Computer Science and Application 计算机科学与应用, 2018, 8(1), 1-8
Published Online January 2018 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2018.81001

Hans汉斯

Research on Recommendation Algorithm Based on User's Attributes and Score Similarity Factors

Peisheng Shi, Jun He, Li Shu, Hao Yin, Junkai Feng

School of Computer, Sichuan University, Chengdu Sichuan
Email: gtshipeisheng@163.com

Received: Jan. 1st, 2018; accepted: Jan. 12th, 2018; published: Jan. 19th, 2018

Abstract

In order to make the user receive more accurate and more personalized recommendation information, this paper improves the influence of the current recommendation system due to the sparse data and the cold start problem. This paper takes the basic attributes of the user, the score timestamp and the user's rating, The similarity factor of the project is combined with the cooperative filtering algorithm, and a cold start recommendation algorithm based on the combination of basic attributes and similarity factors is proposed. This method exhibits better recommendation accuracy and good adaptability to data sparseness by comparing experiments with traditional methods on the Movie Lens dataset.

Keywords

Recommendation Algorithm, Collaborative Filtering, Cold Start, Similarity Factor, User-Similarity

基于用户属性与评分相似因子的推荐算法研究

石佩生，何　军，舒　莉，尹　皓，冯俊凯

四川大学计算机学院，四川　成都
Email: gtshipeisheng@163.com

收稿日期：2018年1月1日；录用日期：2018年1月12日；发布日期：2018年1月19日

摘　要

为了使用户接收更准确和更加个性化的推荐信息，改善当前推荐系统因为数据稀疏、冷启动问题带来的

文章引用: 石佩生, 何军, 舒莉, 尹皓, 冯俊凯. 基于用户属性与评分相似因子的推荐算法研究[J]. 计算机科学与应用, 2018, 8(1): 1-8. DOI: 10.12677/csa.2018.81001

诸多影响。本文采取了将用户基本属性、评分时间戳与用户的评分、偏好、评价项目的相似因子相结合的协同过滤算法，提出了基于用户基本属性与评分相似因子相结合的冷启动推荐算法。通过与传统方法在Movie Lens数据集上的对比实验，该方法展现出更好的推荐精度和对数据稀疏情况的良好适应性。

关键词

推荐算法，协同过滤，冷启动，相似因子，用户相似度

1. 引言

随着 WEB2.0 的到来，互联网信息呈现井喷式增长并成为一个海量的信息空间，这些信息远超用户接受范围，使用户迷失在这些杂乱的信息中。用户需要的信息无法被快速有效地找到，这就是所谓的网络信息超载(Information Over-load)问题，为解决这些信息超载问题，帮助用户快速准确地找到自己想要的东西，推荐系统(Recommendation System, RS)应运而生。推荐系统通过分析用户的过往信息记录，然后主动地把用户在未来可能需要的信息(用户喜欢的电影、商品、新闻等)推荐出来。推荐系统的核心在于研究推荐算法[1]上，根据推荐算法的不同，目前个性化推荐系统分为协同过滤(Collaborative Filtering, CF)推荐系统、基于内容(Content-based)的推荐系统、基于规则(Rule-based)的推荐系统[2] [3]。其中应用最广泛、最成功的当属协同过滤推荐系统，而协同过滤算法又主要分为：基于内存的协同过滤算法(memory-based CF)、基于模型的协同过滤算法(model-based CF) [4] [5]。而相似度计算是推荐系统的根本所在，比较常见的相似度计算方法有：余弦相似性(cosine)、皮尔逊相关系数(Pearson correlation)、修正的余弦相似性(adjusted cosine)、Jaccard 相似度(Jaccard) [6] [7] [8]，在相似度计算后选取相似邻居进行评分预测，最后经 top-n 推荐[9]，得出最终的推荐列表。

虽然推荐系统得到了广泛的应用，但也面临着很多问题。随着互联网用户与项目的增多，当用户或者项目的评分矩阵极度稀疏甚至未评分，或者当用户在一个新领域中没有任何过往记录时，如何将当前领域用户需要的信息推荐给他，如果不能够很快速地给一个新用户推荐其感兴趣的信息，那么用户可能会认为该领域的信息对他没有价值，这样可能导致用户不再使用该推荐系统，就出现了推荐系统中的冷启动问题[10] [11] [12] [13] [14]。所以针对这一问题，本文提出了基于用户基本属性与评分相似因子相结合的冷启动推荐算法。

2. 本文算法

本文利用对用户基本属性的相似度与评分相似因子相结合的方法，以此来挑选更合适的相似邻居。

$$Sim(u,v)=\lambda Sim_{Attr}(u,v)+(1-\lambda)Sim_{Score}(u,v) \tag{1}$$

$Sim_{Attr}(u,v)$ 为用户基本属性的相似度，$Sim_{Score}(u,v)$ 为用户评分的相似度；λ 为用户的基本属性在相似度计算中所占的权重比例，$(1-\lambda)$ 为用户对项目的评价占整个相似度计算的权重比例。

对于新用户的冷启动问题，在新用户完全未评价任何项目前，需要把用户基本属性的相似度权重调到 1。而考虑到随着新用户对项目评分的增多，应该加大对于评分相似性的重视程度，削减属性对于最终相似度的不利影响，以便用户相似度计算的重点由用户属性逐渐转移到用户评分上面，并且要保证这

是一个平滑过渡的过程，以便提高在冷启动条件下向非冷启动转换时可以提高我们的推荐准确度，所以将函数引入权值 λ：

$$\lambda = \frac{2}{1+\exp(I_i)} \tag{2}$$

当评价过的项目 I_i 由 0 逐渐增加的时候，用户基本属性的相似度权重值由 1 逐渐降到接近于 0；而评分相似度的权重值逐渐升到 1。

2.1. 用户基本属性相似度

用户在注册的时候填写的注册信息往往是不完整，比较稀疏的，根据国外的数据统计，对于性别、年龄、职业、爱好的填写率最高而且也是反应个人偏好比较准确的属性，而时间戳 Time 作为用户对项目评分与项目产生时间的差值，可以用来表示用户是否为活跃用户，可以提高对较新的项目推荐机率。可得属性集合 Puser = {Sex, Age, Profession, Interest, Time, ...}。

2.1.1. 用户文本型属性相似值

对于 Sex、Profession、Interest 以及其余文本型的属性来说，若属性相同则相似属性的值为 1，反之，不同相似属性的值为 0。设 $S(k)$为各个属性的相似值，$P_{u_i}^{(k)}$ 表示用户 i 的第 k 个属性，$P_{u_j}^{(k)}$ 表示用户 j 的第 k 个属性，则有：

$$S(k) = \begin{cases} 1, & P_{u_i}^{(k)} = P_{u_j}^{(k)} \\ 0, & P_{u_i}^{(k)} \neq P_{u_j}^{(k)} \end{cases} \tag{3}$$

2.1.2. 用户数值型属性相似值

对于 Age、Time 等这类数值型的属性就采取函数分段处理得方法，从而得到更合适的相似值。

① 关于年龄 Age 的分段函数，设 Age_u 与 Age_v 为用户 u、v 的年龄，则有相似值 S_{Age} 为：

$$S_{Age} = \begin{cases} 1, & Age_u = Age_v \\ \dfrac{1}{|Age_u - Age_v|}, & Age_u \neq Age_v \end{cases} \tag{4}$$

S_{Age} 的取值为 $\left\{1, \frac{1}{2}, \frac{1}{3}, \cdots, \frac{1}{n}\right\}$。

② 关于评分时间分段取值，$T_u = t_u - t_i$；t_u、t_i 分别为用户 u 发布评论时间、项目 i 生成时间，T_u 为两者差值，差值越小证明用户 u 越活跃。

$$S_{Time} = \begin{cases} 1, & 0 \le T_u \le 1\,\text{week}; \\ \dfrac{1}{2}, & 1\,\text{week} \le T_u \le 1\,\text{month}; \\ \dfrac{1}{3}, & 1\,\text{month} \le T_u \le 1\,\text{year}; \\ 0, & T_u > 1\,\text{year or } T_u = 0; \end{cases} \tag{5}$$

根据各个属性对于真实相似度的影响不同，我们设置不同的权值系数 ω_i，以缓解不同属性对结果造成的相似度误差。这样，当 Sim_{Attr} 的值越大，说明用户间的相似性越高，反之，则越小。

$$Sim_{Attr} = \sum_{i \in Attr} \omega_i \times S_{Attr}(u, v, i) \tag{6}$$

$$\sum_{i=1}^{n} \omega_i = 1 \tag{7}$$

举例说明：用户 u_1 = {男，24，计算机，科技，2017/2/13}，用户 v_1 = {女，27，土木，音乐，2017/3/1}。

$Sim_{Attr} = 0.2\times0 + 0.4\times1/3 + 0.2\times0 + 0.1\times0 + 0.1\times1/2 = 0.183.$

用户 u_2 = {男，24，计算机，音乐，2017/2/13}，用户 v_2 = {男，22，计算机，IT 互联网，2017/4/16}。

$Sim_{Attr} = 0.2\times1 + 0.4\times1/2 + 0.2\times1 + 0.1\times0 + 0.1\times1/3 = 0.633.$

直观的来看，第一组用户相似程度较低，而第二组用户相似程度较高。最终的属性相似度运算结果也基本与观察相符合。

2.2. 用户评分相似度

用户评分对于精确推荐至关重要，而仅仅通过用户对项目的单一的评分关系来确定用户之间的相似度是不准确的，所以本文引入了基于评分的多种相似因子算法，将相似因子算法相结合可得到较为准确的评分相似度。

2.2.1. 用户评分的相似因子

由于用户对某项目的评分是呈非线性变化趋势的，所以通过引入 sigmoid 函数来表现用户相似度的非线性变换，使得用户的差异变得相对平滑。引入的用户评分相似因子算法如下：

$$Sim_1(u,v) = 1 - \frac{1}{1+\exp\left(-\left|R_{up} - R_{vp}\right|\right)} \tag{8}$$

R_{up} 与 R_{vp} 分别表示用户 u、v 对项目 p 的评分，若用户 u 与用户 v 对于项目之间的评分差越小，则式(8)的值越大，表示用户 U 与用户 V 的相似性取值越高，$Sim_1(u,v)$ 的取值范围为(0, 1/2)。

2.2.2. 用户偏好的相似因子

若用户不喜欢某项目就会给出其评分区间相对较低的分数，而喜欢该项目的话，分数就会较高。但由于用户的个人偏好与评分标准各有不同，为了尽量找到偏好的平衡点，引入用户各自的项目评分的平均分，为了消除因为评判标准不同带来的消极影响，用户偏好相似因子的算法则为：

$$Sim_2(u,v) = \frac{1}{1+\exp\left(-\left(R_{up} - \overline{R_u}\right)\left(R_{vp} - \overline{R_v}\right)\right)} \tag{9}$$

① $\left(R_{up} - \overline{R_u}\right), \left(R_{vp} - \overline{R_v}\right)$ 为一正向偏移、一负向偏移，且均较大偏移值。则 $Sim_2(u,v)$ 取值较小，此相似因素较低。

② $\left(R_{up} - \overline{R_u}\right), \left(R_{vp} - \overline{R_v}\right)$ 两正向偏移或者两负向偏移，且有较大偏移值。则 $Sim_2(u,v)$ 取值趋近于 1，此种情况下相似因素较高。

③ 若 $\left(R_{up} - \overline{R_u}\right), \left(R_{vp} - \overline{R_v}\right)$ 一正向偏移、一负向偏移，但是偏移值较小，则 $Sim_2(u,v)$ 趋近于 1/2，则相似因素趋于中间值附近。

2.2.3. 用户评价项目的相似因子

$$Sim_3\left(U_i, U_j\right) = \frac{\left|I_{U_i} \cap I_{U_j}\right|}{\left|I_{U_i} \cup I_{U_j}\right|} \tag{10}$$

用户 U_i 与用户 U_j 共同评分的项目越多，则 $Sim_3\left(U_i, U_j\right)$ 数值越大，相似因子越大，相似程度越高。

2.2.4. 相似因子综合计算

结合上诉三种相似因子算法，可以得到基于评分的最终的相似度计算方法：

$$Sim_{score}(u,v)=\left(\frac{1}{I_{uv}}\sum_{i\in I_{uv}}Sim_1(u,v,i)\cdot Sim_2(u,v,i)\right)\cdot Sim_3(u,v) \tag{11}$$

2.3. 本文算法设计

本文算法设计流程图如图 1 所示：

输入：用户基本属性矩阵、用户评分矩阵

输出：推荐集合

步骤：

Step 1：根据用户基本属性计算出属性相似度矩阵；

Step 2：根据用户历史行为记录计算出评分相似度矩阵；

Step 3：根据用户属性与评分相似度矩阵计算出最终用户相似度矩阵；

Step 4：根据用户最终相似度矩阵得到待测用户 u 最相似的 K 个近邻，通过式(12)可以得到相应的预测评分，以获得 TOP-N 推荐方案。

$$\overline{r_{ui}}=\frac{\sum_{k\in N_K}Sim(u,i)\cdot r_{ki}}{\sum_{k\in N_K}Sim(u,i)} \tag{12}$$

N_K 代表与用户相似度最高的 K 个邻居；r_{ki} 表示用户 k 对项目 i 的评分。

3. 实验数据展示

3.1. 数据集

实验采用了 MovieLens 100k 数据集，该数据集包括了 943 名用户对 1682 部电影的 100,000 条评价，评分取值为[1, 2, 3, 4, 5]，数值升高表示用户的喜好程度越高。每位用户至少评价了 20 部电影，这些用户属性包括了：年龄、性别、职业、所在地等信息，评分信息中包括了时间戳信息。本文采用这些信息作

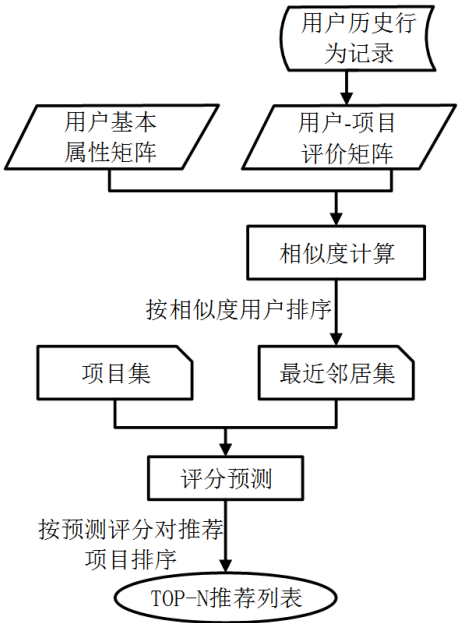

Figure 1. Algorithm recommendation flow chart in this paper

图 1. 本文算法推荐流程图

为用户基本属性，并将各个特征属性的权重值按统计结果分别设置为(0.2, 0.4, 0.2, 0.1, 0.1)。

3.2. 评价标准

本文采用协同过滤推荐算法中最常见的推荐质量度量方法：平均绝对误差MAE (Mean Absolute Error, MAE)，作为度量本文算法优劣的标准。MAE 通过计算预测评分与实际评分之差的绝对值来度量预测的准确性。MAE 的值越小，表明预测越准确，推荐效果越好。

$$\mathrm{MAE}=\frac{\sum_{i=0}^{N}\left|\overline{r_{ui}}-r_{ui}\right|}{N} \tag{13}$$

r_{ui} 表示用户 u 对一系列项目的实际评分；N 为项目个数，$\overline{r_{ui}}$ 表示为项目的预测分数。

3.3. 实验结果与分析

3.3.1. 预测准确率

因为考虑到用户的样本数对最终的 MAE 会产生影响，所以我们将样本数设为了 200 人和 500 人进行计算，并将本文算法与传统的 person 相似度算法、修正的 cosin 相似度算法相比较，如图 2、图 3 所示。

由图 2、图 3 可以得出，用户样本数在为 200、500 的时候，本文算法的准确性和预测效果明显优于传统算法；当样本数从 200 增加到 500 的时候，本文算法与两种传统算法的准确性都提高了很多；而且 MAE 随最近邻个数 K 的增而降低，当最近邻居个数 K 在 30 的时候，三种算法的效果基本趋于稳定，说明本文算法的稳定性也很好。综合来看，本文算法在提高推荐质量方面有较好表现。

3.3.2. 数据的稀疏度

从图 4 可以看出来，本文算法与其他两种算法虽然均受到了样本数据的稀疏度影响，但是本文算法的预测准确性依然领先于传统的 Person 算法和修正的 Cosin 算法。随着样本数据稀疏度增高，MAE 迅速上涨，使得推荐质量逐渐降低。当稀疏度接近 100%，甚至达到冷启动标准的时候，本文算法 MAE 大约为 0.87，达到了样本数为 200 时的推荐水平，反映出本文能有效的适应不同的数据稀疏度环境，预测准确度有良好的稳定性。

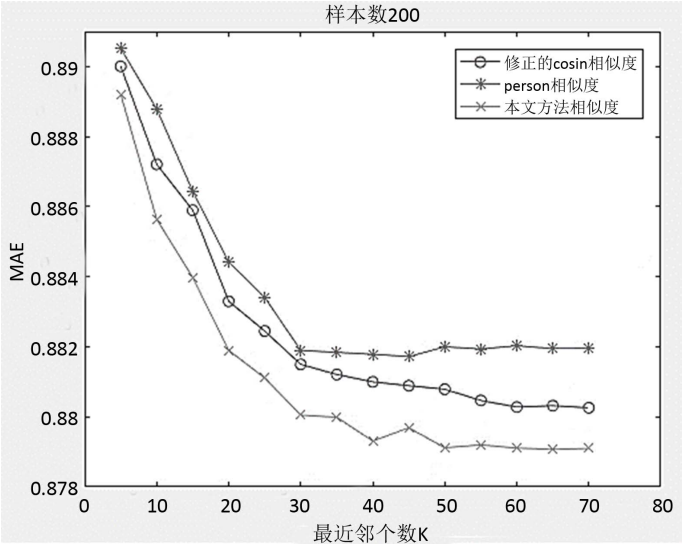

Figure 2. The effect comparison of the algorithm
图 2. 算法的效果对比

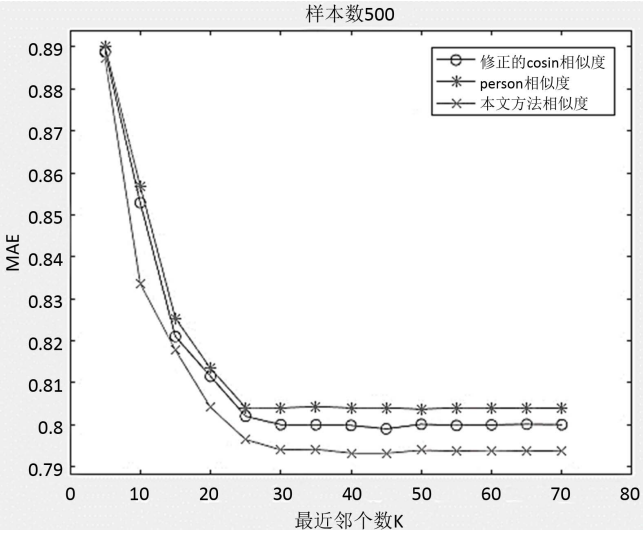

Figure 3. The effect comparison of the algorithm
图 3. 算法的效果对比

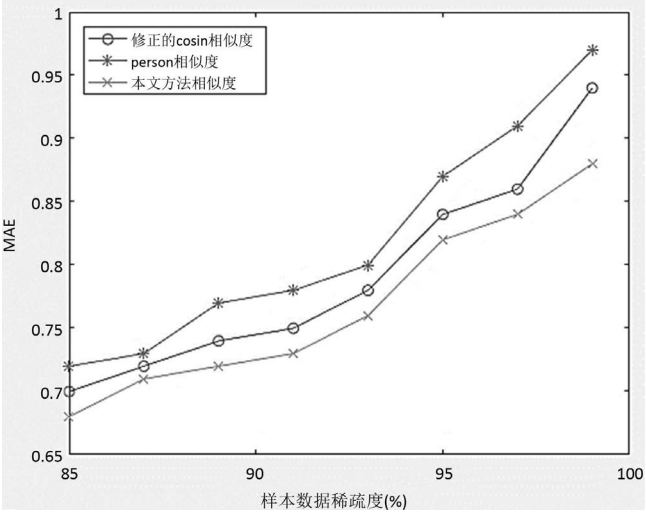

Figure 4. MAE of the algorithm
图 4. 算法的 MAE

4. 结束语

本文提出了基于用户基本属性与评分相似因子的结合算法，将用户基本属性赋予权重值，然后同用户评分信息的三个相似因子相结合，得到一种新的相似度计算方法。实验表明新算法使得用户的相似度计算更加精确，而且在用户冷启动阶段向着非冷启动阶段过渡的时候比较平滑，不会出现大幅度的变化情况，具有较好的实际应用价值。

由于本文只利用了用户的部分显性属性，并未结合项目属性和用户的一些隐性属性，接下来可以利用这些方面进行进一步的研究。

参考文献 (References)

[1] Ioannis, K. and Vassilios, S. (2009) On Social Networks and Collaborative Recommendation. *Proceedings of the* 32*nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, New York, 195-202.

[2] Sarwar, B.M., Konstan, J.A., Borchers, A., *et al.* (1998) Using Filtering Agents to Improve Prediction Quality in the Grouplens Research Collaborative Filtering System. *Proceedings of the* 1998 *ACM Conference on Computer Supported Cooperative Work*, November 14-18 1998, Seattle, 345-354. https://doi.org/10.1145/289444.289509

[3] Pazzani, M.J. and Billsus, D. (2007) Content-Based Recommendation Systems. *The Adaptive Web*, 325-341. https://doi.org/10.1007/978-3-540-72079-9_10

[4] Cacheda, F., Carneiro, V., Fernandez, D., *et al.* (2011) Comparison of Collaborative Filtering Algorithms: Limitations of Current Techniques and Proposals for Scalable, High-Performance Recommender Systems. *ACM Transactions on the Web*, **5**. https://doi.org/10.1145/1921591.1921593

[5] Sarwar, B., Karypis, G., Konstan, J., *et al.* (2001) Item-Based Collaborative Filtering Recommendation Algorithms. *Proceedings of the* 10*th International Conference on World Wide Web*, May 1-5 2001, Hong Kong, 285-295. https://doi.org/10.1145/371920.372071

[6] Liang, C.Y. and Leng, Y.J. (2014) Collaborative Filtering Based on Information-Theoretic Co-Clustering. *International Journal of Systems Science*, **45**, 589-597. https://doi.org/10.1080/00207721.2012.724109

[7] Bobadilla, J., Serradilla, F. and Bernal, J. (2010) A New Collaborative Filtering Metric That Improves the Behavior of Recommender Systems. *Knowledge-Based Systems*, **23**, 520-528. https://doi.org/10.1016/j.knosys.2010.03.009

[8] Reina, D.G., Toral, S.L., Johnson, P., *et al*. (2014) Improving Discovery Phase of Reactive Ad Hoc Routing Protocols Using Jaccard Distance. *The Journal of Super-Computing*, **67**, 131-152. https://doi.org/10.1007/s11227-013-0992-x

[9] Deshpande, M. and Karypis, G. (2004) Item-Based Top-N Recommendation Algorithms. *ACM Transactions on Information Systems*, **22**, 143-177. https://doi.org/10.1145/963770.963776

[10] Park, Y.J. and Tuzhilin, A. (2008) The Long Tail of Recommender Systems and How to Leverage It. *Proceedings of the* 2008 *ACM Conference on Recommender Systems*, October 23-25 2008, Lausanne, 11-18. https://doi.org/10.1145/1454008.1454012

[11] Martinez, L., Perez, L.G. and Barranco, M.J. (2009) Incomplete Preference Relations to Smooth out the Cold-Start in Collaborative Recommender Systems. 2009 *IEEE Annual Meeting of the North American Fuzzy Information Processing Society* (*NAFIPS*), June 14-17 2009, Cincinnati, 1-6. https://doi.org/10.1109/NAFIPS.2009.5156454

[12] Gunawardana, A. and Meek, C. (2008) Tied Boltzmann Machines for Cold Start Recommendations. *Proceedings of the* 2008 *ACM Conference on Recommender Systems*, October 23-25 2008, Lausanne, 19-26. https://doi.org/10.1145/1454008.1454013

[13] Gunawardana, A. and Meek, C. (2009) A Unified Approach to Building Hybrid Recommender Systems. *Proceedings of the* 3*rd ACM Conference on Recommender Systems*, October 23-25 2009, New York, 117-124. https://doi.org/10.1145/1639714.1639735

[14] Park, S.T. and Chu, W. (2009) Pairwise Preference Regression Forcold-Start Recommendation. *Proceedings of the* 2008 *ACM Conference on Recommender Systems*, October 23-25 2008, Lausanne, 21-28.