DOI:10.13306/j.1672-3813.2009.03.006
第6卷第3期 复杂系统与复杂性科学 Vol.6 No.3
2009年9月 COMPLEX SYSTEMS AND COMPLEXITY SCIENCE Sep. 2009

文章编号:1672-3813(2009)03-0001-10

个性化推荐系统评价方法综述

刘建国[1,2,4],周 涛[1,2,4],郭 强[2,3],汪秉宏[1,4]

(1.中国科学技术大学近代物理系理论物理研究所,合肥 230026;2.弗里堡大学物理系,瑞士 弗里堡 CH-1700;
3.大连民族学院理学院,辽宁 大连 116600;4.上海理工大学复杂系统科学研究中心,上海 200093)

摘要:根据推荐系统任务的不同,介绍了不同的准确性度量指标以及各自的优缺点;介绍了准确度之外的其它指标,例如推荐多样性、覆盖率等;指出了目前评价指标存在的缺陷,以及未来可能的改进方向。

关键词:个性化推荐系统;准确率指标;推荐多样性;覆盖率

中图分类号:TP274.2;N94 文献标识码:A

Overview of the Evaluated Algorithms for the Personal Recommendation Systems

LIU Jian-guo[1,2,4], ZHOU Tao[1,2,4], GUO Qiang[2,3], WANG Bing-hong[4]

(1. Department of Modern Physics, University of Science and Technology of China, Hefei 230026, China;
2. Department of Physics, University of Fribourg, Fribourg CH-1700, Switzerland;
3. School of Science, Dalian Nationalities University, Dalian 116600, China;
4. Research Center of Complex Systems, University of Shanghai for Science and Technology, Shanghai 200093, China)

Abstract: According to the system tasks, we introduce the different accuracy metrics as well as their respective advantages and disadvantages. Furthermore, other metrics are also introduced, such as the diversity, coverage and so on. Finally we point out the defects of the current evaluation, as well as the possible future directions to improve.

Key words: personalized recommendation systems; accuracy metric; recommendation diversity; coverage

1 引言

个性化推荐系统通过建立用户与产品之间的二元关系,利用用户已有的选择过程或相似性关系挖掘每个用户潜在感兴趣的对象,进而进行个性化推荐,其本质就是信息过滤[1-3]。一个完整的推荐系统由3部分组成:收集用户信息的行为记录模块、分析用户喜好的模型分析模块和推荐算法模块,其中,推荐算法模块是推荐系统中最为核心的部分。目前的推荐算法主要包括协同过滤推荐算法[4-7]、基于内容的推荐算法[8-10]、基于用户-产品二部图关系的推荐算法[11-14]及混合推荐算法[15-16]。每年都会涌现大量新的推荐算法,其作者都声称自己的算法在某些方面是最好的。然而,针对特定目的,清晰地鉴别算法的优劣是一个

收稿日期:2008-12-05

基金项目:国家重点基础研究发展计划(973计划)(2006CB705500);国家自然科学基金(10635040,60744003,10532060,10472116);辽宁省教育厅基金(20060140)

作者简介:刘建国(1979-),男,山西临汾人,博士后,主要研究方向为个性化推荐算法、复杂网络。

十分具有挑战性的课题。在不同应用背景下,如何选取恰当的评价准则对算法的表现进行评估,研究者们目前还没有达成共识。

评价推荐算法本质上是十分困难的。1)不同的算法在不同数据集上的表现不同。例如,基于用户的协同过滤算法在用户数量远远大于产品数量的系统上表现不错;反之,这样的算法就不是很实用。相关的影响因素还包括打分稀疏性、打分尺度,以及数据集的其他特性;2)评价的目的也不尽相同,许多研究工作注重于评价算法预测打分的准确度。然而,更早时候,Shardanand U[17]就已经认识到度量系统的错误推荐更有意义。还有一些工作研究了准确度之外影响用户满意度的指标[18-19]。这些指标包括:覆盖率[19]、产品流行性、推荐列表多样性[12]等;3)对不同的数据是否需要在线用户的测试?当打分数据非常稀疏的时候,对电子商务系统的推荐算法进行离线评价就非常困难。这是因为离线测试的时候,通常的做法是把用户选择过的产品随机划分为训练集和测试集,利用训练集的数据预测用户的喜好,进而利用测试集对预测结果进行评估。因此,离线测试只能评价用户"已经打分"的产品。对于用户没有选择或并不知道的产品,推荐算法就无法对这些产品进行评估。如果让在线用户评价,用户可以给每个被推荐的产品打分,从而得到推荐算法的准确度;4)选择哪些指标进行综合评价也十分困难。这 4方面的因素直接决定了评价的客观性和合理性。

本文简要回顾了推荐系统已有的推荐指标。首先,介绍了 6种评价推荐算法准确度的指标,指出了不同指标的应用环境以及各自的优缺点。其次,介绍了准确度之外的一些评价指标,包括推荐产品的流行性、覆盖率、系统发现新鲜产品的能力以及用户的满意度等。这些指标有助于从准确性之外全面对推荐系统的表现进行客观评估。最后指出了目前的评价指标共同面临的问题,以及可能的发展方向。

2 准确度评价指标

绝大多数的推荐系统都利用准确度评价推荐算法的好坏。假设用户可以考察所有产品的信息,并且可以根据自己对产品的偏好程度对产品进行排序,那么准确度可以定义为推荐算法的预测排名与用户的实际排名的贴近度。由于不同系统的任务是不一样的,而且评价指标缺乏标准化,因此很难对不同系统的推荐算法进行比较。针对不同的系统,已有的准确度指标有:预测准确度、分类准确度、排序准确度、预测打分关联、距离标准化指标和半衰期效用指标。

2.1 预测准确度

预测准确度考虑推荐算法的预测打分与用户实际打分的相似程度。在预测分值显示给用户的系统中,预测准确度十分重要。例如,在 MovieLens系统[20]中,系统给出其它用户评价电影的"星"的平均数,而且给出对某个用户的预测"星"数。预测准确度能够度量系统中预测"星"数与用户实际给出的"星"数的差别。即使一个推荐系统能够为用户提供正常的电影排序,但是由于预测打分不准也可能导致系统不可信。预测准确度的一个经典度量方法是度量系统的预测打分与用户的实际打分的平均绝对误差(Mean Absolute Error,简称 MAE)[17,21-22]:

$$\mathrm{MAE}=\frac{1}{c}\sum_{\alpha=1}^{c}|v_{\alpha}-r_{\alpha}| \tag{1}$$

其中,c为系统中用户 α打分产品的个数,r_{α}为用户的实际打分,v_{α}为系统的预测打分。推荐算法的准确度是所有用户准确度的平均。平均绝对误差有两个优点:1)计算方法简单,易于理解;2)每个系统的平均绝对误差唯一,从而能够区分两个系统平均绝对误差的差异。在有些系统中,用户只在意推荐列表前端的预测误差,而对系统的整体误差并不是很在意,这时也不适合采用预测准确度进行评估。平均绝对误差在用户偏差的程度比较小时也并不适用,因为用户只关心把好产品错归为坏产品,或者把坏产品错归为好产品的比例。例如,以 3.5个星为界区分好坏,那么把 4预测成了 5,或者把 3预测成了 2都对用户没有影响。

与平均绝对误差相关的其它指标有平均平方误差(Mean Squared Error,简称 MSE)和标准平均绝对误差(Normalized Mean Absolute Error,简称 NMAE)[7]。平均平方误差定义为

$$\mathrm{RMSE}=\sqrt{\frac{1}{N}\sum_{\alpha=1}^{n}|v_{\alpha}-r_{\alpha}|^{2}} \tag{2}$$

其中, n为系统中用户 —产品对 (,i α)的个数。平均平方误差在求和之前对系统预测打分与用户打分误差进行平方, 因此打分误差越大, 其对平均平方误差的影响会比平均绝对误差更大。标准平均绝对误差的定义为 NMAE= MAE/(r_{max}— r_{min}), 其中, r_{min}和 r_{max}分别为用户打分区间的最小值和最大值。标准平均绝对误差在打分值的区间内作标准化, 从而可以在不同的数据集上对算法的效果进行比较。如果系统中只有用户的二元选择信息, 即喜欢或不喜欢, 则并不适合采用预测准确度对系统进行评价, 这类系统更适合用分类准确度度量系统的推荐质量。

2 2　分类准确度

分类准确度定义为推荐算法对一个产品用户是否喜欢判定正确的比例。因此, 当用户只有二元选择时, 用分类准确度进行评价较为合适。应用于实际的离线数据时, 分类准确度可能会受到打分稀疏性的影响。当评价一个推荐列表的质量时, 列表中的某些产品很可能还没有被该用户打分, 因此会给最终的评价结果带来偏差。一个评价稀疏数据集的方法就是忽略还没有打分的产品, 那么推荐算法的任务就变成了“预测已经打分的产品中排名靠前的产品”。另外一个解决数据稀疏性的方法就是假设存在默认打分, 常常对还没有打分的产品打负分[21]。这个方法的缺点就是默认打分常常与实际的打分相去甚远。第 3种方法是计算用户打分高的产品在推荐列表中出现的次数, 即度量系统在多大程度上可以识别出用户十分喜欢的产品。这种方法的缺点是容易把推荐系统引向偏的方向: 一些方法或者系统对某数据集中已知的数据表现非常好, 但是对未知的数据表现十分差。

分类准确度评价并不直接评价算法的打分是否准确。如果分类的信息准确无误, 与实际打分存在偏差也是允许的。下面简要介绍广泛使用的分类准确度指标: 准确率, 召回率以及相关的指标。Billsus D等人[23–26]最先把准确度和召回率引入到推荐系统中, 对推荐系统进行评价。假设系统中的产品总数为 N, 被推荐产品的总数为 N_s, 其中 $N_s = N_{rs} + N_{is}$, N_{rs}和 N_{is}分别为被推荐产品中用户喜欢和不喜欢的产品数。相应地, N_m和 N_{in}分别为未被推荐产品中用户喜欢和不喜欢的产品数。而 $N_n = N_m + N_{in}$为未被推荐的产品数。显然, $N = N_s + N_n$。准确率和召回率可以根据表 1进行计算。如果用户的打分没有分为两类, 则需要进行转换。例如 MovieLens数据为 5分制[20], 通常 3 ~ 5分被认为是用户喜欢的, 1 ~ 2分被认为是用户不喜欢的。准确率定义为系统的推荐列表中用户喜欢的产品和所有被推荐产品的比率:

$$P = \frac{N_{rs}}{N_s} \tag{3}$$

准确率表示用户对一个被推荐产品感兴趣的可能性。召回率定义为推荐列表中用户喜欢的产品与系统中用户喜欢的所有产品的比率。

$$R = \frac{N_{rs}}{N_r}, \quad N_r = N_{rs} + N_m \tag{4}$$

表 **1**　准确率、召回率数据表

	被推荐产品数	未被推荐产品数
用户喜欢产品数	N_{rs}	N_m
用户不喜欢产品数	N_{is}	N_{in}
N	N_s	N_n

召回率表示一个用户喜欢的产品被推荐的概率。准确率和召回率可以有两种计算方法: 以用户为中心和以系统为中心。以用户为中心的方法中分别计算每个用户的准确率和召回率, 再对所有的用户进行平均。这种方法的重点在于考虑用户的感受, 保证每个用户对系统表现的贡献强度是一致的。以系统为中心的方法以考察系统的总体表现为目的, 不需要对所有用户做平均。准确率和召回率的定义依赖于用户喜欢和不喜欢的产品分类。如何定义用户是否喜欢一个产品, 尤其是用户是否喜欢一个没有打分的产品还是十分困难的。在推荐系统中, 无法知道用户是否喜欢某些未知的产品, 所以召回率在纯粹意义上讲并不适合度量推荐系统。因为召回率需要知道每个用户对未选择产品的喜好, 然而这与推荐系统的初衷是相悖的。

SarwarBM等[25]提出了一种计算召回率的方法。首先, 把用户的打分数据随机地分为训练集和测试集。算法利用训练集数据预测用户感兴趣的产品并给出推荐列表。进而, 把测试数据中出现在推荐列表前 L个的比例作为召回率。尽管这一方法非常有效, 但是召回率与系统的实际情况存在着偏差。因为每个用户打过分的产品只是所有数据的一小部分, 因此测试集中用户喜欢的产品也只是所有用户喜欢产品中的一小部分。尤

其需要指出的是这个指标严重依赖于每个用户打过分的产品中用户喜欢的产品的比例。如果一个用户仅给一小部分产品打了分，那么系统的测试召回率低并不意味着系统的表现不好。这是因为系统推荐了很多用户没有打分的产品，这些产品中很可能有很多用户原本就非常喜欢。因此，对一个用户打分非常稀疏的数据集，其测试集由于取样过少而无法反映全部数据的真实特性，根据测试集算出指标误差过大，这种情况下即使对于同一数据集而言，不同算法的指标也不具有比较的意义。

利用准确率和召回率对推荐系统进行评价的最大问题在于它们必须要一起使用才能全面评价算法的好坏。Basu C等[24]提出了一种方法可以同时计算准确率和召回率。把用户的打分数据随机地分为训练集和测试集，利用训练集数据得到推荐列表，再分别计算算法的准确率和召回率。如果假设测试集中用户喜欢和不喜欢的产品的分布与其在所有产品的分布一致，那么准确率和召回率将与实际值很接近。Cleverdon CW[27]发现准确率和召回率存在负相关关联，并且取决于推荐列表的长度。因此，如果系统不能返回一个固定长度的推荐列表，就必须提供一个准确率/召回率向量来度量系统的表现。在实际应用的时候，如果系统的任务是发现所有用户喜欢的产品，召回率就变得很重要，因此需要在一定的召回率水平下考虑准确率。为了同时考察准确率和召回率，PazzaniM等把二者综合考虑提出了F指标[8]。F指标定义为

$$F=\frac{2PR}{P+R} \tag{5}$$

其中，P为准确率，R为召回率。由于F指标把准确率和召回率统一到一个指标，因此得到了广泛的应用。

另外一个度量系统分类准确度的重要指标就是ROC曲线[28-29]。关于ROC曲线，有两个不同的概念。Swets JA[28-29]称之为相对工作特征(Relative Operating Characteristic)，然而，更多的人称之为受试者工作特征(Receiver Operating Characteristic)[30]。对于推荐系统，Herlocker J等介绍了一种ROC曲线的画法[18]：1)确定用户对每个产品感兴趣与否。2)根据预测结果为用户提供一个推荐列表，从图的原点开始，如果预测的产品符合用户喜好，画一个竖线；如果预测的产品不符合实际，画一个横线；如果预测产品还没有被打分，那么抛弃这个产品，并不影响曲线。一个最好的预测系统产生一个竖的ROC线，随机预测产生从原点到右上角的直线。如果所有用户喜欢的产品都排在不喜欢的产品前面，那么就得到了一条完美的ROC曲线。ROC曲线不适合描述多通道打分系统。但是ROC曲线下的面积与起点和终点构成的长方形的面积之比可以评价系统的辨别好坏产品的能力，这一比例称之为Swet A指标[28-29]。Hanley JA[30]指出ROC曲线下的面积等于系统能够在两个产品中选择正确的概率。

Swet A指标的优点是可以用一个数值表征系统的表现，并且不受推荐列表长度的限制。缺点是：1)需要分析每个用户潜在感兴趣的产品；2)只考虑面积。因此，如果推荐列表的某一片段预测结果都正确，那么这一片段中产品的排序前后位置对曲线下的面积没有影响。但是，实际上推荐列表前面的负面影响对用户是十分巨大的；3)识别两条曲线面积的不同需要大量的数据点以保证统计结果的准确度。

2.3 排序准确度

排序准确度用于度量推荐算法产生的列表符合用户对产品排序的程度。对于排列顺序要求严格的系统，排序准确度十分重要。不同于分类准确度，排序准确度指标更适合于评价需要给用户提供一个排序列表的推荐系统。排序准确度对于只需要知道分类准确度的系统来说太敏感了。例如，尽管前10个都是用户感兴趣的，但排序准确度可能很低，因为用户最感兴趣的产品排在了第10位。周涛等[12]提出用平均排序分(average ranking score)度量推荐系统的排序准确度，具体定义如下：

$$r_i=\frac{L_i}{N} \tag{6}$$

其中，N为训练集中用户未选择的产品个数，L_i为预测集中待预测产品i在推荐列表中的位置。例如，如果训练集中用户未选择的产品有100个，测试集中用户喜欢的某个产品在推荐列表中排名第10，那么这个产品的排序分就是$r_i=10/100=0.1$，用户喜欢的所有产品排序分的平均值可以度量系统的准确度。排序分越小，说明系统趋向于把用户喜欢的产品排在前面。反之，则说明系统把用户喜欢的产品都排在了后面。由于平均

排序分不需要额外的参数, 而且不需要事先知道用户对产品的喜好打分, 因此可以很好刻画不同算法对同一数据集的效果。

2.4 预测打分关联

预测打分关联分析系统的打分排序与用户实际的打分排序之间的关联关系, 常常用于刻画推荐系统的准确度。与预测准确度不同的地方在于, 预测打分关联不考虑预测打分与用户打分各单项的偏差, 而是考虑两者之间整体的相关程度。推荐系统中, 3个常用的相关性描述有 Pearson关联[31]、Spearman关联[32]和 Kendall's Tau[33]。Pearson关联度量两个向量的相关度, 定义如下:

$$C=\frac{\sum (x-\bar{x})(y-\bar{y})}{n\sqrt{\prod (x-\bar{x})^2}\sqrt{\prod (y-\bar{y})^2}} \tag{7}$$

其中, x和 y为两个向量中对应位置的打分值, n为向量的维度。Spearman关联与 Pearson系数一样, 只不过 x和 y由打分值变成了相应的排序位置。Kendall's Tau是另一种计算排名相关性的方法, Tau越大表示系统预测结果越好, 反之, 则不好。下面为一种计算 Tau的近似方法:

$$Tau=\frac{C-D}{\sqrt{(C+D+TR)(C+D+TP)}} \tag{8}$$

其中, C为系统预测正确的喜好偏序数, D为预测错误的喜好偏序数, TR为用户打分相同的产品数, TP为具有相同预测值的产品数。假设有 3个产品 X, Y, Z, 用户的打分为 5, 4, 3, 系统的预测打分为 5, 3, 4, 那么用户对产品的喜好偏序关系为 X> Y> Z, 而系统给出的预测序列为 X> Z> Y, 则系统预测正确的序列为 X> Y, X> Z, 因此 C= 2。预测错误的序列为 Y> Z, 因此 D= 1。用户打分相同的产品为 TR= 0, 系统预测值相同的产品数为 TP= 0, 因此, Tau= 1/3。尽管简单, 上述相关性指标在推荐系统中应用广泛。

预测打分关联的优点是: 可以比较多通道打分系统的排名, 计算简单且对全部系统只返回一个值。但是不同的计算方法也有各自的缺点。例如, Kendall's Tau的缺点是给每个等距离交换赋予相等的权重。因此, 在推荐列表中排名第 1与第 2的差别和排名 1 000与 1 001的差别一样。而实际上, 用户可能只关心排名前 10的产品, 而永远不会检查排在 1 000的产品。因此, 排名 1与 2之间的差别对用户的影响更大。Spearman相关对"弱排序"解决得并不好。所谓弱排序指的是至少两个产品的打分是一样的, 反之, 每个产品打分都不同的排序叫做完全排序。由于系统会把得分相同的产品排在不同的位置, Spearman对不同的排序的反馈值不一样。但是这并不合理, 因为用户并不关心他打分相同的产品是如何排序的。Kendall's Tau也有类似的问题。

2.5 距离标准化指标

1995年, Yao Y Y[34]首次提出距离标准化指标, 简称为 NDPM, 在推荐系统中, NDPM的核心思想为: 对比系统预测打分排名与用户实际排名的偏好关系, 对基于偏好关系的度量进行标准化, 具体定义如下:

$$NDPM=\frac{2\bar{c}+c^{u}}{2c^{i}} \tag{9}$$

其中, $\bar{c}$为系统排序与用户排序相冲突的个数, 例如, 系统认为用户喜欢 1超过 2, 而用户却说正好相反; c^{u}为相容的个数; c^{i}为用户排序中有偏好关系的产品总数。NDPM与 Spearman系数和 Kendall's Tau相似, 但是 NDPM的结果更精确。Balabanovc M和 Shoham Y将 NDPM指标用于评价 FAB系统的准确度[7], 取得了非常好的效果。

2.6 半衰期效用指标

推荐系统为用户呈现一个排序的产品列表, 但多数用户并不愿意深入浏览这个列表。在 Internet网页推荐系统中, 设计者声称绝大多数的 Internet用户不会深入浏览搜索引擎返回的结果, 而且用户愿意浏览推荐列表的函数呈指数衰减, 这里将衰减强度描述为一个半衰参数。用户 i的期望效用 R_i定义如下[24, 35]:

$$R_i=\sum_j \frac{\max(r_{ij}-d,0)}{2^{(j-1)/(h-1)}} \tag{10}$$

其中，r_{ij}为用户 i 对推荐列表中排名第 j 的产品的打分，d为默认打分，h为半衰期。系统的半衰期由所有用户半衰期的平均值得到。为了得到一个高的半衰期效用值，系统必须把用户打分高的产品赋予高的打分值。缺点是如果实际的效用函数不是指数衰减的，那么系统的半衰期效用与用户的实际感受差别就会很大。例如，如果用户常常在推荐列表前 20个产品中搜索，那么效用函数只应该对前 20个产品赋值，而后的都应设为 0。半衰期效用指标有如下两个缺点：1）系统中的弱排序使得即使对同一个系统排序，其结果也不同；2）因为 max函数的缘故，所有打分小于默认值的产品的作用相同。

3 准确度之外的评价指标

在实际应用中，我们已经发现准确率高的推荐系统并不能保证用户对推荐系统呈现的结果满意[18]。推荐系统不仅需要高的准确率，还需要得到用户的认可，而后者才是更本质的。例如，系统推荐了非常流行的产品给用户，使得准确度非常高，但是这些信息用户很可能从其他渠道早已得到，因此用户不会认为这样的系统是有价值的。一般而言，系统推荐非流行的产品会使得系统的准确度降低，但这时用户反而容易发现一些新奇的，自己找不到的产品。本节介绍除了准确率之外度量推荐系统的评价指标，包括推荐的流行性和多样性、覆盖率、新鲜性和意外性以及用户的满意度等度量指标。

3.1 推荐列表的流行性和多样性

除了准确度之外，周涛等[12]提出利用推荐产品的平均度和平均海明距离对推荐产品的流行性以及不同推荐列表的多样性进行度量。一个产品的度就是被收藏或购买的次数，产品度越大，说明越流行。如果系统趋向于推荐流行的产品，那么被推荐产品的平均度会很高；反之，平均度会很低。一般而言，被推荐产品的平均度小的系统相对更好。另外，个性化推荐系统设计的宗旨就是针对不同用户的需求给出不同的推荐，因此对不同用户推荐的产品也需要表现出相当的多样性。而准确度高的推荐系统不一定能照顾到不同用户的不同需求。基于此，周涛等[12]提出利用平均海明距离度量推荐系统中推荐列表的多样性。用户 i 和 j 推荐列表的海明距离被定义如下：

$$H_{ij}=1-Q_{ij}/L \tag{11}$$

其中，L为推荐列表的长度，Q_{ij}为系统推荐给用户 i 和 j 的两个推荐列表中相同产品的个数。推荐列表的多样性定义为 H_{ij}的平均值 $\langle H\rangle$。推荐列表多样性的最大值为 1，即所有用户的推荐列表完全不一样；最小值为 0，意味着所有用户的推荐列表一模一样。

利用准确度，产品的平均度和海明距离可以对推荐算法进行综合评价[12]。最近的研究表明，可以找到准确度很高，而且推荐列表的多样性又很大的算法。

3.2 覆盖率

覆盖率定义为可以预测打分的产品占所有产品的比例[18-19]。低覆盖率的系统只能对有限的产品进行评估，因此对用户的意义不大。在推荐系统中，覆盖率尤其重要，因为只有覆盖率高才有可能尽可能多地找到用户感兴趣的产品。覆盖率最简单的计算方法就是随机地选取若干用户—产品对，对每一个用户—产品对都做一次预测，衡量一下可预测的产品占所有产品的比例。正如准确率和召回率必须同时使用一样，覆盖率必须结合准确率进行使用，因为推荐系统不能仅仅为了提高覆盖率而给出一个差的准确率。

3.3 新鲜性和意外性

一些推荐系统具有非常高的准确率和相对合理的覆盖率，但是仅仅有这些，系统可能还是对用户没有任何帮助。例如，如果某购物推荐系统向没有购买牛奶的用户推荐牛奶，在统计上，这或许非常准确：每个人都可能购买牛奶。然而，人们都很熟悉牛奶，即使系统不推荐，用户也会知道是否需要购买。因此，最佳的方案是向用户推荐他们从未购买过，但是感兴趣的产品。音乐或电影推荐系统也是如此，给用户推荐流行的产品，无疑会提高系统的准确率，但是用户不会从系统中得到任何新的信息。

Swearingen K等[36]发现用户喜欢系统推荐他们熟悉的产品。这很奇怪，因为推荐系统并没有给用户任何全新的信息。然而，这些用户熟悉的产品会增加用户对推荐系统的信心，这一点非常重要。另外，用户更

愿意买熟悉的产品, 而不是全新的东西。与此相对应的是用户免费下载资料, 他们更愿意选择新鲜的推荐。一些推荐系统试图预测用户对一个产品熟悉的程度。对某些任务, 系统推荐用户熟悉的产品; 其它情况下, 很少或者不推荐用户熟悉的产品。因此, 我们介绍两种新的指标度量推荐系统: 新鲜性和意外性[36]。推荐用户感到意外的产品会帮助用户发现一些他还没有发现的可能感兴趣的产品。在这里新鲜性和意外性具有本质的不同。例如, 考虑一个电影推荐系统, 这个系统只考虑用户喜欢的导演信息。如果系统给用户推荐了他喜欢的导演执导的一个自己并不熟悉的电影, 这个电影就是新鲜的, 但是并不是意外的。如果系统推荐了一个新导演的电影, 那么系统提供了一个意外的推荐。从定义的角度来讲, 意外的也是新鲜的。区别新鲜性和意外性在推荐系统中是非常重要的, 因为基于内容的推荐系统不会给出意外性的推荐, 只会给出新鲜性的推荐。

SarwarBM等[37]还讨论了如何修改用户的推荐列表使得新的推荐列表具有新鲜性和意外性。一个简单的方法就是建立一个独立的流行产品列表, 在把用户的推荐列表呈现给用户之前, 把那些出现在用户推荐列表中的流行产品删除, 未删除的产品在推荐列表中的位置自动向前移动。因为每个用户都有不同的经历和爱好, 因此每一个用户删除的流行产品应该是不同的。另一个方法是在用户群中把每个用户喜欢某个产品的概率除以群中所有人喜欢这个产品的概率之和, 再重新排序, 这样可以判断此用户是否比群体中的其它人更喜欢某个产品。流行的产品将推荐给目前没有选择它的用户, 而如果非流行的产品符合某个用户的口味, 就会被推荐。这个方法将大大地改变每个用户的推荐列表, 帮助用户发现令他们意外的他们喜欢的产品。

定义一个指标来度量系统的意外性十分困难, 因为意外性是度量推荐系统提供给用户的产品既感兴趣又意外的程度。一个好的意外性指标应该从用户兴趣随时间变化的角度进行推荐。产品在多大程度上属于用户没有买过的类别? 如果推荐给用户, 用户是否会满意? 好的意外性指标希望推荐系统能很好地识别以前不知道的产品。尽管新鲜性和意外性的重要性在 2001年就被提出[36-37], 直到目前, 还没有系统地定量地研究新鲜性和意外性指标的工作。

3.4 用户的满意度

大多数推荐系统都对推荐给用户的产品进行预测打分。CosleyD等[38]的研究发现系统的预测打分值也影响着用户的打分。他们同时发现, 系统给出的错误打分会降低用户对系统的满意度。这些关于用户评价的研究显示用户对推荐系统的预测准确度非常敏感。SweringenK和 SinhaR[39]在 3个推荐系统上的实验系统地研究了用户满意度对推荐系统的作用。首先, 赋予系统足够的打分, 使得可以对用户进行推荐。接着让用户从头至尾浏览推荐列表, 直到用户发现一个感兴趣的产品。这个实验能让我们知道, 推荐准确度是否真的影响用户的满意度。用户在使用推荐系统的时候必须信任推荐系统, 如果系统能向用户解释为什么给用户推荐这些产品, 就非常有助于增强用户对系统推荐结果的信心。用户在使用推荐系统的同时也在评价推荐系统。例如, 仅仅一个电影的名字并不能说服用户点击看这部电影。因此, 系统提供信息的种类和质量也决定了用户是否认可推荐的结果。另外, 上述实验中用户的反馈信息表明, 系统的满意度并不是用户选择系统的根本原因。以 Amazon和 MediaUnbound为例, 用户更愿意从 Amazon网站购买东西。然而, 在实际调查中, 用户认为 MedaUnbound更有用, 可以更好地理解用户的喜好。进一步的研究表明, Amazon更多地向用户推荐用户熟悉的产品, 这或许是 Amazon成功的原因之一。因此, 用户会在某个目的下选择某个系统, 而在另一个目的下选择另一个系统。ChoiceStream的调查结果显示, 用户的满意度对推荐系统的错误推荐非常敏感。通常有 43%的错误推荐是用户已经购买或拥有的产品, 41%的错误推荐是并不适合用户的推荐。

总之, 用户对推荐系统的满意度不仅仅取决于系统的准确度, 而是更多地取决于系统在多大程度上可以帮助用户完成任务。因此, 如果想要度量用户对一个推荐系统的评价, 首先这个系统必须对自身的任务有一个清晰的定义, 进而, 针对特定的任务选择适当的指标对推荐算法进行评价。

4 小结

有效地对推荐系统进行评价是一个非常具有挑战性的课题。本文简要介绍了推荐系统的各种指标及其优缺点,进而介绍了准确性之外评价推荐系统的其它指标。这些指标有些是基于推荐算法的,有些独立于推荐算法,从系统或者用户的角度对推荐系统进行评价。虽然目前为止有些指标还难以量化,只能进行定性的讨论,但是无疑对推荐系统的客观、合理评价具有重要的指导意义。随着 Web2.0技术和互联网技术的迅猛发展,个性化推荐系统的应用越发广泛。实际应用的巨大需求为推荐系统的研究工作带来了巨大的推动力。然而,推荐系统的评价指标至今仍是“乱花渐欲迷人眼”。目前为止,从纷繁复杂的指标群当中找到一个合适指标对系统或算法进行评价还是十分困难的,尤其在需要对比不同系统优劣的时候,更是难以找到可以综合评价系统表现的指标。希望本文的工作可以促进推荐系统研究工作者对推荐系统评价指标的了解,并根据自身系统的任务,选择合适的指标进行进一步的研究工作。

虽然目前已经有很多的评价指标,但是所有这些评价指标都面临着一些共性的问题。这些问题的解决对于推荐系统的研究工作具有非常重要的意义。总结起来,推荐系统的评价工作可以从以下方面继续进行深入研究:

1)用户对算法准确度的敏感度。一个错误的预测会大大降低用户对推荐系统的信心。对于不同的指标,准确率改变多少用户就可以察觉系统的改变?用户对哪个准确度指标最敏感?用户对准确度的感受受到哪些因素影响?其它统计特性如何影响用户的满意度,例如覆盖率、新鲜性等。如果以上问题可以回答,那么我们就可以用离线数据对推荐系统进行测试。

2)算法对不同领域的普适性。不同的推荐算法在不同的数据集上的表现不同。对于某个推荐算法,在什么类型的数据上可以发挥最好的效果。这些需要进行深入的研究。如果这个问题解决了,那么系统设计者可以根据自身数据集的特点选择最适合的算法进行工作。

3)广义的质量评价。大部分评价指标只重视准确度,忽略了覆盖率,新鲜性系统发现新鲜产品的能力以及用户的满意度等特性。因为用户总是同时从多个方面综合评价实际系统,因此准确度高的算法在实际应用中表现却不一定好。是否能把这些指标进行结合,提出一个综合性评价指标,这样系统设计者就可以模仿用户直接对系统进行评价。

4)个人隐私的保护。推荐系统的本质是利用用户现有的选择信息或者配置文件,发掘用户的兴趣、爱好。用户如果希望得到推荐系统的帮助,必须共享一些个人的隐私数据。对系统来说,不仅需要有效保护用户的个人隐私,而且需要在尽可能少利用用户隐私数据的情况下做出准确、合理的推荐。反过来,用户只有在确认系统可以有效保护个人的隐私数据的情况下,才愿意使用推荐系统。因此,未来的准确度指标应该结合个人隐私数据保护的水平进行使用。

5)推荐系统的鲁棒性研究。推荐系统在实际投入应用后,有些恶意用户希望用自己的选择信息破坏系统中正常的用户—产品二元关系。以期降低系统的准确度,改变系统提供给正常用户的推荐列表,从而达到破坏系统本身或抬高某些产品被推荐程度的目的。随着推荐系统的日益广泛使用,系统鲁棒性的研究日益重要。只有经得起这种恶意攻击考验的系统才具有持久的生命力。

感谢张子柯、金慈航、吕琳媛对文章的认真修改和中肯意见。

参考文献:

[1] Resnick P, Iakovou N, Sushak M, et al. GroupLens: an open architecture for collaborative filtering of netnews[DB/OL]. [2008-10-12]. http://www.si.umich.edu/~presnick/papers/cscw94/GroupLens.htm.

[2] Hill W, Stead L, Rosenstein M, et al. Recommending and evaluating choices in a virtual community of use[DB/OL]. [2008-10-12]. http://sigchi.org/chi95/proceedings/papers/wch_bdy.htm.

[3] 刘建国, 周涛, 汪秉宏. 个性化推荐系统研究进展[J]. 自然科学进展, 2009, 19(1): 1—15.
Liu Jianguo, Zhou Tao, Wang Binghong. Progress of the personalized recommendation systems[J]. Progress of Nature and Science, 2009, 19(1): 1—15.

[4] Liu R R, Jia C X, Zhou T, et al. Personal recommendation via modified collaborative filtering[J]. Physica A, 2009, 388: 462—468.

[5] Konstan J A, Miller B N, Maltz D, et al. GroupLens: applying collaborative filtering to usenet news[J]. Comm ACM, 1997, 40(3): 77—87.

[6] Linden G, Smith B, York J. Amazon. com recommendations: item-to-item collaborative filtering[J]. IEEE Internet Computing, 2003, 7(1): 76—80.

[7] Balabanovic M, Shoham Y. Fab: content-based, collaborative recommendation[J]. Comm ACM, 1997, 40(3): 66—72.

[8] Pazzani M, Billsus D. Learning and revising user profiles: the identification of interesting web sites[J]. Machine Learning, 1997, 27: 313—331.

[9] Mooney R J, Bennett P N, Roy L. Book recommending using text categorization with extracted information[DB/OL]. [2008—10—12]. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.16.4905.

[10] Chang Y J, Shen J H, Chen T J. A data mining-based method for the incremental update of supporting personalized information filtering[J]. Journal of Information Science and Engineering, 2008, 24(1): 129—142.

[11] Zhou T, Ren J, Medo M, et al. Bipartite network projection and personal recommendation[J]. Phys Rev E, 2007, 76: 046115.

[12] Zhou T, Jiang L L, Su R Q, et al. Effect of initial configuration on network-based recommendation[J]. Europhys Lett, 2008, 81: 58004.

[13] Zhang Y C, Medo M, Ren J, et al. Recommendation model based on opinion diffusion[J]. Europhys Lett, 2007, 80: 68003.

[14] Zhou T, Su R Q, Liu R R, et al. Ultra accurate personal recommendation via eliminating redundant correlations[DB/OL]. [2008—10—12]. http://arxiv.org/abs/0805.4127.

[15] Soboroff I M, Nicholas C K. Combining content and collaboration in text filtering[DB/OL]. [2008—10—12]. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.39.8019.

[16] Yoshii K, Goto M, Komatani K, et al. An efficient hybrid music recommender system using an incrementally trainable probabilistic generative model[J]. IEEE Transactions on Audio Speech and Language Processing, 2008, 16(2): 435—447.

[17] Shardanand U, Maes P. Social information filtering: algorithms for automating "word of mouth"[C]// Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems. New York: ACM Press, 1995: 210—217.

[18] Herlocker J, Konstan J A, Terveen L, et al. Evaluating collaborative filtering recommender systems[J]. ACM Transactions on Information Systems, 2004, 22(1): 5—53.

[19] Geyer-Schulz A, Hahsler M, Wien W, et al. Evaluation of recommender algorithms for an internet information broker based on simple association rules and on the repeat-buying theory[DB/OL]. [2008—10—12]. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.19.911.

[20] Dahlen B J, Konstan J A, Herlocker J L, et al. Jumpstarting movielens: user benefits of starting a collaborative filtering system with "dead data"[DB/OL]. [2008—10—12]. http://www.bibsonomy.org/bibtex/24433e6aa3be2cdad117bfb5fd7a757a1/bsmyth.

[21] Breese J S, Heckerman D, Kadie C. Empirical analysis of predictive algorithms for collaborative filtering[DB/OL]. [2008—10—12]. http://www.cs.pitt.edu/~mrotaru/comp/rs/Breese%20UAI%201998.pdf.

[22] Herlocker J L, Konstan J A, Borchers A, et al. An algorithmic framework for performing collaborative filtering[C]// Hearst M A, Gey F F, Tong R. Proceedings of the 22nd International Conference on Research and Development in Information Retrieval (SIGIR'99) (Aug). New York: ACM Press, 1999: 230—237.

[23] Billsus D, Pazzani M J. Learning collaborative information filters[C]// Rich C, Mostow J. Proceedings of the 15th National Conference on Artificial Intelligence (AAAI—1998). Menlo Park, Calif: AAAI Press, 1998: 46—53.

[24] Basu C, Hirsh H, Cohen W W. Recommendation as classification: using social and content-based information in recommendation[C]// Rich C, Mostow J. Proceedings of the 15th National Conference on Artificial Intelligence (AAAI—1998). Menlo

Park, Calif: AAAI Press, 1998: 714—720.
[25] Sarwar B M, Karypis G, Konstan J A, et al. Analysis of recommendation algorithms for e-commerce[C]//Proceedings of the 2nd ACM Conference on Electronic Commerce (EC'00). New York: ACM Press, 2000: 285—295.
[26] Sarwar B M, Karypis G, Konstan J A, et al. Application of dimensionality reduction in recommender system— a case study[DB/OL]. [2008—10—12]. http://robotics.stanford.edu/~ronnyk/WEBKDD2000/papers/sarwar.pdf.
[27] Cleverdon C W, Mills J, Keen M. Factors Determining the Performance of Indexing Systems[M]. England: Cranfield (Beds), 1966.
[28] Swets J A. Information retrieval systems[J]. Science, 1963, 141: 245—250.
[29] Swets J A. Effectiveness of information retrieval methods[J]. Amer Doc, 1969, 20: 72—89.
[30] Hanley J A, Mcneil B J. The meaning and use of the area under a receiver operating characteristic (ROC) curve[J]. Radiology, 1982, 143: 29—36.
[31] Rodgers J L, Nicewander W A. Thirteen ways to look at the correlation coefficient[J]. The American Statistician, 1988, 42: 59—66.
[32] Spearman C. The proof and measurement of association between two things[J]. Amer J Psychol, 1904, 15: 72—101.
[33] Kendall M. A new measure of rank correlation[J]. Biometrika, 1938, 30: 81—89.
[34] Yao Y Y. Measuring retrieval effectiveness based on user preference of documents[J]. JASIS, 1995, 46: 133—145.
[35] Heckerman D, Chickering D M, Meek C, et al. Dependency networks for inference, collaborative filtering, and data visualization[J]. JMach Learn Res, 2000, 1: 49—75.
[36] Swearingen K, Sinha R. Beyond algorithms: an HCI perspective on recommender systems[DB/OL]. [2008—10—12]. http://www.citeulike.org/user/lschiff/article/375842.
[37] Sarwar B M, Karypis G, Konstan J, et al. Item-based collaborative filtering recommendation algorithms[DB/OL]. [2008—10—12]. http://www10.org/cdrom/papers/pdf/p519.pdf.
[38] Cosley D, Lam S K, Albert I, et al. Is seeing believing: how recommender interfaces affect users' opinions[DB/OL]. [2008—10—12]. http://www.grouplens.org/papers/pdf/conform—chi03.pdf.