**基于item协同过滤算法实现**

原创 2016年03月28日 15:18:23

* 标签：
* [协同过滤算法](http://so.csdn.net/so/search/s.do?q=%E5%8D%8F%E5%90%8C%E8%BF%87%E6%BB%A4%E7%AE%97%E6%B3%95&t=blog) /
* [matlab实现代码](http://so.csdn.net/so/search/s.do?q=matlab%E5%AE%9E%E7%8E%B0%E4%BB%A3%E7%A0%81&t=blog)
* 1470

基于item协同过滤算法：利用的是item组成相似性矩阵，这里我们是基于电影协同过滤算法实现

现在要给所有的用户推荐电影

首先我们要计算电影之间的相似度，相似度用皮尔逊相关来求电影之间的相似性。

求出相识度如下表所示

比如我们要给A推荐电影，那么根据A看过的电影与其他电影相似度，进行加权评分，得出要给A推荐的电影。

如表是我们要给所有用户推荐的电影，其中0代表该用户看过电影，不需要推荐。

从表中我们可以看出给A推荐最高的电影是4，即寻龙诀电影推荐给A用户。

代码如下所示：

电影相似度代码：

**[plain]** [view plain](https://blog.csdn.net/jcm666666/article/details/50998193) [copy](https://blog.csdn.net/jcm666666/article/details/50998193)

1. load('score.mat');
2. [rowsize,colsize] = size(A);
3. sim = zeros(colsize);
4. %% Pearson相关系数求电影相似性
5. for i = 1:colsize-1
6. for j = i+1:colsize
7. sum1=0;
8. sum2=0;
9. sum3=0;
10. sum4=0;
11. sum5=0;
12. temp = A(:,i)&A(:,j);
13. index = find(temp);
14. for k = 1:size(index,1)
15. sum1 = sum1 + A(index(k),i)\*A(index(k),j);
16. sum2 = sum2 + A(index(k),i);
17. sum3 = sum3 + A(index(k),j);
18. sum4 = sum4 + A(index(k),i)^2;
19. sum5 = sum5 + A(index(k),j)^2;
20. end
21. sum23 = (sum2 \* sum3)/k;
22. sum42 = sum4 - sum2^2/k;
23. sum53 = sum5 - sum3^2/k;
24. if (sum42~=0&&sum53~=0)
25. sim(i,j)=(sum1 - sum23)/sqrt(sum42 \* sum53);
26. end
27. end
28. end

给用户推荐电影评分代码：

**[plain]** [view plain](https://blog.csdn.net/jcm666666/article/details/50998193) [copy](https://blog.csdn.net/jcm666666/article/details/50998193)

1. %% 给用户推荐电影评分
2. predict\_score = zeros(rowsize,colsize);
3. for i = 1:rowsize
4. %% 找到用户已评价的电影的索引
5. find\_temp = find(A(i,:));
6. %% 找到用户没有评价的电影索引
7. ufind\_temp = find(A(i,:)==0);
8. %% 预测用户没有评价电影的评分值
9. for j = 1:size(ufind\_temp,2)
10. %% 利用某用户对所有评价电影乘以未评价电影相似度
11. for k=1:size(find\_temp,2)
12. predict\_score(i,ufind\_temp(j)) = predict\_score(i,ufind\_temp(j)) + A(i,find\_temp(k))\*sim(find\_temp(k),ufind\_temp(j));
13. end
14. end
15. end

欢迎大家一起探讨学习