**基于自适应的近邻选择机制的社交推荐方法**

Sajad Ahmadian, Majid Meghdadi ∗, Mohsen Afsharchi

摘要：推荐系统是一种为用户提供个性化推荐的技术。在电商网站和在线共享社区中提供高质量的推荐是一个很重要的问题，可以帮助用户在选择商品时做出有效的选择。协同过滤是一种很重要的推荐系统，它基于评级或者使用模式（如购买）为用户提供特定的推荐产品。然而，用户的预测评级质量和近邻选择是推荐系统中的重要问题。为用户选择合适的近邻集可以提高推荐过程中评级预测的准确性。在本论文中，提出了一种基于自适应近邻选择机制的街角推荐方法。该方法首先采用了聚类算法计算用户的初始近邻集。在这个步骤中，将用户之间的历史评级和社交信息组合起来形成用户的初始近邻集。然后，将这些近邻集用来预测不可见项目的初始评级。此外，使用基于用户之间的历史评级和社交信息的可靠性度量来评估初始预测评级的质量。接着，提出了一个直线度模型，该模型从用户的初始近邻中剔除掉无用的用户，并为用户生产一个新的自适应近邻集。最后，使用用户的最新调整近邻集来预测未知的新评级，并将最感兴趣的项重新记录到当前用户中。在三个真实数据集上的实验结果表明了该方法明显的优于其他几种最先进的推荐算法。

关键词：推荐系统，自适应的近邻选择机制，置信，可靠性，信任

1 介绍

近些年，互联网上的数据量以前所未有的速度在增长，信息超载给互联网带来了困难。因此，用户在可选项中很难找到有用的信息。推荐系统（RSs）用于为用户提供个性化的推荐来面对信息过载的问题。换言之，RSs帮助用户从大量的可选项中找到感兴趣的条目（如书籍，电影，新闻等）。为了通过RSs提供高质量的推荐结果，RSs需要预测以及比较产品的效用，然后根据这个来判断推荐那些产品( Ricci, Rokach, Shapira, & Kantor, 2011 )。

协同过滤（CF）是推荐系统算法中的一种，它用于利用过去的行为或现有用户的意见信息来预测未评级的项目并找到当前系统用户最有可能喜欢或者感兴趣的项目。基于CF的方法的基本思想是，如果用户在过去有着相同的兴趣，那么他们在未来同样也会有相近的选择( Cechinel, Sicilia, Alonso, & Barriocanal, 2013; Domingues, Jorge, & Soares, 2013; Park, Park, Jung, & Lee, 2015 )。在这个方法中，使用给定的用户-项目评分矩阵作为输入来计算用户之间的相似度。此外，这些方法利用相似值来确定一组用户之间的差异，并形成当前用户的近邻集。因此，在这些方法中，使用当前用户的近邻集中的用户意见为当前用户提供适合的推荐。基于CF的算法通常分为两大类，一类为基于内存的方法，一类为基于模型的方法。在基于内存的方法中，原始用户-项目评级矩阵存储在内存中，直接用于查询。与之相反，在基于模型的方法中，原始数据经过基于项目的过滤或某些降维方法利息处理后，再使用预先计算或学习模型在需要进行预测时运行。在其他文献中提出了一些基于模型的方法，包含了概率模型( Javari & Jalili, 2015; Ma, Zhang, Liu, Li, & Yuan, 2016 )，聚类模型( Bilge & Polat, 2013; Tsai & Hung, 2012 )，降维模型( Hernando, Bobadilla, & Ortega, 2016; Hong, Zheng, & Chen, 2016 )，模式挖掘技术( Tsai & Lai, 2015 )，潜在语义模型( Hofmann, 2004 )，马尔科夫决策过程模型( Shani, Brafman, & Heckerman, 2005 )。尽管这些基于内存的方法相较于基于模型的方法使用更加广泛，但这些方法面临着数以千万用户以及数以百万项的系统的可伸缩性问题( Ricci et al., 2011 )。

基于CF的方法通常有着几个缺点导致系统的性能下降。这些方法使用用户-评级矩阵来计算用户之间的相似度，以形成当前用户的近邻集( Kaššák, Kompan, & Bieliková, 2016; Liu, Hu, Mian, Tian, & Zhu, 2014 )。因此，基于CF的方法需要大量包含用户对大多数项目评分的数据来做出有效的推荐。然而，在真正的应用系统中，用户-评级矩阵往往是非常稀疏的，这表明了用户通常只为少数的项目提供评价信息。这个问题被称为数据稀疏性问题，它降低了基于CF的方法找出当前用户近邻集的性能。此外冷启动问题可以看成是数据稀疏性问题的一个特例。这个问题是关于那些还没有对任何项目进行评价的新用户或已经对一些项目进行评价的用户。另外，推荐的项目未经评级或购买是基于CF的方法的另一个挑战，因为这些项目无法获得足够多的反馈信息。恶意攻击是降低基于CF的方法系统可靠性的另一个问题( O’Mahony, Hurley, & Silvestre, 2005 )。恶意攻击发生在当用户试图故意地去影响系统功能时。换言之，恶意的用户可能会试图通过在推荐列表中经常包含某个项目的方式来影响推荐系统的行为。

为了解决上述提到的基于CF方法的问题，文章提出了几种方法。信任感知推荐系统旨在利用用户信任网络中的社交信息来提高基于CF的推荐方法的系统性能( Fang, Guo, & Zhang, 2015; Kim & Phalak, 2012; Yan, Zheng, Chen, & Wang, 2013 )。在这些系统中，可将信任近邻的评价作为起点，用于预测未评级的项目并向用户提出建议。因此，这些系统能够缓解冷启动问题，提高用户覆盖率。此外，用户之间的信任声明有助于增强推荐系统对恶意攻击的鲁棒性。由此，期望的信任关系到一个虚假的信息不能轻易的注入到一个推荐系统的数据库中。

用户选择近邻是推荐系统中最重要的问题之一，它对系统预测评级的准确性有很大的影响。大多数的推荐算法使用用户的已识别近邻集来预测未知的项目，并向用户提供推荐。但是，这些用户的近邻可能并不能预测所有未知的项目，从而导致了推荐过程中预测评级的准确性降低。本文提出了一种基于自适应近邻选择机制的社交推荐方法，以提高评级预测的准确性，解决推荐系统中近邻的选择问题。为此，采用了基于每对用户之间的相似值和信任信息组合的聚类算法来计算用户的初始近邻集。然后，利用用户的初始近邻集来预测未知项的初始评级，并根据用户之间的相似值和信任声明的可靠性度量来预估预测评级的质量。在此基础上，提出了一种自适应的近邻选择机制，为初始近邻集可靠性较低的用户计算新近邻集来预测目标项。为了这一目标，文章提出了一种新的用户间的置信度模型，用于从用户的初始近邻集中识别和移除无用用户。最后，使用用户的新调整近邻集来预测未知项的最终评级，并将用户最感兴趣的项目推荐给当前用户。

本文的剩余部分组织如下：相关的研究内容将在第二节中进行综述。第三节中介绍提出的方法。在第四节中，通过对三个已知的数据集的多次试验，将所提出的算法与目前最先进的算法进行比较。最后，在第五部分讨论总结性的意见。

2 相关工作

大多数的协同过滤算法使用了二维矩阵，其中包含了用户对某些特定项目的评分。然而，在现实应用中，包含这些信息的矩阵不足以让推荐系统为用户提供准确的推荐( Lee, Kahng, & Lee, 2015; Moon, Kim, & Ryu, 2013 )。因此，增加社交因子或信任信息等可以提高推荐系统的性能，这类系统被称为信任感知推荐系统( Massa & Avesani, 2007 )。在信任感知方法中，信任信息可以分为显式信任和隐式信任两类。在显式方法中，基于社交网络中用户之间预先构建社交联系，能显式的收集用户的信任信息( Bedi & Vashisth, 2014; Yan et al., 2013 )。反之，在隐式方法中，信任信息可以隐式的从用户的评级信息中推断出来( Alahmadi & Zeng, 2015 )。

在Guo，Zhang，and Thalmann(2014)中，提出了一种新的方法，将社交信任信息纳入到用户的推荐中。为此，对给定的用户的可信近邻的评级进行合并用来补充和表示用户的偏好，并找到具有类似偏好的其他用户。且以评级数量和好评与差评的比例作为衡量合并评级质量的置信度。在Deng，Huang，and Xu(2014)中，提出了一种基于矩阵分解方法的推荐算法用来评估社交网络中用户之间的信任程度。此外，该方法还提出了一种扩展的随机游走算法来获得推荐结果。Yang，Guo，and Liu(2013)提出了一种基于贝叶斯推理的在线社交网络算法，该算法通过一系列用户的相关评级历史记录的条件概率衡量用户之间的评级相似度。

此外，将社交信息纳入到局域矩阵因子分解的推荐算法中，有助于提高这些系统在为目标用户预测未知项的能力。Yang，Lei，Liu，and Liu(2013)提出了一种改进协同过滤的方法，该方法通过整合两个稀疏信息，包含用户给出的历史评级数据和相同用户之间的社交信任网络。该方法利用了矩阵分解技术，将用户信任信息映射到低维的潜在特征空间中，更合理地反映用户对自己观点的交互影响。Jamali and Ester(2010)提出了一种基于矩阵分解技术的模型社交网络推荐方法。为此，将信任传播机制引入到基于矩阵法的推荐方法中。信任传播机制是社交推荐算法中的一个重要现象。在Guo, Zhang, and Smith (2015b)中，提出了一种基于信任的矩阵分解算法，该算法基于用户信任和项目评级的显性和隐性影响。作者指出，在推荐算法中不仅要考虑评级和信任的显性影响，还要考虑评级和信任的隐性影响。因此，他们提出了一种方法，通过进一步合并可信用户对当前用户的项目预测的显式和隐式影响，来发掘评级项的显式和隐式影响。

聚类算法可以在推荐系统中用于将用户/项目分到多个集群中。此外，推荐系统需要在考虑用户/项目所属的同一集群中的用户/项目的情况下进行预测。Pereira and Hruschka(2015)在已有的聚类算法的基础上，提出了一种最小化系统退化的混合方法。该方法将协同过滤与人口结构信息相结合，解决了新用户无法得到评级的冷启动问题。此外，在Birtolo and Ronca(2013)中，作者提出了一个包含两种基于聚类的协同过滤算法的聚类框架。其中一种算法(IFCCF)是基于系统项的模糊聚类，另一种算法(TRACCF)是基于系统用户的信任感知聚类协同过滤。在Guo, Zhang, and Smith (2015a)中，提出了一种多视图聚类方法，该方法从评级模式和社交信任关系的角度对用户进行了迭代聚类。Moradi、Ahmadian and Akhlaghian(2015)提出了一种基于全新的图聚类算法和用户间信任声明的推荐方法。为此，我们将识别出的集群作为一组近邻，用以预测未知项并向当前用户提出建议。此外，每个集群的初始中心是自动确定的，因此，该算法不需要设置预定义的集群数量。

在上述所有推荐方法中，都使用了静态近邻选择机制来预测当前用户未知项的评级。换句话说，用户的初始近邻集直接用于预测未知项的评级，而不考虑这些近邻集中每个用户的有效性。然而，为用户构建的初始近邻集可能并不能预测所有的未知项，从而降低了推荐过程中预测评级的准确性。本文提出的方法的主要贡献是利用自适应近邻选择机制来提高预测评级的准确性，解决了推荐系统中的近邻选择问题。为此，使用用户的初始近邻集来预测未知项的初始评级，然后使用文章提出的可靠性度量来评估预测评级的质量。并在此基础上，提出了一种基于用户间置信度模型的自适应近邻选择机制，用来为那些在初始近邻集在预测目标项是可靠性较低的用户计算一个新的近邻集。

3 提出的方法

在这一部分中，我们提出了一种新的推荐系统——基于自适应近邻选择的数据推荐算法。该方法首先使用聚类算法计算用户的初始近邻集(Moradi et al.， 2015)。然后，使用用户的初始近邻集来预测未知项的初始评级。此外，使用基于相似值和用户之间的信任声明的可靠性度量来评估初始预测评级的质量。然后，利用自适应近邻选择机制，从用户的初始近邻集中选取一个子集，进行可靠度的预测评级。为此，提出了用户之间的信任模型，从用户的初始近邻集中去除无用的用户。最后，使用用户的新调整过近邻集来预测未知项的新评级，并将最感兴趣的项推荐给当前用户。关于所提到的方法的具体步骤将在下面的小节中进行讨论。

3.1 初始近邻选择

在该方法的第一步中，使用用户聚类算法来初始化用户的最近项。用户聚类算法背后的主要思想是将原始用户集分到几个具有相似属性的集群中。因此，同一集群中的用户彼此相似，可以在评级预测中作为最近邻集使用。关于初始近邻选择步骤的更多细节将在以下小节中讨论。

3.1.1 构造

在此步骤中，将给定的用户集映射到图中，其中*V*表示用户集，表示用户对间的边集，*W*表示每对用户之间的相似度权重。在该方法中，结合相似度值和信任声明来计算用户之间的最终相似度权重。使用皮尔逊相关系数函数作为公式（1）计算用户之间的相似度值：

公式（1）

其中，表示用户给的项目的评级分，表示用户给出评级分的平均值，而表示用户与用户共同给项目集的评级分。此外，利用公式（2）计算用户之间的置信值：

公式（2）

其中，为用户与用户间的信任传播距离( Massa & Avesani, 2007 )，为用户之间可允许的最大传播距离，公式（3）为最大传播距离的计算公式：

公式（3）

其中，和分别为推荐系统中信任网络的大小和平均程度( Yuan, Guan, Lee, Lee, & Hur, 2010 )。最后，我们根据公式（4）来计算用户之间的相似度和置信值的组合值作为最终的相似度权重：

其它

公式（4）

其中，和为用户之间的相似度值和置信值，使用公式（1）和公式（2）进行计算。

3.1.2 查找初始中间集

图片包含 文字

描述已自动生成

**图3-1 计算初始中心集算法的伪代码**

在这个步骤中，会自动的找到每个集群的初始中心，因此在集群算法中不需要设置预定义的集群数量。为此，我们采用基于图的方法(Bahmani, Kumar， & Vassilvitskii, 2012)来寻找一个子图作为聚类方法的初始中心集。该方法对基于图论的方法进行了改进，找到最稀疏的子图作为聚类方法的初始中心集。设是无向的。

在图构造步骤中构造的加权图(见3.1.1节)。子图的密度可以被定义为公式(5)：

公式（5）

其中，为子图的边集，为边的权值。节点的加权程度由公式(6)计算：

公式（6）

其中，是节点和之间的边，是边的权值。

找到集群初始中心集的过程在图3-1(即Algorithm 1)中表示。该算法首先给出图G和计算当前的密度。然后,候选节点为能从图中删除已识别的基于阈值(即)的候选节点。为此,权重最高的节点是删除候选节点列中的一部分。该算法保证至少有一个子图包含大约k个节点。最后，如果得到的子图是非空的，则对剩下的图执行算法。需要注意的是，这一步的主要思想是找到一个密度最小的子图。因为，这个思想保证了找到一个中间距离最大的中心集来作为聚类算法的初始中心集。因此,这一步的输出为顶点的子集，其大小至少为k且密度最小。

3.1.3 用户初始近邻集的计算

在这一步中，我们使用了一种高效的机制来计算用户的初始近邻集。为此，在初始中心集上进行不断迭代，为使得集群得到更好的中心集。这一步是很必要的，因为识别出的初始中心集可能不是聚类算法得到的最佳结果。此外，那些相关联的成员小于阈值的集群将会与其他集群合并。需要注意的是，用户数量较少的集群可能会导致评级预测的精度下降。因此，合并后的集群(只有少数成员的集群)的成员将被分配到最近的中心集。用户的初始近邻集计算过程在在图3-2（即Algorithm 2)表示。在该算法中,首先将每个用户分配给最近的聚类中心基于初始中心集(如，在Algorithm 1中给出)中,。利用用户与中心之间的相似度值为用户找到最近的聚类中心，该聚类中心值由皮尔逊相关系数函数计算得到。然后，根据Algorithm 2的第5-6行计算聚类的新中心。此外，执行合并过程是为了将其基数小于参数的集群与其他集群合并，并为用户形成更优的集群(即Algorithm 2中第7-11行)。最后，结果中的集群的成员均被视为属于其集群的用户的初始近邻集(即Algorithm 2中第12-15行)。

图片包含 文字, 屏幕截图

描述已自动生成

**图3-2 计算用户的初始近邻集的伪代码**

3.2 自适应的近邻选择机制

在此节中，为了提高在预测过程中预测评级的准确性，提出了一种新的近邻选择机制。值得注意的是，在现有的大多数推荐算法中，用户的初始近邻集被直接用在于当前用户的近邻集来预测未知项，而不去考虑预测评级的质量。然而，一些初始的近邻对于预测所有未知项或许是没有用的。换言之，最重要的是判断和细化用户的初始近邻集，以便为用户提供更优的推荐。因此，所提出的自适应近邻选择机制是用来选择初始近邻集的子集，这有助于我们预测未知项。为此，将使用可靠性度量来评估初始预测评级的质量。在此基础上，本节提出了一种用户间的置信模型，将无用的用户从初始近邻集中移除。

3.2.1 计算可靠性度量值

在这一步中，使用可靠性度量值来计算初始预测评级的可靠性值(Hernando, Bobadilla, Ortega，&Tejedor, 2013)。这些初始评级是利用了在提出的方法中的用户初始近邻选步骤中形成的初始近邻集来进行预测的(见3.1节)。因此，我们可以利用公式(7)来计算当前用户对项的初始评级：

公式（7）

其中，为当前用户的平均评级分,指初始近邻集的某个子集的用户对项的评分,为用户和用户之间的相似性权重，再通过公式(4)得到。

另一方面，我们提出的可靠性度量是基于相似值和用户之间的信任声明的组合，用于评估初始预测评级的质量。为此，本文采用了正负两个因子来计算可靠性度量。计算可靠性度量的整个过程如下所述。

步骤1：可靠性度量的正因子由公式(8)得到：

公式（8）

其中，是给定推荐系统中的中位数，是用户与其初始近邻集相似度权值的总和，使用公式（9）计算得出：

公式（9）

其中，为用户对第项评级的近邻集合，为用户与用户之间的相似度权重，可由式(4)计算得到。

步骤2：可靠性度量的负因子由公式(10)得到：

公式（10）

其中，

公式（11）

是给定推荐系统中值的中值，和分别是给定推荐系统中项目评级的最大值和最小值。此外，在可在公式(12)中得到：

公式（12）

其中，为当前用户的平均评级分数,指的是一组一对项目评级的用户的近邻集,指用户对项的初始评级，可通过公式（7）得到。为用户和用户之间相似度权重，可通过公式（4）计算得到。

步骤3：最后，利用公式(13)来计算用户即)对项的初始评级的置信程度：

公式（13）

其中，和分别为采用公式（8）和公式（10）计算的可靠度度量的正、负因子。

3.2.2 近邻的自适应

在初始评级预测步骤(即公式(7))中，初始近邻集中的所有用户都被作为当前用户的最近近邻来预测未知项。这些初始近邻是基于用户的已识别集群来确定的，这些已识别的集群在提出的方法中的初始邻居近邻步骤中计算出来的(参见3.1节)。然而，对于给定的当前用户，一些初始近邻或许会无法预测某些未知项。因此，我们提出一种能够评估初始预测评级的算法来帮助系统向当前用户高效的推荐产品。

在此步骤中，我们提出了一种基于可靠性度量的自适应近邻选择机制，用以提高初始预测评级的准确性。因此，可靠性度量被用来评价未知项的初始预测评级的质量。为此,当前用户的可靠性度量值和项被用来在公式13中计算且当值低于一个阈值时,当前用户的初始近邻集将被替换为一个新的近邻集。接着，利用这个经过调整后的近邻集来预测初始预测评级的新评级分。为了计算出当前用户和项的一组新的自适应近邻，我们在提出的可靠性度量(即公式(13))的基础上，提出了用户之间的置信模型。提出的用户之间的信任模型可由公式(14)计算得到:

公式（14）

其中,是用户和用户之间的置信度,为当前用户的可靠性度量值再用项再经过了公式（13）计算得到的。是用户给定项的评级分数,是由用户给定的评级评级分数,是用户和用户同时给项集的评级分。

在计算出当前用户和其初始近邻集的用户之间的置信值之后，将无用的用户从用户的初始近邻集中删除。为此，一个阈值被用来识别哪些属于无用的用户。换句话说, 如果置信值低于阈值，用户将从当前用户的初始近邻中被删除。因此，用户的初始近邻集是基于用户之间的置信模型进行调整的，以用来适应当可靠性度量值小于阈值的初始预测评级。也就是说，我们提出的近邻选择机制可以为用户形成自适应的近邻集，从而提高了预测过程的准确性和效率。

3.3 推荐算法

在步骤近邻的自适应中(参见3.2.2节)，我们提出了一组基于当前用户的比初始近邻集更加有效的近邻集。在推荐算法这一步中，我们使用当前用户的近邻集和公式（15）来预测的即将进行评级的未知项：

公式（15）

其中，是当前用户对项的最新评级分，是用户对给定项的评级分，是当前用户的平均评级分，是aa是用户a的自适应近邻集的子集且已对项进行评级的，表示用户与用户之间的相似度权重，由公式(4)计算得到。该算法在预测了用户未知项的最新评级后，选择最前面的项作为推荐列表向用户进行推荐。该算法的伪代码如图3-3所示(即Algorithm 3)。

图片包含 文字, 屏幕截图

描述已自动生成

**图3-3 推荐算法伪代码**

4 实验结果

在本节中，我们将根据三个已知的数据集对我们所提出的方法(即SRANS)的性能进行实证及评估，并将结果与其他基于不同评估指标的推荐方法进行比较。下面是对比较方法的简要说明：

* KCF是一种使用了K-Means聚类算法的基线方法，根据用户的历史评分(即公式（1）)对用户进行聚类。同时，将识别出的聚类作为用户的近邻，利用协同过滤的方法预测未知项的评分。
* TKCF是KCF方法的一个变体，它使用相似值和信任声明的组合作为用户之间的相似度权重(即公式（4）)来进行评级预测。
* TRACCF是一种基于模型的协同过滤方法，它同时考虑了聚类和信任信息，为用户预测未知项(Birtolo & Ronca, 2013)。
* IFCCF是一种使用模糊C-Means聚类算法对所获得的评分来对项目进行聚类的算法。此外，得到的聚类将用于预测当前用户对未知项的评级分(Birtolo & Ronca, 2013)。
* MV是一种基于历史评级和信任信息对用户进行聚类的多视图K-medoids方法(Guo et al.， 2015a)。
* DGCTARS是一种使用一种图形聚类算法将用户分成几个集群，然后去利用这些识别出来的集群来预测在推荐过程中当前用户的评级的推荐方法(Moradi et al.，2015)。
* TrustSVD是一种基于置信的矩阵分解技术，通过进一步的合并那些受信任用户对项目预测的显式和隐式影响，涉及评级项目的显式和隐式影响(Guo et al.， 2015b)。
* SocialMF是一种基于将置信传播纳入矩阵分解模型的推荐方法，来在社交网络中为用户提供推荐(Jamali & Ester, 2010)。
* TrustMF是一种基于模型的方法，它使用矩阵分解技术将用户按照他们之间的置信关系映射到低维的潜在因子空间(Yang, Lei et al.， 2013)。
* SRANS是一种基于自适应近邻选择机制来为社交网络用户提供推荐的方法。

实验中使用的数据集的描述、参数设置、评价指标、性能比较、参数的敏感性分析和统计分析将在以下小节中进行介绍。

4.1 数据集

在实验中我们使用了三个知名的数据集来验证本文提出的算法的有效性。这些数据集包括了Epinions 、Flixster 和FilmTrust 。Epinions的数据集是从Epinions.com网站上提取的，它的用户将能够查看项目，并为它们分配数值评级，范围从1 (最小)到5 (最大)。此外，还体现了用户与其他用户之间的信任关系。该数据集中用户之间的信任声明值分别为0和1，这个值体现了两个用户之间的信任关系是否存在。Epinions数据集中的用户数量为49,290个，且都在139,738个不同的项目中至少评分过一次。为了简单起见，我们从原始数据集以及用户评级和信任信息中随机抽取了10000个用户。Flixster数据集是一个社交电影网站，用户可以在其中与其他用户建立社交关系，并可以共享他们对已有电影的评分，这些电影根据步长0.5从0.5 (最小)扩展到4.0 (最大)。Flixster数据集中的信任声明的值是不可用的；因此，我们使用了用户之间的社交关系作为信任声明。在实验中，我们对Flixster数据集中的一个子集进行了抽样，并且随机选择了10000个用户，他们对项目和信任声明进行了相应的评级。最后，FilmTrust数据集是一个社交网站，它的用户能够对已有的电影进行评级打分，评分值在0.5 (最小)到4.0 (最大)之间且步长为0.5。此外，信任声明的值在1到10之间，由于共享策略，这些值依旧是不可用的。因此，在实验中，我们使用用户之间的链接信息作为信任声明。该数据集包括1986个用户、2071部电影、35497个电影评分和1853个信任评级分(Guo et al.，2014)。

4.2 参数设置

在我们提出的方法中，在我们需要比较的方法中需要去初始化多个参数。这些参数是,和，这些参数分别代表了可靠性阈值,阈值的置信值,推荐列表的大小。应该注意的是,参数和都是用作阈值来在我们提出的算法中计算可靠性度量值以及置信度量，可分别(参见3.2.2节)。在本次实验中,参数,和在所有数据集中均设为 ，，。在算法中使用的参数和参数使用我们给定的默认值，，，这些默认值通常来说都能带来适合的结果。此外，我们在算法1中使用了参数，我们分别对Flixster、Epinions和FilmTrust数据集使用了、和的参数设置。实验中其余的方法都需要设置参数，为了更加公平的比较，这些参数的取值都是根据相应的文献中所公布的最优值来确定的。

4.3 评价指标

在当前的工作中，我们使用了传统的留一法过程来比较本文所提出方法与其他方法的性能差异(Massa & Avesani, 2007)。为此，我们采用了5次交叉验证的方法来比较推荐算法的结果。换句话说，每个数据集分为5份，再每次运行四份折叠作为训练集，其余的部分作为测试集。五分执行并测试所有的数据集份和这些运行后的平均结果报告作为实验的最终结果。在留一法中，测试集中的一个评级评审结果被用作未知项项的评级结果，并通过使用关于其余评级评审的信息来预测它的评级是否满足质量。然后将预测评级与实际评级进行比较，计算两者之间的差值，将这个差值作为预测误差。需要注意的是，该过程是对测试集中的所有评级都进行了重复测试，用以计算评估指标。在本次实验中，采用了平均绝对误差(MAE)、均方根误差(RMSE)、精度（Precision）、查全率（Recall）和F1等评价指标来评价该方法有效与否。

MAE和RMSE是评估预测评级准确性的两个指标。为此，将预测评级与实际评级进行比较，并将两者之间的差异视为预测误差。再对所有预测评级分数重复此过程，然后将计算出所有值的平均值作为最终预测误差。MAE指标和RMSE指标分别使用公式（16）以及公式（17）进行计算：

公式（16） 公式（17）

其中，和分别为项目的实际评分和预测评分。表示推荐算法预测的评级总数。

另一方面，为了计算、和指标，评级被转换为二进制的格式，包括了相关和不相关的项目(Herlocker, Konstan, Terveen， & Riedl, 2004)。Epinions数据集提供了5点规模评级。此外，Flixster和FilmTrust数据集包含的评级范围为0.5到4.0(步长为0.5)。因此，在实验中，对于所有使用的数据集，大于3(即)的评分值将被视为相关项，剩余的评分值被视为非相关项。精度度量定义为精确或准确性的度量，其计算方法是计算从所有项目中检索到的相关项目的比例，使用公式（18）进行计算：

公式（18）

此外，Recall度量值是一种完整的度量过程，其计算方式是检索到的所有相关项目中相关项目的比例。这个度量定义为公式（19）：

公式（19）

需要注意的是，和在本质上是明显冲突的(Herlocker et al.， 2004)。换句话说，如果推荐结果的数量增加，那么相关条目的数量和也会增加。另一方面，同样也会随之下降。为了获得和的适当加权组合，将度定义为公式（20）：

公式（20）

4.4 性能比较

在本节中，我们将在三个真实的数据集上进行一系列的实验，以演示基于两个数据视图(包括所有用户和冷用户)的方法相对于其他方法的性能。在所有用户的视图中，我们使用了数据集中的所有数据，而在冷用户视图中，我们将考虑评级分低于5的冷启动用户来评估的推荐算法。表1-6显示了提出的SRANS方法与其他基于MAE和RMSE度量的最先进方法的比较结果。需要注意的是，这些结果是基于所有推荐算法的不同邻域大小(即N = 20,40,60,80,100)。在TrustSVD、SocialMF和TrustMF方法中，我们使用用户的信任网络来形成不同大小的近邻集。通过表1和表2显示，与Epinions数据集上的其他方法相比，所提出的方法在所有用户和冷用户视图的MAE和RMSE度量结果方面获得了最好的效果。例如，SRANS方法在所有用户视图中获得MAE值0.772和RMSE值1.052和在冷用户视图MAE值0.813和RMSE值1.083且都是在基于N = 20的情况下,同时在同一例子中第二个最优方法TrustSVD方法得到了在所有用户视图MAE值0.866和RMSE值1.113和冷用户视图下MAE值0.898和RMSE值1.233的结果。

此外，我们还基于MAE和RMSE测量的实验在Flixster和FilmTrust数据集上重复进行，得到的结果如表3-6所示。表3和表4的结果表明，对于所有用户和冷用户视图，本文提出的方法在MAE和RMSE度量值下都优于其他方法。此外，在表5和表6显示了在FilmTrust数据集上MAE和RMSE测量的实验结

**表4.1 在Epinions数据集上的MAE度量值、不同的近邻集大小（N）以及五倍的交叉验证结果**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Algorithms | 所有用户 | | | | | 冷用户 | | | | |
| N=20 | N=40 | N=60 | N=80 | N=100 | N=20 | N=40 | N=60 | N=80 | N=100 |
| KCF | 1.167 | 1.144 | 1.098 | 1.082 | 1.057 | 1.384 | 1.312 | 1.297 | 1.253 | 1.239 |
| TCKF | 1.142 | 1.103 | 1.086 | 1.069 | 1.054 | 1.293 | 1.276 | 1.249 | 1.225 | 1.208 |
| TRACCF | 1.025 | 1.002 | 0.984 | 0.972 | 0.956 | 1.175 | 1.159 | 1.143 | 1.128 | 1.107 |
| IFCCF | 0.986 | 0.963 | 0.941 | 0.923 | 0.891 | 1.108 | 1.087 | 1.052 | 1.032 | 1.009 |
| MV | 1.003 | 0.995 | 0.973 | 0.957 | 0.929 | 1.096 | 1.082 | 1.041 | 1.017 | 0.995 |
| DFCTARS | 0.954 | 0.932 | 0.916 | 0.885 | 0.864 | 0.965 | 0.952 | 0.934 | 0.918 | 0.896 |
| TrustSVD | 0.866 | 0.839 | 0.817 | 0.805 | 0.793 | 0.898 | 0.882 | 0.864 | 0.845 | 0.832 |
| SocialMF | 0.921 | 0.908 | 0.887 | 0.862 | 0.839 | 0.943 | 0.927 | 0.895 | 0.887 | 0.864 |
| TrustMF | 0.893 | 0.871 | 0.856 | 0.839 | 0.817 | 0.928 | 0.905 | 0.876 | 0.862 | 0.849 |
| SRANS | **0.772** | **0.743** | **0.698** | **0.672** | **0.664** | **0.813** | **0.786** | **0.753** | **0.742** | **0.735** |

**表4.2 在Epinions数据集上的RMSE度量值、不同的近邻集大小（N）以及五倍的交叉验证结果**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Algorithms | 所有用户 | | | | | 冷用户 | | | | |
| N=20 | N=40 | N=60 | N=80 | N=100 | N=20 | N=40 | N=60 | N=80 | N=100 |
| KCF | 1.594 | 1.571 | 1.537 | 1.506 | 1.487 | 1.653 | 1.645 | 1.627 | 1.612 | 1.596 |
| TCKF | 1.564 | 1.529 | 1.509 | 1.488 | 1.472 | 1.634 | 1.632 | 1.625 | 1.609 | 1.593 |
| TRACCF | 1.343 | 1.317 | 1.294 | 1.265 | 1.228 | 1.372 | 1.346 | 1.325 | 1.297 | 1.245 |
| IFCCF | 1.243 | 1.201 | 1.189 | 1.154 | 1.125 | 1.287 | 1.248 | 1.216 | 1.185 | 1.159 |
| MV | 1.275 | 1.253 | 1.231 | 1.207 | 1.192 | 1.295 | 1.274 | 1.253 | 1.219 | 1.216 |
| DFCTARS | 1.119 | 1.107 | 1.098 | 1.069 | 1.055 | 1.272 | 1.221 | 1.195 | 1.163 | 1.151 |
| TrustSVD | 1.113 | 1.101 | 1.087 | 1.053 | 1.051 | 1.233 | 1.188 | 1.165 | 1.127 | 1.123 |
| SocialMF | 1.271 | 1.189 | 1.146 | 1.115 | 1.072 | 1.258 | 1.208 | 1.182 | 1.153 | 1.146 |
| TrustMF | 1.134 | 1.112 | 1.105 | 1.086 | 1.069 | 1.241 | 1.201 | 1.174 | 1.138 | 1.129 |
| SRANS | **1.052** | **1.033** | **1.012** | **1.002** | **0.993** | **1.083** | **1.065** | **1.042** | **1.027** | **1.018** |

果。从表5可以看出，除了TrustSVD方法得到了在所有用户视图下的最佳的MAE值外，本文提出的算法的性能优于其他推荐方法。且在这种情况下，本文所提出的方法可以得到次优的MAE值。此外，表6的结果表明，对于所有用户视图，与其他方法相比，本文所提出的方法获得了第三好的RMSE值。然而，表5和表6的结果也表明，对于冷用户视图来说，与其他方法相比，本文里所提出的方法得到了更好的MAE和RMSE值。

**表4.3 在Flixster数据集上的MAE度量值、不同的近邻集大小（N）以及五倍交叉验证结果**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Algorithms | 所有用户 | | | | | 冷用户 | | | | |
| N=20 | N=40 | N=60 | N=80 | N=100 | N=20 | N=40 | N=60 | N=80 | N=100 |
| KCF | 1.195 | 1.146 | 1.105 | 1.038 | 0.996 | 1.242 | 1.218 | 1.175 | 1.163 | 1.148 |
| TCKF | 1.167 | 1.115 | 1.063 | 0.998 | 0.965 | 1.219 | 1.183 | 1.161 | 1.147 | 1.127 |
| TRACCF | 1.183 | 1.124 | 1.085 | 1.001 | 0.979 | 1.201 | 1.178 | 1.153 | 1.125 | 1.094 |
| IFCCF | 1.144 | 1.098 | 1.057 | 0.953 | 0.924 | 1.196 | 1.172 | 1.146 | 1.098 | 1.052 |
| MV | 1.135 | 1.072 | 1.004 | 0.946 | 0.912 | 1.192 | 1.115 | 1.108 | 1.062 | 1.014 |
| DFCTARS | 1.128 | 1.012 | 0.973 | 0.885 | 0.857 | 1.188 | 1.143 | 1.102 | 1.031 | 0.985 |
| TrustSVD | 1.097 | 0.963 | 0.875 | 0.817 | 0.798 | 1.172 | 1.105 | 1.031 | 0.964 | 0.913 |
| SocialMF | 1.109 | 0.985 | 0.913 | 0.852 | 0.815 | 1.175 | 1.112 | 1.023 | 0.987 | 0.945 |
| TrustMF | 1.131 | 1.055 | 0.997 | 0.931 | 0.897 | 1.183 | 1.119 | 1.078 | 1.021 | 0.976 |
| SRANS | **0.975** | **0.883** | **0.791** | **0.748** | **0.709** | **1.007** | **0.941** | **0.885** | **0.843** | **0.805** |

**表4.4 在Flixster数据集上的RMSE度量值、不同的近邻集大小（N）以及五倍的交叉验证结果**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Algorithms | 所有用户 | | | | | 冷用户 | | | | |
| N=20 | N=40 | N=60 | N=80 | N=100 | N=20 | N=40 | N=60 | N=80 | N=100 |
| KCF | 1.395 | 1.345 | 1.312 | 1.304 | 1.293 | 1.468 | 1.432 | 1.397 | 1.371 | 1.367 |
| TCKF | 1.376 | 1.342 | 1.305 | 1.271 | 1.258 | 1.445 | 1.415 | 1.392 | 1.364 | 1.353 |
| TRACCF | 1.353 | 1.324 | 1.287 | 1.252 | 1.235 | 1.376 | 1.359 | 1.317 | 1.285 | 1.258 |
| IFCCF | 1.294 | 1.275 | 1.246 | 1.203 | 1.187 | 1.344 | 1.306 | 1.262 | 1.237 | 1.216 |
| MV | 1.253 | 1.228 | 1.197 | 1.185 | 1.164 | 1.352 | 1.314 | 1.287 | 1.255 | 1.232 |
| DFCTARS | 1.217 | 1.193 | 1.164 | 1.136 | 1.125 | 1.276 | 1.249 | 1.213 | 1.194 | 1.175 |
| TrustSVD | 1.175 | 1.129 | 1.075 | 1.004 | 0.945 | 1.203 | 1.176 | 1.134 | 1.117 | 1.095 |
| SocialMF | 1.196 | 1.154 | 1.105 | 1.047 | 0.996 | 1.221 | 1.197 | 1.168 | 1.149 | 1.126 |
| TrustMF | 1.242 | 1.217 | 1.182 | 1.168 | 1.152 | 1.255 | 1.228 | 1.211 | 1.191 | 1.173 |
| SRANS | **0.995** | **0.963** | **0.941** | **0.925** | **0.897** | **1.104** | **1.041** | **1.007** | **1.003** | **0.998** |

**表4.5 在FilmTrust数据集上的MAE度量值、不同的近邻集大小（N）以及五倍的交叉验证结果**

此外，基于Precision、Recall和F1度量值中报告了Epinions、Flixster和FilmTrust数据集的实验结果在表7-10。这些实验是根据参数top\_N的不同值(即top\_N = 5,10,15)在所有用户视图下进行的。另一方面，对于冷用户视图，实验是基于top\_N = 5来执行的，因为这些用户没有对超过5个项目进行评级打分。表7报告了基于所有用户视图在Epinions数据集上的Precision、Recall和F1度量的实验结果。从这些结果可以看到,本文提出的方法在所有的Precision、Recall和F1度量值和所有值参数top\_N是优于其他的推荐方法。对Flixster和FilmTrust数据集重复这些实验并将结果分别显示在表8和9。从这些结果中可以明显看出，不同的基于Precision、Recall和F1度量以及所

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Algorithms | 所有用户 | | | | | 冷用户 | | | | |
| N=20 | N=40 | N=60 | N=80 | N=100 | N=20 | N=40 | N=60 | N=80 | N=100 |
| KCF | 0.965 | 0.897 | 0.824 | 0.776 | 0.745 | 1.069 | 1.002 | 0.945 | 0.923 | 0.889 |
| TCKF | 0.983 | 0.912 | 0.878 | 0.816 | 0.759 | 1.086 | 1.021 | 0.964 | 0.937 | 0.895 |
| TRACCF | 0.854 | 0.808 | 0.751 | 0.685 | 0.648 | 1.024 | 0.955 | 0.902 | 0.846 | 0.803 |
| IFCCF | 0.789 | 0.774 | 0.721 | 0.668 | 0.637 | 0.942 | 0.896 | 0.828 | 0.781 | 0.736 |
| MV | 0.931 | 0.875 | 0.819 | 0.752 | 0.734 | 1.008 | 0.951 | 0.895 | 0.842 | 0.798 |
| DFCTARS | 0.815 | 0.795 | 0.739 | 0.673 | 0.643 | 0.997 | 0.928 | 0.861 | 0.819 | 0.764 |
| TrustSVD | **0.769** | **0.748** | **0.693** | **0.637** | **0.611** | 0.896 | 0.853 | 0.806 | 0.761 | 0.706 |
| SocialMF | 0.876 | 0.823 | 0.765 | 0.698 | 0.651 | 0.965 | 0.907 | 0.845 | 0.784 | 0.742 |
| TrustMF | 0.837 | 0.797 | 0.746 | 0.679 | 0.645 | 0.923 | 0.882 | 0.521 | 0.774 | 0.715 |
| SRANS | 0.775 | 0.752 | 0.706 | 0.643 | 0.614 | **0.872** | **0.805** | **0.753** | **0.698** | **0.662** |

**表4.6 在FilmTrust数据集上的RMSE度量值、不同的近邻集大小（N）以及五倍的交叉验证结果**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Algorithms | 所有用户 | | | | | 冷用户 | | | | |
| N=20 | N=40 | N=60 | N=80 | N=100 | N=20 | N=40 | N=60 | N=80 | N=100 |
| KCF | 1.118 | 1.045 | 1.008 | 0.976 | 0.941 | 1.289 | 1.242 | 1.191 | 1.148 | 1.094 |
| TCKF | 1.134 | 1.088 | 1.021 | 0.986 | 0.954 | 1.294 | 1.243 | 1.196 | 1.154 | 1.105 |
| TRACCF | 1.009 | 0.994 | 0.954 | 0.918 | 0.881 | 1.265 | 1 .236 | 1 .175 | 1 .112 | 1 .067 |
| IFCCF | 0 .992 | 0.938 | 0.891 | 0.857 | 0.839 | 1.184 | 1.147 | 1.034 | 0.978 | 0.942 |
| MV | 1.125 | 1.063 | 1.014 | 0.982 | 0.949 | 1.218 | 1.164 | 1.095 | 1.022 | 0.987 |
| DFCTARS | 1.002 | 0.991 | 0.943 | 0.906 | 0.875 | 0.949 | 1.218 | 1.164 | 1.095 | 1.022 |
| TrustSVD | **0.986** | **0.927** | **0.884** | **0.851** | **0.827** | 1.141 | 1.108 | 1.005 | 0.953 | 0.912 |
| SocialMF | 1.103 | 1.007 | 0.962 | 0.923 | 0.885 | 1.206 | 1.152 | 1.063 | 1.002 | 0.965 |
| TrustMF | 0.998 | 0.952 | 0.916 | 0.891 | 0.863 | 1.158 | 1.152 | 1.011 | 0.926 | 0.931 |
| SRANS | 0.995 | 0.942 | 0.898 | 0.863 | 0.844 | **1.056** | **1.004** | **0.965** | **0.919** | **0.862** |

**表4.7 所有用户对Epinions数据集的Precision、Recall和F1估计，不同的top\_N值和5倍交叉验证**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Algorithms | Precison | | | Recall | | | F1 | | |
| P@5 | P@10 | P@15 | R@5 | R@10 | R@15 | F1@5 | F1@10 | F1@15 |
| KCF | 0.935 | 0.925 | 0.915 | 0.551 | 0.612 | 0.691 | 0.693 | 0.737 | 0.787 |
| TCKF | 0.939 | 0.923 | 0.912 | 0.563 | 0.632 | 0.665 | 0.704 | 0.751 | 0.769 |
| TRACCF | 0.954 | 0.943 | 0.935 | 0.582 | 0.646 | 0.712 | 0.723 | 0.767 | 0.808 |
| IFCCF | 0.971 | 0.957 | 0.952 | 0.675 | 0.718 | 0.786 | 0.796 | 0.821 | 0.861 |
| MV | 0.968 | 0.948 | 0.945 | 0.658 | 0.685 | 0.748 | 0.783 | 0.795 | 0.835 |
| DFCTARS | 0.951 | 0.949 | 0.942 | 0.751 | 0.788 | 0.832 | 0.839 | 0.861 | 0.884 |
| TrustSVD | 0.975 | 0.961 | 0.958 | 0.776 | 0.831 | 0.872 | 0.864 | 0.891 | 0.913 |
| SocialMF | 0.958 | 0.946 | 0.939 | 0.758 | 0.809 | 0.847 | 0.846 | 0.872 | 0.891 |
| TrustMF | 0.969 | 0.954 | 0.949 | 0.769 | 0.827 | 0.859 | 0.857 | 0.886 | 0.902 |
| SRANS | **0.981** | **0.975** | **0.966** | **0.788** | **0.852** | **0.887** | **0.874** | **0.909** | **0.925** |

有用户视图参数top\_N的取值不同，本文所提出的方法可以获得比其他方法更好的性能。此外，对同样对冷用户视图进行的实验，表10显示了所有Epinions、Flixster和FilmTrust数据集的结果。从表10的结果可以看出，该方法对于基于Precision、Recall和F1度量的冷用户视图是有效的，并且与其他方法相比，该方法可以获得更优的结果。

**表4.8 所有用户对Flixster数据集的Precision、Recall和F1估计，不同的top\_N值和5倍交叉验证**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Algorithms | Precison | | | Recall | | | F1 | | |
| P@5 | P@10 | P@15 | R@5 | R@10 | R@15 | F1@5 | F1@10 | F1@15 |
| KCF | 0.825 | 0.812 | 0.775 | 0.684 | 0.697 | 0.715 | 0.748 | 0.751 | 0.744 |
| TCKF | 0.848 | 0.819 | 0.786 | 0.716 | 0.728 | 0.743 | 0.776 | 0.771 | 0.764 |
| TRACCF | 0.876 | 0.851 | 0.827 | 0.738 | 0.756 | 0.772 | 0.801 | 0.801 | 0.799 |
| IFCCF | 0.913 | 0.894 | 0.863 | 0.765 | 0.783 | 0.798 | 0.832 | 0.835 | 0.829 |
| MV | 0.927 | 0.915 | 0.881 | 0.771 | 0.789 | 0.815 | 0.842 | 0.847 | 0.847 |
| DFCTARS | 0.935 | 0.926 | 0.892 | 0.765 | 0.783 | 0.798 | 0.832 | 0.835 | 0.829 |
| TrustSVD | 0.952 | 0.938 | 0.921 | 0.813 | 0.842 | 0.863 | 0.877 | 0.887 | 0.891 |
| SocialMF | 0.946 | 0.932 | 0.913 | 0.809 | 0.831 | 0.846 | 0.872 | 0.879 | 0.878 |
| TrustMF | 0.941 | 0.929 | 0.904 | 0.805 | 0.825 | 0.837 | 0.868 | 0.874 | 0.869 |
| SRANS | **0.964** | **0.942** | **0.934** | **0.821** | **0.852** | **0.878** | **0.887** | **0.895** | **0.905** |

**表4.9 所有用户对FilmTrust数据集的Precision、Recall和F1估计，不同的top\_N值和5倍交叉验证**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Algorithms | Precison | | | Recall | | | F1 | | |
| P@5 | P@10 | P@15 | R@5 | R@10 | R@15 | F1@5 | F1@10 | F1@15 |
| KCF | 0.828 | 0.807 | 0.765 | 0.738 | 0.755 | 0.782 | 0.780 | 0.781 | 0.773 |
| TCKF | 0.811 | 0.809 | 0.791 | 0.753 | 0.778 | 0.819 | 0.781 | 0.793 | 0.805 |
| TRACCF | 0.825 | 0.815 | 0.786 | 0.816 | 0.839 | 0.864 | 0.824 | 0.827 | 0.823 |
| IFCCF | 0.849 | 0.834 | 0.815 | 0.834 | 0.861 | 0.886 | 0.841 | 0.847 | 0.849 |
| MV | 0.834 | 0.819 | 0.798 | 0.827 | 0.857 | 0.873 | 0.831 | 0.838 | 0.834 |
| DFCTARS | 0.839 | 0.827 | 0.801 | 0.866 | 0.896 | 0.915 | 0.852 | 0.861 | 0.854 |
| TrustSVD | 0.851 | 0.829 | 0.823 | 0.857 | 0.885 | 0.921 | 0.854 | 0.856 | 0.869 |
| SocialMF | 0.842 | 0.812 | 0.806 | 0.846 | 0.852 | 0.881 | 0.844 | 0.832 | 0.842 |
| TrustMF | 0.845 | 0.824 | 0.812 | 0.851 | 0.873 | 0.892 | 0.848 | 0.848 | 0.851 |
| SRANS | **0.859** | **0.838** | **0.832** | **0.847** | **0.901** | **0.938** | **0.866** | **0.869** | **0.882** |

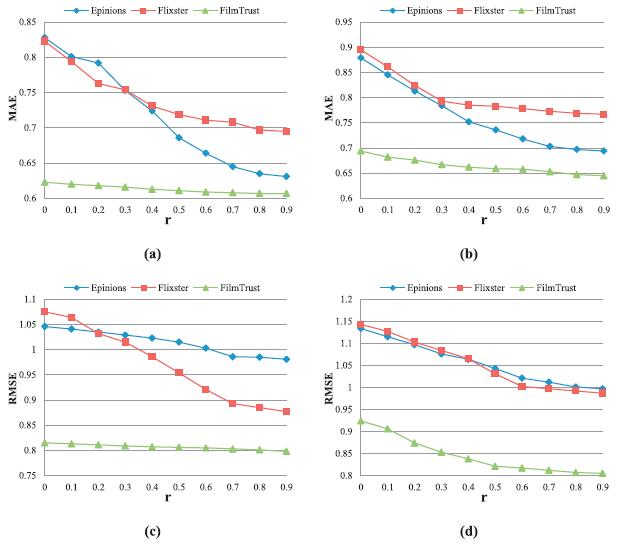
**表4.10 冷用户对Epinions、Flixster、FilmTrust数据集的Precision、Recall和F1估计，top\_N值和5倍交叉验证**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Algorithms | Epinions | | | Flixster | | | FilmTrust | | |
| P@5 | R@5 | F1@5 | P@5 | R@5 | F1@5 | P@5 | R@5 | F1@5 |
| KCF | 0.743 | 0.565 | 0.642 | 0.625 | 0.553 | 0.587 | 0.631 | 0.706 | 0.666 |
| TCKF | 0.756 | 0.572 | 0.651 | 0.641 | 0.586 | 0.612 | 0.645 | 0.752 | 0.694 |
| TRACCF | 0.769 | 0.593 | 0.669 | 0.662 | 0.628 | 0.645 | 0.662 | 0.761 | 0.708 |
| IFCCF | 0.812 | 0.684 | 0.743 | 0.713 | 0.665 | 0.688 | 0.698 | 0.805 | 0.748 |
| MV | 0.786 | 0.647 | 0.709 | 0.726 | 0.683 | 0.704 | 0.617 | 0.781 | 0.722 |
| DFCTARS | 0.795 | 0.746 | 0.769 | 0.742 | 0.697 | 0.719 | 0.684 | 0.842 | 0.755 |
| TrustSVD | 0.828 | 0.769 | 0.797 | 0.759 | 0.739 | 0.749 | 0.712 | 0.858 | 0.778 |
| SocialMF | 0.804 | 0.721 | 0.761 | 0.754 | 0.721 | 0.737 | 0.678 | 0.826 | 0.745 |
| TrustMF | 0.819 | 0.752 | 0.784 | 0.749 | 0.718 | 0.733 | 0.689 | 847 | 0.759 |
| SRANS | **0.841** | **0.782** | **0.881** | **0.763** | **0.756** | **0.759** | **0.731** | **0.864** | **0.792** |

4.5 参数的敏感性分析

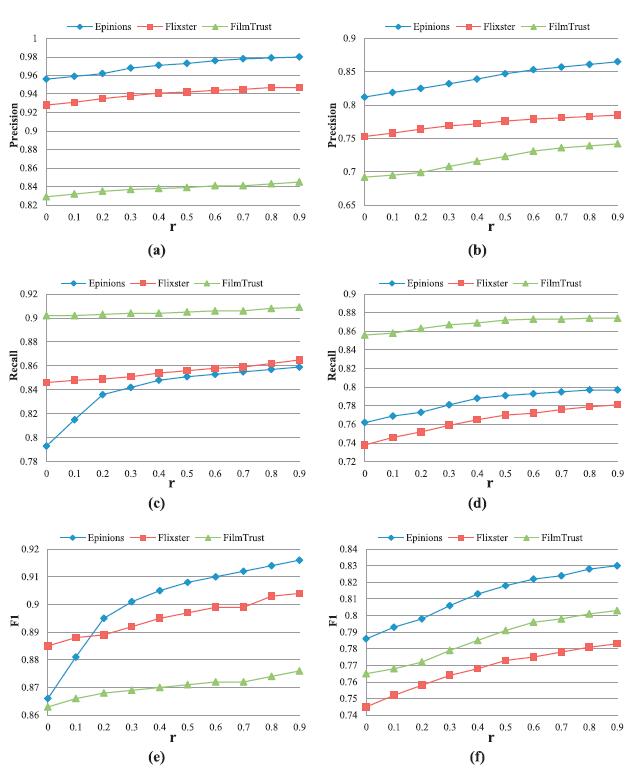
在本节中，我们做了几个实验用来展示不同的输入参数的对所提出的方法性能的影响。参数是本算法中近邻自适应选择步骤中重要的参数(见3.2.2)，其作为计算可靠性度量的阈值。图4显示了参数的不同值对所有用户以及冷用户视图在Epinions、Flixster和FilmTrust数据集上的MAE和RMSE度量并经过提出的算法后的性能结果。从图中可以看出，当参数随着步长0.1从0增加到0.9时，本方法的MAE和RMSE的值在不断减小。因此，我们可以从中得出结论，参数的值越高，本文里所提出的方法在所有用户和冷用户视图的评级预测过程的准确性方面的性能就越好。

此外，对于不同的参数的Precision、Recall和F1度量评价通过所提出的算法的性能的影响在图5中所展示。从这些结果中可以看出，当步长为0.1，参数的值从0增加到0.9，这将导致了在Epinions、Flixster和FilmTrust数据集上推荐算法的Precision、Recall和F1度量值的增加。因此，可以认为，参数的值越高，对于所有用户来说，基于精度、召回率和F1度量的冷用户视图，本文所提出的方法的性能更好。需要值得注意的是，在这些实验中，对于所有用户视图和冷用户视图，参数的取值分别设置为。



**图4-1 参数对系统性能的影响：（a）所有用户的MAE，（b）冷用户的MAE，（c）所有用户的RMSE，（d）冷用户的RMSE**

该算法的所需的另一个参数是参数，可以作用在评级的准确性预测这个过程中。该参数用于本文所提算法的近邻自适应选择步骤，将作为提出的置信模型的阈值(见3.2.2节)。换句话说,参数用于从最初当前用户的近邻集中删除那些无用的用户，接着为目标项计算出一个新的自适应近邻集来预测出一个新的评级分。不同的参数值的影响的评估是基于Epinions，Flixster, FilmTrust数据集的使用来进行评估的。图6报告基于MAE和RMSE度量以及所有用户和冷用户视图的参数的不同值的情况下所执行的实验的结果。如图所示，当的值从0增加到0.9时，步长为0.1时，所有用户和冷用户视图的MAE和RMSE度量值都会减小。结果表明，参数的值越高，对预测结果的准确性指标有积极的影响。另一方面，在图7中所展示出了不同的参数值对所有用户和冷用户视图以及对Epinions、Flixster和FilmTrust数据集的Precision、Recall和F1度量的影响。在这些实验中，所有用户视图和冷用户视图的参数的值分别设置为和。基于这些结果表明了，当参数的值增大时，Precision、Recall和F1度量的值也随之增大。



**图4-2 参数对系统性能的影响：（a）所有用户的Precision，（b）冷用户的Precision，（c）所有用户的Recall，（d）冷用户的Recall，（e）所有用户的F1，（f）冷用户的F1**

因此，较高的参数θ对基于上述所有用户和冷用户视图的评估度量在本文提算法下进行的性能都有积极影响。

图片包含 文字, 地图

描述已自动生成

**图4-3 参数对系统性能的影响：（a）所有用户的最大允许误差，（b）冷用户的最大允许误差，（c）所有用户的最大允许误差，（d）冷用户的最大允许误差**

4.6 统计分析

本节通过统计显著性检验来确定实验结果是否具有统计显著性。为此，对结果进行了弗里德曼检验(Friedman, 1937)，这是一个非参数检验，用来衡量算法在多个数据集上的统计差异。因此，推荐算法需要根据每个数据集的不同评价指标进行单独排序，结果最好的方法获得第1级，次好的结果获得第2级，以此类推。弗里德曼检验采用公式(21)计算:

公式（21）

其中，

公式（22）

其中，是数据集的个数，是算法的个数，是算法在所有数据集上的平均秩。

带和自由度的费雪分布被用于弗里德曼检验。另一方面,弗里德曼的零假设测试意味着所有方法执行等同于显著性水平。因此，当的计算值小于临界值时，我们接受原假设。否则，弗里德曼检验将拒绝原假设。需要注意的是，实验结果展示的是使用了三种不同的数据集和十种推荐方法(即4.4节)。因此，在实验中将参数和分别设为。此外，将显著性水平的值设为0.05。且的费雪分布的临界值等于。表11给出了基于不同评价指标的推荐方法之间的弗里德曼检验结果。需要注意的是，这些结果是根据表1-10中的结果计算出来的。由表11可知，的计算值均大于所有评价指标的临界值2.46。因此，拒绝原假设，可以得出这些结果具有统计显著性的结论。

图片包含 文字, 地图

描述已自动生成

**图4-4 参数对系统性能的影响：（a）所有用户的精确度，（b）冷用户的精确度，（c）所有用户的召回，（d）冷用户的召回，（e）所有用户的F1，（f）冷用户的F1**

**表4.11 基于不同的评价指标的弗里德曼检验结果比较**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Measures |  |  |  | Significance |
| Mae | 24.81 | 22.65 | 2.46 | + |
| RMSE | 24.01 | 16.06 | 2.46 | + |
| Precision | 24.27 | 17.78 | 2.46 | + |
| Recall | 25.72 | 40.19 | 2.46 | + |
| F1 | 25.63 | 37.42 | 2.46 | + |

5 结论和未来工作

推荐系统是信息检索、信息过滤等众多研究领域的重要工具。虽然在针对推荐系统方面已经开发了大量的算法去实现，但是这些算法仍然存在着一些不足的地方。本文提出了一种全新的基于自适应结论选择机制的社交推荐算法，用以提高推荐系统在预测准确性和近邻选择方面的性能。通过对在三个真实数据集的实验结果表明，在大多数情况下，本文所提出的算法是优于其他最先进的方法。未来的工作将着眼于将额外的信息，如用户之间的不信任声明，纳入到我们提出的基于自适应近邻选择机制算法中，用以提高推荐系统的性能。此外，如果通过考虑模糊概念和社区检测技术来识别用户的初始近邻集，可以提高该算法的性能。最后，本文提出的算法可以应用于其他类型的推荐系统，如上下文感知推荐系统中，以提高这些系统的性能。

**参考文献**

[1] Alahmadi, D. H. , & Zeng, X. J. (2015). ISTS: Implicit social trust and sentiment based approach to recommender systems. Expert Systems with Applications, 42 , 8840-8849.

[2] Bahmani, B. , Kumar, R. , & Vassilvitskii, S. (2012). Densest subgraph in streaming and mapreduce. In The 38th international conference on very large data bases: Vol. 5 (pp. 454-465) .

[3] Bedi, P. , & Vashisth, P. (2014). Empowering recommender systems using trust and argumentation. Information Sciences, 279 , 569-586 .

[4] Bilge, A. , & Polat, H. (2013). A scalable privacy-preserving recommendation scheme via bisecting k-means clustering. Information Processing & Management, 49 , 912-927 .

[5] Birtolo, C. , & Ronca, D. (2013). Advances in clustering collaborative filtering by means of fuzzy C-means and trust. Expert Systems with Applications, 40 , 6997-7009 .

[6] Cechinel, C. , Sicilia, M. A. , Alonso, S. S. , & Barriocanal, E. G. (2013). Evaluating collaborative filtering recommendations inside large learning object reposito- ries. Information Processing & Management, 49 , 34-50 .

[7] Deng, S. , Huang, L. , & Xu, G. (2014). Social network-based service recommendation with trust enhancement. Expert Systems with Applications, 41 , 8075-8084 .

[8] Domingues, M. A. , Jorge, A. M. , & Soares, C. (2013). Dimensions as virtual items: improving the predictive ability of top-N recommender systems. Information Processing &Management, 49 , 698-720 .

[9] Fang, H. , Guo, G. , & Zhang, J. (2015). Multi-faceted trust and distrust prediction for recommender systems. Decision Support Systems, 71 , 37-47 .

[10] Friedman, M. (1937). The use of ranks to avoid the assumption of normality implicit in the analysis of variance. Journal of the American Statistical Association, 32 , 675-701 .

[11] Guo, G. , Zhang, J. , & Smith, N. Y. (2015a). Leveraging multiviews of trust and similarity to enhance clustering-based recommender systems. Knowledge-Based Systems, 74 , 14-27 . Guo, G. , Zhang, J. , & Smith, N. Y. (2015b). TrustSVD: Collaborative filtering with both the explicit and implicit influence of user trust and of item ratings. In Proceedings of the 29th AAAI conference on artificial intelligence (pp. 123-129) .

[12] Guo, G. , Zhang, J. , & Thalmann, D. (2014). Merging trust in collaborative filtering to alleviate data sparsity and cold start. Knowledge-Based Systems, 57 , 57-68 .

[13] Herlocker, J. L. , Konstan, J. A. , Terveen, L. G. , & Riedl, J. T. (2004). Evaluating collaborative filtering recommender systems. ACM Transactions on Information Systems, 22 , 5-53 . Hernando, A. , Bobadilla, J. , Ortega, F. , & Tejedor, J. (2013). Incorporating reliability measurements into the predictions of a recommender system. Information Sciences, 218 , 1-16 .

[14] Hernando, A. , Bobadilla, J. , & Ortega, F. (2016). A non negative matrix factorization for collaborative filtering recommender systems based on a Bayesian probabilistic model. Knowledge-Based Systems, 97 , 188-202 .

[15] Hofmann, T. (2004). Latent semantic models for collaborative filtering. ACM Transactions on Information Systems, 22 , 89-115 .

[16] Hong, F. X. , Zheng, X. L. , & Chen, C. C. (2016). Latent space regularization for recommender systems. Information Sciences, 360 , 202-216 .

[17] Jamali, M. , & Ester, M. (2010). A matrix factorization technique with trust propagation for recommendation in social networks. In Proceedings of the 4th ACM conference on eecommender systems (RecSys) (pp. 135-142) .

[18] Javari, A. , & Jalili, M. (2015). A probabilistic model to resolve diversity-accuracy challenge of recommendation systems. Knowledge and Information Systems, 44 , 609-627 .

[19] Kassak, O. , Kompan,M. , & Bielikova, M. (2016). Personalized hybrid recommendation for group of users: Top-N multimedia recommender. Information Processing & Management,52,459-477.

[20] Kim, Y. A. , & Phalak, R. (2012). A trust prediction framework in rating-based experience sharing social networks without a web of trust. Information Sciences, 191 , 128-145 .

[21] Lee, S. , Kahng, M. , & Lee, S. G. (2015). Constructing compact and effective graphs for recommender systems via node and edge aggregations. Expert Systems with Applications, 42 , 3396-3409 .

[22] Liu, H. , Hu, Z. , Mian, A. , Tian, H. , & Zhu, X. (2014). A new user similarity model to improve the accuracy of collaborative filtering. Knowledge-Based Systems, 56 , 156-166 .

[23] Ma, B. , Zhang, N. , Liu, G. , Li, L. , & Yuan, H. (2016). Semantic search for public opinions on urban affairs: A probabilis tic topic modeling-based approach. Information Processing & Management, 52 , 430-445 .

[24] Massa, P. , & Avesani, P. (2007). Trust-aware recommender systems. ACM conference on recommender systems .

[25] Moon, H. S. , Kim, J. K. , & Ryu, Y. U. (2013). A sequence-based filtering method for exhibition booth visit recommendations. International Journal of Information Management, 33 , 620-626 .

[26] Moradi, P. , Ahmadian, S. , & Akhlaghian, F. (2015). An effective trust-based recommendation method using a novel graph clustering algorithm. Physica A, 436 , 462-481 .

[27] OMahony, M. P. , Hurley, N. J. , & Silvestre, G. C. M. (2005). Recommender systems: Attack types and strategies. In Proceedings of the 20th national conference on American association for artificial intelligence (AAAI-05) (pp. 334-339) .

[28] Park, Y. , Park, S. , Jung, W. , & Lee, S. (2015). Reversed CF: A fast collaborative filtering algorithm using a k-nearest neighbor graph. Expert Systems with Applications, 42 , 4022-4028 .

[29] Pereira, A. L. V. , & Hruschka, E. R. (2015). Simultaneous co-clustering and learning to address the cold start problem in recommender systems. Knowl- edge-Based Systems, 82 , 11-19 .

[30] Ricci, F. , Rokach, L. , Shapira, B. , & Kantor, P. B. (2011). Recommender systems handbook . Springer Science, Business Media .

[31] Shani, G. , Brafman, R. I. , & Heckerman, D. (2005). An MDP-based recommender system. Journal of Machine Learning Research, 6 , 1265-1295 .

[32] Tsa, C. I,F. , & Hung, C. (2012). Cluster ensembles in collaborative filtering recommendation. Applied Soft Computing, 12 , 1417-1425 .

[33] Tsai, C. Y. , & Lai, B. H. (2015). A Location-Item-Time sequential pattern mining algorithm for route recommendation. Knowledge-Based Systems, 73 , 97-110 .

[34] Yan, S. , Zheng, X. , Chen, D. , & Wang, Y. (2013). Exploiting two-faceted web of trust for enhanced-quality recommendations. Expert Systems with Applications, 40 , 7080-7095 .

[35] Yang, B. , Lei, Y. , Liu, D. , & Liu, J. (2013). Social collaborative filtering by trust. In Proceedings of the 23th international joint conference on artificial intelligence (IJCAI) (pp. 2747-2753) .

[36] Yang, X. , Guo, Y. , & Liu, Y. (2013). Bayesian-inference-based recommendation in online social networks. IEEE Transactions on Parallel and Distributed Systems, 24 , 642-651 .

[37] Yuan, W. , Guan, D. , Lee, Y. K. , Lee, S. , & Hur, S. J. (2010). Improved trust-aware recommender system using small-worldness of trust networks. Knowl- edge-Based Systems, 23 , 232-238 .

本文译自:

Sajad Ahmadian,Majid Meghdadi,Mohsen Afsharchi. A social recommendation method based on an adaptive neighbor selection mechanism[J]. Information Processing and Management,2018,54(4).