改进的增量奇异值分解协同过滤算法

顾 晔,吕红兵
GU Ye,LV Hongbing
浙江大学 计算机学院,杭州 310027
College of Computer Science,Zhejiang University,Hangzhou 310027,China

GU Ye,LV Hongbing.Improved algorithm of incremental singular value decomposition collaborative filtering.Computer Engineering and Applications,2011,47(11):152-154.

Abstract: As a kind of programs and algorithms, recommendation systems provide personalized recommendations by measuring the preference levels of users(customers) on the given commodities.More broadly, recommender systems attempt to profile user preferences and model the interaction between users and products.Compared with other singular value decomposition methods,the improved incremental singular value decompositon allows the singular value decomposition of a user-item rating matrix to be learned based on single observation presented serially,and produces singular vector pairs one at a time,each time forms the most significant one at present.The algorithm has minimal memory requirements,high scalability and is particularly suitable for handling large data sets.The technique is demonstrated on the Netflix dataset.
Key words: singular value decomposition;recommender system;collaborative filtering

摘 要:推荐系统作为一种程序算法,是通过度量用户对给定商品的的喜好程度做个性化推荐。广泛地说,推荐系统试图总结出用户的个人喜好,并在用户和商品之间建立一种关系模型。与其他奇异值分解方法相比,改进的增量奇异值分解协同过滤算法基于一系列评分值对用户-商品矩阵进行分解,每次产生一对当前最重要的特征向量。算法有着最小的内存需求,扩展性高,特别适合处理大规模数据集;算法的有效性在Netflix数据集上得到了验证。
关键词:奇异值分解;推荐系统;协同过滤
DOI:10.3778/j.issn.1002-8331.2011.11.043 **文章编号:**1002-8331(2011)11-0152-03 **文献标识码:**A **中图分类号:**TP311

1 介绍

推荐系统[1]通过分析用户对商品的兴趣模式,做出符合用户口味的个性化商品推荐。实质上,推荐系统就是尝试找到用户偏好,并在用户和商品之间找到交互关系。由于推荐系统能够给用户带来更多的便利和全新的购物体验,在国内外众多电子商务领先者如亚马逊和阿里巴巴都有着广泛深入的应用。

一般来说,推荐系统通常使用两种不同的策略。基于内容的方法(Content-Based Approach)为每个用户和商品建立一个文档来刻画他们的特征。比如一个电影文档可以包括该电影的年代,演员和内容简介等。一个用户文档可以包括用户的性别、年龄、职业等人口特征。这样,程序就可以搜索用户和与之匹配的商品。但是,基于内容的方法需要搜集额外的信息,而这些信息往往不容易得到。

协同过滤(Collaborative Filtering,CF)[2]依靠的是用户过去行为,而并不需要建立一个文档。这种方法在收集用户判断(比如评分)的基础上,分析哪些用户具有相似的爱好,以此作为推荐的依据。和基于内容的方法相比,协同过滤具有前者没有的一些优点[3]:(1)支持过滤不容易分析内容的商品。(2)基于特征与爱好的过滤。(3)具有发现意想不到的巧妙推荐。但是,协同过滤推荐系统仍然存在两个重要问题需要克服[4]:(1)提高协同过滤算法的应用数据的规模。现代商业系统中的用户、商品数量是非常多的,如何提高协同过滤算法在大规模数据上的实时计算变得非常重要。(2)提高商品推荐精度。如果推荐系统一直提供用户并不喜欢的商品,用户会对推荐系统失去兴趣。正是由于协同过滤有如此良好的特性,以及现实存在的诸多问题,因而引起了研究者的研究兴趣[3],并得到了广泛的商业应用[4-5]。

从更抽象的层次来看,协同过滤可以看作是矩阵填充(Matrix Completion)问题:用户对商品的评分组成一个评分矩阵,每个用户可能仅对其中一小部分商品做过评价,因此这个矩阵通常是非常稀疏的。目标就是根据已知的元素来估计缺失的矩阵元素值。一般情况下,使用特征值分解(Singular Value Decomposition,SVD)是为了对原问题进行降维处理,在一个子空间内重新表达数据。然而由于CF中的矩阵大多数元素未知,在这里使用SVD几乎是为了达到相反的目的——估计矩阵中未知元素来拓展原来已有的数据。

2 相关工作

协同过滤系统中最常用的是最近邻接点(K-Nearest Neighborhood,KNN)方法。由于这种方法简单、直观,因此广

作者简介:顾晔(1984—),男,硕士研究生,研究方向为模式识别与人工智能;吕红兵(1966—),男,副教授,硕导。E-mail:ghtwht.gy@gmail.com
收稿日期:2009-10-13;**修回日期:**2010-01-25

泛使用在早期CF系统中。基于用户的KNN方法通过使用统计学方法来找到一组被称为邻居的相似用户集合来估计未知评分。具体地说,为了计算用户u对商品i的评分评分r_{ui},找到一组为商品i打过分,并且和当前用户相似的"邻居"集合$N(u;i)$,那么待估r_{ui}则为这些邻居的加权平均分:

$$r_{ui}=\frac{\sum_{v\in N(u;i)}s_{uv}r_{vi}}{\sum_{v\in N(u;i)}s_{uv}} \tag{1}$$

式(1)中,相似度s_{uv}作为算法的核心,是选择邻居$N(u;i)$的依据,通常使用pearson相关系数或商品空间两用户的余弦夹角。基于商品的KNN和基于用户的方法类似,而文献[4]指出,基于商品的KNN能够极大地提高算法速度,同时提供更好的预测精度。

尽管KNN方法实现起来相对简单,但还有一些需要克服的缺点。首先是计算量太大,这主要体现在s_{uv}的计算过程中。现代商业数据库中的用户和商品数量都是百万甚至千万级别的,要在这么多数据中作两两比较,即使计算复杂度是$O(n^2)$,也仍需要很长时间,而且存储相似度数据需要占用大量的内存。总之,KNN方法的扩展性受到了很大的约束。另外,使用这种邻居信息往往不能适应真实数据情况。考虑极端的情况,如果一件商品和另一件商品非常相似,在计算时希望这件相似的商品得到所有的权重,这时KNN还是必须得使用一些没用的数据来使结果产生较大的偏差。

基于奇异值分解的协同过滤算法在生成评分预测的时候仅需要做一些简单的算术运算,因此具有很好的在线性能,但在计算SVD过程中代价很高,而且还要面对矩阵稀疏的问题。文献[6]提出了使用折叠(folding-in)技术的增量SVD算法,极大地提高了奇异值分解速度,但是由于增量SVD空间特征向量的非正交性,使得计算精度下降。Eigenstate[7]使用PCA分解结合递归聚类的方法来估计评分。它从原矩阵中抽出一部分评分数据形成一个子集,在这个子集组成的相对完整的矩阵基础上使用PCA方法,从而解决了矩阵稀疏问题。然而这种忽略其他评分数据的做法降低了预测精度,而且在实际情况下,如何抽取数据是个不容易解决的问题。其他方法依赖于通过插值来填充未知评分数据,从而使得矩阵完整。比如文献[8]使用相关商品正则化后的平均评分填充缺失数据,文献[9]提出了在逐步迭代提高插值精确度的同时做奇异值分解的方法。在一个典型的CF应用中,需要插值的缺失元素远远多于已知的评分数据,因此这些依赖插值的方法会因为扭曲的插值而使得结果不精确。同时,插值计算量大,在面对海量数据时变得无能为力。

3　改进的增量奇异值分解的算法

3.1　背景

给出一个$m\times n$用户-商品评分矩阵$\boldsymbol{R}$,SVD计算原矩阵的最优秩为f的近似矩阵$\boldsymbol{R}^f$,它可以表示成两个矩阵$\boldsymbol{P}_{m\times f}$和$\boldsymbol{Q}_{n\times f}$的乘积,其中$f\leqslant m,n$。即在所有秩为$f$的矩阵中,$\boldsymbol{R}^f=\boldsymbol{P}\boldsymbol{Q}^{\mathrm{T}}$使得Frobenius范数$\|\boldsymbol{R}-\boldsymbol{R}^f\|$最小。在这个意义上来看,矩阵$\boldsymbol{R}^f$抽取了原数据$f$个最重要的特征,丢掉了一部分可能被认作是噪音的不重要数据。每个未知的评分可以由$\boldsymbol{R}^f_{ui}$来表示,也就是矩阵$\boldsymbol{P}$的第u行和矩阵$\boldsymbol{Q}$的第i行的内积。因此,可以把$\boldsymbol{P}$看成是用户特征矩阵,$\boldsymbol{Q}$为商品特征矩阵。在信息获取(Information Retrieval)领域,这就是熟知的隐含语义索引[10](Latent Semantic Indexing)。

3.2　本文的方法

将基于SVD的技术用于CF问题,会面临矩阵稀疏的问题。传统的SVD计算需要知道矩阵中的所有元素。文献[6-9]提出了使用各种方法来解决矩阵分解中的元素缺失问题。这里给出如下方法。

设所有已知的评分r_{ui}组成一个集合为K,在这个集合范围内,SVD的目标是最小化下式:

$$\sum_{(u,i)\in K}(r_{ui}-\boldsymbol{p}_u\boldsymbol{q}_i^{\mathrm{T}})^2 \tag{2}$$

其中$\boldsymbol{p}_u$是$\boldsymbol{P}$的第u行,对应着用户u、$\boldsymbol{q}_i$是$\boldsymbol{Q}$的第i行,对应着商品i。这里一个重要的问题就是避免对相对数据量比较小的商品和用户产生过适应现象,因此需要对上面的模型作正则化处理,加上惩罚项来约束$\boldsymbol{p}_u$和$\boldsymbol{q}_i$。于是式(2)变成:

$$\sum_{(u,i)\in K}(r_{ui}-\boldsymbol{p}_u\boldsymbol{q}_i^{\mathrm{T}})^2+\lambda(\|\boldsymbol{p}_u\|^2+\|\boldsymbol{q}_i\|^2) \tag{3}$$

为了最小化式(3),分别固定$\boldsymbol{Q}$和$\boldsymbol{P}$,得到一系列可解决的最小二乘问题。设$\hat{r}_{ui}$为预测评分,Δ_{ui}为真实值和预测值之间的误差。使用简单的梯度下降方法,得到特征向量中一个元素的解:

$$\Delta_{ui}=r_{ui}-\hat{r}_{ui} \tag{4}$$

$$p_{uk}=p_{uk}+l_{\mathrm{rate}}\times(\Delta_{ui}q_{ik}-\lambda p_{uk}) \tag{5}$$

$$q_{ik}=q_{ik}+l_{\mathrm{rate}}\times(\Delta_{ui}p_{uk}-\lambda q_{ik}) \tag{6}$$

式中l_{rate}为学习速率,$\boldsymbol{k}$表示第k个特征向量。对该元素来说,当训练达到最少迭代次数,或者误差停止下降的时候,停止当前特征向量的训练,开始训练下一个元素。当特征向量训练完后,评分的预测仅需计算两个对应用户、商品特征向量的内积,即:

$$\hat{r}_{ui}=\boldsymbol{p}_u\boldsymbol{q}_i^{\mathrm{T}} \tag{7}$$

由于特征向量维数f固定,因此预测的时间复杂度仅为$O(1)$。在用梯度方法对当前单个特征向量元素求偏导时,和其余特征向量相关的项均为0,因此回避了之前几个算法中需要迭代或插值求解矩阵中未知元素的问题。另外,从式(7)也可以看出,每次迭代使用一个评分数据,整个训练过程只需要存储特征向量的内存空间,因此特别适合于有非常大数据集的情况。

这种算法每次迭代更新仅需要很少的计算量,求解每个特征向量都是相同的时间。和批处理矩阵分解相比,数据量的多少并不影响更新计算速度。算法每次产生一个特征向量,并且保证该向量是目前为止最显著的特征向量。这意味着每次从矩阵中提取出来的都是最重要的特征,保证了算法的精度。与其他增量方法相比,这种学习方法区别在于前者收敛于整个数据集的特征值分解,而不是目前所见数据为止的一个最优解[11]。

4　实验

4.1　Netflix数据集

将算法在Netflix数据集上进行测试。Netflix数据集有超过1亿条的电影评分数据,包含了480 189个用户对17 770部电影的1~5之间的评分,评分高则表示用户喜欢这部电影。这是目前为止所知道的最大的公开可用商业数据集。为了测试

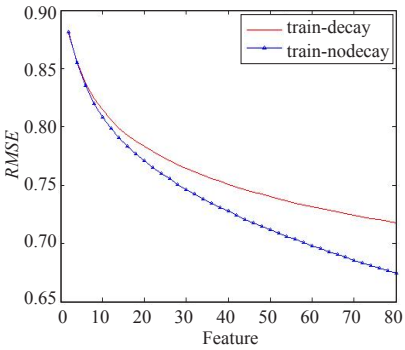

图1 训练集上*RMSE*随*f*的变化曲线图

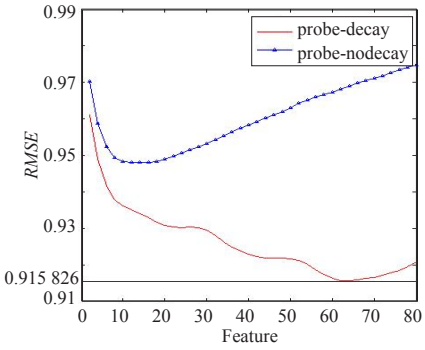

图2 探测集上*RMSE*随*f*的变化曲线图

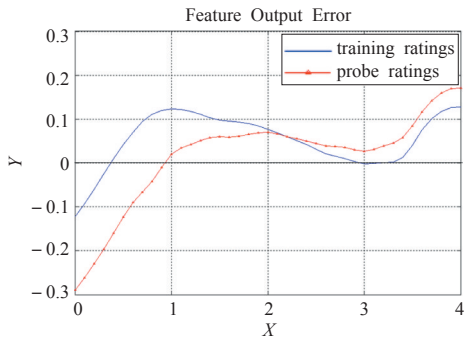

图3 特征误差曲线图

各种不同的算法,Netflix提供了两个测试数据集。一个是探测集(Probe Set),这个集合在原1亿条训练数据集中抽取140万条评分数据,因此在这个集合中,用户对电影的真实评分是知道的。另一个是测验集(Qualifying Set),测验集由280万条数据组成,但是没有提供电影评分的真实数据。两个数据集都挑选了一些难以预测的用户,让人们对他们的评分进行猜测。使用误差的均方根(RMSE)作为验证算法的标准:

$$RMSE=\frac{\sum_{(u,i)\in TestSet}(r_{ui}-\hat{r}_{ui})^2}{|TestSet|} \tag{8}$$

在对模型进行训练之前,将探测集从训练集中剥离,在剩下的数据集上训练出来的模型在探测集上进行预测评估。实验表明,如果包含探测集来训练,最后得到的*RMSE*将比没有包含探测集得到的结果降低0.03左右,这是一个很大的提高。因此,这种剥离探测集训练模型的方法将更接近真实预测情况。

4.2 结果与讨论

在使用SVD时,首先需要确定的是分解后矩阵的秩,也就是特征向量的维数*f*。设置的维数越多,则从数据中能得到更多的信息,相应的预测结果也应该更精确,然而实际情况下,在训练集上训练得到的模型往往容易产生过适应的现象。图1给出了在训练集上不同维数对应的*RMSE*。可以看到随着维数的增加,无论是否有正则项,*RMSE*一直在下降。图2给出了探测集上*RMSE*随维数变化的情况。带正则项的曲线在*f*=65时,*RMSE*达到最低0.915 826,接下来随着*f*的增大开始上升,出现了过适应的情况。而不带正则项的曲线RMSE一直比前者高,而且出现过适应的情况很早,说明正则项对解决过适应,提高结果精度都有很大帮助。

实际上,当使用更多地维数的时候,可以适当增大正则项系数λ来避免出现严重的过适应。另外从两幅图中还可以看出,曲线对应几个特征向量的*RMSE*下降得很快,越是后面的特征向量,曲线也越趋于平缓,这也从侧面印证了算法每次计算出来的都是最显著的特征向量。

即使在训练过程中加上了正则项,过适应仍然存在。当把要预测的评分和平均误差分别作为*X*轴和*Y*轴时,可以画出一个特征的"形状"。这里的误差指的是预测评分和真实值之间的偏离。图3是用实验中第一个得到的特征数据画出的图,把1~5的评分转到0~4的范围内。从图中可以看出,低的评分倾向于被预测得更低,而高的评分则倾向于被预测得更高。由此可以对式(2)作非线性优化,比如使用sigmoid函数,即$G(x)$=sigmoid(x),以此来修正特征向量过适应情况。

算法性能方面,使用的是Intel酷睿E2160 CPU,2 GB RAM的普通PC,对1亿条数据迭代120次训练一个特征向量需要10 min,而预测一个评分在0.1 ms之内。因此可以说,算法在保持较高准确度的前提下,仍具有很好的性能表现。

5 结论

本文提出了一种改进的增量奇异值分解算法。在Netflix数据集上的实验表明,该算法精度高,性能优异,特别适合处理大规模数据集。但是,如何发现并解决在训练阶段模型的过适应问题,仍值得继续探究。

参考文献:

[1] 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.

[2] 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.

[3] Herlocker J L, Konstan J A, Borchers A, et al.An algorithmic framework for performing collaborative filtering[C]//Proceedings of the 22nd ACM SIGIR Conference on Information Retrieval, 1999: 230-237.

[4] Sarwar B, Karypis G, Konstan J, et al.Item-based collaborative filtering recommendation algorithms[C]//Proceedings of the 10th International Conference on the World Wide Web, 2001: 285-295.

[5] Linden G, Smith B, York J.Amazon.com recommendations: Item-to-item collaborative filtering[J].IEEE Internet Computing, 2003, 7: 76-80.

[6] Berry M W, Dumais S T, O'brian G W.Using linear algebra for intelligent information retrieval[J].SIAM Review, 1995, 37(4).

[7] Goldberg K, Roeder T, Gupta D, et al.Eigentaste: A constant time collaborative filtering algorithm[J].Information Retrieval, 2001, 4: 133-151.

[8] Sarwar B M, Karypis G, Konstan J A, et al.Application of dimensionality reduction in recommender system-a case study[C]//Proceedings of Confrence on WEBKDD, 2000.

[9] Kim D, Yum B.Collaborative filtering based on iterative principal component analysis[J].Expert Systems with Applications, 2005, 28: 823-830.

[10] 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.

[11] Gorrell G.Generalized Hebbian algorithm for incremental singular value decomposition in natural language processing[C]//Proceedings of Confrence on EACL, 2006.