

文章编号:1001-9081(2014)11-3140-04

doi:10.11772/j.issn.1001-9081.2014.11.3140

基于共同评分项目数和用户兴趣的协同过滤推荐方法

王雪霞*, 李青, 李季红

(上海大学 计算机工程与科学学院, 上海 200444)

(*通信作者电子邮箱 wxx116815@126.com)

摘要:在推荐系统中,为了在一定程度上减少用户评分数据稀疏对推荐效果的负面影响,提出了一种基于用户共同评分项目数和用户兴趣的协同过滤推荐算法。此算法将用户共同评分项目数和用户兴趣相似度相结合,使用户之间的相似度计算更加准确,为目标用户提供更好的推荐结果。仿真实验结果表明:所提算法比基于 Pearson 相似度计算方法的算法推荐效果更优,具有更小的平均绝对误差(MAE),表明了其有效性和可行性。

关键词:稀疏数据;共同评分项目数;用户兴趣;协同过滤;Pearson 相似度

中图分类号: TP301.6 **文献标志码:**A

Collaborative filtering recommendation based on number of common items and common rating interest of users

WANG Xuexia*, LI Qing, LI Jihong

(School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China)

Abstract: In order to reduce the negative impacts of sparse data, a new collaborative filtering recommendation algorithm was put forward based on the number of common rating items among users and the similarity of user interests. The similarity calculations were made to be more credible by combining the number of common rating items among users with the similarity of user interests, so as to provide better recommendation results for the target user. Compared with the method based on Pearson similarity, the new algorithm provides better recommendation results with smaller Mean Absolute Error (MAE). In conclusion, the new algorithm is effective and feasible.

Key words: sparse data; number of common rating items; user interest; collaborative filtering; Pearson similarity

0 引言

随着互联网的发展,特别是 Web 2.0 的出现,人们正处于一个信息爆炸的时代。搜索引擎^[1]虽然在一定程度上解决了信息筛选问题,但还远远不够。当用户无法准确描述自己的需求时,搜索引擎的筛选效果将大打折扣。在此背景下,推荐系统的出现,一方面帮助用户发现对自己有价值的信息,另一方面让信息能够展现在对它感兴趣的人群中,从而实现信息提供商与用户的双赢。

人们根据推荐算法的不同把推荐系统分成不同的类别,主要有基于人口统计学的推荐、基于内容的推荐、协同过滤推荐和混合推荐算法。其中,协同过滤推荐算法作为目前推荐系统中应用最广泛、最成功的推荐技术之一^[2-3],得到了研究者的广泛关注。相似度计算是协同过滤推荐技术的核心,在实际应用当中,用户和项目数量十分庞大,但用户仅仅会对一小部分项目进行评分,这导致了用户项目矩阵非常稀疏,增加了相似度计算的难度和不准确性。

关于相似度计算问题,很多学者进行了相关研究。Herlocker 等^[4]最早提出了用户相似度的调整参数和邻居用户的选取阈值,并通过实验证明引入这些参数后提高了推荐准确性。Bobadilla 等^[5-6]提出采用多因素的相似度来衡量用

户之间的相似性,并使用智能算法确定各因素的权重系数,如遗传算法,该方法的缺点是权重系数计算比较费时间。Massa 等^[7-8]提出基于信任的相似性度量,由于信任的可传递性,因此可以解决数据的稀疏性问题,但是这种方法需要信任网络或者社交网络结构^[9],所以可能不太适合当前只涉及项目评分的系统。

为了减少数据稀疏性影响,提高相似度计算的准确性,本文将用户共同评分项目数和用户兴趣相似性度量引入到传统的协同过滤推荐算法中,使目标用户取得更好的最近邻居集,从而获得更好的推荐效果。

1 传统推荐算法

传统的协同过滤推荐算法通过用户的最近邻居集合产生最终的推荐列表。主要包括两部分^[10]:1)针对目标用户,找到最相似的邻居集合;2)运用最近邻居集合中的用户评分预测目标用户对未知项目的评分。

协同过滤推荐算法中最核心的部分是找到准确的邻居集合,即相似度计算是协同过滤算法的核心。常见的相似度计算方法有 Jaccard 系数、Pearson 相关系数和余弦相似度系数^[11]。

1) Jaccard 系数:该相似度是基于两个用户共同评分项目

收稿日期:2014-06-05;修回日期:2014-07-31。

作者简介:王雪霞(1988-),女,山东烟台人,硕士研究生,主要研究方向:推荐系统、复杂网络; 李青(1962-),男,湖北嘉鱼人,教授,博士生导师,CCF 会员,主要研究方向:并行计算、复杂网络、复杂系统建模、计算机模拟; 李季红(1987-),男,浙江杭州人,硕士,主要研究方向:并行计算、推荐系统。

的个数衡量他们之间的相似性。

$$Jaccard(x, y) = \frac{\#(A_x \cap B_y)}{\#(A_x \cup B_y)} \quad (1)$$

其中: A_x, B_y 分别表示 x 和 y 的评分项目集。

2) Pearson 相关系数: 该相似度主要是衡量两个用户公共评分的相关程度。

$$sim(x, y) = \frac{\sum_{i \in C_{x,y}} (ra_{x,i} - \bar{ra}_x)(ra_{y,i} - \bar{ra}_y)}{\sqrt{\sum_{i \in C_{x,y}} (ra_{x,i} - \bar{ra}_x)^2} \sqrt{\sum_{i \in C_{x,y}} (ra_{y,i} - \bar{ra}_y)^2}} \quad (2)$$

其中: $C_{x,y}$ 为用户的公共评分集, \bar{ra}_x 和 \bar{ra}_y 分别表示用户 x 、 y 的平均评分, $ra_{x,i}$ 和 $ra_{y,i}$ 表示用户 x 、 y 对项目 i 的评分。

3) 余弦相似度: 该相似度是将用户评分看作一个向量, 计算两个用户之间的相似度就是计算评分向量夹角的余弦值。

$$\cos(x, y) = \frac{\mathbf{r}_x \times \mathbf{r}_y}{|\mathbf{r}_x| \times |\mathbf{r}_y|} \quad (3)$$

其中: \mathbf{r}_x 和 \mathbf{r}_y 分别为 x 、 y 的评分向量。

根据前面提到的三种相似度计算方法, 可以得到用户之间的相似度; 然后选择与目标用户 u 最相似的用户集 N , 最终得到目标用户 u 对项目 i 的预测评分 $P_{u,i}$ 。预测评分公式如式(4)所示:

$$P_{u,i} = \bar{R}_u + \frac{\sum_{v \in N} sim(u, v) \times (R_{v,i} - \bar{R}_v)}{\sum_{v \in N} (1 \cdot sim(u, v))} \quad (4)$$

其中: \bar{R}_u 表示目标用户 u 的平均评分, $sim(u, v)$ 表示用户 u 、 v 之间的相似度, $R_{v,i}$ 表示用户 v 对项目 i 的评分, \bar{R}_v 表示用户 v 的平均评分。

2 新的协同过滤推荐算法

2.1 基于共同评分项目数的相似性度量

传统相似度计算不能准确地衡量用户之间的相似性。Jaccard 相似度计算方法考虑了用户的共同评分项目数, 共同评分项目越多, 两个用户相似的可能性越大。但由于大规模用户项目数据的稀疏性, 其计算出来的值相对太小, 区分度不大。

而 Pearson 相似度计算方法, 当共同评分项目数为 2 时, Pearson 相关系数只有 1 或者 -1 两个值, 所以传统的 Pearson 相似度计算方法会让公共评分项目数比较少的用户占优势, 如表 1 所示。

表 1 共同评分项目数不同的用户-项目

User	Item1	Item2	Item3
User1	3.0	2.0	1.0
User2	1.0	2.0	3.0
User3	1.0	—	—
User4	2.0	—	1.0
User5	3.0	2.0	1.0

表 1 中 User1 与 User5 共有 3 个共同评分项, 并且评分趋势相似, 同时 User1 与 User4 只有 2 个共同评分项, 虽然趋势也相似, 但是 User4 对 Item2 的喜恶程度是不确定的, 因此直观地看 User1 与 User5 更相似, 但是 Pearson 相似度计算结果

却是 User1 与 User4 更相似。

同样的场景, 当某两个用户共同观看了 200 部电影, 他们不一定给出完全相近的评分, 但只要他们趋势相似就一定比另一位只观看了 2 部相同电影的相似度高。但是事实并非如此, Pearson 相似度计算方法计算出的结果明显高于观看了相同的 200 部电影的相似度。

如上所述, 在现实的系统中, 由于用户评分数据的极端稀疏性, 导致很多用户间的共同评分项目个数少之又少, 此时使用 Pearson 相似度计算方法显然是不合理的。

但是研究表明 Pearson 相似度计算方法能够很好地描述两组数据的变化趋势, 不受绝对数值的影响。由表 2, 可以直观地认为 User2 和 User3 更为相似, 它们的重叠评分数目一致, 趋势也相同, 记录 User1 虽然也满足上述的条件, 但是它的整体数值很低。利用 Pearson 相似度计算方法计算它们之间的相似度: User1 与 User2 为 0.9899, User2 与 User3 为 0.9899, User1 与 User3 为 0.9999。可看出 Pearson 相似度计算方法对绝对数值并不敏感, 它只是描述了两组数据变化的趋势。

表 2 评分趋势相似的用户-项目

User	Item1	Item2	Item3	Item4
User1	1.0	2.0	3.0	4.0
User2	40.0	50.0	70.0	80.0
User3	50.0	60.0	70.0	80.0

通过上述对 Jaccard 与 Pearson 相似度计算方法的优缺点分析, 可知传统的 Jaccard 和 Pearson 相似度计算方法很难发挥良好作用, 目标用户不能得到推荐或者满意的推荐。但是把二者的优点相互结合, 可以弥补二者各自的不足, 因此提出了与共同评分个数相关的新相似度计算方法。传统的 Pearson 相似度计算方法是从公共评分数值的角度定量地来度量两个用户之间的相似性。而公共评分个数是从非数值角度定性地来衡量公共评分非数值方面的相似性, 它的引入是基于 Jaccard 相似度计算方法中公共评分项目数越多, 则两个用户间相似的可能性越大的原则。

$$sim_{co-pears}(x, y) = sim_{pearson}(x, y) \cdot f(x, y) \quad (5)$$

$$f(x, y) = 1 - \frac{1}{\#(A_x \cap B_y)^\alpha} \quad (6)$$

其中: $sim_{pearson}(x, y)$ 表示 Pearson 相关系数; A_x 和 B_y 分别表示 x 和 y 的评分项目集; $\#(A_x \cap B_y)$ 表示用户 x 和 y 的共同评分个数; 为了防止共同评分数较少时 $f(x, y)$ 增长过快, α 的取值设为 $0 < \alpha < 1$ 。

当用户 x 和 y 的公共评分项目个数越多, 二者相似的可能性越大, 所以 $f'(x, y) > 0$ 。当 $\#(A_x \cap B_y)$ 从 5 增加到 10 对相似性的影响肯定比从 60 增加到 65 的影响要大, 所以 $f''(x, y) < 0$ 。

2.2 基于用户兴趣的相似性度量

使用 Pearson 相似度计算方法计算相似度要求用户之间至少有两个共同评分项目才能参与相似度计算。显然, 在数据极端稀疏的情况下, 这个方法存在不足之处。同时, 用户之间的相似度不仅与项目评分数值和共同评分项目的个数相关, 还与用户的兴趣相关, 并且根据用户的兴趣进行推荐是推荐系统的关键所在。因此引入了基于用户兴趣的相似度计算方法。当两个用户具有相同的兴趣爱好时, 就可以简单地认

为他们之间具有较高的相似性。

定义用户 u 的兴趣向量为:

$$\text{Interest}_u = (i_{u,1}, i_{u,2}, \dots, i_{u,n}) \quad (7)$$

$$i_{u,j} = s_{u,j}/s_u \quad (8)$$

其中: $i_{u,j}$ 表示用户对某类项目的喜爱程度, $s_{u,j}$ 表示用户 u 评价 j 类项目的总数, s_u 表示用户 u 评价各类项目的总数。

那么,两个用户之间的兴趣相似性度量可以使用式(9)余弦相似度计算:

$$\text{sim}_{\text{interest}}(u, v) = \cos(u, v) = \frac{\text{Interest}_u \times \text{Interest}_v}{|\text{Interest}_u| \times |\text{Interest}_v|} \quad (9)$$

2.3 共同评分项目数与用户兴趣结合的相似性度量

使用加权的方法,集成基于共同评分项目数的相似性度量和基于用户兴趣的相似性度量,克服传统相似性度量的不足,从而使用户之间的相似度计算更加准确,为目标用户提供更好的推荐列表,如式(10)所示:

$$\text{sim}(x, y) = (1 - \lambda) \cdot \text{sim}_{\text{co-peerson}}(x, y) + \lambda \cdot \text{sim}_{\text{interest}}(x, y) \quad (10)$$

其中参数 λ 的取值范围设定为 $0 < \lambda < 0.5$, 这是由于基于共同评分项目数的相似度计算是从数据集的全局出发,统计全面;基于用户兴趣的相似度计算方法是为了弥补 Pearson 相似度计算方法至少有两个共同评分项目的不足,同时为了降低算法的时间复杂度仅选取有限数量的最高评分进行兴趣统计。

2.4 算法描述

基于前文提出的相似度计算方法,在此给出新的协同过滤推荐算法描述。

算法 1 基于共同评分项目数和用户兴趣的协同过滤推荐算法。

输入 用户信息文件、项目信息文件、参数 α 、参数 λ 、目标用户。

输出 目标用户预测评分。

过程如下:

- 1) 计算用户的平均评分;
- 2) 计算用户的共同评分项目数;
- 3) 计算基于共同评分项目数的相似性 $\text{sim}_{\text{co-peerson}}(x, y)$;
- 4) 选取固定数量的用户评分较高的多个项目,求得用户的兴趣向量;
- 5) 计算基于用户兴趣的相似性值 $\text{sim}_{\text{interest}}(x, y)$;
- 6) 两种相似度加权相加得到新的相似度 $\text{sim}(x, y)$;
- 7) 选择最近邻居集。

3 实验与分析

3.1 数据集

实验使用美国明尼苏达大学 GroupLens 研究项目组提供的 MovieLens 数据集。选用的数据集包含 943 个用户对 1682 部电影 100 000 多评分信息及用户和项目的多种基本信息;该数据集的稀疏性为 93.695%;选择其中的 80% 作为训练集,20% 作为测试集。

3.2 评价标准

统计精度度量方法中的平均绝对误差(Mean Absolute Error, MAE)被广泛用于评价协同过滤推荐系统的推荐质

量^[12]。因此,推荐质量评价采用了常见的平均绝对误差 MAE。在测试集上先用推荐系统预测出用户的评分,然后根据测试集中用户的实际评分,计算出二者的偏差,即为 MAE 的值。

假设预测用户评分值为 $\{p_1, p_2, \dots, p_n\}$, 对应的实际评分值为 $\{q_1, q_2, \dots, q_n\}$, 则 MAE 的计算公式为:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |p_i - q_i| \quad (11)$$

3.3 结果与分析

在完成用户的共同评分项目数统计的前提下,通过实验并最终确定 α 的取值。 $\alpha \geq 1$ 时,当用户共同评分项目数较少的时候,式(6)值增长太快;而当用户共同评分项目数较多的时候,式(6)值几乎保持不变,不能很好地调节 Pearson 相关系数的值。当 $0 < \alpha < 1$ 时,部分取值对应的式(6)的图像如图 1 所示。

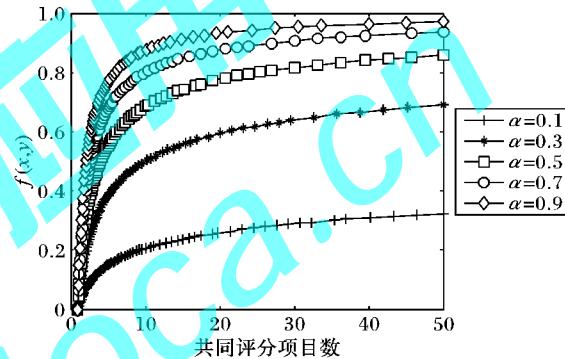


图 1 不同 α 取值对应的调节函数取值

由图 1 可知, $\alpha = 0.1, 0.3$ 或 0.5 的时候, 曲线取得了良好的变化率, 符合建模调节函数的初衷。因此, 实验过程中设置 $\alpha = 0.1, 0.3$ 或 0.5 。

由图 2 可知, 当 $\alpha = 0.3$ 时, 基于共同评分项目数的相似度计算方法的 MAE 曲线效果最好, 因此在后续实验中设定 $\alpha = 0.3$ 。

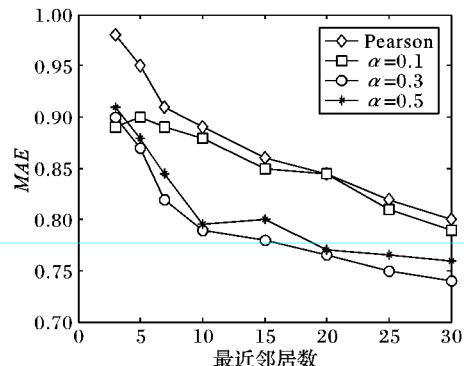
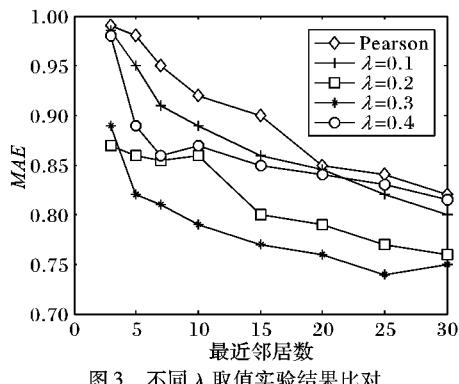


图 2 不同 α 取值对应的 MAE 曲线

实验中对传统的协同过滤算法以及基于共同评分项目数和用户兴趣的协同过滤算法 λ 取不同值进行了比对, 因为 $0 < \lambda < 0.5$, 所以实验中 λ 分别取 $0.1, 0.2, 0.3, 0.4$ 。由图 3 可知, 在不同邻居数、不同的 λ 取值条件下, 本文算法与传统的基于 Pearson 相似度计算方法的算法相比, 均具有较小的 MAE 值, 其中 $\lambda = 0.3$ 的时候值最优。由此推理, 基于用户共同评分项目数和用户兴趣的相似性度量方法能够提高预测结果。

图 3 不同 λ 取值实验结果对比

4 结语

本文针对传统相似性度量方法的不足,提出基于评分相似性、共同评分项目数和用户兴趣相似性的协同过滤推荐算法,并通过一系列实验分析比较算法的准确性。实验结果表明基于共同评分项目数和用户兴趣相似度的推荐算法具有良好的准确性。

目前提出的推荐算法与时间戳等因素无关,近年来很多研究者开始进行动态推荐技术^[13]的相关研究。如果在新的推荐算法中加入时间戳,动态地计算用户的兴趣,也许可以取得更好的推荐效果。这也将是下一步的研究工作。

参考文献:

- [1] ZHANG X. Search engine technology and research [J]. Modern Information, 2004, 24(4): 142–146. (张兴华. 搜索引擎技术及研究[J]. 现代情报, 2004, 24(4): 142–146.)
- [2] ADOMAVICIUS G, TUZHILIN A. Toward the next generation of recommender system: a survey of the state-of-the-art and possible extensions [J]. IEEE Transactions on Knowledge and Data Engineering, 2005, 17(6): 734–749.
- [3] LI Y, WU C, LAI C. A social recommender mechanism for e-commerce : Combining similarity , trust and relationship [J] . Decision

(上接第 3111 页)

务为例,进行了简单的性能测试。测试结果表明,使用了本文提出的缓存策略后,在降低费用的同时能够显著提高数据读取的速度。利用本文方法,可在学校、政府部门或企业建立低廉的本地缓存系统。

参考文献:

- [1] WU J, FU J, PING L, et al. Study on the P2P cloud storage system [J]. Acta Electronica Sinica, 2011, 39(5): 1100–1107. (吴吉义, 傅建庆, 平玲娣, 等. 一种对等结构的云存储系统研究[J]. 电子学报, 2011, 39(5): 1100–1107.)
- [2] WU J, ZHANG J, FU J, et al. Study on data redundancy scheme in Kademlia cloud storage system [J]. Telecommunications Science, 2011, 27(2): 68–73. (吴吉义, 章剑林, 傅建庆, 等. 基于 Kademlia 的云存储系统数据冗余方案研究[J]. 电信科学, 2011, 27(2): 68–73.)
- [3] YANG Z, ZHAO B, XING Y, et al. AmazingStore: available, low-cost online storage service using cloudlets [C]// IPTPS 2010: Proceedings of the 9th International Workshop on Peer-to-Peer Systems. Berkeley: USENIX, 2010.
- [4] GHARAIBEH A, AL - KISWANY S, RIPEANU M. ThriftStore : finessing reliability trade-offs in replicated storage systems [J]. IEEE Transactions on Parallel and Distributed Systems, 2011, 22 (6): 910–923.
- [5] LI J, YUAN P. Study on cloud storage scheme based on distributed open source management service (PPStore) [J]. Computer Applications and Software, 2013, 28(10): 208–210. (李建国, 袁平鹏. 一种基于分布式开放资源管理服务的“云存储”(PPStore)方案研究[J]. 计算机应用与软件, 2011, 28(10): 208–210.)

- [6] HERLOCKER L J, KONSTAN A J, RIEDL T J. An empirical analysis of design choices in neighborhood-based collaborative filtering algorithms [J]. Information Retrieval, 2002, 5(4): 287–310.
 - [5] BOBADILLA J, ORTEGA F, HERNANDO A, et al. A collaborative filtering approach to mitigate the new user cold start problem [J]. Knowledge-Based Systems, 2012, 26: 225–238.
 - [6] BOBADILLA J, ORTEGA F, HERNANDO A, et al. Improving collaborative filtering recommender system results and performance using genetic algorithms [J]. Knowledge-Based Systems, 2011, 24: 1310–1316.
 - [7] MASSA P, AVESANI P. Trust-aware collaborative filtering for recommender systems[C]// Proceedings of the 2004 OTM Confederated International Conferences on the Move to Meaningful Internet Systems: CoopIS, DOA, and ODBASE, LNCS 3290. Berlin: Springer-Verlag, 2004: 492–508.
 - [8] MASSA P, AVESANI P. Trust metrics on controversial users: balancing between tyranny of the majority and echo chambers [J]. International Journal on Semantic Web and Information Systems, 2007, 3(1): 39–64.
 - [9] WANG X, LI X, CHEN G. Complex network theory and its application [M]. Beijing: Tsinghua University Press, 2006: 18–46. (汪小帆, 李翔, 陈关荣. 复杂网络理论及其应用[M]. 北京: 清华大学出版社, 2006: 18–46.)
 - [10] ZHAO X, NIU Z, CHEN W. Interest before liking: two-step recommendation approaches [J]. Knowledge-Based Systems, 2013, 48: 46–56.
 - [11] GAN M, JIANG R. Improving accuracy and diversity of personalized recommendation through power law adjustments of user similarities [J]. Decision Support Systems, 2013, 55(3): 811–821.
 - [12] LÜL, MEDO M, YEUNG H C, et al. Recommender systems [J]. Physics Reports, 2012, 519(1): 1–49.
 - [13] XIANG L. Researches on the key technologies of dynamic recommendation system [D]. Beijing: Chinese Academy of Sciences, 2009. (项亮. 动态推荐系统关键技术研究[D]. 北京: 中国科学院, 2009.)
- tions and Software, 2011, 28(10): 208–210. (李建国, 袁平鹏. 一种基于分布式开放资源管理服务的“云存储”(PPStore)方案研究[J]. 计算机应用与软件, 2011, 28(10): 208–210.)
 - [6] LI D, LIU P, DING K, et al. Distributed cache strategy in cloud storage based on solid state disk [J]. Computer Engineering, 2013, 39(4): 32–35. (李东阳, 刘鹏, 丁科, 等. 基于固态硬盘的云存储分布式缓存策略[J]. 计算机工程, 2013, 39(4): 32–35.)
 - [7] Amazon simple storage service API reference [EB/OL]. [2013-12-10]. <http://docs.aws.amazon.com/AmazonS3/latest/API/Welcome.html>.
 - [8] Google drive API reference [EB/OL]. [2013-12-10]. <https://developers.google.com/drive/v2/reference>.
 - [9] Kingsoft cloud storage [EB/OL]. [2014-05-10]. <http://www.ksyun.com/product/ks3>.
 - [10] Baidu cloud storage [EB/OL]. [2014-05-10]. <http://developer.baidu.com/cloud/stor>.
 - [11] OAuth [EB/OL]. [2013-12-10]. <http://en.wikipedia.org/wiki/OAuth>.
 - [12] TANG B, FEDAK G. Analysis of data reliability tradeoffs in hybrid distributed storage systems [C]// IPDPS 2012: Proceedings of the 17th IEEE International Workshop on Dependable Parallel, Distributed and Network-Centric Systems. Piscataway: IEEE Press, 2012: 1546–1555.
 - [13] Amazon S3 pricing [EB/OL]. [2014-05-10]. <http://aws.amazon.com/cn/s3/pricing/>.