基于项目聚类的全局最近邻的协同过滤算法

韦素云 业 宁 朱 健 黄 霞 张 硕
(南京林业大学信息科学技术学院 南京 210037)

摘 要 用户评分数据极端稀疏的情况下,传统相似性度量方法存在弊端,导致推荐系统的推荐质量急剧下降。针对此问题,提出了一种基于项目聚类的全局最近邻的协同过滤算法。该算法根据项目之间的相似性进行聚类,使得相似性较高的项目聚成一类,在项目聚类集的基础上,计算用户的局部相似度,使用一种新的最近邻用户全局相似度作为衡量用户间相似性的标准;其次,给出了一种利用重叠度因子来调节局部相似度的方法,以更准确地刻画用户之间的相似性。实验结果表明,该算法可以提升预测结果的准确性,提高推荐质量,特别是在数据较为稀疏时,改善尤为明显。

关键词 推荐系统,协同过滤,聚类,全局相似性,重叠度因子
中图法分类号 TP391.4 **文献标识码** A

Collaborative Filtering Recommendation Algorithm Based on Item Clustering and Global Similarity

WEI Su-yun YE Ning ZHU Jian HUANG Xia ZHANG Shuo
(College of Information Science and Technology, Nanjing Forestry University, Nanjing 210037, China)

Abstract When facing with the extreme sparsity of user rating data, traditional similarity measure method performs poor work which results in poor recommendation quality. To address the matter, a new collaborative filtering recommendation algorithm based on item clustering and global nearest neighbor set was proposed. Clustering algorithm is applied to cluster items into several classes based on the similarity of the items, and then the local user similarity is calculated in each cluster, at last a newly global similarity between nearest neighbor users is used to measure user similarity. In addition, the factor of overlap is introduced to optimize the accuracy of the local similarity between users. The experimental results show that this algorithm can improve the accuracy of the prediction and enhance the recommendation quality, which shows good result on the condition of the extreme sparse data.

Keywords Recommendation systems, Collaborative filtering, Clustering, Globe similarity, Overlap

1 引言

协同过滤[1]是在电子商务系统中使用非常广泛的一类推荐算法,其基本思想是基于与目标用户具有相同(或相似)兴趣偏好的其它客户的观点向其提供商品推荐或评分预测。

推荐系统中普遍存在数据稀疏性[2]、冷启动[3]和可扩展性[4]等问题。随着协同过滤推荐系统应用的不断深入,许多研究者提出了一些新的方法来改进推荐系统的不足。基于项目的协同过滤算法[5]根据用户对最近邻居项目的评分来预测其对目标项目的评分,由于项目之间的相似性相对比较稳定[6],因此可以通过离线计算来提高推荐速度。Zeng 等人[7]提出用户-类别评分矩阵,每个矩阵元素值为该类别所有项目评分之和,由于项目类别数远小于项目数,因此可以降低矩阵维数、增加矩阵数据密度和改善冷启动问题。Billsus 等人[8]使用奇异值分解(SVD)将 $m\times n$ 阶评分矩阵进行分解,得到与其最近似的、秩为 $k(k\ll \min(m,n))$ 的重构矩阵,然后基于该低阶近似矩阵进行协同过滤推荐。Goldberg 等人[9]将主成份分析用于笑话推荐系统 Jester 中,提出了基于主成份分析的协同过滤算法。Aggarwal 等人[10]提出了基于图论的 Horting 技术,图中节点代表用户,边代表两个用户之间的相似度,通过搜索近邻节点并综合近邻节点的评分来生成推荐。Papagelis 等人[11]根据用户评分及信任推导来建立社会网络,从而在无共同评分项的用户之间产生用户相似性的传递关联。聚类[12]技术通过减小最近邻搜寻空间来提高协同过滤可扩展性。

在协同过滤算法中,相似性度量是否准确关系到推荐质量。由于数据的极端稀疏性,传统的度量方法存在一定的弊端,系统的推荐精度往往会很低。因此,为了提高系统的推荐质量,如何有效进行相似性计算成为需要解决的关键问题。本文提出了一种基于项目聚类的用户最近邻全局相似性协同

到稿日期:2012-02-08 返修日期:2012-07-20 本文受国家 973 项目(2012CB114505),国家杰出青年基金项目(31125008),江苏省自然科学基金项目(BK2009393),江苏省青蓝工程学术带头人项目(CXLX11_0525),南京林业大学科技创新项目(163070079),江苏高校大学生创新计划项目(164070742)资助。

韦素云(1981—),女,硕士,讲师,主要研究方向为数据挖掘、个性化推荐技术,E-mail:weisuyun@163.com;业 宁(1967—),博士,教授,主要研究方向为数据挖掘、机器学习;朱 健 本科生;黄 霞 本科生;张 硕 本科生。

过滤算法，给出一种度量用户相似性的新方法，即根据项目间的相似性对项目进行聚类，将相似性较高的项目聚成一类，计算用户局部相似性，构造用户相似性的全局相似性度量方法。采用重叠度因子，根据用户共同评分的项目数量动态调整相似性，保证只有共同评分项目较多且评分相似的用户才有可能成为邻居用户，以弥补传统用户相似度计算中存在的不足。在 Movielens 标准数据集上的实验结果表明，本文算法在提高预测准确度的同时，在用户评分数据极端稀疏的情况下，能够利用局部相似性获得小规模邻居集，提高了推荐质量。

2 基于项目聚类的局部最近邻用户相似性

2.1 项目相似性

项目的相似性表示两个不同项目之间的相似程度，如果两个不同的用户对两个项目的共同评分趋向一致，说明他们对这两个项目感兴趣程度也趋向一致。可以使用 Pearson 相关系数来计算项目间的相似性，设项目 i 与项目 j 共同评分的用户集合用 U_{ij} 表示，则项目 i 和 j 之间的相似性 $\mathrm{sim}(i,j)$ 如式(1)所示。

$$\mathrm{sim}(i,j)=\frac{\sum_{u\in U_{ij}}(r_{ui}-\bar{r}_i)(r_{uj}-\bar{r}_j)}{\sqrt{\sum_{u\in U_{ij}}(r_{ui}-\bar{r}_i)^2}\sqrt{\sum_{u\in U_{ij}}(r_{uj}-\bar{r}_j)^2}} \tag{1}$$

式中，$\bar{r}_i$ 和 $\bar{r}_j$ 分别表示项目 i 和项目 j 在用户集 U_{ij} 上的平均评分，即

$$\bar{r}_i=\frac{1}{|U_{ij}|}\sum_{u\in U_{ij}}r_{ui},\ \bar{r}_j=\frac{1}{|U_{ij}|}\sum_{u\in U_{ij}}r_{uj} \tag{2}$$

2.2 基于项目相似性的聚类算法

利用聚类算法对 n 个项目进行聚类，使得具有较高相似性的项目聚成一类，而不同类中的项目差别较大。设项目集 $I=\{i_1,i_2,\cdots,i_n\}$，计算项目间的相似性，使用 K-MEANS 聚类算法，以相似性作为项目间的类别距离，最终生成聚类集合 $C=\{c_1,c_2,\cdots,c_k\}$，其中同一类 $c_i(i=1,2,\cdots,k)$ 包含的项目特征尽可能相似。

算法 1 基于项目相似性的聚类算法

输入：项目集 $I=\{i_1,i_2,\cdots,i_n\}$，用户评分矩阵 R_{mn}

输出：聚类项目集 $C=\{c_1,c_2,\cdots,c_k\}$

1. 任意选择 k 个项目，将其用户评分值向量作为初始的聚类中心，记为 $CC=\{w_1,w_2,\cdots,w_k\}$；
2. 聚类集合 $C=\{c_1,c_2,\cdots,c_k\}$ 初始化为空；
3. Repeat

   For 项目 $i_l(i_l\in I)$ do

   For 聚类中心 $w_j(w_j\in CC)$ do 计算 $\mathrm{sim}(i_l,w_j)$；

   $\mathrm{sim}(i_l,w_{j^*})\geqslant\mathrm{sim}(i_l,w_j),j\in\{1,\cdots,k\}$

   $c_{j^*}=c_{j^*}\cup i_l$

   For 聚类 $c_j(c_j\in C)$ do

   更新聚类中心 $w_j=\sum_{i_l\in c_j}i_l/|c_j|$

   计算误差函数 $E=\sum_{j=1}^{k}\sum_{i_l\in c_j}|i_l-w_j|^2$

Until E 不再改变

2.3 局部最近邻用户相似性

在 k 个项目聚类的基础上，本文提出局部相似度的概念，引入重叠度因子，并将其融合到计算用户局部相似度的公式中。

局部相似性表示用户 u 和用户 v 在项目聚类集 c_j 上的评分相似性，设用户 u 和 v 在项目聚类集 c_j 上共同评分的项目集合为 $I_{uv}^j=I_u\cap I_v\cap c_j$，则用户 u 和 v 在项目聚类集 c_j 的局部相似性 $\mathrm{sim}_j(u,v)$ 如式(3)所示。

$$\mathrm{sim}_j(u,v)=\frac{\sum_{i\in I_{uv}^j}(r_{ui}-\bar{r}_u^j)(r_{vi}-\bar{r}_v^j)}{\sqrt{\sum_{i\in I_{uv}^k}(r_{ui}-\bar{r}_u^j)^2}\sqrt{\sum_{i\in I_{uv}^k}(r_{ui}-\bar{r}_v^j)^2}} \tag{3}$$

式中，$\bar{r}_u^j$、$\bar{r}_v^j$ 分别表示用户 u 和 v 对聚类集 c_j 中所有项目的平均评分。

为了避免传统方法中共同评分项目数稀少但评分非常相似、用户相似度较高的不合理现象，引入项目重叠度因子对局部相似度计算进行修正，如式(4)所示。

$$\mathrm{sim}_j(u,v)=\frac{\min(|I_u\cap I_v\cap c_j|,\gamma)}{\gamma}\mathrm{sim}_j(u,v) \tag{4}$$

式中，$|I_u\cap I_v\cap c_j|$ 表示用户 u 和 v 在聚类集 c_j 上共同评分的项目数量；设置参数 γ，当用户共同评分的项目数小于 γ，即数据相对稀疏时，共同评价的项目数越多，因子值越大，从而保证只有共同评分项目较多且评分相似的用户才有可能成为邻居用户。

3 基于全局最近邻的协同过滤

3.1 用户模型

协同过滤算法的输入数据通常表示为 $m\times n$ 的评分矩阵 R_{mn}，如式(5)所示。其中，行表示 m 个用户，列表示 n 个项目，矩阵中的第 u 行第 i 列所对应元素 r_{ui} 表示用户 u 对项目 i 的评分，它一般通过用户现实提交兴趣评价级别而获得。

$$R_{mn}=\begin{bmatrix}r_{11}&r_{12}&\cdots&r_{1n}\\r_{21}&r_{22}&\cdots&r_{2n}\\\vdots&\vdots&\ddots&\vdots\\r_{m1}&r_{m2}&\cdots&r_{mn}\end{bmatrix} \tag{5}$$

3.2 全局最近邻用户相似性

根据第 2 节局部最近邻用户相似性计算方法，在 k 个项目聚类集上，计算得到 k 个调整后的用户 u 和 v 之间的局部相似性 $\mathrm{sim}_j(u,v),j=1,2,\cdots,k$。由此，定义用户 u 和 v 之间的全局相似性度量公式作为用户 u 和 v 之间相似性的度量标准，如(6)所示。

$$\mathrm{sim}(u,v)=\sum_{j=1}^{k}\mathrm{sim}_j(u,v) \tag{6}$$

3.3 产生推荐

利用通过全局最近邻用户相似性度量得到的目标用户的最近邻，可以计算两类推荐结果：目标用户对任意项目 i 的评分、top-N 推荐集。

(1)目标用户对任意项目 i 的评分：设目标用户 u 的最近邻集合为 $N_u=\{v_{i1},\cdots,v_{iK}\}$，$u\notin N_u$，则目标用户 u 对项目 i 的预测评分 P_{ui} 如式(7)所示。

$$P_{ui}=\bar{r}_u+\frac{\sum_{v\in Nu}\mathrm{sim}(u,v)(r_{ui}-\bar{r}_v)}{\sum_{v\in Nu}\mathrm{sim}(u,v)} \tag{7}$$

式中，$\bar{r}_u$ 和 $\bar{r}_v$ 分别表示用户 u 和 v 对项目的平均评分值。

(2)top-N 推荐集：分别计算用户 u 对不同项目的评分后，取评分值最高并且不在用户已评分的项目集合中的 N 个项目作为 top-N 推荐集。

3.4 基于全局最近邻的协同过滤算法

为了有效解决用户评分极端稀疏情况下传统相似性度量

方法存在的问题，本文设计了一个基于全局最近邻相似性的协同过滤推荐算法。其通过计算不同用户之间的全局相似性，寻找相似性最高的 K 个邻居，利用最近邻居集合计算用户对未评分项目的预测评分，最后产生推荐集。

算法 2 基于全局最近邻协同过滤算法(GBCF)

输入：用户评分矩阵 R_{mn}，推荐集元素个数 N

输出：目标用户 u 的推荐集 top-N

1. 根据算法 1 生成聚类项目集 $C=\{c_1,c_2,\cdots,c_k\}$。
2. for 任意用户 u 和 v do
   利用式(4)计算 k 个局部相似性 $sim_j(u,v)(j=1,\cdots,k)$。
   根据式(6)获得全局相似性 $sim(u,v)$。
   更新相似度矩阵 R_{sim}，即 $R_{sim}=R_{sim}\cup sim(u,v)$。
3. for 每个用户 u do
   寻找最近邻居集合 $N_u=\{v_{i1},\cdots,v_{iK}\}$，$u\notin N_u$，且满足 $sim(u,v_{i1})\geqslant\cdots\geqslant sim(u,v_{iK})$
4. for 每个用户 u do
   利用式(7)计算对未评分项目 i 的预测评分 P_{ui}；
   对预测评分进行升序排序；
   取前 N 个值所对应的项目组成推荐集 top-N。

4 实验结果及分析

4.1 数据集

本文实验采用 MovieLens 站点(http://movielens.umn.edu)提供的数据集。这个数据集由美国 Minnesota 大学的 GroupLens 工作组创建并维护。本文选取的是 100k 的公开数据集，其具有 10 万条记录，包括 943 个用户对 1682 部电影的评分。其中，每个用户至少对 20 部电影进行了评分，评分值范围为 1～5，5 表示最喜欢，1 表示最不喜欢，用户通过评分的数值表达了自己的兴趣爱好。实际评分数据的密度为 $100000/(943*1682)=6.3\%$，说明此数据是相当稀疏的。从数据集中随机抽取 500 个用户，将实验数据的评分矩阵进一步划分为训练集和测试集，引入变量 x 表示训练集占整个数据集的百分比。例如，$x=0.8$ 表示随机地将数据集中的 80% 作为训练集，剩下的 20% 作为测试集。在本文的实验中，均采用 $x=0.8$。

4.2 推荐质量的度量标准

评价推荐系统推荐质量的度量标准主要包括统计精度度量方法和决策支持精度度量方法两类。统计精度度量方法中的平均绝对误差(MAE)是一种常用的衡量推荐结果的度量方法。该标准是通过比较预测值与用户实际的评分值之间的偏差来衡量预测结果的准确性。MAE 越小，表明推荐质量越高。绝大多数实验都采用 MAE 作为衡量推荐结果的参考标准，因此本文也采用平均绝对偏差 MAE 作为度量标准。设预测的用户评分集合表示为 $\{p_1,p_2,\cdots,p_n\}$，对应的实际用户评分集合为 $\{q_1,q_2,\cdots,q_n\}$，则平均绝对偏差 MAE 的定义如式(8)所示。

$$MAE=\frac{\sum_{i=1}^{n}|p_i-q_i|}{n} \tag{8}$$

4.3 实验参数的调整

我们的算法中有 2 个参数需要选择：聚类个数 k、重叠度因子参数 γ。

使用 K-MEANS 聚类算法对项目进行聚类时，聚类数目的选取对算法很重要。为了选取合适的聚类数，我们选用不同的 k 值(10,20,30,40,50)进行聚类来验证预测的性能。设定最近邻居数为 20，实验结果如图 1 所示。从图 1 中可以看出，聚类数会影响预测的性能，当聚类数过小时，类信息过于普遍化，无法表示不相似项目之间的特征；而当聚类信息过大时，类信息过于个性化，无法表示相似项目间的相似性。当聚类数为 30 时，预测性能最优。

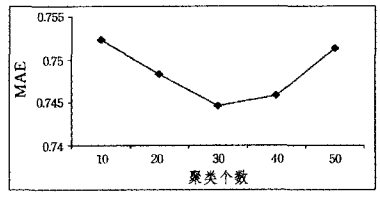

图 1 不同聚类数对 MAE 的影响

重叠度因子参数 γ 是为了调整用户相似性预先设定的参数，可以使相似性的计算更加合理，利用它可以调整那些用户共同评分数据很少但是相似性却很高的情况。图 2 表示本文提出的基于项目聚类的用户最近邻全局相似性协同过滤算法(GBCF)与基于用户协同过滤算法(UBCF)中参数 γ 对预测的性能的影响，设定最近邻居数为 20。实验结果表明，随着 γ 的增加，MAE 逐渐减小，说明预测评分越来越接近实际评分。

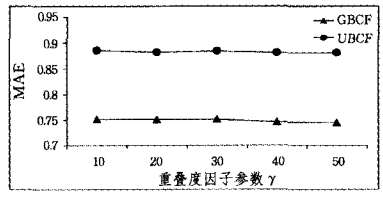

图 2 重叠度因子参数 γ 对 MAE 的影响

4.4 实验结果比较

最近邻居的个数会在很大程度上影响算法的性能，在实验中，将最近邻居的个数从 3 递增到 40。计算本文提出基于项目聚类的用户最近邻全局相似性度量方法(ICGS)与基于用户协同过滤算法(UBCF)、基于项目协同过滤算法(IBCF)的 MAE 的值，实验参数按上节讨论的最优值来设置，实验结果如图 3 所示。

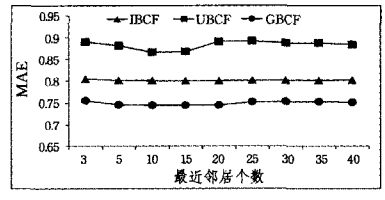

图 3 不同邻居数对 MAE 的影响

当最近邻居数为 3 时，本文的算法明显优于基于用户协同过滤算法，这是因为在数据稀疏，只能获取较少邻居用户情况下，基于用户协同过滤算法不能够准确地描述出用户与目标用户之间的相似性，无法区别在已有的邻居集中真正对预测结果有价值的用户，以至于不能够在预测时分配适当的权重。在本文的算法中，通过引入重叠度因子计算局部相似度，可以弥补基于用户协同过滤算法的不足，提高预测结果的准确性。随着邻居集用户增加，本文的算法一直优于基于用户协同过滤算法和基于项目协同过滤算法，表明本文的算法确实对推荐结果的准确性有明显的提升作用。

结束语 在协同过滤算法中，相似性度量直接影响到算法的预测精度与推荐质量。在本文提出的基于项目聚类的用户最近邻全局相似性协同过滤算法中，计算项目间的相似性，通过聚类算法将相似性较高的项目聚成一类，在每个聚类项目集合上计算用户之间的局部相似性，形成最近邻用户全局相似度计算方法，以提高相似度和预测评分的准确性。同时还根据用户共同评分的项目数量，引入重叠度因子，并将其融合到计算用户局部相似度的公式中，来进一步加强相似度的准确性。从实验结果可以看出，本文的算法具有比传统推荐算法更好的推荐质量。

参 考 文 献

[1] Goldberg D, Nichols D, Oki B, et al. Using collaborative filtering to weave an information tapestry [J]. Communications of the ACM, 1992, 35(12): 61-70

[2] Sarwar B M, Karypis G, Konstan J, et al. Application of dimensionality reduction in recommender system — A Case Study [C]// Proceedings of the ACM Web KDD Web Mining for E-Commerce Work shop. Boston, MA, United States, 2000: 82-90

[3] Massa P, Avesani P. Trust-Aware Collaborative Filtering for Recommender Systems[J]. Lecture Notes in Computer Science, 2004, 3290: 492-508

[4] Vincent S Z, Boi F. Using Hierarchical Clustering for Learning the Ontologies used in Recommendation Systems[C]// Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San-Jose, California, USA, 2007: 599-608

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

[6] Sarwar B M. Sparsity, scalability, and distribution in recommender systems [D]. Minneapolis, MN: University of Minnesota, 2001

[7] Zeng C, Xing C, Zhou L. Similarity measure and instance selection for collaborative filtering [C]// Proceedings of the 12th International Conference on World Wide Web. New York: ACM Press, 2003: 652-658

[8] Billsus D, Pazzani M J. Learning collaborative information filters [C]// Proceedings of the 15th International Conference on Machine Learning. San Francisco: Morgan Kaufmann Publishers, 1998: 46-54

[9] Goldberg K, Roeder T, Gupta D, et al. Eigentaste: a constant time collaborative filtering algorithm [J]. Information Retrieval, 2001, 4(2): 133-151

[10] Aggarwal C C, Wolf J L, Wu K L, et al. Horting hatches an egg: a new graph-theoretic approach to collaborative filtering[C]// Proceedings of the 5th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, ACM Press, 1999: 201-212

[11] Papagelis M, Plexousakis D, Kutsuras T. Alleviating the sparsity problem of collaborative filtering using trust inferences [C]// Proceedings of the 3rd International Conference on iTrust 2005. Berlin: Springer-Verlag, 2005: 224-239

[12] Ungar L, Foster D. Clustering methods for collaborative filtering [C]// Proceedings of the Workshop on Recommendation Systems at the 15th National Conference on Artificial Intelligence. Menlo Park, CA: AAAI Press, 1998: 112-125

(上接第 124 页)

于仿真数学模型的参数修正作用较大；有效历史测试数据库可供开发人员建立针对性的经验统计模型。这种混合测试应该与仿真软件开发过程互相协调。

参 考 文 献

[1] 陈翔，顾庆，王新平，等. 组合测试研究进展[J]. 计算机科学，2010，37(3)：1-5

[2] 查日军，张德平，聂长海，等. 组合测试数据生成的交叉熵与粒子群算法及比较[J]. 计算机学报，2010，10：1896-1908

[3] 赵亮，王建民，孙家广. 统计测试的软件可靠性保障能力研究[J]. 软件学报，2008，19(6)：1379-1385

[4] 陆文，徐锋，吕建. 一种开放环境下的软件可靠性评估方法[J]. 计算机学报，2010，3：452-462

[5] 张德平，聂长海，徐宝文. 软件可靠性评估的重要抽样方法[J]. 软件学报，2009(09)：2859-2867

[6] 吴重光，等. 我国石油化工仿真技术 20 年成就与发展[J]. 系统仿真学报，2009(21)：6689-6696

[7] 赵一丁，等. FCCU 反应再生仿真模型的研究[J]. 计算机仿真，2007(06)：74-77

[8] 赵一丁，等. 催化反应再生生产在线专家系统研究[J]. 计算机仿真，2007(07)：138-141

[9] Chari K, Hevner A. System test Planning of software: An optimization approach [J]. IEEE Transon Software Engineering, 2006, 32(7): 503-509

[10] 顾庆，唐宝，陈道蓄. 一种面向测试需求部分覆盖的测试用例集约简技术[J]. 计算机学报，2011(5)：879-888

[11] 姜瑛，辛国茂，单锦辉，等. 一种 Web 服务的测试数据自动生成方法[J]. 计算机学报，2005(4)：568-577

[12] 刘新忠，徐高潮，胡亮，等. 一种基于约束的变异测试数据生成方法[J]. 计算机研究与发展，2011，4：617-626

[13] 刘哲，张为群，肖魏娜. 一种基于模糊评估分层模型的构件可测试性评价方法[J]. 计算机科学，2011，38(5)：113-115

[14] Kuball S, May J. Test—Adequacy and statistical testing: Combing different ProPerties of atest-set[C]// Proc. of thelsthlnt, ISymP. on Software Reliability Engineering (ISSRE 2004). 2004: 161-172

[15] 路晓丽，董云卫. Web 应用软件的结构测试研究[J]. 计算机科学，2010，37(12)：110-113

[16] 山红红，等. 两段提升管催化裂化技术动力学特点分析[J]. 中国石油大学学报：自然科学版，2007(01)：52-56

[17] Chernak Y. Validating and improving test-case effectiveness[J]. IEEE Software, 2001, 18(1): 81-86

[18] Miller S D, DeCarlo R A, Mathur A P. Modeling and control of the incremental software test process [C]// Proc. of the 28th Annual Int'l Computer Software and Applications Conf(COMPSAC 2004). 2004: 156-159