Computer Communications 41 (2014) 1–1

[0](http://dx.doi.org/10.1016/j.comcom.2013.06.009)



Contents lists available at

ScienceDirec

[t](http://www.sciencedirect.com/science/journal/01403664)

Computer Communications

journal homepage:

www.elsevier.com/locate/comco

[m](http://www.elsevier.com/locate/comcom)



关于采用协同过滤算法的社交推荐系统的研究



Xiwang Yang a,⇑, Yang Guo b, Yong Liu a, Harald Steck c,1

a Polytechnic Institute of NYU, Brooklyn, NY, USA b Bell Labs, Alcatel-Lucent, Holmdel, NJ, USA c Netflix Inc., Los Gatos, CA, USA

|  |  |  |
| --- | --- | --- |
| a r t i c l e | i n f o | a b s t r a c t |

|  |  |
| --- | --- |
| Article history:  Received 22 July 2012  Received in revised form 4 June 2013  Accepted 26 June 2013 Available online 16 July 2013  Keywords:  Social network  Recommender system  Collaborative filtering | 推荐在我们的日常生活中扮演着日益重要的角色。 推荐系统会自动向用户建议可能感兴趣的物品。 最近的研究表明，可以利用来自社交网络的信息来提高建议的准确性。 在本文中，我们提出了基于协作过滤（CF）的社交推荐系统的调查。 我们简要介绍推荐系统和不使用社交网络信息的传统方法的任务。 然后，我们介绍推荐系统如何采用社交网络信息作为提高准确度的附加输入。 我们将基于CF的社交推荐系统分为两类：基于矩阵分解的社交推荐方法和基于邻域的社交推荐方法。 对于每个类别，我们会调查并比较几种有代表性的算法。  2013 Elsevier B.V. All rights reserved. |

1. Introduction

通信网络为人们访问信息提供了便利。 但同时，网络信息的丰富性也带来了“信息超载”问题。 例如，如果有人想购买数码相机，在做出购买决定之前，阅读并比较有关数码相机的所有在线评论将是一件令人抓狂的体验。 推荐系统通过自动向用户建议可能符合其兴趣的商品来处理信息超载问题。 准确的建议使用户能够快速找到中意的物品，而不会被无关的信息淹没。 供应商也很热衷于推荐那些符合他们网站访问者兴趣的产品，并希望他们满意以变成回头客。 难怪，在Netflix竞赛[19]中，仅提高了10％的推荐准确性就获得了100万美元的奖励。

|  |
| --- |
| Corresponding author.  E-mail addresses: xy271@nyu.edu (X. Yang), Yang.Guo@alcatel-lucent.com (Y. Guo), yongliu@poly.edu (Y. Liu), hsteck@netflix.com (H. Steck). 1 Part of this work was done while at Bell Labs, Alcatel-Lucent, Murray Hill, NJ.  0140-3664/$ - see front matter 2013 Elsevier B.V. All rights reserved.  <http://dx.doi.org/10.1016/j.comcom.2013.06.009> |

推荐系统（RS）起源于认知科学，近似理论和信息检索等几个相关的学科。由于推荐的重要性日益增加，自90年代中期以来它已成为一个独立的研究领域[1]。 一般而言，RS向用户建议那些可能对他有利的项目。 一般而言，推荐方法有两种变体：基于内容的和基于协作过滤（CF）的方法[1,2]。 CF方法可以进一步分为基于模型的CF和基于邻域的CF [2,3]。 基于模型的方法使用用户-项目评级来学习得出预测模型。 总体思路是用系统中表示用户与项目互动的因素来表示用户和项目的潜在特

⇑

征，例如用户的偏好类别和项目的类别。 相反，基于邻域的CF方法使用系统中存储的用户项目评级来直接预测新项目的评级。

在线社交网络（OSN）为进一步提高RS的准确性提供了新的契机。在现实生活中，人们通常会在购买产品或使用服务之前向社交网络中的朋友寻求建议。社会学和心理学领域的发现表明，人类倾向于与类似的其他人构建关系，也称为homophily [25]。由于稳定和持久的社交联系，人们更愿意与他们的朋友分享他们的个人意见，并且通常信任来自朋友的推荐，而不是来自陌生人和供应商的推荐。流行的在线社交网络，例如Facebook [21]，Twitter [20]和Youtube [17]，为人们交流和建立虚拟社区提供了新途径。在线社交网络不仅可以让用户更轻松地分享彼此的意见，而且还可以作为开发新的RS算法的平台，以自动化现实生活中的社交网络中的其他手动和轶事社交建议。

社交RS通过将OSN中的用户之间的社会兴趣和社会信任作为额外的输入来改进传统RS的准确性。 例如，由于社交兴趣，用户可以阅读关于事件的特定新闻文章，仅仅是因为事件发生在她的家人住的地方; 由于社交信任，用户可能喜欢Facebook上她的好友推荐的歌曲。 可以基于用户u关于用户v的明确反馈（例如通过投票）来建立一对朋友之间的社交信任，或者可以从隐式反馈（例如，交互/通信/交互的频率和数量） u和v之间的电子邮件交换）。 不同的社交RS算法以不同方式探索社交网络和嵌入式社交信息。

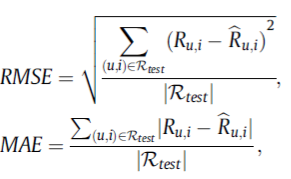
在这项调查中，我们关注基于CF的社交RS，因为大多数现有的社交推荐系统都是基于CF。在对传统的基于CF的RS进行分类后[2,3]，我们将基于CF的社会RS划分为两大类：基于矩阵分解（MF）的社会推荐方法和基于邻域的社会推荐方法。在基于MF的社交推荐方法中，用户 - 用户社交信任信息与用户 - 项目反馈历史（例如，评级，点击，购买）集成以提高传统基于MF的RS的精确度，其仅将用户项反馈数据。基于邻域的社会推荐方法包括基于社交网络遍历（SNT）的方法和最近邻（NN）方法。基于SNT的算法在社交网络中遍历并查询她的邻居中的直接和间接朋友之后，为用户合成推荐。 NN方法将传统的CF邻域与社会邻域相结合，并预测项目的评级或推荐项目列表。

本调查的其余部分安排如下。 我们在第2节中正式提出了RS的任务。第3节简要介绍了传统的基于CF的RS。然后我们将在线社交网络作为第4节中的附加RS输入引入。基于MF的社会推荐方法在第5节中进行了调查。第6节对基于社区的社会推荐方法进行了调查。我们在第7节中对基于CF的社会推荐方法进行了比较。第8节对调查进行了总结。.

1. 推荐系统的任务

推荐系统通常向用户提供她可能感兴趣的推荐项目的列表，或者预测她可能更喜欢每个项目的多少。 这些系统可帮助用户决定适当的项目，并简化在集合中查找首选项目的任务。

文献的主体一直关注于预测评级值的准确性。 为此，测试数据被表示为用户项目评级矩阵，其中u0表示用户数量，i0表示项目数量。 Rui是用户u对于物品i的评级。 通常，用户项目评级矩阵R中有很多缺失值。在商业系统中R的稀疏度通常大于99％[42]。 表1示出了关于六个用户（表示为u1至u6）和七个项目（表示为i1至i7）的玩具评级矩阵。 每个用户对一些项目进行评级，以表达她对每个项目的兴趣。 评级通常是采用数值五星级，其中一颗和两颗星代表负面评级，三颗星代表矛盾心理，而四颗和五颗星代表正面评级。 RS算法预测矩阵中缺失的评级，并且如果她对该项目的预测评级是例如四颗或五颗星则向用户推荐项目。

通常，评级数据集被分成训练集和测试集，其中训练集用于模型拟合和参数调整，并且测试集用于评估RS。 让预测评级的矩阵表示为Rb 2 Ru0i0。 为了评估RS的准确性，最流行的评估指标是均方根误差（RMSE）和平均绝对误差（MAE）：

其中Rtest是所有用户项对的集合; 在测试集中。 RMSE / MAE越低，预测评级越接近实际评级。

商业RS算法通常为用户提供她可能更喜欢的k个推荐项目的列表，也被称为Top-k推荐，而不是向用户呈现预测的项目评级。除了RMSE和MAE之外，top-k RS的直接准确性度量还有top-k命中率，精确度和标准化贴现累积增益（NDCG）[43]等。为了计算top-k命中率或召回率，对于每个用户u，我们根据预测评级 ，i或投票值对这些项目进行排序。这里我们以预测评级为例。如果预测评级是连续的，则排名列表是唯一的。否则，关系可能会随机破坏。如果项目被发现有吸引力或有趣（例如，测试数据中指定的评级高于特定阈值），则该项目被定义为与测试集中的用户相关。例如，Netflix [19]数据中的评级值范围为1到5而[9]中的作者认为5星评级是相关的（即用户肯定喜欢这些项目），而其他评级值和缺失评级值被视为无关紧要。其他选择导致类似的结果。现在，top-k命中率或召回率可以定义为测试集中位于排名列表top-k中的相关项目的分数，记为N(k,u);从用户u的测试集合中的所有相关项目中，用N( u)表示。对于每个用户u，top-k命中率由下式给出



|  |
| --- |
| 它可以聚合到所有用户，以获得平均top-k,例如，测试集的命中率或召回率 |

这可以解释为其中相关项目的分数推荐给用户的k项。 对于给定用户和固定k，精确度与召回成正比。 我们聚合了所有精度用户获得测试集的平均精度如下：

其中u0是用户数量。 NDCG是信息检索的另一个准确性度量，其中推荐项目的收益相对于其在整个推荐清单中的位置/排名以对数形式打折[43]。 具体来说，假设每个用户你有一个gu的增益;当我推荐项目i时，用户u的k个项目列表的折扣累积增益（DCG）被定义为：

其中ij表示有序建议列表中的第j个项目，对数基准b是一个自由参数，通常在2到10之间。以2为底的对数通常用于确保所有位置都打折扣。

用户u的NDCG是DCG的归一化版本，由以下公式给出：

其中DC @k(u)是理想的DCG @ k（u），即项目按降序排列，相对于实部Ru, i，并且列表在位置k被截断。

k项列表的平均DCG定义如下：

类似地，k个项目列表的平均NDC被定义为：

目前，大多数RS算法已经被评估并按其预测能力排列，即它们准确预测用户选择的能力。 然而，现在人们普遍认为预测的准确性是至关重要的，但对于现实世界的好RSs本身并不足够[57,58]。 在许多应用中，人们使用RS不仅仅是对他们口味的准确预测。 用户也可能有兴趣发现新的和多样的项目，偏离他们的日常选择。 在提出良好建议时，RS保留用户的隐私也很重要。 如果要推荐的项目具有高度动态性，例如新闻文章，则RS的响应性至关重要。 因此，识别可能影响RS在特定应用环境中的成功的相关属性集非常重要。

3. 基于反馈数据的推荐系统

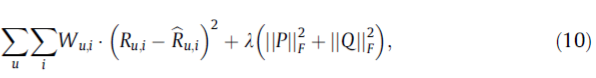
社交网络信息仅在最近才可用来改进推荐系统。在概述使用社交网络信息的各种方法之前，我们简要回顾以下两个主要变体：基于内容的方法和协作过滤方法。

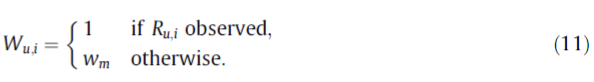
基于内容的方法的基本思想是使用项目的属性来预测用户对其的兴趣。例如，对于一本书，可以使用作者的名字，流派，关键字和标签。然后将这些属性与目标用户的口味相匹配。

协作过滤的关键思想是使用每个用户的反馈。关于用户的反馈，可以区分明确的反馈（例如，用户为项目分配评级）和隐式反馈（例如，用户点击链接，收听歌曲或购买项目）。当关于足够多的用户及其反馈的数据可用时，其可用于确定相似的用户（例如，收听同一首歌曲）;然后可以在类似用户中推荐项目：虽然类似的用户在他们的歌曲集合中有很大的重叠，但是每个用户可能已经收听了一些额外的歌曲;那些额外的歌曲可以推荐给其他类似的用户。作为基于其过去行为的相似性来识别类似用户的这种想法的补充，物品之间的相似性可以类推地推断，即，当它们是由相同用户购买时。发现这种基于协同过滤的基本思想可以在文献中提出非常准确的建议[6,7,44,47]。由于大多数现有的社交推荐系统都是基于CF的，所以本节将重点讨论基于CF的传统RS。

文献中的许多工作集中于使用明确的反馈数据，特别是分配给项目（例如，电影，歌曲，餐馆）的评分（例如，从一星到五星级）。其任务是预测用户评价的其他项目的评价值，推荐准确度通常以RMSE或MAE来衡量。遵循协作过滤方法的基本思想，已经提出了各种最近邻居方法，其可以分为用户 - 用户和项目 - 项邻近模型及其组合，例如参见[6]。发现最准确的方法之一是矩阵分解[24,5,6,13]。这种方法在以前被用于计算机视觉[16]和文本分析[15]。矩阵分解的最基本方法是奇异值分解，但已经开发了许多更复杂的方法，例如[24,5,6,13]。其基本思想是将用户和项目映射到低维空间，并确定这个潜在空间中用户和项目之间的相似性。例如，预测评级矩阵可以如下建模

矩阵和，其中j0 <<i0; u0确定（低）排名（例如50）; 是（全局）偏移量。本质上代表潜在的用户配置文件，而捕获潜在的项目配置文件。 使用梯度下降方法，可以确定它们，例如，通过最小化给定评估值Ru,i和模型为用户u和项目i预测的值之间的平方误差:

为了防止过度拟合，将最后一项加入以规范学习矩阵P和Q; > 0是正则化参数，Frobenius范数由表示。 有多种方式来指定训练权重Wu,i。 一个简单而有效的选择是

 当目标是优化观察评级的RMSE时，则wm = 0.如果目标是为了获得所有项目的良好排序（如通过精度，召回或NDCG测量），那么通过为未观测到的Ru,i输入一个低值一个小值wm> 0 是有利的[9,53]，并结合为未观测到的Ru,i输入一个低值。

矩阵分解方法也可以描述为一个概率图模型[8]，如图1所示。通过组合矩阵Q和P获得评分值Ru, i。对于矩阵Q中的条目的先验分布 P用和表示; 这导致了上面等式中的L2正则化项，详见[8]。 以表示的评级值之前的先验产生重量wm。

Fig. 1.

The BaseMF graphical model.

矩阵分解也与可以与邻域方法结合[6]。 有条件限制的玻尔兹曼机[14]是另一个非常成功的模型。 通过使用不同模型的集合可以进一步提高推荐的准确性，其预测在最终混合步骤中进行组合。 已经开发了各种混合方法，例如， 见[4]。

使用隐式反馈在文献中受到的关注较少。 该领域著名的出版物包括[10,12,11]，其目的在于根据用户过去的观看行为向用户推荐电视节目，例如他们在每种类型的电视节目上花费了多少时间。 由于在实际应用中，像这样的隐式反馈通常比明确的反馈数据丰富得多，通常部署基于隐式反馈的RS。

4. 社交网络作为额外的RS输入

现在我们调查了RS算法如何采用来自社交网络的信息。我们假设用户在底层社交网络中连接，无论是通用社交网络，如Facebook [21]，还是特定领域的推荐社交网络，如用于电影推荐和Epinions的Flixster [23]为广泛的产品推荐。我们将基础社交网络表示为有向图G = (U, F），其中U是具有|U| = u0的用户集合，并且F是友谊链接的集合。社交网络信息由矩阵表示。每个用户你都有一个你信任或跟随的直接邻居的，同时，你被一组用户信任/跟随。用户u与用户v（例如，用户u信任/知道/关注用户v）之间的定向和加权的社交关系由正值表示; 1.一个缺席或不可观察的社会关系由Su, v = sm表示，其中典型的sm = 0。社会权重Su，v可以解释为用户在社交网络中信任或知道用户v的程度。它可以基于用户u关于用户v的明确反馈（例如通过投票）或者从隐式反馈（例如，交互/通信的程度）推断。通常，社会信任Su, v是非负的。在特殊情况下，它也可能带有负值，明确地建模两个用户的冲突口味。 图2示出了六个用户之间的社交网络的玩具示例，其中每个用户都有一组朋友。 每个有向的友谊链接都由一个积极的信任值加权。 所有用户对之间的社会信任被表2中所示的矩阵S捕获。在本文中，信任网络和社交网络可以互换地用作通用术语。



Fig. 2. Social Network Graph.

Table 2

User-User Trust Matrix.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | u1 | u2 | u3 | u4 | u5 | u6 |
| u1 |  | 0.9 | 0.7 |  |  |  |
| u2 | 0.8 |  |  |  |  | 0.7 |
| u3 |  | 0.8 |  |  |  |  |
| u4 |  |  | 0.2 |  | 0.9 |  |
| u5 |  |  |  | 0.8 |  |  |
| u6 | 0.6 |  |  |  | 0.4 |  |

4.1. Social circles for RS

Most existing social RSs mine a social network as a whole. Recently, authors of [52] proposed circle-based recommendations in online social networks. It is obvious that a user’s social life, being online or offline, is intrinsically multifaceted. Intuitively, a user trusts different subsets of friends in different domains. For example, in the context of multi-category recommendation, a user u may trust user v in the Cars category while not trusting v in the Kids’ TV Show category. Therefore, u should care less about v0s ratings in the Kids’ TV Show category than in the Cars category. Ideally, if we know a user’s trust circles in different categories, we probably should only use her trust circles specific to the category for which we want to predict ratings. Unfortunately, in most existing multi-category rating data-sets, a user’s social connections are mixed together from all categories. Even if the circles were explicitly known, e.g. Circles in Google+ [22] or Facebook [21], they may not correspond to particular item categories that a recommender system may be concerned with.

In [52], the authors proposed a set of algorithms to infer category-specific circles of friends, and to infer the trust value on each link based on users’ rating activities in each category. They infer the circles of friends from rating (or other feedback) data concerning items that can be divided into different categories (or genres etc.). The basic idea is that a user may trust each friend only concerning certain item categories but not regarding others. They divide a social network S of all trust relationships into several sub-networks SðcÞ, each of which concerning a single category c of items.

Definition – Inferred Circle: Regarding each category c, a user v is in the inferred circle of user u, i.e., in the set CðucÞ, if and only if the following two conditions hold:

Su;v > 0 in the (original) social network, and

NðucÞ > 0 and NðvcÞ > 0 in the rating data, where NuðcÞ denotes the number of ratings that user u has assigned to items in category c. Otherwise, user v is not in the circle of u concerning category c, i.e., v R CuðcÞ.

This is illustrated in a toy example in Fig. 3. They further proposed a set of algorithms to construct the trust value Sðuc;vÞ of user u to each friend v in her trust circle CðucÞ. When it comes to predict users’ rating of an item in one category, only a circle that corresponds to this item’s category is used as social network input. Recommendation methods that use only trust information within a social circle, instead of the whole social network, are applicable to all social RS algorithms outlined in the following sections.

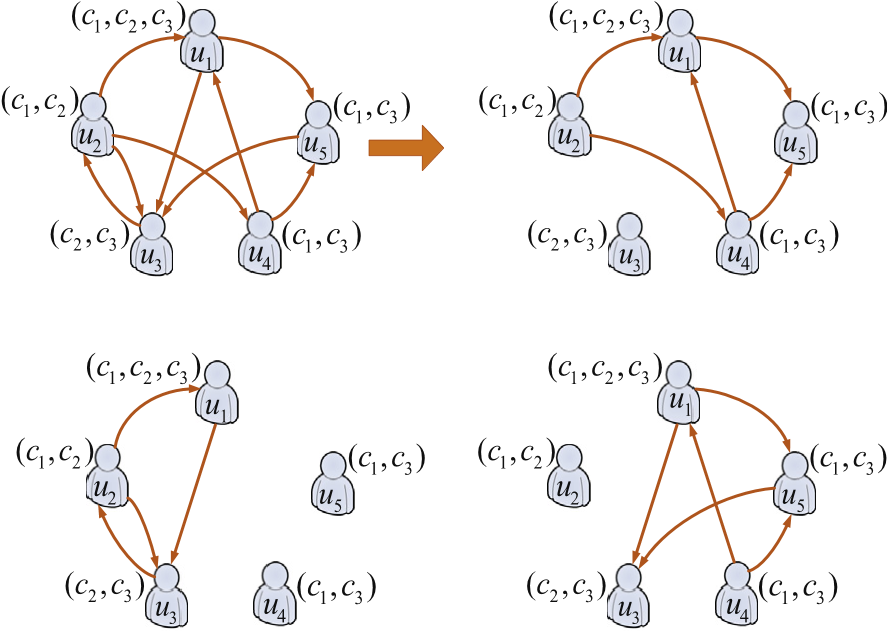


Fig. 3. Illustration of inferred circles, each user is labeled with the categories in which she has ratings. (a): the original social network; (b), (c) and (d): inferred circles for categories c1; c2 and c3 respectively.

1. Matrix factorization based social recommendationapproaches

The general idea of matrix factorization (MF) is to model the user-item interactions with factors representing latent characteristics of the users and items in the system, like the preference class of users and the category class of items. This model is then trained using the available data, and later used to predict ratings of users for new items.

Numerous social matrix-factorization (MF) based RSs have recently been proposed as to improve recommendation accuracy [45,46,44,47–56]. The common rationale behind all of them is that a user’s taste is similar to and/or influenced by her trusted friends in the social network.

In the following subsections, we review some of the existing MF approaches in the literature that combine rating data with social network information [45,46,44,47,48]. We also review some of the nearest-neighborhood based approaches [40,53], which combine the traditional CF neighborhood with social neighborhood.

5.1. Social recommendation (SoRec) model

Social Recommendation (SoRec) was introduced in [45]. The graphical model of SoRec model is presented in Fig. 4. In SoRec, trust between users in a social network is integrated into the recommender systems by factorizing the social trust matrix S. In this model, the social network matrix S may be slightly modified as follows [45]: sffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffi

Su;v ¼ Su;v þdv ; du þ dv

Fig. 4. The SoRec graphical model.

where dþu is the out-degree of user u in the social network (i.e. the number of users who u follows/trusts), and dv is the in-degree of user v in the network (i.e. the number of users who follow/trust user v). The predicted user-item rating-matrix is obtained from the model as follows:

Rb ¼ rm þ QP>; ð12Þ

with matrices P 2 Ri0j0 and Q 2 Ru0j0, where j0  i0; u0 is the rank; and rm 2 R is a (global) offset. In addition to the rating data, also the social network information is used for training this model. The social relationships are predicted as follows:

bS¼ sm þ QZ>; ð13Þ

where Z 2 Ru0j0 is a third matrix in this model, besides P and Q. Note that the matrix Q is shared among the two Eqs. (12) and (13). Due to this constraint, one can expect Q (i.e., the user profile Qu for each user u) to reflect information from both user-item ratings and user-user social trust as to achieve accurate predictions for both. Note that the matrix Z is not needed for predicting rating values, and hence may be discarded after the matrices P and Q have been learned. Both (12) and (13) are combined as follows in the training objective function to optimize RMSE: X Ru;i Rbu;i þ X Su;v bSu;v

2 2

ðu;iÞ obs: ðu;vÞobs:

þ k jjPjj2F þ jjQjj2F þ jjZjj2F ; ð14Þ

where jj jjF denotes the Frobenius norm of the matrices, and k is the usual regularization parameter. Note that obs. means observed. As to optimize ranking, the training of this model can be modified as detailed in paper [53]. Analogous to Eq. (10), the training function is modified to account for all items (instead of RMSE on the observed ratings) for an improved top-k hit-ratio on the test data:

XX b 2 XX 2

Wu;i Rou&;i i Ru;i þ WðuS;vÞ u b

S ;v Su;v

allu alli allu allv

þ k jjPjj2F þ jjQjj2F þ jjZjj2F : ð15Þ

Rou&;i i equals the actual rating value of user u to item i if it is observed in the training data; otherwise the value Rou&;i i ¼ rm is imputed. The training weights are

Wu;i ¼ 1 if Ru;i observed; ð16Þ wm otherwise:

The term concerning the social network is analogous to the first term concerning the ratings. In particular, the absent or unobserved social links are treated analogous to the missing ratings in AllRank [9], i.e. the value sm with weight wðmSÞ is imputed. Like

Wu;i in (16), WðuS;vÞ is defined as follows:

ðSÞ ( 1 if Su;v observed;

Wu;v ¼ c ðSÞ ð17Þ wm otherwise;

where cP 0 determines the weight of the social network information compared to the rating data. Obviously, c ¼ 0 corresponds to the extreme case where the social network is ignored when learning the matrices P and Q. As c increases, the influence of the social network increases. The effect is that the user profiles Qu and Qv of two users u and v become more similar to each other if they are friends. While only positive social relationships are considered in the original model [45], we note that this model allows also for negative values of Su;v, representing distrust among users. This objective function can be optimized using the popular (stochastic) gradient descent method.

In paper [53], this modified training procedure was experimentally found to achieve higher top-k hit-ratios than the (modified) models outlined in the following sections. This is remarkable, because this model was found to perform rather poorly regarding the RMSE metric when compared to the models outlined in the following sections.

5.2. Social trust ensemble (STE) model

Recommendation with the Social Trust Ensemble (STE) was introduced in [46]. This approach is a linear combination of the basic matrix factorization approach [8] and a social network based approach.

The predicted ratings are obtained from a model comprising the matrices P 2 Ri0j0 and Qu0j0:

Rbu;i ¼ rm þ aQuP>i þ ð1 aÞ X Su;vQvP>i ; ð18Þ

v2Fþu

where Fþu is the set of user u’s direct friends. The predicted rating for item i by user u consists of three terms. The first two terms are the same as in the traditional CF: global offset rm and prediction based on user u and item i’s latent features. The last term is a weighted sum of the predicted ratings for item i from all of user u’s friends. It captures the social influence. The trade-off between the feedback data (ratings) and the influence from social network is determined by a 2 ½0; 1. Obviously, the social influence is ignored for a ¼ 1, while a ¼ 0 assigns the highest possible weight to the social influence. Intermediate values of a result in a weighted combination of the information from both sources. The training objective function to optimize RMSE is as follows:

|  |  |
| --- | --- |
| Ru;i Rbu;i þ k jjPjj2F þ jjQjj2F :  ðu;iÞobs:  Eq. (18) can be rewritten using matrix notation: | ð19Þ |
| Rb ¼ rm þ SaQP>; | ð20Þ |

X 2

where Sa ¼ aI þ ð1 aÞS, and I is the identity matrix. When the objective is ranking, in [53], the training function was modified as follows:

XXWu;i Rou&;i i Rbu;i2 þ kjjPjjF2 þ jjQjj2F; ð21Þ

allu alli

where jj jjF denotes the Frobenius norm. Wu;i and Rou&;i i are defined as in the previous section. Again, this training objective function can be optimized efficiently using stochastic gradient descent.

5.3. Social MF model

Social Matrix Factorization (SocialMF) was proposed in [44], and was found to outperform SoRec and STE with respect to RMSE. The SocialMF model addresses the transitivity of trust in social networks, as the dependence of a user’s feature vector on the direct neighbors’ feature vectors can propagate through the network, making a user’s feature vector dependent on possibly all users in the network (with decaying weights for more distant users). Each of the rows of the social network matrix S is normalized to 1, resulting in the new matrix Swith Su;v / Su;v, and PvSu;v ¼ 1 for each user u. The predicted ratings are obtained from the model, comprising the matrices P 2 Ri0j0 and Qu0j0, as follows:

Rb ¼ rm þ QP>: ð22Þ

The training objective function to optimize RMSE is as follows:

X Ru;i Rbu;i2 þ bXjjðQu X Su;vQvÞjj2

ðu;iÞobs: allu v2Fþu

2 2 þ k jjPjjF þ jjQjjF ; ð23Þ

where the second term in the objective function ‘‘forces’’ user u’s latent feature Qu to be similar to the (weighted) average of his/her friends’ profiles Qv (measured in terms of the square error). The tradeoff between the feedback data (ratings) and the social network information is controlled by bP 0. Obviously, the social network information is ignored for b ¼ 0, while increasing the value of b shifts the tradeoff more and more towards the social network information.

When the goal is ranking, in [53], the training function (23) was modified as follows as to better optimize the top-k hit ratio (instead of RMSE): XXW Ro&i Rbu;i2 þ bXjjðQu X Su;vQvÞjj2 u;i u;i

all u alli allu v2Fþu

2 2 þ k jjPjjF þ jjQjjF : ð24Þ

Also this modified training function can be optimized efficiently by means of (stochastic) gradient descent.

5.4. Similarity-based social regularization

Authors of [47] proposed social regularization as to incorporate social network information into the training procedure. They coined the term Social Regularization to represent the social constraints on recommender systems. For example, in the previous

SocialMF model, the social regularization part is

X X 2

b jjðQu Su;vQvÞjjF: ð25Þ

allu v2Fþu

Authors of [47] named Eq. (25) as average-based regularization–a user’s latent feature is constrained to be similar to the weighted average of whom he follows. They further proposed individual-based regularization:

X X 2

b simðu;vÞjjQu  QvjjF; ð26Þ

alluv2Fþu

where a user’s latent feature is constrained to be similar to his/her followees, weighted by their similarities. Similarity between users can be derived by calculating the Pearson Correlation Coefficient (PCC) or Vector Space Similarity (VSS) of commonly rated items between them. However, due to data sparsity, the number of commonly rated items between friends could be very small or even zero. To address this problem, authors of [48] improved the prediction accuracy by employing adaptive social similarities in the social regularization part. They calculate similarity between users based on their latent features. They demonstrated that latent feature based similarity function outperforms VSS and PCC similarity metric on Epinions [18] data set.

5.5. Circle-based recommendation

Similarity-based Social Regularization treats different friends differently. Circle-based recommendation further extends this idea, as only a subset of friends are taken into account when performing rating prediction in a specific circle. When applying circle-based recommendation to the SocialMF [44] model, one can build a separate MF model for each category. Using rating data only in category c and trust values in the corresponding circles CðucÞ, the training objective function to minimize RMSE becomes:

LðcÞðRðcÞ;QðcÞ;PðcÞ;SðcÞ Þ ¼ 12 X Ruðc;iÞ Rbuðc;iÞ2 þ 2b XjjðQðucÞ

ðu;iÞobs: allu

X ðcÞ ðcÞ 2 Su;v Qv Þjj v

k ðcÞ 2 ðcÞ 2

þ 2 jjP jjF þ jjQ jjF ; ð27Þ

where Rðuc;iÞ is the real rating of item i in category c; Rbðuc;Þi is the predicted rating for item i:

Rbðuc;iÞ ¼ rðmcÞ þ QuðcÞPiðcÞ>; ð28Þ

where they define the global bias term rðmcÞ as the average value of observed training rating in category c. The summation in Eq. (27) extends over all observed user-item pairs ðu; iÞ in category c. Note that this model only captures user and item profiles in category c, i.e., QðcÞ and PðcÞ. PðcÞ 2 Rið0cÞj0, where ið0cÞ is the number of items in category c and QðcÞ 2 Ru0j0.

As an alternative training objective function, one can also use rating data from all categories, instead of only the ratings in category c. The only difference from Eq. (27) is that the first line is replaced by

 X b 2

Ru;i Ru;i ; ð29Þ

ðu;iÞobs:

where the summation extends over all observed user-item pairs ðu; iÞ from all categories. They train a separate model for each category c, i.e., QðcÞ and PðcÞ, with PðcÞ 2 Ri0j0, and QðcÞ 2 Ru0j0. The difference from Eq. (28) is to substitute rðmcÞ by rm, which is the average value of all observed ratings in the training set.

1. Neighborhood based social recommendation approaches

Neighborhood based approaches use the stored ratings directly in the prediction/recommendation. We first review some social network traversal (SNT) based approaches which traverse the source-user’s neighborhood in the social network and query the rating of the target item. Then, we review some of the nearestneighborhood based approaches [40,53], which combine the traditional CF neighborhood with social neighborhood.

* 1. Social network traversal based approaches

Given a social network, some RS algorithms predict a user’s rating for an item by traversing the user’s neighborhood and querying the item ratings of her direct and indirect friends. We call them Social Network Traversal (SNT) based approaches.

6.1.1. Trust weighted prediction

Trust has recently been identified as an effective means to utilize social network information as to improve recommendation accuracy. Empirical studies in [28,29] found a correlation between trust and user similarity. Various techniques have been proposed to incorporate trust into CF approaches [26,27,30–37]. For instance, [30] attempts to address the rating sparsity issue using the trust relationship. It was shown that even simple binary trust relations can increase the coverage and thus the number of recommendations that can be made. [33] investigates the use of trust to better cluster users, thus improving recommendation accuracy. Typically, the rating similarity between friends is quantified by a numerical value, with larger values indicating higher levels of trust. Then recommendations are calculated for a user as a function of the ratings and the associated trust values of his friends.

MoleTrust [32] is such a SNT approach. Users are connected in a trust network, where the trust relationship is explicitly issued by users. MoleTrust considers all raters up to a maximum-depth given

as input. Maximum-depth is independent of any specific user and item. Also, to compute the trust value between indirectly connected users u and v in MoleTrust, backward exploration is performed. The trust value from u to v is the aggregation of trust values between user u and users directly trusting v weighted by the direct trust values:

P Sw;vSu;w

Su;v ¼ Pw2Fv Sw;v : ð30Þ

w2Fv

Only users within maximum-depth, and which have rated the target item, are considered. We denote this set by T. The rating prediction for target user u of item i in MoleTrust [32] is calculated as:

bu;i ¼ Ru þ Pv2TPSu;vðRSvu;i;v RvÞ; ð31Þ

R

v2T

where Su;v is user u0s trust of user v, and Ru is user u0s average rating.

Golbeck designed a trust metric called TidalTrust [27], working in a breadth-first search fashion. TidalTrust works as follows: First, the system searches for raters that the source-user knows directly. If there is no direct connection from the source to any rater, the system moves one step out to find connections from the source to raters that are two hops away. This process repeats until a path is found. The opinions of all raters at that depth are considered. Second, using TidalTrust, the trust value is calculated for each rater at the given depth. As to infer the trust value of user u to v, who are not directly connected, TidalTrust aggregates the trust value from u’s direct neighbors to v, weighted by the direct trust values from u to its direct neighbors:

Pw2Fþu Su;wSw;v

Su;v ¼ P þ Su;w : ð32Þ

w2Fu

Once the raters have been selected, the rating prediction is calculated as the weighted average of all raters’ ratings:

bu;i ¼ PP SuS;vuR;vv;i

R v2T ; ð33Þ

v2T where T is the set of raters within maximum-depth.

6.1.2. Bayesian inference based prediction

Different from trust-based recommendation, authors of [38] proposed to use conditional probability distributions to capture the similarity between friends in social networks. Probability distributions carry richer information than trust values, and allow one to employ Bayesian networks to conduct multiple-hop recommendation in online social networks.

In [38], each pair of friends ðu;vÞ measures their rating similarity by a set of conditional distributions pðujvÞ and pðvjuÞ, each of which is one user’s rating distribution given the other user’s ratings. pðujvÞ is calculated by taking out the commonly rated items between user u and v, then given user v0s rating value, we calculate the rating distribution of user u on the commonly rated items. When a user wants a recommendation rating for an item, he sends out a rating query to his direct friends in the social network. Upon receiving a query for an item, a user returns its rating if he has rated the item before; otherwise the query is relayed to her friends. As to avoid loops, a user only responds to the first query from a requester, and ignores the following queries relayed through other paths. A query will be dropped after a pre-defined number of hops as to limit the range of query flooding. Fig. 5 depicts the recommendation Bayesian tree that consists of three types of users: (i) query initiator S at the root, (ii) recommenders fLig, who have rated the item and respond to the query with their ratings. They are leaves in the tree; (iii) intermediate

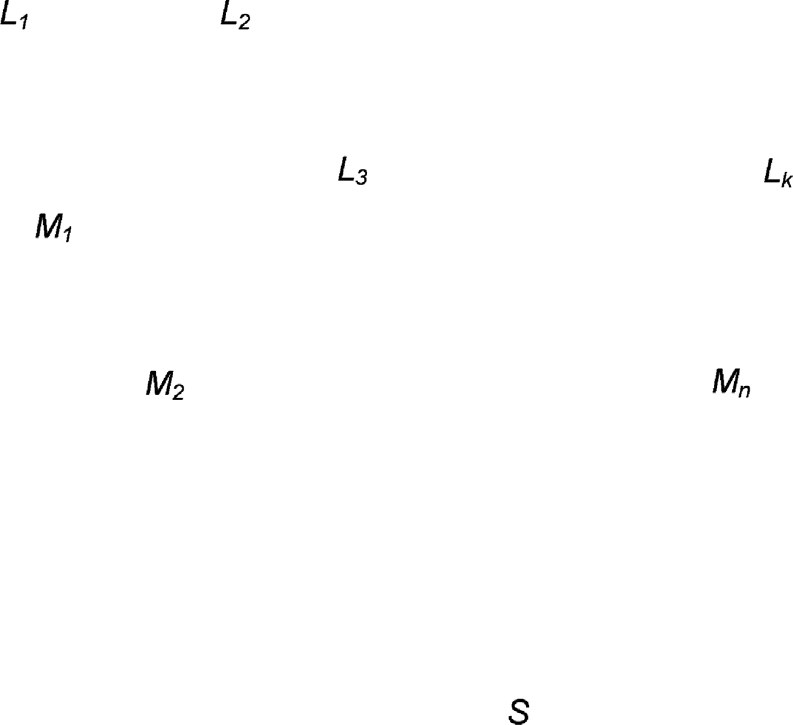


Fig. 5. Recommendation Propagation Tree as a Bayesian network.

users fMjg, who forward and aggregate queries and responses between the initiator and the raters. Specifically, an intermediate user collects the recommendation information from his children, aggregates the information according to Bayesian calculation, and relays the aggregated information to his parent. Finally, given the structure of the recommendation propagation tree and the ratings on all leaf users, the querying initiator computes the final recommendation rating.

6.1.3. Random walk based approaches

Some other social RS algorithms employ random walks in online social networks as to compute recommendation ratings [39– 41]. Authors of [39] proposed the so-called TrustWalker, which performs a random walk in online social networks as to query a user’s direct and indirect friends’ ratings for the target item as well as similar items. Since both ratings from similar users and ratings of similar items are considered, TrustWalker is a combination of the trust-based approach and item-item similarity based approach. Item-item similarity can be calculated using user rating information or item content information. Specifically, TrustWalker consists of two major components: random walk in the trust network and probabilistic item rating selection on each visited node. During the random walk, a user’s direct and indirect friends are visited in the trust network. Whenever a friend is visited, if she has rated the target item, her rating is logged; if she has not rated the target item, but has rated an item similar to the target item, her rating is logged with certain probability. The probability of using a rating of a similar item in place of a rating for the target item increases as the length of random walk increases. This probabilistic item rating selection aims to avoid going too deep in the network when no user in a close neighborhood has rated the target item.

They employ the Pearson Correlation Coefficient of ratings expressed for two items to calculate the similarity value between them,

P C ðRu;i RuÞRu;j Ru u2U

corrði;jÞ ¼ ~~qffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffiffi~~Pu2UC Ru;i Ru2Pu2UC Ru;j Ru2 ; ð34Þ

where UC is the set of users who have rated both i and j; Ru;i and Ru;j

are ratings of u assigned to items i and j respectively. Ru is the average rating issued by user u. Values of the Pearson correlation are in the range of ½1; 1. Only items with positive correlation with the target item are considered. The similarity value is then calculated as:

1

simði;jÞ ¼ jUcj corrði;jÞ; ð35Þ

1 þ e 2

where jUcj is the number of users who rated both i and j. TrustWalker improves the prediction precision by preferring raters within a shorter distance and improves the coverage by considering ratings for similar items in addition to the target item.

The same authors extended TrustWalker to recommend top-k items for a source user u in [40]. Starting from user u, a random walk is performed in the trust network. Each random walk stops at a certain user. Then the items rated highly by that user will be considered as the recommended items, ordered according to the ratings expressed by that user. Several random walks are performed to gather more information and compute a more confident recommendation rating. The estimated rating of each item is the average of ratings for that item over all sampled raters. At the end, items with the highest estimated ratings are chosen as top-k recommended items.

6.2. Nearest neighbor (NN) methods

A NN based algorithm works by identifying the so-called neighbors of a source-user, a prediction of item preferences or a list of recommended items for him or her can be produced. In CF-based social RSs, a NN approach combines the traditional CF neighborhood with social neighborhood.

Authors of [40] proposed an approach, namely Trust-CF, to incorporate social network into Nearest Neighbor (NN) based top-k recommender systems. In Trust-CF, Breadth First Search (BFS), starting from a source user u, is performed to traverse the social network multiple times to obtain a set of trusted neighbors, namely trusted neighborhood. Meanwhile, it constructs a collaborative filtering (CF) neighborhood, consisting of users who are close to the source user u in terms of the Pearson Correlation Coefficient (PCC), which is obtained by computing the Pearson Correlation Coefficient of ratings from two users on their commonly rated items. The items rated highly by users in either neighborhoods are considered to be candidates for top-k recommendation. Trust-CF calculates the predicted rating for a candidate item as the weighted average of all observed ratings in the two neighborhoods. The weight for a user in the trusted neighborhood is set to 1=dv, where dv is the depth of user v from user u in the trust network. The weight for a user in the CF neighborhood is the Pearson Correlation Coefficient between this user and the source-user. If an item has predicted ratings from both neighborhoods, two predicted ratings are combined using weighted average with weights proportional to the neighborhood size for this item. Finally, TrustCF sorts all the candidate items by their predicted ratings and recommends the top-k items to the source-user.

Authors of [53] proposed Trust-CF-ULF as to incorporate social network information into top-k recommender systems. The TrustCF-ULF approach is the combination of a user latent feature space based collaborative filtering approach (CF-ULF) and a social network based approach. CF-ULF uses MF (i.e., AllRank [9]) to obtain the user latent features. The users are then clustered in the user latent feature space using the Pearson Correlation Coefficient. The k1 users nearest to the source user u are identified. Then they find k2 closest neighbors from the trust neighborhood which are not in the k1 set. Later on, users in the combined neighborhood vote for their relevant items. The weight for a user in the trusted neighborhood is the same as in the Trust-CF approach. The weight for a user in the

CF-ULF neighborhood is the Pearson Correlation Coefficient between this user and the source-user in the user latent feature space. Finally, Trust-CF-ULF sorts all the candidate items by their received voting values, and recommends the top-k items to the source user.

Table 3

Comparison of matrix factorization based social recommendation approaches for item rating prediction task.

Prediction

Representative approaches

Accuracy Training complexity

|  |  |  |
| --- | --- | --- |
| SoRec | High | Oðu0ðr þsÞj0KÞ |
| STE | High | Oðu0rs2j0KÞ |
| SocialMF | High | Oðu0ðr þs2Þj0KÞ |
| Social Regularization | High | Oðu0rsj0KÞ |
| CicleRec | High | Oðu0ðr þ sc2c0Þj0KÞ |
| Table 4 list recommendation task. | Comparison of matrix factorization based social recommendation approaches for ite | m |

Representative approaches Prediction

Accuracy

Training complexity

|  |  |  |
| --- | --- | --- |
| SoRec | High | Oðu0ðr þs þ j0Þj20KÞ |
| STE | High | Oðu0ði0s2j0 þrj20 þ j30ÞKÞ |
| SocialMF | High | Oððu0ðs2j0 þrj20 þ j30ÞKÞ |
| Social Regularization | N/A | N/A |
| CicleRec | N/A | N/A |

7. Approach comparison

Now we make a comparison of the surveyed approaches and provide a high-level summary. The comparison is focused on model-training complexity and accuracy. For the comparison of training complexity, we only outline the complexity of model-based approaches, i.e., MF-based approaches in our paper. Model complexity is the complexity of learning the model parameters during the training step. We focus on two recommendation tasks here, one is the item rating prediction task, and the other one is the item list recommendation task. Model complexity is bound to a specific optimization method. The optimization method used for the training rating prediction models is gradient descent. The optimization method used for training the item list recommendation models is alternating least squares. The accuracy metric in the rating prediction task is RMSE/MAE and the accuracy in item list recommendation task is the top-k hit ratio.

First, we compare the MF-based approaches. The comparison of different MF-based approaches surveyed in this paper is depicted in Table 3 and Table 4 for the item rating prediction task and the item list recommendation task separately. In addition to the notations introduced in Section 3, we introduce some new notations here. r is the average number of ratings per user and s is the average number of friends per user in the social network. sc is the average number of friends per user in a social circle, c0 is the number of circles in the social network and K is the number of iterations needed for the training of the model to converge. Among MF-based approaches, we do not report the complexity and accuracy of Social Regularization and CircleRec in the recommendation task as there is no existing work on employing them for the top-k recommendation task. For model training complexity calculation, readers can refer to model’s original papers for details.

Then we make a comparison of neighborhood based social recommendation approaches. In the item rating prediction task, accuracies of MoleTrust and TidalTrust are Low, and accuracies of Bayesian Inference and Random Walk are Medium. In the item list recommendation task, Trust-CF’s accuracy is Medium and TrustCF-ULF’s accuracy is High. Among neighborhood based approaches, the accuracies of Trust-CF and Trust-CF-ULF in the item rating prediction task are not available. The accuracies of MoleTrust, TidalTrust, Bayesian Inference and Random Walk in the item list recommendation task are not available.

We can see that, generally, model based approaches perform well in both item rating prediction and item list recommendation tasks (if available), while neighborhood based approaches enjoy the advantage of easy implementation.

8. Conclusions

In this paper, we presented a survey of CF-based social recommender systems. We first gave a short overview of the task of recommender systems and the traditional recommendation algorithms. We then presented how social network information can be adopted by recommender systems as additional input for improved accuracy. We classify CF-based social recommender systems into two categories: matrix factorization based social recommendation approaches and neighborhood based social recommendation approaches. Both types of approaches are surveyed and compared.

Current work on social recommender systems has demonstrated the effectiveness of incorporating social network information to improve recommendation accuracy. Given the increasing popularity of online social networks, new recommendation algorithms will be needed to better mine various kinds of newly available social information. Most of the surveyed algorithms are trained and tested offline. One of the next steps will be to test and improve their performance in real online social networks, with real-time user experience feedback. Finally, privacy in online social networks has attracted more and more user awareness. Privacypreserving social recommender systems are another interesting direction for future work.

References

1. G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, IEEE Transactions on Knowledge and Data Engineering 17 (6) (June 2005), [http:// dx.doi.org/10.1109/TKDE.2005.99.](http://dx.doi.org/10.1109/TKDE.2005.99)
2. F. Ricci, L. Rokach, B. Shapira, P.B. (Eds.), Kantor Recommender Systems Handbook, first ed., 2011, XXX, 842 p. 20 illus
3. [X. Su, T.M. Khoshgoftaar, A survey of collaborative filtering techniques, Advances in Artificial Intelligence 2009 (2009) 19. Article ID 421425](http://refhub.elsevier.com/S0140-3664(13)00172-2/h0010).
4. M. Jahrer, A. Toscher, R. Legenstein, Combining predictions for accurate recommender systems, in: Proc. of KDD ’10, Washington, DC, USA, 2010, pp. 693–702.
5. [R. Keshavan, A. Montanari, S. Oh, Matrix completion from noisy entries, Journal of Machine Learning Research 11 (2010) 2057–2078](http://refhub.elsevier.com/S0140-3664(13)00172-2/h0030).
6. Y. Koren, Factorization meets the neighborhood: a multifaceted collaborative filtering model, in: ACM Conference on Knowledge Discovery and Data Mining, 2008, pp. 426–34.
7. Y. Koren, Collaborative filtering with temporal dynamics, in: Proc. of KDD ’09, Paris, France, 2009, pp. 447–456.
8. R. Salakhutdinov, A. Mnih, Probabilistic matrix factorization, in: NIPS 2008, vol. 20.
9. H. Steck, Training and testing of recommender systems on data missing not at random, in: ACM Conference on Knowledge Discovery and Data Mining, 2010, pp. 713–722.
10. Y. Hu, Y. Koren, C. Volinsky, Collaborative filtering for implicit feedback datasets, in: IEEE International Conference on Data Mining (ICDM), 2008.
11. Y. Xin, H. Steck, Multi-value probabilistic matrix factorization for IP-TV recommendations, in: ACM Conference on Recommender Systems (RecSys), 2011.
12. R. Pan, Y. Zhou, B. Cao, N. Liu, R. Lukose, M. Scholz, Q. Yang, One-class collaborative filtering, in: IEEE International Conference on Data Mining (ICDM), 2008.
13. A. Paterek, Improving regularized singular value decomposition for collaborative filtering, in: KDDCup, 2007.
14. R. Salakhutdinov, A. Mnih, G. Hinton, Restricted Boltzmann machines for collaborative filtering, in: International Conference on Machine Learning (ICML), 2007.
15. S. Deerwester, S. Dumais, G. Furnas, R. Harshman, T. Landauer, K. Lochbaum, L. Streeter et al., Latent semantic analysis/ indexing, [<http://lsa.colorado.edu/>.](http://lsa.colorado.edu/)
16. [D.D. Lee, H.S. Seung, Learning the parts of objects by non-negative matrix factorization, Nature 401 (1999) 788–791](http://refhub.elsevier.com/S0140-3664(13)00172-2/h0040).
17. YouTube, available at [<http://www.youtube.com>.](http://www.youtube.com/)
18. Epinions, available at [<http://www.epinions.com/>.](http://www.epinions.com/)
19. NetflixPrize, available at [<http://www.netflixprize.com/>.](http://www.netflixprize.com/)
20. Twitter, available at [<http://twitter.com>.](http://twitter.com/)
21. Facebook, [<http://www.facebook.com>.](http://www.facebook.com/)
22. Google Plus, [<http://plus.google.com/>.](http://plus.google.com/)
23. Flixster, [<http://www.flixster.com>](http://www.flixster.com/).
24. S. Funk, Netflix update: try this at home, 2006, [<http://sifter.org/simon/ journal/20061211.html>.](http://sifter.org/simon/journal/20061211.html)
25. M. McPherson, L. Smith-Lovin, J.M. Cook, Birds of a Feather: homophily in social networks, Annual Review of Sociology 27 (1) (2001) 415–444, [http:// dx.doi.org/10.1146/annurev.soc.27.1.415.](http://dx.doi.org/10.1146/annurev.soc.27.1.415)
26. J.A. Golbeck, Computing and applying trust in web-based social networks, PhD. Dissertation, 2005.
27. J. Golbeck, J. Hendler, FilmTrust: movie recommendations using trust in webbased social networks, in: Proceedings of the IEEE Consumer Communications and Networking Conference, Jan. 2006, pp. 282–286.
28. C.-N. Ziegler, G. Lausen, Analyzing correlation between trust and user similarity in online communities, in: Proceedings of Second International Conference on Trust Management, 2004, pp. 251–265.
29. C.-N. Ziegler, J. Golbeck, Investigating interactions of trust and interes[t](http://refhub.elsevier.com/S0140-3664(13)00172-2/h0050) similarity, Decision Support Systems 43 (2) (2007) 460–475. http:/[/ dx.doi.org/10.1016/j.dss.2006.11.003](http://refhub.elsevier.com/S0140-3664(13)00172-2/h0050).
30. P. Massa, B. Bhattacharjee, Using trust in recommender systems: an experimental analysis, in Proceedings of iTrust2004 International Conference, 2004, pp. 221–235.
31. P. Massa, P. Avesani, Trust-aware recommender systems, in: Proceedings of the 2007 ACM Conference on Recommender Systems, 2007.
32. P. Massa, P. Avesani, Controversial users demand local trust metrics: an experimental study on epinions.com community, in: Proceedings of the 25th American Association for Artificial Intelligence Conference, 2005.
33. T. DuBois, J. Golbeck, J. Kleint, A. Srinivasan, Improving recommendation accuracy by clustering social neworks with trust, in: Proceedings of the ACM RecSys 2009 Workshop on Recommender Systems and the Social Web, Oct. 2009.
34. J. O’Donovan, B. Smyth, Trust in recommender systems, in: Proceedings of the 10th International Conference on Intelligent User Interfaces, 2005.
35. F. Walter, S. Battiston, F. Schweitzer, A model of a trust-base[d](http://refhub.elsevier.com/S0140-3664(13)00172-2/h0055) recommendation system on a social network, Autonomous Agents an[d Multi-Agent Systems 16 (1) (2007) 1573–7454](http://refhub.elsevier.com/S0140-3664(13)00172-2/h0055).
36. K. Sarda, P. Gupta, D. Mukherjee, S. Padhy, H. Saran, A distributed trust-based recommendation system on social network, in: Proceedings of the 10th Second IEEE Workshop on Hot Topics in Web Systems and Technologies, 2008.
37. P. Bedi, H. Kaur, S. Marwaha, Trust based recommender system for semantic web, in: Proc. of IJCAI ’07, 2007, pp. 2677–2682.
38. X. Yang, Y. Guo Y. Liu, Bayesian-inference based recommendation in online social networks, in: IEEE Transactions on Parallel and Distributed Systems (TPDS), 2013, pp. 642–651.
39. M. Jamali, M. Ester, Trustwalker: a random walk model for combining trustbased and item-based recommendation, in: ACM Conference on Knowledge Discovery and Data Mining (KDD), 2009.
40. M. Jamali, M. Ester, Using a trust network to improve top-n recommendation, in: ACM Conference on Recommender Systems (RecSys), 2009.
41. M.A. Tayebi, M. Jamali, M. Ester, U. Glasser, R. Frank, CrimeWalker: a recommendation model for suspect investigation, in: RecSys ’11 Proceedings of the fifth ACM conference on Recommender systems, 2011.
42. [B. Sarwar, G. Karypis, J. Konstan, J. Reidl, Item-based collaborative filtering](http://refhub.elsevier.com/S0140-3664(13)00172-2/h0060) recommendation algorithms, in: WWW’ 01: Proceedings of the 10t[h International Conference on World Wide Web, ACM, New York, NY, USA, 2001, pp. 285C–295](http://refhub.elsevier.com/S0140-3664(13)00172-2/h0060).
43. [K. Jarvelin, J. Kekalainen, Cumulated gain-based evaluation of ir techniques, ACM Transactions on Information Systems 20 (4) (2002) 422C–446. http:// doi.acm.org/10.1145/582415.582418.](http://refhub.elsevier.com/S0140-3664(13)00172-2/h0065)
44. M. Jamali, M. Ester, A matrix factorization technique with trust propagation for recommendation, in: Social Networks Proceedings of the 2010 ACM conference on Recommender systems(RecSys), 2010.
45. H. Ma, H. Yang, M.R. Lyu, I. King, Sorec: social recommendation using probabilistic matrix factorization, in: International Conference on Information and Knowledge Management (CIKM), 2008.
46. H. Ma, I. King, M.R. Lyu, Learning to recommend with social trust ensemble, in: ACM Conference on Research and Development in Information Retrieval

(SIGIR), 2009.

1. H. Ma, D. Zhou, C. Liu, M.R. Lyu, I. King, Recommender systems with social regularization, in: ACM International Conference on Web Search and Data Mining (WSDM), 2011.
2. L. Yu, R. Pan, Z. Li, Adaptive social similarities for recommender systems, in: RecSys ’11 Proceedings of the Fifth ACM Conference on Recommender Systems, 2011.
3. [F. Liu, H.J. Lee, Use of social network information to enhance collaborative filtering performance, Expert System Applications 37 (7) (2010) 4772–4778.](http://refhub.elsevier.com/S0140-3664(13)00172-2/h0070)
4. S.-H. Yang, B. Long, A.J. Smola, N. Sadagopan, Z. Zheng, H. Zha, Like alike: joint friendship and interest propagation in social networks, in: Proceedings of the 20th International Conference on World Wide Web (www2011).
5. J. Noel, S. Sanner, K.-N. Tran, P. Christen, L. Xie, E. Bonilla, E. Abbasnejad, N.D. Penna, New objective functions for social collaborative filtering, in: Proceedings of the 21st International Conference on World Wide Web (www2012).
6. X. Yang, H. Steck, Y. Liu, Circle-based recommendation in online social networks, in: Proceedings of 2012 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD’12).
7. X. Yang, H. Steck, Y.Guo, Y. Liu, On Top-k recommendation using social networks, in: ACM Conference on Recommender Systems (RecSys’12), 2012.
8. [Q. Yuan, S. Zhao, L. Chen, Y. Liu, S. Ding, X. Zhang, W. Zheng, Augmenting](http://refhub.elsevier.com/S0140-3664(13)00172-2/h0075) collaborative recommender by fusing explicit social relationships[, Recommender Systems & the Social Web workshop, RecSys, 2009](http://refhub.elsevier.com/S0140-3664(13)00172-2/h0075).
9. V. Vasuki, N. Natarajan, Z. Lu, I. Dhillon, Affiliation recommendation using auxiliary networks, in: ACM Conference on Recommender Systems

(RecSys’2010), 2010.

1. P. Symeonidis, E. Tiakas, Y. Manolopoulos, Product recommendation and rating prediction based on multi-modal social networks, in: Proceedings of the 5th ACM Conference in Recommender Systems (RecSys’2011), Chicago, IL, 2011.
2. S.M. McNee, J. Riedl, J.A. Konstan, Being accurate is not enough: how accuracy metrics have hurt recommender systems, in: Proceeding CHI EA ’06 CHI ’06 Extended Abstracts on Human factors in Computing Systems.
3. [J.L. Herlocker, J.A. Konstan, L.G. Terveen, J.T. Riedl, Evaluating collaborative filtering recommender systems, ACM Transactions on Information Systems 22 (1) (2004) 5–53](http://refhub.elsevier.com/S0140-3664(13)00172-2/h0080).