第31卷 第21期 计 算 机 工 程 2005年11月
Vol.31 № 21 Computer Engineering November 2005
·博士论文· 文章编号：1000—3428(2005)21—0007—03 文献标识码：A 中图分类号：TP39

基于领域知识的个性化推荐算法研究

张丙奇
（中国科学院计算技术研究所，北京 100080）

摘 要：提出了利用领域知识进行相似度计算的协同过滤算法，使用户在评分的共同项目很少或为零的情况下也能找到最近邻进行协同推荐。实验结果表明，该算法解决了传统协同过滤算法中相似性度量方法“过严”的问题，在过滤初期显著地提高了推荐质量。
关键词：个性化；推荐系统；协同过滤；领域知识；平均绝对误差

A Collaborative Filtering Recommendation Algorithm Based on Domain Knowledge

ZHANG Bingqi
(Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080)

【Abstract】 A novel method which exploring hierarchical domain knowledge to computer the user similarity is proposed. This method can find user's neighborhood and get predictions by neighbor's rating even they have no common rating items. The experiment results show that it can provide better prediction results than traditional collaborative algorithms at such condition.
【Key words】 Personalization; Recommendation system; Collaborative filtering; Domain knowledge; MAE

随着WWW的快速发展和网上信息呈指数级的增长，如何快速、高效地获取和利用信息已成为人们关注的焦点。个性化服务是服务提供商根据用户访问网站的历史中表现出的行为和兴趣，动态调节站点的结构和内容，自动适应用户的浏览和向用户推荐可能感兴趣的内容的过程。个性化服务可使用户快速、准确地得到所需要的信息。

协同过滤，又叫社会信息过滤，是个性化系统中应用最广泛和最成功的推荐技术，其本质上是将“word－of－mouth”口头推荐的过程自动化，关键思想是认为对一些项目具有相同兴趣的用户对另一项目也具有相同的兴趣[1]。在协同过滤中，用户最近邻的查找是否正确，直接影响系统的推荐质量。

协同过滤适用于不同种类的对象（Web页面、文本、视频、书籍、CD等）的推荐。Tapestry是最早利用协同过滤技术的推荐系统，在此系统中，当前用户需要明确声明自己的观点来形成用户群。GroupLens是基于用户评分的自动化协同过滤系统，它主要用于电影和新闻的推荐。其它的典型系统包括Ringo，Fab，Amazon.com，eBay等。

然而，协同过滤系统刚建立时，由于参与的用户和用户评分项目很少导致用户评分项的交集很小，传统协同过滤中的相似性计算方式在这种情况下很难找到最近邻进行推荐，这被称为系统的“冷启动”现象。针对这种情况，本文提出了利用领域知识进行相似度计算的协同推荐算法：通过领域知识，计算用户评分项的相似性。数据实验结果说明本方法在用户评分项目较少的情况下，推荐效果优于传统的协同过滤方法，显著提高了推荐质量。

1 传统的协同过滤算法

协同过滤也称为面向用户行为的技术，它通过分析用户的历史数据，生成与当前用户行为、兴趣最相近的用户集，然后利用他们对项目的评分来预测当前用户对项目的评分产生推荐列表，即 *top*－*N* 推荐。

常用的协同过滤推荐算法主要包括基于用户的协同推荐、基于项目集的协同推荐、基于模型的协同推荐等。Breese[2]等对各种协同推荐算法及其性能进行了深入的分析和对比。

在协同推荐系统中，输入的数据是用户评分矩阵（见表1），可用一个 $M \times N$ 矩阵 R（M，N）来表示，其中 M 行代表 M 个用户，N 列代表 N 个项目，元素 $R_{i,j}$ 表示用户 i 对项目 j 的评分。根据项的类型和系统的实现方式，有不同的用户评分方法。比如用1表示用户对某项目感兴趣，0表示不感兴趣；也可用1，2，3，4，5等离散值表示对该商品的偏爱度。用离散值表示用户的偏爱度时，一般采用奇数个等级。

表1 用户评分矩阵

	$Item_1$	…	$Item_j$	…	$Item_N$
$User_1$	$R_{1,1}$	…	$R_{1,j}$	…	$R_{1,N}$
….	…	…	…	…	…
$User_i$	$R_{i,1}$	…	$R_{i,j}$	…	$R_{i,N}$
….	…	…	…	…	…
$User_M$	$R_{M,1}$	…	$R_{M,j}$	..	$R_{M,N}$

基于协同过滤技术的推荐过程分为两个阶段：最近邻的发现，推荐集合的产生。

(1) 最近邻的发现

协同过滤算法的核心是根据用户评分矩阵发现当前用户的最近邻，即：对当前用户 u，要产生一个依照用户相似度大小进行排列的邻居的集合 $N=\{N_1, N_2, \cdots, N_k\}$，$u$ 不属于 N，$Sim(u,N_i) > Sim(u,N_{i+1})$，i＝1，…，k－1。

传统的相似度计算方式有多种，主要包括如下几种：余弦（*Cosine*）相似度计算方法和 *Pearson* 相似度计算方法。

基金项目：国家“863”计划基金资助项目（2002AA142110）
作者简介：张丙奇 (1972—)，男，博士生，主研方向：网络信息处理与个性化服务
收稿日期：2004-08-26 **E-mail：**zhangbq@software.ict.ac.cn

其定义分别为:

余弦相似性：

$$Sim(i,j)=\cos(\vec{i},\vec{j})=\frac{\vec{i}*\vec{j}}{|\vec{i}\|\vec{j}|}$$

其中，$\vec{i},\vec{j}$ 是用户 i，j 的评分向量。

Pearson 相似系数：

$$Sim(i,j)=corr(\vec{i},\vec{j})=\frac{\sum_{k\in I_{i,j}}(R_{i,k}-\overline{R_i})(R_{j,k}-\overline{R_j})}{\sqrt{\sum_{k\in I_{i,j}}(R_{i,k}-\overline{R_i})^2}\sqrt{\sum_{k\in I_{i,j}}(R_{j,k}-\overline{R_j})^2}}$$

其中，$I_{i,j}$ 表示用户 i，j 共同评分的项目集合，$\overline{R_i}$，$\overline{R_j}$ 表示用户 i，j 对项目的平均评分。

(2) 推荐集的产生

基于用户的最近邻，可以计算两类推荐结果：用户对任意项的偏爱度和 *top*－*N* 推荐集。

1)用户对任意项的偏爱度：已知用户 i 和用户已评分项集 I_i，则用户 i 对任意项 k，（$k\notin I_i$）的偏爱度(评分预测)为

$$P_{i,k}=\overline{R_i}+\frac{\sum_{m=1}^{n}sim(i,m)*(R_{m,k}-\overline{R_m})}{\sum_{m=1}^{n}sim(i,m)}$$

其中，$\overline{R_i}$ 是用户 i 的平均评分，$sim(i,m)$ 是用户 I 与最近邻集合中的用户 m 的相似系数，$R_{m,k}$ 是用户 m 对项 k 的评分，$\overline{R_m}$ 是用户 m 的平均评分，n 是最近邻的个数。

2) *top*－*N* 推荐集：分别计算用户 i 对不同项目的偏爱度之后，取偏爱度最高的、并且不在 I_i 中的 N 个项作为 *top*－*N* 推荐集。

2 传统协同过滤算法中相似度计算分析

在协同过滤系统中，最近邻的查找是否精确，直接影响推荐效果。而最近邻的查找需要通过用户相似度的计算得到。传统协同过滤中的相似度计算方法，如上面的 *Cosine* 相似度和 *Pearson* 系数都是基于用户评分项的交集之上，用户评分项被表示成欧式空间内的 n 维向量，n 是项数，并且假设项之间互相独立。

考虑电影网站的电影分类体系和用户对电影的评分情况，如图 1 所示。

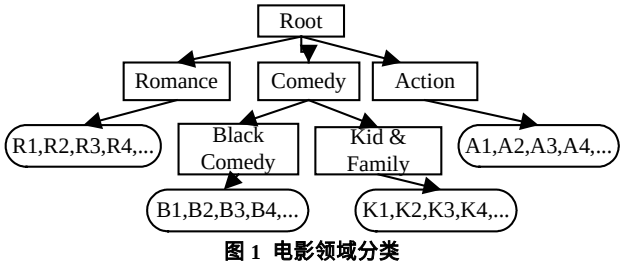

图 1 电影领域分类

用户 *A* 评分的项目为{*R1*,*R2*}，用户 *B* 评分的项目为{*R3*,*R4*}，用户 *C* 的评分项目为{*A1*,*A2*}，用户 *D* 的评分项为{*R1*,*K1*},用户 *E* 的评分项为{*R2*,*R3*,*R4*,*K1*,*K2*,*K3*}。

采用 *Cosine* 相似度或 *Pearson* 系数的相似度计算结果，会发现用户 *A* 与用户 *B* 之间、用户 *B* 与用户 *C* 之间的相似度均为 0。

但是，参考图 1 中的电影领域知识，从直观上发现，用户 *A* 和用户 *B* 应该具有部分的相似度，因为他们都喜欢 Romance 类的电影。而用户 *A* 和用户 *C* 的相似度应小于用户 *A* 和用户 *B* 的相似度，因为他们既没有评价相同的项目，也没有评价同类的项目。

当两个用户评分的共同项较少或者根本没有共同的评分项目时，如果采用传统的相似度计算方法，用户间的相似度会很小或为零。这种情况在系统刚建立时最为突出，因为在系统刚建立时，只有很少的用户评价了很少的项目，数据的极端稀疏使得用户共同评分的项目很少，这时用户就无法找到最近邻进行协同推荐。

3 基于领域知识的协同过滤算法

为了改进传统的基于“交集”的向量方式相似度计算的缺陷，本文提出利用领域知识的协同推荐算法：首先，利用 Web 站点的领域知识计算用户评价项目之间的相似度，这种相似度更适合人们的直观上的感觉；通过项集合的相似性计算用户之间的相似性，得到最近邻进行推荐。

(1) 基于领域知识的项目之间的语义相似度计算方式

基于语义的相似度计算方法是依赖领域的，需要领域对象内在关系和结构等知识的支持。而一般 Web 网站，特别是电子商务网站，都有相关领域的分类体系的描述，如图 1，是一个电影网站中关于电影的分类体系描述。

利用领域知识提供的语义进行相似度计算已经得到许多研究者的关注。文献[3，4]讨论了利用语义知识计算对象之间相似度问题。项目之间的语义相似度计算主要参考这两篇文章介绍的方法。

1) 相关符号定义

U：将领域分类知识的表示为图 1 所示的根为 *Root* 的树 U，U 中节点的标签互异，U 的叶子节点可在任何层次上出现。

L_U：是 *U* 中所有节点的标签的集合。

LL_U：是树 *U* 的所有叶子节点的标签，即是用户评选项目的集合。在图 1 中 LL_U={*R1* ,*R2*,⋯,*B1*, *B2* , ⋯ ,*K1*,*K2*, ⋯, *A1* ,*A1*,⋯}。每个用户的评分项目集合是 LL_U 的一个子集。

$depth(n)$：表示从根节点开始到节点 *n* 的路径的长度。

LCA（m,n）：表示节点 *m*，*n* 最低公共祖先，即 *m*，*n* 的具有最大路径的公共祖先。任何两个节点至少有一个公共祖先－根节点，*m*，*n* 的不同公共祖先的路径不同。

2) 叶子节点（项）l_i,l_j 的相似度定义为

$$Sim(l_i,l_j)=\frac{2*depth(LCA(l_i,l_j))}{depth(l_i)+depth(l_j)}$$

从上面的定义可以看出，两个节点的相似度属于[0,1], 当且仅当 $l_i=l_j$ 时，相似度为 1。

(2) 评分项目之间的相似性－用户对项的评分的影响

$$Sim(R_{a,i},R_{n,j})=Sim(l_i,l_j)*(1-|R_{a,i}-R_{n,j}|/MaxRating)$$

其中，$R_{a,i},R_{n,j}$ 分别是用户 *a* 对项 *i*，用户 *n* 对项 *j* 的评分。

(3) 用户之间相似性计算

有了评分项目之间相似度的计算方式，就可以根据用户评分项的集合计算用户的相似度。集合之间相似度的计算需要考虑相似的对称性性质和集合元素个数对相似计算结果的影响。例如图 1 中用户 A 和用户 D 的相似性应大于用户 A 与用户 E 的相似性。另外，用户之间的相似度应该满足以下基本条件：

1) 用户和自身的相似度为 1；

2) 用户 *A*，*B*，*A* 与 *B* 的相似度应等与 *B* 与 *A* 的相似度，即满足对称性；

3) 假设用户评分项目集合都有 *n* 个元素，其中 *m*（*m*<*n*）个项目相同，又假设两个项目的相似度只能是 0（不同）或 1（相同），

那么这两个用户的相似度应该是 m/n。

通过在集合的元素之间建立对应的关系来消除集合元素数目的影响，并保证计算结果满足以上的条件。具体方法为：

1)首先计算两个用户评分项集合的所有项目两两间相似度；

2)从所有的相似度值中选择最大的一个，将这个相似度值对应的两个项目对应起来，并累加该最大值；

3)从所有的相似度值中删去那些已经建立对应关系的项目的相似度值；

4)重复上述第 2 步和第 3 步，直到所有的相似度值都被删除；

5)没有建立起对应关系的项目与空元素对应；

6)求平均相似度（累加和除以对应项目个数），返回。

上面的算法首先按照相似度建立起两个集合中项目的一一对应关系，然后计算用户评分项目集合的相似度：用户评分项目集合的相似度等于其元素对的相似度的加权平均。因为评分项目集合的元素之间都是平等的，所以所有的权值取相同，那么项目集合的相似度等于其项目对的相似度的算术平均。

(4) 产生推荐集

通过本文的方法得到当前用户的最近邻集合后，推荐方式和传统协同过滤方法一样：

$$P_{i,k}=\overline{R_i}+\frac{\sum_{m=1}^{n} sim(i,m)*(R_{m,k}-\overline{R_m})}{\sum_{m=1}^{n} sim(i,m)}$$

其中，$\overline{R_i}$ 是用户 i 的平均评分，$sim(i,m)$ 是用户 i 和最近邻集合中的用户 m 的相似系数，$R_{m,k}$ 是用户 m 对项 k 的评分，$\overline{R_m}$ 是用户 m 的平均评分。n 是最近邻的个数。

4 数据实验与结果分析

(1) 数据集说明

本文采用 MovieLens 站点提供的数据集合对本文提出的算法与传统协同过滤算法进行比较。Movielens 是明尼苏达州立大学计算机科学系 Grouplens 研究小组搜集的用于研究协同过滤算法的数据集，它包括 943 个用户对 1 682 部电影的 100 000 个评分（评分值：1~5）记录，每个用户至少评价了 20 部电影，并且包含了用户的简单人口统计学信息和电影的分类信息。利用该数据集提供的电影分类作为领域知识。

(2) 评测标准

本文采用平均绝对误差 *MAE*（*mean absolute error*）作为评测的标准。*MAE* 是常用的评价推荐算法质量的标准，它通过计算用户实际评分与预测评分之间的偏差衡量预测的准确性，*MAE* 的值越小，推荐的质量越高。平均绝对误差 *MAE* 的计算方法如下：

$$MAE=\frac{1}{m_a}\sum_{j=1}^{m_a}|p_{a,j}-r_{a,j}|$$

其中，m_a 是用户评分结合中项目的数量，$p_{a,j}$ 是预测的当前用户对项目 j 的评分，$r_{a,j}$ 是当前用户对项目 j 的实际评分。

(3) 测试方法与实验结果

首先，把数据集中分为源训练集和源测试集两部分，每个用户在源训练集和源测试集中同时出现，并且保证源测试集中每个用户的评分项目为 10 个。图 2 是源训练集中用户评分项数目分布情况，其中的横轴表示评分项目数，纵轴表示源训练集中评分项数小于等于横轴数字的用户数。从图 2 中可以看出，大多数用户的评分项目数小于等于 150。

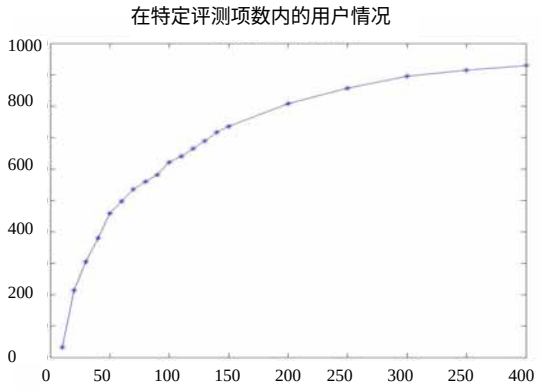

图 2 源训练集中的用户分布

为了检验本算法的有效性，以传统的协同过滤算法作为参照，其中相似度的度量采用 Pearson 系数。本文的算法主要针对系统的“冷启动”现象，所以测试的主要目的在于系统启动时用户和用户评分项目很少时推荐的性能。以源训练集中用户评分的项目数小等于某个阈值为条件，找到符合条件的用户，以这些用户在源训练集和源测试集中的评分数据作为测试集和训练集（例如阈值为 N 时，训练集为源训练集的一个子集，其中每个用户的评分项数小于等于 N，测试集是源测试集的一个子集，其中每个用户属于训练集并且评分项目数为 10），比较两种算法的推荐质量，研究本方法在解决系统“冷启动”方法中的作用。实验采用的阈值 N 取 20 到 150，间隔为 10。最近邻的个数为 50。

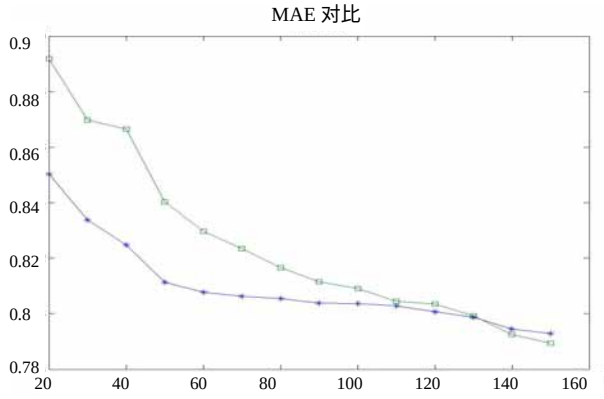

图 3 推荐算法精度－MAE 比较

如图 3 所示，横轴表示训练集中用户评分项目数目最大值，纵轴为 MAE。‘*’－本文提出的方法，‘◇’－Pearson 方法。在用户评分项目数比较少的情况下，本文提出的算法具有较小的 MAE，说明本文的算法取得了较好的推荐质量。但是随着具有较高评分项目数的用户的增加，本文提出的算法的优势逐渐降低，达到一定的程度后，推荐质量反而不如传统的 Pearson 算法。

另外，图 3 说明，随着用户的增加（训练集中用户评分项数最大值的增大），特别是具有较高评分项目数的用户的增加，两种算法的推荐质量都有明显的提高。

(4) 结果分析

本文提出的算法与传统的协同过滤算法的最大不同在于用户相似度的计算。传统的相似度采用基于“交集”的向量方式，而本文提出的相似度利用了领域知识的支持。采用本文提出的方法，当用户的评分项很少、用户的评分项之间没

（下转第 33 页）

找出不同实体的查准率是 100%，查全率是 93.8%。而当实体匹配的相似度阈值被设定为 0.65 以上时，找出不同实体的查准率和查全率均为 100%。我们也对来自于不同数据库平台 SQL Server 和 ACCESS 中的 Product 数据表中的数据进行了测试，两个 Product 表具有相同的模式结构和属性名字，但部分属性的取值有较大差别。当实体匹配的相似度的阈值被设定在 59%以上时，实体匹配的查准率和查全率均高于 90%。作为对比，我们对随机安排属性权重来进行实体匹配的方法也进行了实验，结果显示本文给出的基于属性信息熵的实体匹配方法更加有效。

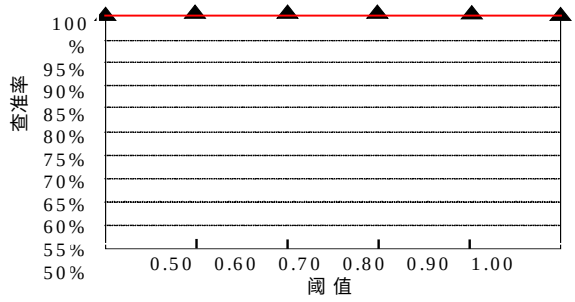

图 1 实体匹配的查准率

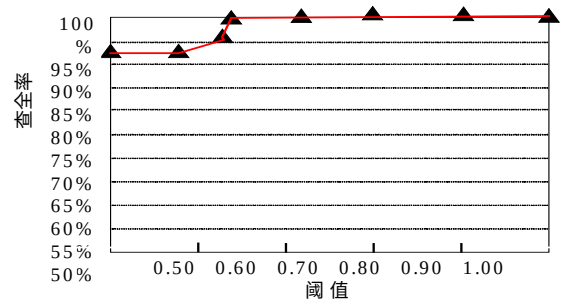

图 2 实体匹配的查全率

3 结论

本文给出了一种基于属性信息熵的实体匹配方法。该方法充分利用了属性的实例值信息，无需从用户处获得属性权重信息，比较客观且容易量化，因此也更容易实现实体匹配的自动化。具体的实验结果显示本文给出的基于属性信息熵的实体匹配方法是很有效的。

属性的信息熵描述了属性取值的不确定性。同样，互信息描述了两个属性的实例值概率分布的相关性，也就是说，两个属性的互信息描述了从一个属性中能够获得到另一个属性信息量的多少，这与数据库中属性之间的函数依赖关系是密切相关的。特别是在属性的函数依赖信息不能直接得到时，通过考察属性之间的互信息，从而获取属性之间的相关性是很有效的。在以后的工作中，我们将使用属性信息熵和互信息来评价属性之间的关系并对属性权重进行量化。有理由相信，进行实体匹配的查全率和查准率将会进一步得到提高。

参考文献

1 Qiang Baohua, Wu Kaigui, Wu Zhongfu. A Data-type-based Approach for Identifying Corresponding Attributes in Heterogeneous Databases. Xi'an, China: In: Proceedings of 2003 International Conference on Machine Learning and Cybernetics, 2003-11

2 Qiang Baohua, Wu Kaigui, Liao Xiaofeng. Similarity Determination on Data Types in Heterogeneous Databases Using Neural Networks. Nanjing, China: In: Proceedings of 2003 International Conference on Neural Networks and Signal Processing, 2003-12

3 Copas J B, Hilton F J. Record Linkage: Statistical Models for Matching Computer Records. J. Royal Statistical Soc.,1990, 153(3): 287-320

4 Dey D, Sarkar S, De P. A Probabilistic Decision Model for Entity Matching in Heterogeneous Databases. Management Science, 1998, 12(10): 1379-1395

5 Dey D, Sarkar S, De P. A Distance-based Approach to Entity Reconciliation in Heterogeneous Databases. IEEE Transaction on Knowledge and Data Engineering, 2002, 14(3)

6 Barron F H, Barrett B E. Decision Quality Using Ranked Attribute Weights. Management Science, 1996, 42(11): 1515-1523

（上接第 9 页）

有交集或交集很小时，也能利用领域知识找到用户的最近邻进行推荐，从而取得较好的推荐效果。

本算法中相似度与传统方法中的相似度相比，计算方法相对更宽松。因为基于向量的相似度计算方式是属性对象之间的严格匹配，而本方法是基于对象类的匹配。从实验上看，当用户评分项数超过某个阈值后，利用传统的相似度计算方法，能为当前用户找到更合适的邻居，推荐效果更佳。因此，一个比较好的选择是在系统启动时，先使用我们的算法，当用户的评测项目数达到一定的程度后，再利用基于传统相似度计算的协同推荐算法，比如 Pearson 方法，会取得更好的推荐效果。

5 结论

本文深入分析了协同过滤系统在系统启动时用户评分项很少、数据极端稀疏的情况下，传统相似度如余弦相似性、Pearson 系数等在计算相近邻时存在的问题，并针对该问题提出了基于领域知识计算用户评分项的相似度的协同过滤算法。实验结果表明，利用领域知识计算相似度可以解决用户评分项很少、数据极端稀疏情况下的系统“冷启动”现象，提高系统的推荐质量和精度。

参考文献

1 Shardanand U, Maes P. Social Information Filtering: Algorithms for Automating "Word of Mouth". In: Proceedings of the ACM CHI'95 Conference on Human Factors in Computing Systems, 1995: 210-217

2 Breese J, Hecherman D, Kadie C. Empirical Analysis of Predictive Algorithms for Collaborative Filtering. In: Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI'98),1998: 43-52

3 Ganesan P, Garcia-molina H. Hierarchical Domain Structure to Compute Similarity. ACM Transaction on System, 2003,21(1): 64-93

4 Rodriguez M A, Egenhofer M J. Determining Semantic Similarity Among Entity Classes from Different Ontologies. IEEE Transactions on Knowledge and Data Engineering,2003, 15(2): 442-456