知识图谱在金融推荐系统中的应用

恒生电子股份有限公司 / 应雄，姜海军，楼承先
E-mail：yingxiong@hundsun.com

近年来，由于金融科技 (FinTech) 的迅猛发展，传统金融领域与金融科技的结合也越来越密切，金融科技产品如雨后春笋般涌现，人们对这些金融产品的关注和参与度也迅速高涨。然而现有的金融科技产品琳琅满目，与这些产品相关的资讯更是众多纷纭，人们要从这海量的资讯中获取自己关注和感兴趣的资讯变得极其繁琐和困难。推荐系统作为解决信息过载问题的有效手段之一，已被广泛应用于各个领域，该系统能够主动的向用户提供需求信息，让每个用户都能够得到具有针对性的推荐结果。然而现有的许多推荐算法存在没有考虑物品本身内涵知识的缺陷，导致对资讯内容分析的不完全和不精确，使得推荐效果不理想。本文的主要贡献在于提出了一个知识图谱和推荐系统的融合模型，在基于经典推荐算法基础上，充分考虑资讯本身内在的语义信息，利用资讯之间的标签关联关系，有效地提升了推荐算法的准确率、召回率和 F 值。

0 概述

自二十世纪九十年代诞生以来，个性化推荐系统一直在互联网的众多领域发挥着巨大的作用，例如亚马逊的电子商务系统、Hulu 的视频推荐系统、Pandora 的音乐推荐系统等。进入二十一世纪以来，移动互联网得到高速发展，用户的移动个人终端设备越来越普及。移动设备往往体积较小，移动端页面承载的信息量较电脑端大大降低；在这样的背景下，展现给每个用户的信息将要求更精准、更符合每个人的个体需求。而信息的碎片化、文化的多样性，也带来了人们对信息的需求越来越个性化。这些因素大大地推动了个性化推荐系统应用的普及和深入。我们相信在不远的将来，推荐系统会像搜索引擎一样，成为互联网产品必不可少的基础性应用。

在这样的背景下，恒生电子也组建团队进行了推荐系统的研发，帮助更多的金融机构更为便捷和快速的把推荐系统应用到各种金融业务场景中。恒生金融资讯推荐系统通过对资讯和用户进行标签化并结合多种经典的推荐算法如：基于用户的协同过滤算法、基于物品（资讯）的协同过滤算法、基于内容的推荐算法等来进行金融资讯、产品的个性化推荐。

但基于经典的推荐算法进行相关相似度计算时，均没有对输入参数自身间的关系及对参数间关系带来关联度或权重的变化做很好的体现。例如，传统资讯间的相似度算法如余弦相似度，通常根据语义分析关键字来进行计算；但通常会忽略关键字间的关联关系及层次结构；本文将通过引入知识图谱到传统的推荐算法中，来优化相关的相似度计算结果，有效提升推荐效果。

1 相关理论

1.1 金融资讯推荐系统

伴随着网络的发展，信息过载问题越发凸显，面对海量的资讯，如何发现资讯内容与用户之间的相关性，找到与用户兴趣爱好相似的资讯内容是个性化资讯推荐系统的关键。推荐系统通过分析用户行为，如用户浏览、用户评论和用户分享，可以发现用户的资讯偏好，给不同用户提供不同的个性化页面展示，提升用户的阅读体验，来提高终端的点击率和转化率。

在推荐系统中，推荐算法是整个推荐系统的灵魂以及核心部分，推荐系统性能的优劣在很大程度上取决于推荐引擎的好坏。目前，主流的推荐算法包括：基于协同过滤的推荐、基于内容的推荐、基于关联规则推荐和组合推荐算法等。

基于内容的推荐 (content-based recommendation) 是指利用用户曾经喜欢的项目的若干属性来分析、查找与该项目相似的项目作为推荐。这类方法一方面通过特征提取的方法来获得内容向量以描述项目模型，另一方面基于用户历史的评分向量来学习用户的兴趣以生成用户模型，这两个模型具有同样的维度定义。系统以这两个矩阵模型作为输入计算得出用户与待推荐内容之间的匹配程度。基于内容的推荐算法的效用函数，即每个用户对项目的预测值可以定义为：

$$score(u,i) = evaluate(profile(u), content(i))$$

其中函数 $profile(u)$ 以用户 u 作为其输入，返回该用户的偏好向量，函数

$content(i)$以待推荐的对象作为输入，返回该对象内容的特征向量。对于项目向量的提取函数$content(i)$，最直接的实现方式是通过文档中抽取的词或短语构造文档的词频向量，并使用该词频向量来表达项目的特征。本文方法是使用经典的 TextRank 方法来获取词频向量。

对于用户向量的生成函数$profile(u)$，存在着多种不同的计算方法，常见的方法包括向量表示法以及基于机器学习的贝叶斯分类算法。本文采用向量表示法，该方法将用户所有评分过的内容向量进行累加，分别计算每一个维度特征的加权平均值，便可得到最终的用户向量。

最终的评分预测函数$evaluate(\ )$以用户向量以及项目内容向量作为输入参数，返回用户和项目内容相似度的评估数值，该数值代表了用户对该项目潜在的兴趣度。该数值通常通过输入的两个向量的夹角余弦进行计算：

$$evaluate(p,c)=\cos(\vec{v_p},\vec{v_c})=\sum_{k=1}^{K}\frac{v_{p,k}v_{c,k}}{\sqrt{\sum_i^N v_{p,i}^{\ 2}}\sqrt{\sum_i^N v_{p,i}^{\ 2}}}$$

协同过滤推荐技术的研究相对较早也较成熟，该算法主要思想可描述为：首先根据用户对物品的行为评分找到兴趣相类似的用户，然后将该物品推荐给兴趣相似的且未对该物品发生过行为的用户。这种方法被称为基于用户的协同过滤（UserCF，user-based collaborative filtering）。另一方面，也可以通过用户行为评分来查找最近邻的项目，然后进行预测。这种方法被称作基于项目的协同过 (ItemCF，item-based collaborative filtering)。协同过滤这个技术是采用最邻近原则，拿基于用户的协同过滤方法举例，首先对用户的历史喜好信息计算他们的距离，然后选择最近邻用户并对该用户对物品的评价值进行加权平均，最后根据用户对物品的偏好程度的预测值进行推荐。因此基于用户的协同过滤方法的关键在于寻找最近邻用户，也就是与该用户兴趣最相似的用户。寻找最近邻用户采用如下方法：在评分矩阵中，记一个行向量代表一个用户，向量中的元素为用户对项目的评分。相似度的计算建立在两个用户共同的评分项上，即在用户向量的交集上进行计算。对于两个向量的相似度，可以采用余弦相似度、调整余弦相似性和相关相似性等方法进行计算。以余弦相似度为例，用户的相似性即两个用户向量的夹角余弦。基于用户的推荐算法可以描述为：

$$similarity(u_a,u_b)=\cos(profile(u_a),$$

$$profile(u_b))=\sum_{k=1}^{K}\frac{P_{a,k}P_{a,k}}{\sqrt{\sum_i^N P_{a,i}^{\ 2}}\sqrt{\sum_i^N P_{b,i}^{\ 2}}}$$

向量 P_a 和 P_b 分别表示用户 a 和用户 b 在公用评分项上的评分。$profile(u)$可以表示为：

$$profile(u_a)=(s_1,s_2,s_3\cdots s_n)$$

S_n 表示了用户 u 对第 n 个项目的历史评分，n 为系统中所有的项目数。计算得到用户 i 与用户 j 之间的相似度$similarity(u_i,u_j)$后，便可以根据当前用户 u_a 对某个项目 i 的评分以及用户之间的相似度来计算其他用户对该项目的喜好程度了：

$$score(u_a,i)=similarity(u_a,u_b)\cdot profile(u_b)_i$$

协同过滤技术存在以下优点：可以更好的共享其他用户的经验，与此同时还可以过滤掉一些复杂的、不容易表述的概念，具有较高的准确性。然而基于协同过滤的资讯推荐系统仍然存在一些缺陷，如系统没有考虑资讯本身内在的语义信息，利用资讯之间的标签关联关系、缺少历史数据而导致的新加入新闻无法被推荐的冷启动问题等。

1.2 知识图谱

知识图谱（Knowledge Graph）的概念由谷歌 2012 年正式提出，旨在实现更智能的搜索引擎，并且于 2013 年以后开始在学术界和业界普及，并在智能问答、情报分析、反欺诈等应用中发挥重要作用。知识图谱本质上是一种叫做语义网络（semantic network）的知识库，即

图 1：知识图谱总体架构

具有有向图结构的一个知识库，其中图的结点代表实体（entity）或者概念（concept），而图的边代表实体 / 概念之间的各种语义关系，比如说两个实体之间的相似关系。

知识图谱的总体架构如图 1 所示。整个架构图主要分为三个部分，第一个部分是知识获取，主要阐述如何从非结构化、半结构化、以及结构化数据中获取知识。第二部是数据融合，主要阐述如何将不同数据源获取的知识进行融合构建数据之间的关联。第三部分是知识计算及应用，这一部分关注的是基于知识图谱的计算功能以及基于知识图谱的应用。

2 融合了知识图谱的金融资讯推荐系统

2.1 融合了知识图谱的推荐过程

自语义网的概念提出，越来越多的开放链接数据和用户生成内容被发布于互联网中。互联网逐步从仅包含网页与网页之间超链接的文档万维网转变为包含大量描述各种实体和实体之间丰富关系的数据万维网。在此背景下，知识图谱 (Knowledge Graph) 于 2012 年 5 月首先由 Google 提出。紧随其后，国内搜狗提出了“知立方”、微软的 Probase 和百度的“知心”。

对于一个典型的知识图谱来说，可以通过有向图表示的三元组，以及三元组之间的相互链接构成一个网状的知识集合，这种三元组携带着实体自身的语义信息。其中实体作为节点，实体之间的关系作为边。以金融行业为例，具体一个上市公司有股东构成、所属行业以及企业类型等等一系列特征；利用这些特征，可以得到类似于图 2 所示

图 2 知识图谱三元组

的某一家上市公司知识图谱的三元组。

在上图中，上市公司实体“恒生电子”和自然人实体“马云”之间通过“实际控制人”关联，构成三元组（恒生电子，实际控制人，马云），再通过三元组之间的相互链接形成知识图谱。而这些特征的综合能够反映一家上市公司的特点。而孤立的看某一些特征，又缺乏足够的意义；知识图谱把这些特征进行关联，能够很好的反映和体现全市场的上市公司的情况及潜在的关联关系。下图为节选的部分金融企业、自然人、金融产品及金融行业的关联关系构成的网状结构。

从图 3 中我们可以看到，各个实体间存在着各种各样的关联关系；而这些实体往往又是金融资讯或者产品的标签化内容之一；通过知识图谱，能够很直观的反映这些实体间的联系。

基于上述思想，在金融资讯推荐系统中引入知识图谱来完善个人与资讯、资讯标签间的潜在逻辑关系，将其运用于用户核心关注资讯分析上，并有效提升推荐效果；有效地弥补了传统推荐系统中只通过语义分析来进行资讯关键字提取导致的关键字扁平化缺陷。

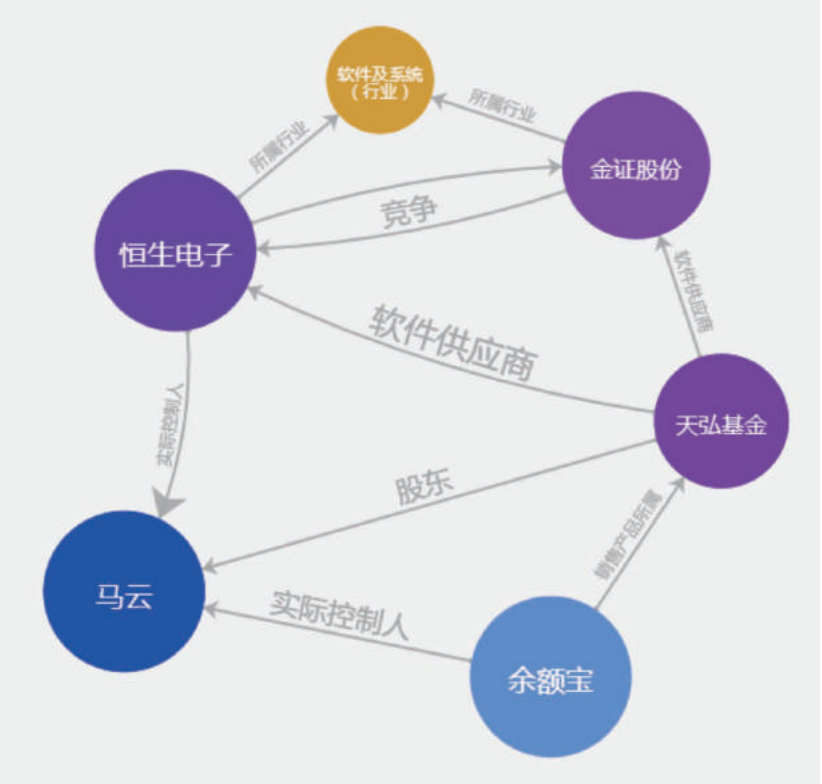

图 3 实体关联关系网状结构图

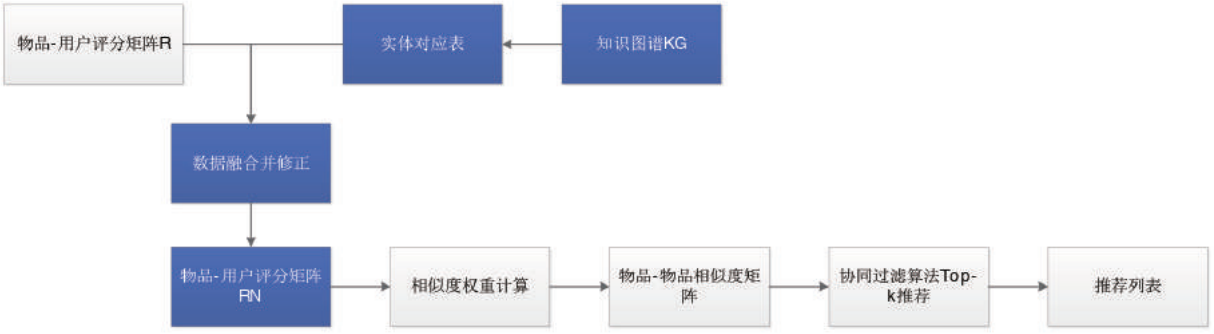

图 4 融合了知识图谱的金融资讯推荐系统流程图

图 4 描述了融合了知识图谱的金融资讯推荐系统的核心处理流程图。

2.2 算法描述

在资讯推荐系统中，资讯相似度是通过计算余弦相似度得到的，即通过计算两篇资讯的 n（n>0）维向量间的夹角的余弦来测算资讯相似度。余弦相似度（绝对值）的取值范围为 [0~1]，夹角的大小与余弦相似度成反比，两个向量间的夹角越小，余弦相似度的值就越大。以计算两篇资讯的相似度为例，若求得余弦值接近 1，则说明这两篇资讯非常相似；若余弦相似度接近 0，则说明两篇资讯差异很大。余弦相似度的计算公式为：

$$\mathrm{sim}\left(x_i', x_j'\right) = \cos(x_i', x_j') = \frac{x_i' \cdot x_j'}{\left\|x_i'\right\|_2 \cdot \left\|x_j'\right\|_2}$$

上式中，x_i'、x_j' 分别表示资讯融合知识图谱后的标签新向量。

在实际资讯推荐系统中，不同用户的评分标准不尽相同。例如，在给定了评分区间为 1 到 5 分的情况下，对于某个用户 A 来说评分在 3 分以上的就是自己感兴趣的资讯，而对于另一用户 B 来说，评分在 4 分以上的才是其感兴趣的。此处的余弦相似度未能考虑到用户评分标准差异的问题，所以在直接计算不同用户的余弦相似度时会导致最终预测结果发生很大的偏差。修正的

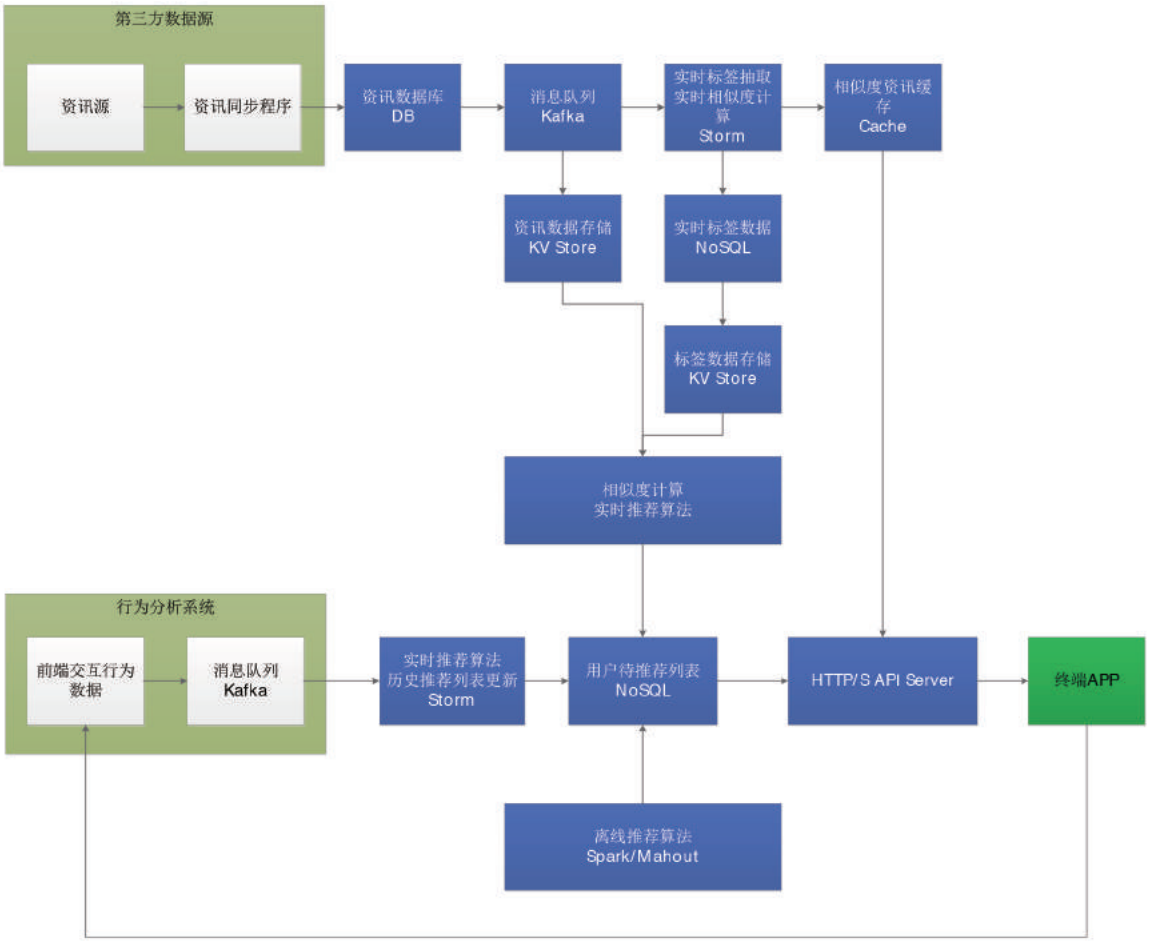

图 5 推荐系统架构

余弦相似性度量方法通过减去用户对物品的平均评分，有效地改善了这一问题。修正的余弦相似度计算公式如下：

$$\mathrm{sim}\left(x_i', x_j'\right) = \frac{\sum_{u \in U}\left(R_{u,x_i} - \overline{R_u}\right)\left(R_{u,x_j} - \overline{R_u}\right)}{\sqrt{\sum_{u \in U}\left(R_{u,x_i} - \overline{R_u}\right)}\sqrt{\sum_{u \in U}\left(R_{u,x_j} - \overline{R_u}\right)}}$$

3 效果评估及分析

3.1 测试运行环境

系统的主要架构如上图 5 所示。主要思想采用结合了实时处理和离线处理的 Lambda 架构。实时流部分主要通过 Kafka 消息队列、Storm 实时处理框架结合 Redis 内存存储来进行实时处理；离线任务主要通过 Spark 计算框架及 Mahout 机器学习算法来进行离线的大数据量运算来完成。

机器编号	R1	R2	R3	R4	R5	R6
操作系统	CentOS 6.5 x64					
CPU	Intel Xeon 4*8C					
内存	128G					
虚拟化数	/	/	/	3	3	3
部署角色	CDH 5.7.1 Kafka 2.10-0.9.0.0 JStorm 2.1.1 Mahout 0.9 ES 2.3.3			Redis 3.2.5 Mongo 3.0.6 DataSync DataAnalysis	Redis 3.2.5 DataApi DataSync DataAnalysis	MySql 5.6 Neo4j 3.2.1 DataApi DataAnalysis

注：DataSync、DataAnalysis、DataApi 为恒生电子自主研发产品 / 模块

表 1 系统软、硬件环境

目前用于测试的硬件主要使用了 6 台物理服务器，采用 Linux 系统。具体软件部署情况大致如上表 1 所示。

3.2 数据集

推荐系统效果评估的实验数据集来自恒生聚源梵思移动资讯，该 App 聚合了各大金融类网站的资讯，用户可以在该 App 上阅读、推荐、评论自己喜欢的资讯。

实验采用离线测试的方法，从后台数据库中选取 100 万条用户行

	用户喜欢	用户不喜欢
系统推荐	true positives（TP）	false negatives（FN）
系统未推荐	false positives (FP)	true negatives (TN)

表 2 混淆矩阵

为记录，10 万篇资讯数据作为实验数据集，包含 5470 个用户，每个用户至少阅读过 100 篇资讯。实验数据集中 90% 作为训练数据集，10% 作为测试数据集。

为了对本文提出的融合了知识图谱的推荐系统方案进行验证，在上一节知识图谱与推荐系统设计与实践的基础上，本文设计了几组实验分别将本文方案与传统推荐算法进行了效果对比，具体评估结果参看 3.4 小节。

3.3 评价指标

对于用户来说，推荐系统向其推荐的列表中的内容并不能保证完全符合其兴趣偏好，推荐系统的推荐效果通常使用以下 3 个指标对其进行评测：准确率（Precision）、召回率（Recall）和 F 值（F-measure）。对于一个未曾发生过行为的新用户或新物品来说，推荐结果有四种可能：系统推荐了符合兴趣偏好，系统推荐了未能满足兴趣偏好，系统未推荐但物品满足用户偏好，系统未推荐但也不满足用户偏好，这四种情况可以用表 2 混淆矩阵表示。

混淆矩阵中的 TP、FN、FP、TN 分别表示这几种可能情况的数目，由表 2 可以看出，用户喜欢的物品数目 T=TP+FP，推荐列表长度 L=TP+FN。对于某一用户 u 来说，推荐准确率为所有“正确被检索的 item(TP)”占所有“实际被检索到的 (TP+FP)”的比例，即：

$$A=\frac{TP}{TP+FP}=1-\frac{FP}{T}$$

召回率 (Recall) 定义为所有“正确被检索的 item(TP)”占所有“应该检索到的 item(TP+FN)”的比例，即：

$$R=\frac{TP}{TP+FN}=1-\frac{FN}{L}$$

对于推荐列表长度不固定的推荐系统可利用 F—measure 反映系统表现，F 值是准确率和召回率的加权平均，均匀地反映了推荐效果，计算公式如下：

$$F-\text{measure}=\frac{2\times A\times R}{A+R}$$

召回率反映了被推荐系统所推荐的物品占用户真正喜欢的物品的比重。准确率反映了推荐系统的推荐水平，能将用户喜欢的物品推荐给用户，而用户不喜欢的物品则不推荐。F 值是准确率和召回率的加权平均，均匀地反映了推荐效果。

3.4 效果分析

这里我们采用 3.2 小结介绍的数据集进行测试，将本文方法与传统的推荐算法进行比较。选取的 k 近邻数分别为 60、80、100、120。对于每一组测试，均循环 10 次并取其平均值。

从图 6 到图 8 结果可以看出，在相同条件下，与传统的推荐算法相比，本文融合了知识图谱的推荐算法明显具有更好的推荐效果。

4 总结

知识图谱作为一种新颖的知识组织与检索技术，涉及图书情报、计算机科学、知识工程和语言学等多个学科的理论与方法，正逐渐受到业界关注。它在知识组织和展现上体现出来的优势是非常显著的，也是适应当前环境的一大趋势。

本文提出了一种在经典推荐算法基础上融合了知识图谱的推荐算法，既利用了资讯本身内在的语义信息，又使用了资讯外在的关键字深度矩阵，能够更加全面地反映资讯的属性。算法通过知识图谱将关联向量深度融合至相似度矩阵中，计算资讯间的语义相似性，并将其运用到推荐系统中，在语义的层面上增强了金融资讯推荐的效果，从而解决了经典推荐算法未考虑语义的深度关联问题。测试表明，将知识图谱的实体关系应用于传统经典推荐算法中能够有效提高推荐算法的效果。

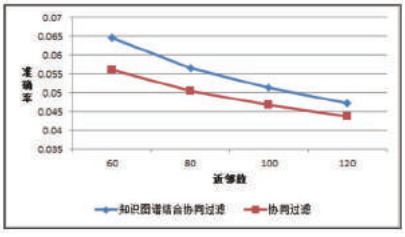

图 6 准确率

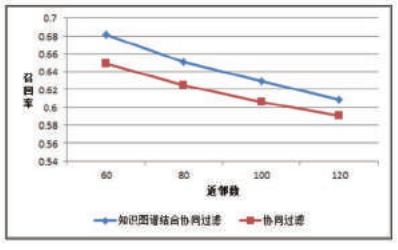

图 7 召回率

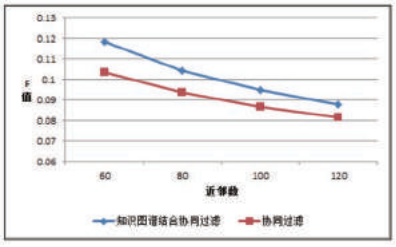

图 8 F 值

参考文献：

[1]（奥地利）詹尼士（Dietmar Jannach），（奥地利）Markus Zanker，（奥地利）Alexander Felfering，（奥地利）Gerhard Friedrich .《推荐系统》[J]. 人民邮电出版社；第 1 版 (2013 年 7 月 1 日)

[2] 邵领 . 基于知识图谱的搜索引擎技术研究与应用 [D]. 电子科技大学，2016.

[3] 牛温佳，刘吉强，石川 等《用户网络行为画像》[J] 电子工业出版社 2016-3-1

[4] 漆桂林，高桓，吴天星 . 知识图谱研究进展 [J]. 情报工程，2017，(01):4-25.

[5] 徐增林，盛泳潘，贺丽荣，王雅芳 . 知识图谱技术综述 [J]. 电子科技大学学报，2016，(04):589-606.

[6] 杨博，赵鹏飞 . 推荐算法综述 [J]. 山西大学学报 (自然科学版)，2011，(03):337-350.