算法、基于知识的推荐算法、基于社区的推荐算法和混合推荐算法。其中，协同过滤的推荐算法是应用最多的算法。

“协同过滤”一词最早是在开发推荐系统 Tapestrym 时提出的，后被广泛地研究与应用。协同过滤的原意是指人们在浏览文档时能够通过分享彼此的评价来对完全陌生的文本进行估计。这种现象有许多猜测，其中一个粗略的假设是：若用户的兴趣爱好相似，他们对同一内容的评价很可能也是相似的。所以人们对内容的评价是他们进行推荐的关键数据，即评分矩阵。对一个给定用户集 $\{u_1,\cdots,u_m\}$ 以及一个项目集 $\{i_1,\cdots,i_n\}$，矩阵中的任意元素 r_{xy} 代表了用户 u_x 对项目 i_y 的评价。因此，矩阵的第 x 行就表示用户 u_x 在所有项目上的标注信息，而第 y 列表示所有的用户对项目 i_y 的标注信息。使用评分矩阵作为数据源的推荐系统在研究与实践中被关注最多的主要是以下两大类。

1. 基于邻域的过滤

早期的协同过滤算法通过一个用户的所有评分(或对一个项目的所有评分)来确定在其所属集中的相似性,即用户与用户之间的相似性矩阵(或项目之间的相似性矩阵),然后根据相似性预测用户对目标项的评分或者进行 top$N$ 推荐。这类方法的基本流程结构被称作基于邻域的框架,实现这类方法需要根据数据集的情况选择适应的距离度量函数,然后依照策略进行预测或推荐。这种框架的实现方法又可分为基于用户(user-based)和基于项目(item-based)两种。基于用户的实现认为具有相似爱好的用户,其评分更可能相似。而基于项目的实现方法认为用户会喜欢他之前评分较高的东西,所以会推荐给用户与高评分项目类似的东西。许多商业推荐系统都实现自基于邻域的框架。基于邻域的框架的优点在于其算法简单,易于实现,且能在一定程度上达到预想的推荐效果。基本的基于用户邻域的评分预测中,相似性的度量函数可使用余弦相似度(cosine similarity)、皮尔逊相关系数(Pearson correlation coefficient)等,可根据数据集自身特性进行调整。邻居的选择策略可选择最相似的几个或者进行相似度加权等。在实际应用中,基于记录的框架也有着自身的缺陷,其中讨论最多的就是大矩阵的数据稀疏性问题。与小规模实验研究不同,真实的评分矩阵的用户和商品的数量是非常巨大的,而用户评价过的商品只有很少一部分。这就使得矩阵特别稀疏,在邻域内无法进行推荐。即使使用扩大邻域等方法进行强行推荐,也得不到理想的结果。

2. 基于模型的推荐算法

在当前基于模型的推荐算法中,一个值得关注的领域是隐语义模型(latent factor model)。这个简单的模型直接作用在观测数据上,而为了平衡过拟合问题,加上了正则化因子。在这个模型中,评分矩阵中的每一个元素都是由用户偏好与项目的内容特征所决定的。其公式为

$$\hat{r}_{ui} = \sum_{k} p_{uk} q_{ki} = q_i^{\mathrm{T}} p_u \tag{2.1}$$

其中,k 是隐含变量的维度;p_u 是用户的偏好;q_i 是项目 i 的隐含内容。所以评分矩阵就可以分解为用户偏好矩阵与项目隐含内容特征矩阵的乘积:$\hat{R} = PQ$。这样,推荐算法就是要解决一个矩阵分解的优化问题。其目标函数为

$$\min_{q^*, p^*} \sum_{(u,i) \in K} (r_{ui} - q_i^{\mathrm{T}} p_u)^2 + \lambda(\|p_u\|^2 + \|q_i\|^2) \tag{2.2}$$

其中，K 是 (u,i) 对的集合，代表了未知的 r_{ui}；λ 是正则化参数。这个系统通过拟合已存在的观测评分来学习参数。而它的目标是通过参数学习到的知识推广到未知的数据。这个系统也可以通过调节正则化参数 λ 来平衡过拟合的问题。这个简单的模型在 2006 年的 Netflix 大奖赛中获得了极好的名次，从而引起了众人的关注。许多优秀的模型都是在此基础上扩展而成的。下面简单介绍几种经典的解法。

1) 随机梯度下降

在 2006 年 Netflix 大赛中获得第三名的 Simon Funk 推广将随机梯度下降(stochastic gradient descent)算法应用到隐语义模型的求解中。在这个算法的训练中，系统随机选择一个评分 r_{ij} 并计算其预测误差：

$$e_{ui} = r_{ui} - q_i^{\mathrm{T}} p_u \tag{2.3}$$

然后系统地使用速度为 γ 的学习效率对用户偏好 p_u 与内容特征进行修正 q_i。

$$q_i \leftarrow q_i + \gamma \cdot (e_{ui} \cdot p_u - \lambda \cdot q_i) \tag{2.4}$$

$$p_u \leftarrow p_u + \gamma \cdot (e_{ui} \cdot q_i - \lambda \cdot p_u) \tag{2.5}$$

这个算法非常容易实现，运算速度非常快，可并行性也很强。

2) 交替最小二乘

若用户偏好 P 与项目的内容 Q 都是未知的，则式(2.2)是非凸的。然而若把其中一个看作固定的，这个优化问题便是一个二次规划问题，具有全局最优解，用随机梯度下降算法就能轻易解决。所以交替最小二乘(alternating least square)借用 EM 算法的思想，首先初始化其中一个矩阵，然后使用最小二乘更新另一个矩阵。这样交替地进行，每一步都会收敛到一个当前状态的全局最优点，所以算法最后也会收敛到一个局部最优点。初始化可使用一些先验的知识或者进行随机填充。

3) 模型的扩展

矩阵分解法实现协同过滤的好处在于其扩展数据模型的能力较强，能满足各种应用的需求。式(2.1)尝试利用用户与对应项目的相互作用来产生不同的评分。然而许多观测值的变化是由用户或者项目自己产生的，这可以被认为是偏见，与相互作用无关。例如，在典型的协同过滤数据中隐含着大规模系统性的趋势，有些用户通常给出比其他用户更高的评分或者有些产品被公认为比其他产品要好。所以简单地把评分解释为 $q_i^{\mathrm{T}} p_u$ 这种相互作用的形式是不够理想的。一个更好的模式是尝试识别哪一部分是由用户或项目的偏见产生的，哪一部分是由用户与项目的相互作用产生的。因此，r_{ij} 包含偏见的一阶逼近可描述为

$$b_{ui} = \mu + b_i + b_u \tag{2.6}$$

r_{ui} 的偏见 b_{ui} 包含了用户与项目的影响，μ 为全局的平均评分，参数 b_i 和 b_u 分别表示项目与用户的影响。例如，用户 A 对电影《泰坦尼克号》的一阶估计包含以下成分：第一部分为电影评分均值 μ，其值为 3.7；第二部分为电影《泰坦尼克号》高出电影平均评分 b_i，其值为 0.5；第三部分为 A 比其他用户更具批判性，平均评分比一般人低 b_u，其值为 0.3。所以 A 对电影的评分估计为 3.9(3.7+0.5−0.3)。若包含偏见则式(2.1)写为

$$\hat{r}_{ui} = \mu + b_i + b_u + q_i^{\mathrm{T}} p_u \tag{2.7}$$

观测到的评分在这里被分解为四部分：全局平均、项目偏见、用户偏见与用户-项目交互。这让信号变化的每一部分在其中都找到了相关的部分。这个系统通过优化以下损失函数进行学习。

$$\min_{q^*,p^*,b^*} \sum_{(u,i),\in K} (r_{ui} - \mu - b_i - b_u - q_i^{\mathrm{T}} p_u)^2 + \lambda(\| p_u \|^2 + \| q_i \|^2 + b_i^2 + b_u^2) \tag{2.8}$$

因为偏见会捕获更多的观测信号，所以精确的建模是至关重要的，包含更多的细节可以取得更好的效果。

推荐系统常需要处理冷启动的问题：许多用户只提供非常有限的评分，以至于对他们做出一般的推测十分困难。解决这类问题的一个办法是使用这些用户的附加信息。推荐系统能通过隐式的反馈来窥测用户的偏好，这种搜集行为的方式不受用户意愿的左右，还能提供准确的评分。例如，零售商能使用客户的购买或者浏览历史来学习他们的趋势。

以一个布尔隐式反馈为例，用户 u 对一些项目 $N(u)$ 具有隐含的偏好。系统通过用户对这些项目隐含的偏好来进行描述。这里需要引入一些新的因素，项目 i 与 $x_i \in \mathbb{R}^f$ 相关，因此，一个用户对 $N(u)$ 中的偏好描述为 $\sum_{i \in N(u)} x_i$，通过归一化之后变为 $|N(u)|^{-0.5} \sum_{i \in N(u)} x_i$。

另外一种信息来自用户属性，如用户资料。简单地考虑 $A(u)$ 的一些属性，即与用户 u 相关的属性集合，可能包括性别、年龄组、邮政编码、收入级别等。这个属性用一个独立的因素向量 $y_a \in \mathbb{R}^f$ 描述。整合了这些信号源的增强矩阵分解模型可表达为

$$\hat{r}_{ui} = \mu + b_i + b_u + q_i^{\mathrm{T}} \left[p_u + |N(u)|^{-0.5} \sum_{i \in N(u)} x_i + \sum_{a \in A(u)} y_a \right] \tag{2.9}$$

以上所述的模型都是静态的。实际上新项目的产生使得产品的接受度与流行度都不断发生变化。相似地，用户的倾向也在不断发展。系统应该把时间效应考虑为一个动态的过程，与偏差、用户-项目相互作用同时发生作用。

实际上，矩阵分解模型也非常适合融合时间效应的影响。把评分分解成独立的成分使得系统能够允许每个部分拥有不同的时态。以下的成分会随着时间的变化发生变化：项目偏见，$b_i(t)$；用户偏见，$b_u(t)$；用户偏好，$p_u(t)$。

时间效应的第一部分解释了项目的流行度会随着时间的变化而改变的现象。例如，电影受欢迎程度的变化会受到一些事件的影响，例如，这部电影的演员突然红了。所以，这些模型把项目的偏见看作一个时间的函数。时间效应的第二部分考虑到用户的评分基线也会随着时间的变化发生改变。用户的偏好也可能随着时间的变化而改变，例如，一个喜欢惊悚小说的人，一年之后可能受到一部热播剧的影响变得喜欢侦探剧。这个模型同时考虑了具有时间效应的用户偏好 $p_u(t)$ 与相对固定的项目特征 q_i 的影响。综合所有的时间因素的影响，式(2.7)对 r_{ij} 的动态预测为

$$\hat{r}_{ui} = \mu + b_i(t) + b_u(t) + q_i^{\mathrm{T}} p_u(t) \tag{2.10}$$

2.2 托攻击模型

攻击规模按攻击意图可以分为推攻击和贬攻击。一般而言，推攻击是为了增加目标项目被推荐的概率，贬攻击则相反。

图 2.1 所示为攻击模型的基本框架[16]。攻击者在实施攻击时，通常会向推荐系统中注入一定数目的攻击用户概貌，从而达到攻击的目的。一组攻击的强度一般由攻击规模(attack size)和填充规模(filler size)来度量。攻击规模一般使用评分系统中攻击用户概貌数目占系统中所有用户概貌数目的百分比来表示。填充规模表示在一条攻击用户概貌中，评分项目的数目与推荐系统中所有项目总数的比值，它主要描述评分项目的稀疏程度。

图 2.1 中的 I_S 表示攻击者根据特定需要而选择的一些项目，I_F 表示攻击者用来伪装自己而选择的填充项目，I_Φ 表示攻击者没有评分的项目，i_t 表示攻击者选择攻击的目标项目。相应地，第二行是相对于第一行项目的评分。

构建攻击模型的方法可以用如下的四元组形式来表示：

$$M = (\chi, \delta, \sigma, \gamma) \tag{2.11}$$