【信息科学与控制工程】　　doi：10.11809/scbgxb2014.05.025

协同过滤算法在电影推荐中的应用

王　越，程昌正

（重庆理工大学 计算机科学与工程学院，重庆　400054）

摘要：推荐系统是很多网站最关心的机器学习应用，因为其准确率的提高对网站收入有直接贡献。构建了一个电影推荐系统，使用基于相似度的 KNN 算法、Baseline 预测、随机梯度下降以及 SVD 共 4 种方法进行预测评分。使用 RMSE 评价标准，对比了不同算法预测精度的差异和不同参数设定下预测精度的变化。

关键词：电影评分预测；RMSE；随机梯度下降；SVD

本文引用格式：王越，程昌正. 协同过滤算法在电影推荐中的应用[J]. 四川兵工学报，2014(5)：86－88.

中图分类号：TP301.6　　**文献标识码**：A　　**文章编号**：1006－0707(2014)05－0086－03

Application of Collaborative Filtering Algorithms in Movie Recommendation

WANG Yue，CHENG Chang-zheng

（School of Computer Science and Engineering，Chongqing University of Technology，Chongqing 400054，China）

Abstract：Recommender system is a highly concerned machine learning application of many Internet companies，for its accuracy has direct contribution to the company's revenue. In this paper，we built a movie rating prediction system to recommend movies to its users. We applied many advanced machine learning related algorithms - KNN，Stochastic Gradient Descent，BaseLine，SVD. Using RMSE evaluation，the simulation result shows different accuracy as well as varying performance under different learning parameters.

Key words：movie rating prediction；RMSE；stochastic gradient descent；SVD

Citation format：WANG Yue，CHENG Chang-zheng. Application of Collaborative Filtering Algorithms in Movie Recommendation[J]. Journal of Sichuan Ordnance，2014(5)：86－88.

推荐系统是网站服务器向用户推荐内容的一组程序。提高推荐系统的效率，可以增加电子商务网站的销售额；或者帮助用户在享用新闻、电影、音乐等资讯的过程中，更容易发现感兴趣的内容，提升用户体验。评分是用户对特定对象喜好程度的量化。在推荐系统的研究中，先使用历史评分预测未知评分，然后按预测分数从高到低的次序向用户推荐内容的方法称为协同过滤[1]（collaborative filtering）。许多音乐、视频的网站上都有“猜你喜欢”的个性化信息，并且鼓励用户对内容打分。使用 MovieLens 数据集，构建了一个电影评分预测系统。结果表明，预测评分可以有效地作为电影推荐的依据，帮助用户发现喜爱的电影。

1　数据预处理与评价标准

考虑到数据的真实性和相关研究的可参照性，选择公共的开放数据集 MovieLens DataSet1 作为模型的训练集和测试集。该数据集包含 100，000 条电影评分记录，其中每位用户至少提供了不同电影的 20 个评分。记录格式是“用户 ID，电影 ID，评分，评分时间”。用户 ID 从 1 到 943，电影 ID 从 1 到 1682，评分采用 5 分制。实验不使用数据集中的时间信息。数据集预先被划分成 uabase 和 uatest 2 部分，前者包含 90570 条记录，用于训练模型；后者包含余下的 9430 条记录用于测

收稿日期：2014－01－12

作者简介：王越（1961—），男，博士，教授，主要从事数据挖掘、数据库技术、嵌入式系统及应用研究。

试模型。评分预测结果的全体最后与真实记录对比,得出准确率的数值评估。实验平台在 Matlab 的免费版本 Octave 上完成。

协同过滤算法需要对评分进行大量数值计算,因此把数据集存储为一个 9431682 的大型矩阵 U

$$U_{ui} = \begin{cases} r_{ui} \\ 0 \end{cases}$$

其中,r_{ui}是用户 u 对电影 i 的评分,未有记录则为 0。显然,矩阵中大部分元素为 0,其稀疏等级为

$$1 - \frac{100\ 000}{943 \times 1\ 682} = 0.936\ 95$$

预处理以后,算法的任务是对测试集中每条记录的评分作预测,预测值与真实值越接近表示算法越成功。按照惯例,评估数值采用 RMSE 公式计算

$$RMSE = \sqrt{\frac{1}{|A_{\text{test}}|}\sum_{(u,i)\in A_{\text{test}}}(\hat{r}_{ui} - r_{ui})^2}$$

其中 A_{test}包含从数据集划分出的 uatest 中的评分记录。预测分数不需要取整,这样做一方面使 RMSE 比取整后更能真实反映不同算法的准确率;另一方面是当按分数高低次序进行推荐的时候,不会出现评分相同的内容。*RMSE* 值越小,反映预测准确率越高。

2　评分预测算法

2.1　KNN

为了应用 KNN 方法,需要计算每部电影两两之间的相似程度。度量方法采用皮尔逊相关系数

$$\text{Similarity}(i_1,i_2) = \frac{\sum_{u\in U(i_1)\cap U(i_2)}(R_{u,i_1} - \bar{R}_u)(R_{u,i_2} - \bar{R}_u)}{\sqrt{(\sum_{u\in U(i_1)\cap U(i_2)}(R_{u,i_1} - \bar{R}_u)^2)(\sum_{u\in U(i_1)\cap U(i_2)}(R_{u,i_2} - \bar{R}_u)^2)}}$$

其中 $U(i_1)\cap U(i_2)$包含同时对电影 i_1 和电影 i_2 进行了评分的用户,$\bar{R}_u$ 是用户 u 给出的所有评分的算术平均值。电影相似度的计算结果组成一个 1 682 × 1 682 的对称矩阵 S,其中 $S_{ij}\in[-1,1]$。

评分预测式为

$$P_{u,i} = \frac{\sum_{k_j\in K\text{ similar Movies}}(S_{i,k_j}\times R_{u,k_j}}{\sum_{k_j\in K\text{ similar Movies}}|S_{i,k_j}|}$$

这里,KNN 方法的准确率取决于不同的 K 值以及“最相似电影”的定义。文献[3]中详细讨论了以上评分预测的计算公式。实验对 uabase 和 ubbase 设定不同的 K 值进行训练和预测,得到以下 RMSE 结果曲线,如图 1 所示。

在本实验中,KNN 方法需要计算 1 682 部电影两两之间的相似度,计算量大,训练时间较长。有些电影获得的评分较少或两部电影之间没有相同的一组用户对其评分,这时它们的相似度无法计算,可设为最小值 -1,该值在 K 邻近选择时不被考虑。

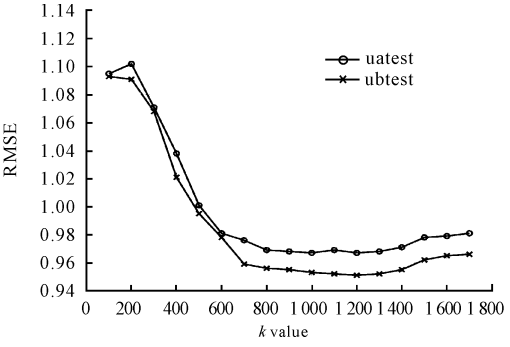

图 1　KNN 的 RMSE 曲线

2.2　BaseLine 算法

文献[2]中介绍了 BaseLine 评分预测算法。评分预测公式是 $P_{u,i} = \mu + b_u + b_i$。$\mu$ 是训练集里面全体评分的算术平均值;b_u 试图反映用户 u 在打分时对于全体平均评分的偏离程度,这里的想法是对不同用户总是打分偏高或者偏低这种现象的考虑;类似地,公式中 b_i 反映电影比起平均来说更受欢迎的程度。在计算上,该模型需要求解以下平方最小问题

$$\min_{b*}\sum_{(u,i)\in K}(r_{ui} - \mu - b_u - b_i)^2 + \lambda_1(\sum_u b_u^2 + \sum_i b_i^2)$$

由于求解以上最小化问题需要对个参数进行最优化,计算复杂度很高,所以 BaseLine 算法一般采用以下脱耦方法来计算模型的全体参数[2]:

$$b_i = \frac{\sum_{u\in R(i)}(r_{ui} - \mu)}{\lambda_2 + |R(i)|}$$

$$b_u = \frac{\sum_{u\in R(u)}(r_{ui} - \mu - b_i)}{\lambda_3 + |R(u)|}$$

使用该方法求解出来的参数存在误差,但它计算复杂度低,实现简单。设定 $\lambda_2 = 25$ 和 $\lambda_3 = 10$ 对 uabase 和 ubbase 进行训练后分别对 uatest 和 ubtest 进行预测,得到以下 RMSE 结果,如表 1 所示。

表 1　BaseLine 算法的 RMSE 结果

DataSet	uatest	ubtest
RMSE	0.966 48	0.977 37

相比 KNN 算法,BaseLine 算法不需要离线计算每部电影两两之间的相关系数,训练时间短,预测精度较高。对于不同的数据集,一般通过试验训练找到参数的最优设定。

2.3　随机梯度下降

随机梯度下降法的评分预测式为[2]

$$\hat{r}_{ui} - q_i^T p_u$$

其中:$q_i\in R^k$ 反映了某部电影各种特征的对全体用户的受欢迎程度;$q_u\in R^k$ 反映了某位用户对所有电影的喜爱程度。对于训练集会有预测误差 $e_{ui} = r_{ui} - \hat{r}_{ui}$。算法使用以下随机梯度下降法迭代参数

$$q_i := q_i + \alpha(e_{ui}\cdot p_u - \beta\cdot q_i)$$

$$p_u := p_u + \alpha(e_{ui} \cdot q_i - \beta \cdot p_u)$$

q_i 和 p_u 初值各分量初始化为[0,1]区间上均匀分布的随机变量。经过若干试验,选择了 $\alpha = 0.02$ 和 $\beta = 0.05$ 作为学习率,对于不同的 k 值,有以下 RMSE 结果曲线,如图 2 所示。

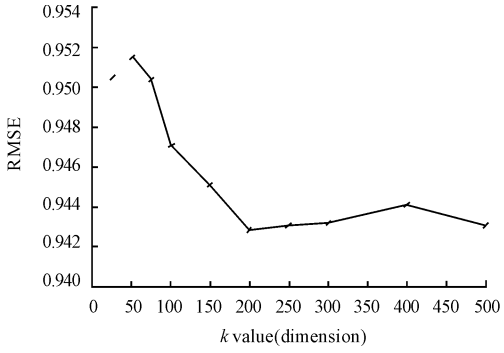

图 2 不同 k 值(维度)下的 RMSE 曲线

随机梯度下降法在训练过程中占用的内存较少,对于每部电影和每位用户,仅需为其保存一个维度为 k 的特征向量,适合数据量特别庞大的场合。算法利用从矩阵分解中获取的隐含语义关系对用户的喜好进行判断[4],获得了比 KNN 和 BaseLine 算法更好的预测精度。

2.4 SVD

奇异值分解(Singular Value Decomposition,SVD)可以看作是综合了 Baseline 预测法和随机梯度下降算法的评分预测算法。文献[2]中给出的评分预测式为

$$\hat{r}_{ui} = \mu + b_i + b_u + q_i^T p_u$$

其中 b_i, b_u, q_i, p_u, μ 的定义与 2.2 小节和 2.3 小节中描述的一致,和的初值仍然采用脱耦方法求得的结果,q_i 和 p_u 的初值仍然是 2.3 小节所述的随机向量。所有被训练参数的迭代公式如下:

$$b_u := b_u + \alpha_1 \cdot (e_{ui} - \beta_1 \cdot b_u)$$

$$b_i := b_i + \alpha_1 \cdot (e_{ui} - \beta_1 \cdot b_i)$$

$$q_i := q_i + \alpha_2 \cdot (e_{ui} \cdot p_u - \beta_2 \cdot q_i)$$

$$p_u := p_u + \alpha_2 \cdot (e_{ui} \cdot q_i - \beta_2 \cdot p_u)$$

实验中,参数设定成 $\alpha_{1,2} = 0.01$, $\beta_{1,2} = 0.04$。在不同的 k 值下的 RMSE 曲线,如图 3 所示。

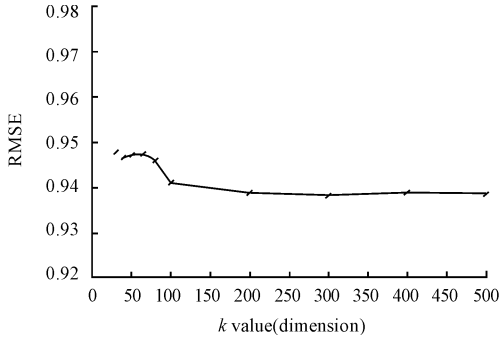

图 3 svd 算法在不同 k 值(维度)下的 RMSE 曲线

3 结束语

在我们的电影推荐系统中,对 MovieLens 数据集采用了 4 种评分预测算法,包括使用皮尔逊相关系数的基于电影相似度 KNN 算法、Baseline 预测器、随机梯度下降以及奇异值分解。其中,随机梯度下降和奇异值分解方法表现了良好的预测精度。这类基于样本随机信息和矩阵分解原理的算法,是机器学习这门学科中较新的理论,值得进一步深入研究和试验。它们的算法实现简洁直观,在高效矩阵计算程序库的支持下,表现出了良好的性能。

参考文献:

[1] Breese J, Heckerman D, Kadie C. Empirical Analysis of Predictive Algorithms for Collaborative Filtering[Z]. Technical Report of Microsoft Research, 1998.

[2] Ricci F, Rokach L, Shapira B, et al. Recommender Systems Handbook: A Complete Guide for Scientists and Practioners [M]. USA: Springer, 2011.

[3] Sarwar B, Karypis G, Konstan J, et al. Item-Based Collaborative Filtering Recommendation Algorithms[C]//Proceedings of the 10th international conference on World Wide Web. [S. l.]: [s. n.], 2001: 285 - 295.

[4] Deerwester S, Dumais S, Furnas G W, et al. Indexing by latent semantic analysis[J]. Journal of the American Society for Information Science, 1990, 41(6): 391 - 407.

[5] Gorrell G. Generalized Hebbian algorithm for incremental singular value decomposition in natural language processing [C]//Proceedings of Conference on EACL. [S. l.]: [s. n.], 2006.

[6] Adomavicius G, Tuzhilin A. Towards the next generation of recommender systems: A survey of the state-of-art and possible extensions[J]. IEEE Transactions on Knowledge and Data Engineering, 2005, 17(6): 734 - 749.

[7] Goldberg D, Nichols D, Oki B M, et al. Using collaborative filtering to weave an information tapestry[J]. Communications of the ACM, 1992(35): 61 - 70.

[8] 刘东辉,彭德巍,张晖. 一种基于时间加权和用户特征的协同过滤算法[J]. 武汉理工大学学报, 2012(5): 144 - 148.

[9] 余肖生,孙珊. 基于网络用户信息行为的个性化推荐模型[J]. 重庆理工大学学报: 自然科学版, 2013(1): 47 - 50.

(责任编辑 杨继森)