一种提高推荐多样性的概率选择模型

张 东 蔡国永 夏彬彬
(桂林电子科技大学计算机科学与工程学院 桂林 541004)

摘 要 传统的推荐算法多以优化推荐列表的精确度为目标,而忽略了推荐算法的另一个重要指标——多样性。提出了一种新的提高推荐列表多样性的方法。该方法将列表生成步骤转换为 N 次概率选择过程,每次概率选择通过两个步骤完成:类型选择与项目选择。在类型选择中,引入项目的类型信息,根据用户对不同项目类型的喜好计算概率矩阵,并依照该概率矩阵选择一个类型;在项目选择中,根据项目的预测评分、项目的历史流行度、项目的推荐流行度3个因素重新计算项目的最终得分,选择得分最高的项目推荐给用户。通过阈值 TR 来调节多样性与精确度之间的折中。最后,通过对比实验证明了该方法的有效性。

关键词 推荐系统,多样性,Top-N推荐,概率选择
中图法分类号 TP391 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.2.016

Improving Recommendation Diversity via Probabilistic Selection

ZHANG Dong CAI Guo-yong XIA Bin-bin
(School of Computer Science and Engineering, Guilin University of Electronic Technology, Guilin 541004, China)

Abstract Typical recommendation algorithms focus on optimizing the accuracy of recommendation lists, however, diversity is also considered as a key property to measure the quality of recommendation lists from both user and system perspective. Many list diversification techniques improve diversity by re-ranking items. In this paper, a new probabilistic selection model for improving the diversity of recommendation lists was proposed. This model transfers the list generation process to N-times probabilistic selection process, and each selection includes two steps: genre selection and item selection. For the genre selection phase, genre information of items is included to compute user-genre probabilistic matrix, and a genre is chosen based on this matrix. For the item selection phase, three properties including estimated score of items, historical popularity of items, and recommending popularity of items are considered for item re-scoring. The item with the highest re-computed score will be selected into the recommendation list. The trade-off between diversity and accuracy can be controlled by changing threshold value TR. Experiments on two movie recommendation datasets show that our model can effectively improve recommendation lists diversity. At the same time, the comparative experiments show that our model outperforms re-ranking method in almost all experimental results, except the case of individual diversity for matrix factorization.

Keywords Recommender system, Diversity, Top-N recommendation, Probabilistic selection

1 引言

通过分析用户的历史行为数据,推荐系统可以从大量的信息空间中找到用户可能会感兴趣的内容,然后将其整合到列表中推荐给用户。现有的推荐算法大多以提高推荐列表的精确度(如准确率、召回率等)为目的。然而,推荐列表的高精确度并不一定意味着高水平的用户满意度,也不一定能提高商品的销量[2,13]。这是由于推荐给用户的列表长度远远小于系统中所有项目的个数,以精确度为衡量指标的推荐算法一方面倾向于推荐流行的项目,而那些被少部分用户购买的、目标用户会喜欢的项目难以被推荐[1];另一方面,产生的推荐列表中的项目与用户过去的行为太相似,从而不能迎合用户的广泛喜好[2,4]。文献[3]的研究表明,推荐系统的使用反而会降低销售的多样性,不利于商业模型。提高推荐列表的多样性有助于解决以上问题。

一般而言,推荐算法给用户推荐项目分为两个步骤:预测用户对项目的评分(评分预测);从已有预测评分的项目中选择 N 个项目推荐给用户(列表生成)。相应地,提高推荐列表多样性的方法也可以分为两类:一类方法在评分预测步骤中进行[5-7],通过考虑多种因素来产生多样化的预测评分;另一类方法在列表生成步骤中进行[8-12],该类方法在已有预测评分的基础上选择多样化的推荐列表。本文方法属于第二类,

到稿日期:2015-05-12 返修日期:2015-06-23 本文受国家自然科学基金(61063039),广西高校高水平创新团队及卓越学者计划,广西可信软件重点实验室基金(kx201202)资助。

张 东(1992-),男,硕士,CCF学生会员,主要研究方向为推荐系统,E-mail:mengjianzhizi@gmail.com;蔡国永(1971-),男,博士,教授,CCF高级会员,主要研究方向为社交媒体分析;夏彬彬(1990-),男,硕士,主要研究方向为自然语言处理。

与第一类方法相比，第二类方法的一个优点是它可以与其他任何具有评分预测步骤的推荐算法相结合，以提高其推荐列表的多样性。

然而，正如文献[5]指出，推荐列表多样性的提高会伴随着精确度的下降，如何在让精确度的下降程度保持在可接受范围内的情况下尽可能地提高推荐列表的多样性是需要进一步研究的工作。针对此问题，本文提出了一种提高推荐列表多样性的概率选择模型。本文第2节介绍推荐列表多样性的研究现状；第3节详细描述本文模型以及具体实现算法；第4节进行实验分析；最后对本文工作加以总结。

2 推荐多样性的研究现状

2.1 多样性的评价指标

推荐列表的多样性可以体现在两方面：个体多样性与整体多样性。前者衡量单个用户推荐列表内的多样性；后者衡量所有用户推荐列表的多样性。对于个体多样性，可以使用Ziegler等人[8]提出的列表中所有项目对之间的相似度来衡量，表示为Intra-List Similarity (ILS)。设$L(u)$表示给用户u推荐的列表，则用户u的推荐列表的个体多样性为：

$$ILS(L(u))=\frac{1}{|L(u)|(|L(u)|-1)}\sum_{i,j\in L,i\neq j}sim(i,j) \quad (1)$$

其中，sim表示项目i和j之间的相似度函数，本文使用余弦相似度并使用用户对项目的打分信息来计算。系统的个体多样性即所有用户的个体多样性的平均值。显然，ILS值越高表示系统给用户推荐列表的个体多样性越低。

对于整体多样性，Adomavicius等人[9]提出了使用推荐给所有用户的不同项目的个数来衡量，表示为$Aggr$：

$$Aggr(S)=|\bigcup_{u\in U}L(u)| \quad (2)$$

其中，S表示整个系统。该方法能直观有效地衡量整体多样性，但是其并不区分项目被推荐的次数。从商业模型角度来看，平衡的商品销售有助于获得更大的利润，因此衡量项目被推荐的次数同样重要。使用$Gini$系数可以进一步衡量不同项目被推荐的次数的销售平衡：

$$Gini(S)=\frac{2}{N-1}\sum_{i=1}^{N}(N+1-i)\frac{rec(i)}{total} \quad (3)$$

其中，N是候选商品的总个数，$rec(i)$是项目i被推荐给不同用户的次数，$total$是推荐给所有用户的不同商品个数。与文献[1,12]相似，本文也使用改进的$Gini$系数，$Gini$值越高表示整体多样性越高。

2.2 提高多样性的方法

针对推荐列表的多样性问题，已经有许多研究者提出了不同的方法来提高推荐列表的多样性。根据长尾效应描述[17]，大部分的评论信息都集中在少量的畅销商品中，而剩余的大部分非畅销的商品仅有很少的评论信息，这部分非畅销的商品即长尾商品。研究发现，通过优先推荐长尾商品能有效提高推荐的多样性。

如引言介绍，一些研究者设计的算法在评分预测步骤提高多样性。周涛等人[5]分析了多样性与精确度之间的困境，提出了一种面向多样性的热传导算法(head-spreading)，并且将该算法与面向精确度的概率传播算法(probabilistic spreading)结合，以调节精确度与多样性之间的折中；Park等人[6]提出了一种新型的聚类方法来增加长尾商品的预测评分；Yin等人[7]提出了基于随机游走的方法来增加长尾商品被推荐的概率。与这些研究相对的是，另外一些研究者在算法的列表生成步骤提高多样性。其中，被广泛使用的是重排序方法。Adomavicius等人[9]提出了一种基于项目流行度的重排序方法，并证明了该方法能在降低少量精确度的同时有效提高推荐列表的多样性。标准的列表生成步骤将项目按照预测评分降序排序，然后选择排序最靠前的N个项目推荐给用户以最大化精确度，该方法的不足是会导致流行度高的商品排在许多用户的Top-N列表中，从而使得整体多样性很低。Adomavicius等人发现改变项目的排序有助于使得更多的项目排在列表前N位，可避免按照评分排序导致的问题，并通过实验证明了重排序方法能有效提高推荐列表的多样性。此外，Vargas S等人[15]也提出了一种新的衡量多样性的指标，并设计了一种基于该指标的重排序方法。

虽然重排序方法[9]能有效提高推荐列表的多样性，但是其在重排序项目时仅考虑项目的流行度，不考虑项目预测评分等相关因素，精确度丢失依旧很大。与具有代表性的重排序方法[9]相比，本文方法不仅同时考虑了项目的预测评分与项目类型，而且将流行度进一步区分为历史流行度与推荐流行度，能在提高多样性的同时更好地保持精确度。此外，与Adomavicius等人的方法类似，本文也将模型应用到预测评分大于TR的项目中，剩余项目使用标准列表生成步骤，以调节精确度与多样性之间的折中。

3 概率选择模型

大多数用户在同一领域内有着不同的喜好，例如在电影领域，有的用户喜欢喜剧，有的用户喜欢悲剧。由于推荐给用户的列表非常短，远远小于项目的总个数，面向精确度的标准列表生成算法不能很好地处理这些不同喜好。一方面，用户可能同时喜欢喜剧、动作、恐怖等类型的电影，但是被推荐的列表中可能全部都是动作片，而其他类型的电影由于预测评分不在最高的N个内而得不到推荐。另一方面，项目的预测评分值的范围一般是固定的(例如1－5)。评分预测结果中最大分数(5分)的项目个数可能大于N，由于只推荐其中的N个项目给用户，剩余具有相同分数的项目得不到推荐。并且，推荐算法往往推荐相同的那几个最大分数的项目给很多用户，因为排序的规则是固定的，例如冒泡排序。因此，根据上述分析，考虑用户对不同项目类型的不同喜好并且重新计算项目的预测评分有助于更好地解决推荐列表的多样性问题。

3.1 算法总体描述

针对推荐列表的多样性问题，本文引入项目类型信息，提出一种概率选择模型(Probabilistic Selection)，将推荐列表的生成过程分为两步：类型选择与项目选择。在类型选择步骤中，根据用户对不同类型的不同程度的喜好来按概率选择类型；在项目选择步骤中，重计算项目的最终得分并选择得分最高的项目。最终，将推荐列表的生成过程转换为如下概率选择过程：

根据概率矩阵M_u选择一个类型g，从类型g中选择最终得分最大的项目i；重复N次，直到得到要推荐的N个项目。

其中，N为推荐列表长度，即推荐给用户的项目个数；M_u为用户-类型概率矩阵，决定用户u以多大的概率选择类型g。

要实现上述概率选择过程需要计算概率矩阵M_u以及重计算项目的最终得分，两者的计算将分别在3.2节和3.3节中详细介绍。

此外，设定一个阈值TR，将上述概率选择过程应用到预测评分大于TR的项目中，而预测评分小于TR的项目则按照标准的列表生成步骤进行选择，这样可以调节多样性与精确度之间的折中。设置最大的TR值，模型将会退化为标准的列表生成步骤。随着TR值的减小，用户将会得到多样性更高而精确度降低的推荐列表。

设U为用户集合，I为具有预测评分的候选项目集合，A为用户-项目评分矩阵，G为项目类型集合，$I^H(u)$为用户u的候选项目集中预测评分大于阈值TR的项目集合，N为给每个用户推荐的项目个数，L为给所有用户的最终推荐列表，则概率选择模型算法的具体描述如下。

算法1 *Probabilistic Selection*

输入：I；A；G；TR

输出：L

第一步：根据评分矩阵A计算用户-类型概率矩阵M_u；

第二步：为每个用户生成列表，具体如下：

FOR each u in U

1. 从I(u)中提取预测评分大于TR的项目到$I^H(u)$；
2. 重计算$I^H(u)$中项目的得分；
3. WHILE (L(u)的长度小于N并且$I^H(u)$的长度大于0)：
   ①根据概率矩阵M_u依概率从G中选择一个类型g；
   ②如果$I^H(u)$中存在属于类型g的项目，则从$I^H(u)$中选择具有最大重计算得分并且属于g的项目i；否则从$I^H(u)$中选择具有最大重计算得分的项目i；
   ③将项目i添加到L(u)中；
   ④从$I^H(u)$中移除项目i；
   END WHILE
4. 判断L(u)的长度，如果长度小于N，则从I(u)中选择剩余的具有最大预测评分的项目添加到L(u)中填满；

END FOR

算法1中，第一步M_u的计算将在3.2节详细介绍，第二步中项目得分的重计算将在3.3节中详细介绍。

3.2 用户-类型概率矩阵计算

用户选择一个类型的概率反映了该用户对该类型的喜欢程度。一般而言，如果一个用户对某一类型中的项目评分很高，并且经常评论该类型中的项目，那么显然该用户喜欢该类型。因此，给定用户集合U与项目类型集合G，可以认为用户$u\in U$选择类型$g\in G$的概率（表示为$p_{u,g}$）由两部分决定：(1)由用户u对项目$i\in g$的评分决定的概率，表示为$p^r_{u,g}$；(2)由用户u对类型g中项目的评论频率决定的概率，表示为$p^f_{u,g}$。$p_{u,g}$的计算式如下：

$$p_{u,g}=\frac{1}{2}(p^r_{u,g}+p^f_{u,g}) \tag{4}$$

其中，概率$p^r_{u,g}$的计算式如下：

$$p^r_{u,g}=\frac{s^r_{u,g}}{\sum_{g\in G}s^r_{u,g}},\ s^r_{u,g}=\sum_{i\in g}(R_{u,i}-\overline{R}_u) \tag{5}$$

其中，$R_{u,i}$为用户u对项目i的评分，$\overline{R}_u$为用户u的平均评分，G为项目类型集合，$s^r_{u,g}$为由用户u对类型g中项目的评分决定的得分。为了保证计算得到的概率值为正，在计算$p^r_{u,g}$前，将所有$s^r_{u,g}$值加上可能为负数的最小的$s^r_{u,g}$值，即$s^r_{u,g}=s^r_{u,g}+\min(s^r_{u,g})$。类似地，概率$p^f_{u,g}$的计算式如下：

$$p^f_{u,g}=\frac{s^f_{u,g}}{\sum_{g\in G}s^f_{u,g}},\ s^f_{u,g}=k_{u,g} \tag{6}$$

其中，$k_{u,g}$为用户u评论过的类型g中的项目个数，$s^f_{u,g}$为由用户u对类型g的评论频率决定的得分。

为每个用户都计算概率$p_{u,g}$，最终可以得到用户-类型概率矩阵M_u，表示如下：

$$M_u=\begin{pmatrix} p_{1,1},p_{1,2},\cdots,p_{1,n} \\ \cdots \\ p_{m,1},p_{m,2},\cdots,p_{m,n}\end{pmatrix} \tag{7}$$

其中，行表示用户，列表示项目类型，m为用户个数，n为项目类型个数，矩阵中的元素为不同用户选择不同类型的概率。矩阵中每一行的和为1，因此用户必然会选择一个类型。

3.3 项目得分重计算

文献[9]将项目的流行度定义为评论过该项目的用户个数，并证明了推荐流行度低的项目能有效提高推荐列表的整体多样性。类似地，本文模型也考虑了项目的流行度。不同的是，我们进一步将流行度分为两类：历史流行度与推荐流行度。前者表示历史数据中评论过该项目的用户个数，后者表示在推荐过程中已经被推荐了该项目的用户个数。

直观地，标准的列表生成算法选择预测评分最大的项目推荐给用户以最大化精确度，因此保持预测评分高的项目被推荐的概率能保持推荐列表的精确度。同时，优先推荐长尾项目（历史流行度低）能有效提高整体多样性。此外，如果一个项目已经被推荐给了很多用户，那么降低其再次被推荐的概率也有助于提高整体多样性。因此，我们根据3个属性（项目的预测评分、历史流行度、推荐流行度）重新计算项目的得分。

给定用户u的候选项目集合$G(u)$，定义$\hat{G}(u)$如下：

$$\hat{G}(u):=\{(\hat{R}_{u,i},P^H_i,P^R_i)\mid i\in G(u)\} \tag{8}$$

其中，$\hat{R}_{u,i}$为用户u对项目i的预测评分，P^H_i为项目i的历史流行度，P^R_i为项目i的推荐流行度。P^H_i和P^R_i的计算式如下：

$$\begin{aligned} &P^H_i=|U^H(i)|,\text{当}\ U^H(i)=\{u\in U\mid \exists R_{u,i}\}\ \text{时} \\ &P^R_i=|U^R(i)|,\text{当}\ U^R(i)=\{u\in U\mid i\in L(u)\}\text{时}\end{aligned} \tag{9}$$

基于$\hat{G}(u)$，用户u对项目i的最终得分计算式如下：

$$score(i)=\begin{cases}\dfrac{1}{P^R_i+1}\left(\dfrac{\hat{R}_{u,i}-\mu^{\hat{R}_{u,i}}}{\delta^{\hat{R}_{u,i}}}-\dfrac{P^H_i-\mu^{P^H_i}}{\delta^{\hat{R}_{u,i}}}\right), & \hat{R}_{u,i}\geqslant TR \\ \hat{R}_{u,i}, & \text{else}\end{cases} \tag{10}$$

其中，μ^*与δ^*分别为均值与标准差。从上式可以看出，预测评分高、历史流行度低、推荐流行度低的项目将会得到更高的得分。此外，只对大于阈值TR的项目重计算得分以控制多样性与精确度之间的折中。

4 实验结果与分析

本文设计了不同的实验来验证所提模型的有效性。实验数据集为MovieLens 1M与Netflix Prize[15]。每个数据集都包含用户对电影的评论信息。对于MovieLens 1M，使用完整的数据集；对于Netflix Prize，由于数据集中不提供电影的分类信息，因此从IMDb上获取电影的分类信息，并且从大量的

原始数据集中随机选择一部分数据进行实验。表 1 描述了两个数据集的详细信息。

表 1　数据集的详细信息

Data set	Users	Movies	Ratings	Genres	Sparsity
MovieLens	6040	3706	1000209	18	4.5%
Netflix	8000	5000	680283	28	1.7%

4.1　初始化

对每个数据集，随机选取其中的 20%作为测试集，剩余的 80%作为训练集。首先，应用 3 种常用的评分预测算法为每个用户预测电影的评分，分别为基于用户的协同过滤(UBKNN)、基于项目的协同过滤(IBKNN)与矩阵分解。对于协同过滤，文献[1]证明了使用非标准化规则可以获得比标准化规则更好的精确度，因此这里也采用非标准化的方法，并且设定邻居个数为 50。对于矩阵分解，采用文献[15]提出的 SVD++，设定因子个数为 50。其次，应用本文提出的模型与重排序方法，在已有预测评分的基础上为每个用户产生推荐列表，列表的长度为 10。两者的精确度与多样性的折中都由阈值 *TR* 控制，通过降低 *TR*，用户将得到多样性更高而精确度更低的推荐列表。最后，使用 2.1 节中的 3 个评价指标(*ILS*、*Aggr*、*Gini*)来衡量推荐结果的多样性，使用准确率(Precisoiin)来衡量精确度。

4.2　实验结果

根据 3 个多样性评价指标，分别将实验结果中多样性与准确率的折中关系描述为 3 部分。其中，PS 代表本文方法，RE 代表重排序方法[9]。

表 2 描述了不同程度的整体多样性(*Aggr*)提高与准确率(*Precision*)降低的关系。表中，Precision Loss 一行表示不同程度的准确率降低，Aggr Gain 两行分别表示重排序方法(RE)与本文方法(PS)在相应准确率降低的情况下整体多样性提高的情况，Standard 一行表示标准列表生成算法的准确率及多样性。从表 2 中可以看出，随着准确率降低程度的增加，推荐列表的整体多样性能获得越来越明显的提高；并且在准确率降低相同的情况下本文方法能获得更大程度的整体多样性增加，明显优于重排序方法。

表 2　不同程度整体多样性(Aggr)提高与准确率降低

Data set	MovieLens 1M								
Methods	UBKNN			IBKNN			SVD++		
Precision Loss	−0.01	−0.05	−0.1	−0.01	−0.05	−0.1	−0.001	−0.005	−0.01
Aggr Gain (RE)	+12	+111	+304	+8	+102	+373	+12	+64	+110
Aggr Gain (PS)	+24	+237	+804	+23	+323	+2226	+74	+389	+733
Standard	Precision:0.268, Aggr:587			Precision:0.224, Aggr:864			Precision:0.067, Aggr:494		
Data set	Netflix								
Methods	UBKNN			IBKNN			SVD++		
Precision Loss	−0.01	−0.05	−0.1	−0.01	−0.05	−0.1	−0.001	−0.005	−0.01
Aggr Gain (RE)	+5	+71	+441	+7	+176	+1373	+92	+107	+132
Aggr Gain (PS)	+12	+297	+2034	+25	+1474	+2873	+1005	+1892	+2145
Standard	Precision:0.192, Aggr:745			Precision:0.169, Aggr:1790			Precision:0.022, Aggr:237		

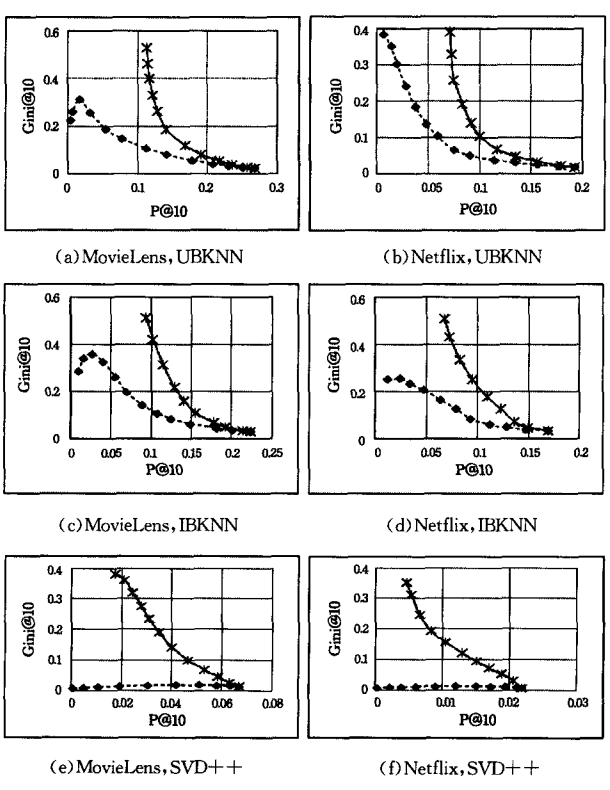

图 1　*Gini* 系数与准确率的关系

图 1 描述了 *Gini* 系数与准确率(Precision)的关系。图中，横坐标表示准确率，纵坐标表示 *Gini* 系数。高的 *Gini* 系数值表示更高的整体多样性与销售平衡。图中曲线由阈值 *TR* 控制，通过降低阈值 *TR*，*Gini* 系数将会升高，相应地准确率会降低。从图 1(a)－图 1(f)中可以看出，随着准确率的降低，应用本文模型与重排序方法的推荐列表的 *Gini* 系数明显增加。而且本文提出的概率选择模型的曲线(实线)在重排序方法的曲线(虚线)的上方，表明在相同准确率的情况下所提方法的 *Gini* 系数优于重排序方法的 *Gini* 系数。

图 2 描述了个体多样性(*ILS*)与准确率(*Precision*)的关系。如 2.1 节所述，低的 *ILS* 值意味着高的个体多样性。从图 2(a)－图 2(f)中可以看出，随着准确率的降低，应用本文模型与重排序方法的推荐列表的个体多样性明显提高。对于基于用户的协同过滤(UBKNN)与基于项目的协同过滤(IBKNN)，本文方法曲线在重排序方法曲线下方，优于重排序方法。对于矩阵分解(SVD++)，本文方法表现不佳，这是由于矩阵分解会为所有用户未看过的电影预测评分，因此重排序算法与本文算法的输入项目候选集非常大。基于流行度进行重排序的方法推荐流行度最低的电影，因此个体多样性高，但是同时也推荐许多相同的流行度最低的电影给所有用户，从而造成整体多样性与销售平衡很低(如表 2 与图 1 所示)。从图 2(e)、图 2(f)可以看出，当准确率下降较少时本文方法与重排序方法效果基本相同；当准确率下降逐渐增大时，重排序方法能发现推荐流行度更低的电影(可能从没有人看过，与其他电影相似度为 0)，因此个体多样性很高，而本文方

法因同时考虑多方面与用户相关的因素，所以个体多样性相对较低，但是同时会获得更好的整体多样性（*Aggr*）与销售平衡（*Gini*）。对于协同过滤，由于它只给与用户过去看过电影相关的电影预测评分，因此考虑了多方面因素的本文方法显然优于重排序方法。

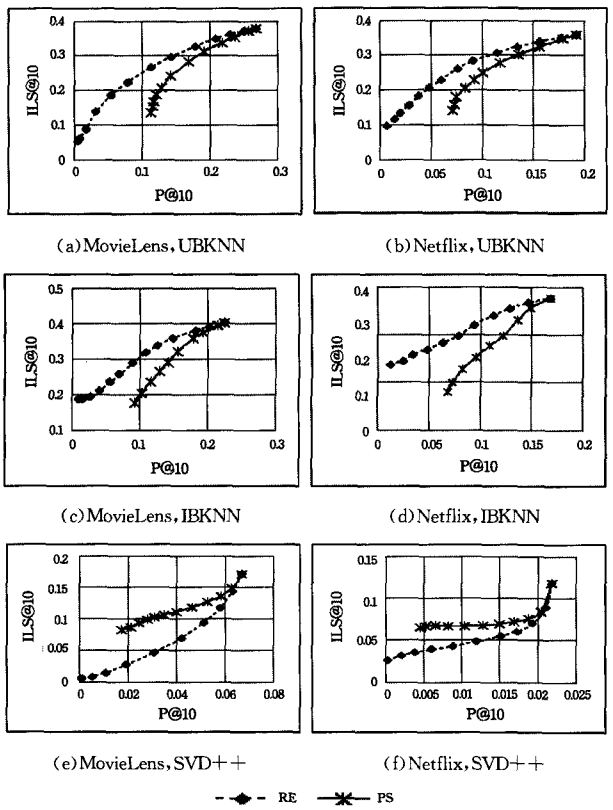

图 2　个体多样性（*ILS*）与准确率的关系

4.3　复杂度分析

从算法 1 中可以发现，在产生推荐列表前，需要计算用户-类型矩阵，在推荐过程中，也需要为每个用户分别重计算一次候选项目的得分，因此对算法进行复杂度分析是有必要的。假设用户评论商品的平均数量为 k，项目候选集的平均长度为 n，推荐列表长度为 N，则算法 1 为一个用户产生推荐列表的时间复杂度为 $O(mk+n+N*n)$，第一项 mk 表示计算用户-类型矩阵的复杂度，第二项 n 表示重计算项目得分的复杂度，第三项 $N*n$ 表示 N 次循环产生推荐列表的复杂度。然而一般情况下，用户-类型矩阵的计算可以线下进行，因此实际推荐过程中的时间复杂度为 $O(nN)$。重排序方法[9]使用快速排序为一个用户产生推荐列表的时间复杂度为 $O(n\log n)$。

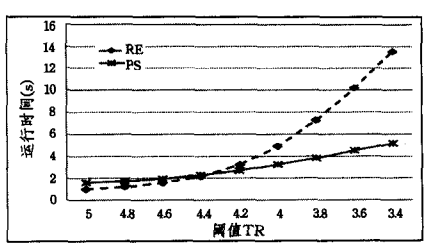

图 3　运行时间

图 3 描述了在同样的实验环境下，本文算法与重排序算法在 UBKNN 预测评分基础上为所有用户产生推荐列表的运行时间（从算法输入数据到产生推荐列表），数据集为 MovieLens 1M，本文算法需要的用户-类型矩阵已事先计算好。从图中可以看出，当阈值较大时，项目候选集较小，因此 $N>\log n$，本文算法产生推荐列表需要更长的时间。随着阈值的减小，项目候选集的增加导致 $N<\log n$，因此重排序算法的运行时间更长。并且本文算法运行时间的增长速度相对于重排序方法更为缓慢，更适用于大型数据集。

结束语　本文将推荐列表的生成过程转换为概率选择过程，提出了一种新的提高推荐列表多样性的方法。通过引入项目类型信息，根据用户对不同项目类型的喜好选择不同的类型，并选择该类型中预测评分高、历史流行度低与推荐流行度低的项目推荐给用户。不同的实验证明了该方法能有效地提高个体多样性、整体多样性与销售平衡。与广泛使用的重排序技术相比，本文所提模型考虑了多方面的因素，在提高多样性的同时能更好地保持精确度。如何更好地提高矩阵分解中的个体多样性是需要深入研究的问题。此外，引入更多的用户与项目的信息以更好地满足用户的广泛喜好也值得进一步研究。

参 考 文 献

[1] Cremonesi P, Koren Y, Turrin R. Performance of recommender algorithms on top-n recommendation tasks[C]// Proc of the fourth ACM Conf on Recommender systems. ACM, 2010: 39-46

[2] McNee S M, Riedl J, Konstan J A. Being accurate is not enough: how accuracy metrics have hurt recommender systems[C]// CHI'06 extended abstracts on Human factors in computing systems. ACM, 2006: 1097-1101

[3] Fleder D, Hosanagar K. Blockbuster culture's next rise or fall: The impact of recommender systems on sales diversity[J]. Management Science, 2009, 55(5): 697-712

[4] Adamopoulos P, Tuzhilin A. On over-specialization and concentration bias of recommendations: Probabilistic neighborhood selection in collaborative filtering systems[C]// Proc of the 8th ACM Conf on Recommender systems. ACM, 2014: 153-160

[5] Zhou T, Kuscsik Z, Liu J G, et al. Solving the apparent diversity-accuracy dilemma of recommender systems[J]. Proc of the National Academy of Sciences, 2010, 107(10): 4511-4515

[6] Park Y J, Tuzhilin A. The long tail of recommender systems and how to leverage it[C]// Proc of the 2008 ACM conf on Recommender Systems. ACM, 2008: 11-18

[7] Yin H, Cui B, Li J, et al. Challenging the long tail recommendation[J]. Proc of the VLDB Endowment, 2012, 5(9): 896-907

[8] Ziegler C N, McNee S M, Konstan J A, et al. Improving recommendation lists through topic diversification[C]// Proc of the 14th Int Conf on World Wide Web. ACM, 2005: 22-32

[9] Adomavicius G, Kwon Y O. Improving aggregate recommendation diversity using ranking-based techniques[J]. IEEE Trans on Knowledge and Data Engineering, 2012, 24(5): 896-911

[10] Vargas S, Castells P. Rank and relevance in novelty and diversity metrics for recommender systems[C]// Proc of the fifth ACM Conf on Recommender Systems. ACM, 2011: 109-116

[11] Adomavicius G, Kwon Y. Maximizing aggregate recommendation diversity: A graph-theoretic approach[C]// Proc. of the 1st Int Workshop on Novelty and Diversity in Recommender Systems (DiveRS 2011). 2011: 3-10

[12] Vargas S, Castells P. Improving sales diversity by recommending

users to items[C]// Proc of the 8th ACM Conf on Recommender systems. ACM, 2014: 145-152

[13] Cremonesi P, Garzotto F, Negro S, et al. Looking for "good" recommendations: A comparative evaluation of recommender systems[M]// Human-Computer Interaction-INTERACT 2011. Springer Berlin Heidelberg, 2011: 152-168

[14] Bennett J, Lanning S. The netflix prize[C]// Proc of KDD cup and workshop. 2007: 35

[15] Vargas S, Baltrunas L, Karatzoglou A, et al. Coverage, redundancy and size-awareness in genre diversity for recommender systems[C]// Proc of the 8th ACM Conf on Recommender systems. ACM, 2014: 209-216

[16] Koren Y, Bell R, Volinsky C. Matrix factorization techniques for recommender systems[J]. Computer, 2009, 42(8): 30-37

[17] Anderson C. The long tail: Why the future of business is selling less of more[M]. Hyperion, 2006

(上接第 50 页)

表 4 基于 OpenMP 的 MNF 并行程序执行时间(s)

数据规模	滤波	协方差	噪声协方差特征值	原始变换矩阵特征值	MNF 变换	总时间
614 * 512	0.44	3.05	0.21	0.21	0.30	4.20
614 * 1087	0.90	6.55	0.23	0.22	0.59	8.48
753 * 1924	1.77	13.62	0.25	0.26	1.28	17.17
781 * 6955	6.39	49.53	0.23	0.23	4.77	61.13

表 5 基于 CPU/GPU 的 MNF 并行程序执行时间(s)

数据规模	滤波	协方差	噪声协方差特征值	原始变换矩阵特征值	MNF 变换	总时间
614 * 512	0.94	0.54	0.31	0.29	0.03	2.11
614 * 1087	0.99	1.11	0.28	0.24	0.06	2.65
753 * 1924	1.02	2.41	0.31	0.24	0.12	4.10
781 * 6955	1.33	9.40	0.28	0.25	0.44	11.71

表 6 基于 OpenMP 的 MNF 并行程序加速比

数据规模	滤波加速比	协方差加速比	MNF 变换加速比	总时间加速比
614 * 512	18.34	7.2	47.76	10.54
614 * 1087	21.91	7.51	54.38	11.84
753 * 1924	21.64	8.17	54.09	12.66
781 * 6955	20.99	8.12	54.22	12.82

表 7 基于 CPU/GPU 的 MNF 并行程序加速比

数据规模	滤波加速比	协方差加速比	MNF 变换加速比	总时间加速比
614 * 512	8.59	40.70	481.65	20.74
614 * 1087	20.36	44.35	569.30	36.63
753 * 1924	37.33	46.23	595.63	50.61
781 * 6955	100.87	42.80	591.28	61.80

从图 6 看出，两种并行方案均取得一定的加速比，其中基于 CPU/GPU 的方案加速比增长随着数据规模的增大尤为明显，这说明了本文提出的并行方案的有效性。OpenMP 加速比的增幅较平稳，主要是由于 GPU 的线程号和数据号一致，在取数据时，一次能取出 cacheline 的数据，可确保访存对齐，且 OpenMP 核数有限。相对于 OpenMP 并行方案，基于 CPU/GPU 异构模式取得的加速比更理想，而且数据规模越大，总加速比越大，其中最大总加速比为 61.8。这验证了 CPU/GPU 异构模式相对于传统同构模式在处理大规模数据上的并行性能优势。

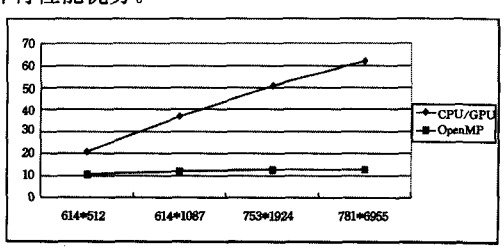

图 6 MNF 降维总加速比(CPU/GPU、OpenMP)

结束语 高光谱遥感影像处理属于计算量大、计算复杂、迭代数多的计算任务。采用异构并行手段能够在保证图像处理精度的同时，有效提高图像的处理速度。本文介绍了高光谱遥感和 CPU/GPU 异构计算模式，并阐述了基于 CPU/GPU 异构模式的高光谱遥感数据处理研究现状和问题，最后面向共享存储型小型桌面超级计算机，引入共享存储的 OpenMP 同构模式作为实验对比基础，设计并实现了基于 CPU/GPU 和 OpenMP 的高光谱遥感影像 MNF 降维的并行算法。实验取得了良好的加速比，其中 CPU/GPU 最大总加速比为61.8，OpenMP 最大总加速比为 12.82，验证了异构模式在高光谱遥感处理领域的发展潜力。

参 考 文 献

[1] Su Hong-jun, Sheng Ye-hua, Yang He, et al. Orthogonal Projection Divergence-Based Hyperspectral Band Selection[J]. Spectroscopy and Spectral Analysis, 2011, 31(5): 1309-1313(in Chinese)
苏红军，盛业华，Yang He，等. 基于正交投影散度的高光谱遥感波段选择算法[J]. 光谱与光谱学分析，2011，31(5)：1309-1313

[2] Tang Yuan-yuan, Zhou Hai-fang, Fang Min-quan, et al. Hyperspectral Remote Sensing Image Data Processing on GPU[J]. Information Security and Technology, 2015, 6(4): 148-152(in Chinese)
汤媛媛，周海芳，方民权，等. 基于 GPU 的高光谱遥感影像数据处理[J]. 信息安全与技术，2015，6(4)：148-152

[3] Ju Tao, Zhu Zheng-dong, Dong Xiao-she. The Feature, Programming Model and Performance Optimization Strategy of Heterogeneous Many-Core System: A Review[J]. Acta Electronica Sinica, 2015, 43(1): 111-119(in Chinese)
巨涛，朱正东，董小社. 异构众核系统及其编程模型与性能优化技术研究综述[J]. 电子学报，2015，43(1)：111-119

[4] 张舒，褚艳利. GPU 高性能运算之 CUDA[M]. 北京：中国水利水电出版社，2009

[5] Setoain J, Tenllado C, Prieto M, et al. Parallel hyperspectral image processing on commodity graphics hardware[C]// 2006 International Conference on Parallel Processing Workshops. 2006: 465-472

[6] Green A A, Switzer B H, et al. A transformatioll for ordering multispectral data in terms of image quality with Implications for noise remaval[J]. IEEE Transactions on Geoscience and Remote Sensing, 1988, 26(1): 65-74