读 paper 笔记-Slope one predictors for online rating-based collaborative filtering. Daniel Lemire and Anna Maclachlan

总结:

文章摘要部分:

总结了 slopeone 算法是以 f(x)=x+b 为目标函数,并有易实现,快速调用,有一定合理性的准确率,并且支持在线查询和动态更新.

开头(Introduction):

首先阐明了作者设计这个算法的目标:1.易于实现和维护;2.快速更新:新输入的评分立刻会对预测产生影响;3.调用查询是有效的 4.缓解冷启动问题:只需要从第一个访问者收到少量评分;5 合理范围内的准率:不会因为在精准率的效提升而对模型的简要性和稳定性做出大的牺牲. Slopeone 算法的基本原理是"流行度差别",它名思义就是,当有两个商品时,减去两个商品的平均评分.本文的主要贡献就是展示 slopeone 算法拥有和以内存为基准的算法一样的准率但对协同过滤任务更协调.

相关的工作部分:

介绍了其他几种主要使用的算法以及三种 slopeone 算法.首先是以内存为基准的算法和以模型为基准的算法.以内存为基准的算法是一个在用户之间的已经过权重平均的相似度测试来预测.它的缺点主要有两个:一是对数据稀疏的伸缩性和敏感程度,因为使用相似度运算,所以不能做快速的在线查询调用;二是冷启动问题:因为算法需要计算用户之间的相似度所以要求有一个最少数量的用户群.在协同过滤的以模型为基准的算法中,一些是以线性代数为基准(PCA,SVD 等);一些是直接应用人工智能技术如贝叶斯算法,神经网络或者聚类.与以内存为基准的算法对比,以模型为基准的算法在调用查询的时间更快但是在学习和更新阶段更昂贵.所以当查询速度很重要的时候, 以模型为基准的算法的优先级更高.

作者在设计算法的预测函数时,主要考虑了两种,一种是以回归为基准做过权重的 f(x)=ax+b;另一种是更为简单的 f(x)=x+b.在经过作者测试后,f(x)=ax+b 的结果相较于以内存为基准的算法并没有显著提升.所以预测函数定为 f(x)=x+b.

作者简要介绍了以模型为基准的算法中的 baseline 算法和以内存为基准的算法中的 Pearson reference 算法.Baseline 算法中的相似度是通过商品评分减去所有商品的平均评分的残差乘积之和得出.然后预测的评分是对得到的相似度进行权重平衡得出.Pearson reference 算法的重点在于计算相关性系数,即商品评分减去所有商品平均评分残差的乘积除以商品评分减去所有商品平均评分的标准差.这个算法的预测的评分中还有一个关键的指数,case amplification power,来对相关性系数做权重平衡,以上的指标确保了这个算法的准确性.

作者接着介绍了三种 slopeone 算法. Slopeone 算法要计算偏差,偏差是由用户对不同商品 i,j 的评分之残差除以所有被评分过的商品 i,j 的数目得到的.而最后用户对商品 j 的预测的评分是用户的平均评分加上刚才的偏差之和除以所有商品 j 的个数的商得到的.由以上的推导,可以得出 slopeone 不依赖用户怎么评价一个单一的商品,但是依赖于用户的平均评分和那些用户评分过的商品.而这个事实引伸出了另一个潜在的问题:观测到的评分的数目并没有被考虑,这样会导致某个被大量评分的商品的权重会过大从而影响算法的预测结果.而 Weighted slopeone 算法就是为解决这个问题而设计了在计算预测评分时在分子分母都乘以商品对应的个数来做权重平衡.最后一种算法是 bi-polar slopeone 算法,这个算法考虑到了一种情况:某商品被大部分用户打高分或者低分会影响最后预测的评分.基于上面的情况,在这个算法中是把用户的平均评分设置为阈值来把商品划分为用户喜欢和不喜欢两类.接下来分别在用户喜欢和不喜欢的两个集中仿照 weighted slopeone 的算法计算出两个预测评分,最后再分别用用户喜欢的商品个数和用户不喜欢的商品个数作为得到的两个预测评分的权重计算对某一商品的预测评分作为结果.

在实验结果部分,作者是把 MAE 作为评价指标来衡量测试过的所有算法的表现.作者在两个数据集 EachMoicve,MovieLens 上实验了三种 slopeone 算法,三种 baseline 算法,以及 Pearson 算法.实现结果显示 slopeone 算法的 MAE 均优于 baseline 算法,并且有大约 1%的提升,与 Pearson 算法(代表以内存为基准的算法).的 MAE 差不多.

结论部分,作者总结了 slopeone 算法可以与成本更高的以内存为基准的算法相媲美,并且完成了文章开头设立的 5 个目标.同时把数据集分离为喜欢和不喜欢两部分能有效提升准确率.

心得

在一些算法中,预设函数并不一定是越复杂越好,往往数据集对预设函数的表现有重要的作用,就如同本文中 slopeone 算法的 f(x)=x+b 要比多用于回归的 f(x)=ax+b 以及更加拓展的 f(x)=ax^2+bx+c 的实际效果要好.

在一些已经提出的算法中,或许仍有小部分算法存在改进或者可提升的空间,可结合实际场景和数据集进行小的改动,或许有更好的的效果.比如本文中的 weighted slopone 算法对可能出现的某商品被评分次数过多而进行的权重平衡,以及 bi-polar slopone 在此基础上划分喜欢和不喜欢两个部分进一步优化来提升准确率.

同时读这篇文章,我个人对创建一个新的能够被广泛化认可的算法有了新的想法,很多现存的应用于同一场景的算法之间更多的是一种 trade-off.有的算法可能运算速度很快,但伸展性和更新能力不强;另一种可能恰恰相反,就如我们的内存和硬盘.未来能够取得新的巨大突破的算法或能结合已有算法的优点,或者应该是用新的数学或者统计原理另起炉灶来实现吧.

最后的感悟就是 BI 的算法其实不一定会非常复杂,但它对能满足实际业务需求的要求很高.