基于聚类算法的个性化搜索研究①

Personalized Search Based on Clustering Algorithms

帅剑平　周娅　(桂林电子科技大学 计算机与控制学院 广西 桂林 541004)

摘　要： 搜索引擎的出现使得用户从信息爆炸性增长的互联网上获取所需的信息成为可能，个性化搜索引擎的研究使搜索结果尽可能满足不同用户的信息需求。文中提出了一种基于改进的 DBSCAN 算法的个性化搜索方法，在全文搜索包 lucene 与开源搜索引擎 Nutch 的基础上，实验证明该方法改善了聚类的结果，提高了用户搜索的准确率。

关键词： 个性化 DBSCAN 聚类 准确率

1　引言

随着计算机技术与网络技术的发展，Internet 上的信息呈爆炸性的增长。搜索引擎为用户提供了一种有效、方便的从 Web 上检索信息的方法。然而，搜索引擎返回的结果往往一半以上的信息没有任何意义，用户只能简单地从选择的搜索引擎所返回的结果中逐个筛选自己感兴趣的内容，这无疑是非常耗时的过程。通常情况下一个词在不同的语境下会有很多不同的意思，例如：对关键词“苹果”进行搜索，一些用户也许想了解苹果作为一种水果所具有的信息，而有些用户关心的则是关于苹果电脑的相关信息。如何从海量的信息中高效全面地获取所需知识及最新信息，如何使得搜索结果能够满足不同的用户的要求，是人们急待要解决的问题。因此，为了提高搜索强擎用户对搜索引擎网站的满意程度，搜索引擎的个性化趋势是搜索引擎未来发展的重要特征和必然方向。

聚类是数据挖掘中一种重要的方法，它是一种非监督学习、基于观察的方法。聚类是按照数据的相似性和差异性，将数据划分为若干组，同组的尽量相似，不同组的尽量相异。聚类算法所具有的特性对个性化搜索引擎的发展有着特殊的意义，本文通过研究聚类算法如何在个性化搜索引擎中聚类数据。

2　相关研究

个性化搜索的研究对于改进搜索结果、返回用户满意的结果方面有着极其重要的作用。文献[1]中的个性化搜索系统研究的重点集中在用户兴趣模型上，通过对用户浏览行为、服务器日志、Cookie logs 等信息的分析处理，用向量空间模型、本体或是其它方法构建兴趣模型，然后计算用户兴趣模型和结果文档相关性，在返回搜索结果给用户之前根据相关性对其进行排序。文献[2]对个性化搜索的查询词进行扩展，其中通过语义本体的方法对查询词进行语义上的扩充，自动创建结构化的兴趣模型，以期能够准确表达用户的搜索意图。文献[3]中通过用概率查询扩展的方法来处理查询词，通过计算查询词和文档之间的概率相关性来获取比较好的查询准确度和数据召回率。

聚类技术在搜索引擎中的研究有着不可替代的地位，人们利用聚类技术在这搜索引擎方面的研究初见成效，如 Vivisimo、Mooter 这类基于聚类的商业搜索引擎。文献[4]提出了基于相关性的聚类算法，定义了三个类别（相关、部分相关各不相关），聚类结果单一，对处于类别边界的数据并没有得到很好的处理，不能获得任意相关度的聚类数据。聚类算法在 web 领域的研究大多数都考虑文档集自身的特性，首先对文

① 基金项目:广西自然青年基金资助项目(桂科青 0832101)
收稿时间:2008-07-19

档特征提取，进而对文档进行聚类，并没有考虑不同用户的信息需求。其中有些聚类算法如 K-means 对噪音数据不敏感和对聚类数据的形状有所要求，容易使得聚类结果并不是很理想。本文改进了 DBSCAN[5] 算法，对搜索结果进行基于用户兴趣模型的聚类。

3 基本密度的聚类算法 DBSCAN

基于密度的聚类算法是一类重要的聚类算法，与基于划分的方法不同的是，它没有去假定聚类中心，而是根据数据集自身的密度分布探测获得类簇。比起基于划分的方法其优势在于可以发现任意形状的簇，而且可以有效去除噪声。众所周知，对于待聚类的数据对象集，通常由一些在特征空间上离散分布的数据对象组成。DBSCAN 是从离散分布的观点来看待数据分布特性的典型算法，它由 Martin Ester 等人在 1996 年提出，它通过检查数据库中每个点的 ε -邻域来寻找聚类。如果一个点 p 的 ε -邻域包含多于 MinPts 个点，则创建一个以 p 作为核心对象的簇。然后，DBSCAN 反复地寻找从这些核心对象直接密度可达的对象，这个过程可能涉及一些密度可达簇的合并，当没有新的点可以被添加到任何簇时该过程结束。

引理 1：P 是数据集合 D 中一个核心对象，则由 P 密度可达的点构成的集合 O 是一个聚类。

引理 2：C 是一聚类，P 是 C 中的任意一个核心对象，则 C 等于由 P 密度可达的点构成的集合 O。

基本 DBSCAN 算法

输入：包含 n 个对象的数据库，半径 ε，最少数目 MinPts

输出：所有生成的簇，达到密度要求

(1) REPEAT

(2) 从数据库中抽取一个未处理过的点

(3) IF 抽出的点是核心点 THEN 找出所有从该点密度可达的对象，形成一簇

(4) ELSE 抽出的点是边缘点(非核心对象)，跳出本次循环，寻找下一点

(5) UNTIL 所有点都被处理

基于密度的聚类算法需要指定聚类参数 ε 和 Minpts 的值，这两个参数的值往往难以设定，重要的是这样两个参数决定了簇的形成，以及哪些点被视为噪声点，当对 ε 和 MinPts 的设置有微小的变化都有可能产生迥异的聚类结果。难以设定且对聚类结果产生致命影响的两个参数，DBSCAN 及部分改进的算法都将设定该参数的任务交给用户；当然，如果用户对整个数据集的分布非常了解的情况下，聚类结果无疑将比较准确；然而这种熟悉所面对的仅仅是专业人员。另外也有其它的参数选取方法：计算数据集中每个数据对象的 k-th 最近距离，绘制 k-dist 图，再由用户指定 ε (大多对应拐点位置)；然而，大量实验后发现：很多数据集的 k-dist 图中有很多类似“拐点”位置，尤其数据高维且海量时，这就使得指定 ε 不可行；同时，绘制 k-dist 图的过程属计算密集型，代价颇高。

4 DBSCAN 算法的改进

与经典的 DBSCAN 算法类似，改进的算法也是一个逐步迭代的过程，针对于 DBSCAN 算法的缺点，从下列方面改善聚类效果。

4.1 ε 和 minpts 的选取

对于用户查询，用一个带权值的向量 Q 表示。采用向量空间模型(Vector Space Model 简称 VSM)来表示用户的兴趣特征，其基本思想是用向量表示文档，每个文档表示为 $((t_1,w_1)(t_2,w_2)\ldots(t_N,w_N))$，其中 t_i 为文档中第 i 个项，w_i 为项 t_i 在文档中的权重，项的权重表示它们在文档中的重要程度。在信息检索中，使用 TF*IDF(Term Frequency; Inverse Document Frequency)算法来计算文档的权值。计算 Q 与兴趣向量的相似性，设定查询与兴趣向量的相似度阀值 T，将用户查询 Q 与兴趣向量的相似度阀值大于 T 所组成的集合表示为 $CS=\{D_1\ldots D_m\}$，$D_i=\{D(c_1,D_i)\ldots D(c_j,D_i)\}$，$D(c_j,D_i)$表示数据库 D_i 中第 c_j 个文档。

计算 Q 与集合 CS 中兴趣向量的欧式距离的平均值，将其作为 DBSCAN 算法的输入参数 ε。欧式距离函数如下：

$$D(X_i,X_j)=\left\|X_i-X_j\right\|=\sqrt{\sum_{k=1}^{n}(X_{ik}-X_{jk})^2}$$

对于 DBSCAN 算法的 MinPts 值的选取，不采用随机选取或是固定值的方法，根据 $sim(Q,Di)\geq T$ 把相似度大于阀值 T 的 K 个用户兴趣向量记为 $S_1,S_2,\cdots,S_k$ 并将 K 值作为密度聚类算法的输入参数 MinPts 的值。

4.2 不同密度簇的识别

DBSCAN 算法是一种通用的聚类算法，能够发现不同大小、不同形状的类簇，并且对噪音数据比较敏感。但是由于全局 ε 和 minpts 值的设定，使得它不能识别不同密度的类簇。 ε 的选值过大时，会把密度相差较大，且相关性并不大的几个类簇聚成一个大的类簇， ε 的选值过小时，当然也会把本该属于一个类簇分裂成不同的类簇，聚类效果不理想。如下图所示

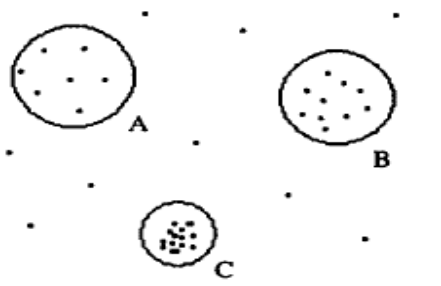

图 1　不同参数值的聚类结果

为此，我们在 4.1 中 ε 和 minpts 确定了的情况下，利用密度区分的方法对符合条件的数据对象进行聚类。

核心对象 P 在 ε 邻域内有 m_0 个数据对象(包含 P 在内)，m_i 其中 i[0,m_0−1]代表这些 m_0 个数据对象在 ε 邻域内的数据对象的个数，相对于 m_0 的值，DBSCAN 算法对 m_i 的取值没有任何限制。为了能够检测出不同密度的聚类，首先给出下列两个条件：

条件 1　(1＋α)m0≥mi>m0

条件 2　m0>mi≥m0/(1＋α)

α 是[0，1)之间的小数，这里取 α 的经验值 0.2，对于任何一个核心对象只要不满足这两个条件就不能聚类扩展。

改进的 DBSCAN 算法具体步骤描述：

(1) 查询 Q 与兴趣向量的相似度计算 $\sum_{k=1}^{m} D(X_i, X_j)$

(2)K=count{D_1...D_m}，dist=D(X_i,X_j) $\frac{}{m}$

(3) 根据(2)得到的 K、dist 值分别作为 DBSCAN 的输入参数 ε 、MinPts

(4) Repeat

(5) 从数据库中抽取一个未处理过的点 P

(6) If　P 是核心点

(7) If (1＋α)m0≥mi>m0 或者 If m0>mi≥m0/(1＋α)

(8) 找出所有从 P 点出发满足(7)的点且未被分配到簇中的点归类

(9) Else 不把能该点指派到簇中

(10) Else 跳出本次循环，寻找下一点

(11) Until 所有点都被处理

5　实验结果

为了评估算法的性能，利用准确率(precision)来进行结果分析。准确率表示为与用户查询相关的文档数和用户查询的总文档的比值。

Precision=n ’ /N

其中 n ’ 表示用户查询相关的文档数； N 表示用户查询的总文档数。

在全文搜索包 lucene 与开源搜索引擎 Nutch 的基础上，设计了基于 DBSCAN 的个性化算法，对 10 个用户的 50 个不同的查询结果进行了试验。

一个用户搜索的准确率我们用此用户 50 次查询的平均准确率来表示，实验结果如下：

查询词	原始的搜索引擎(准确率)	基于 DBSCAN 的个性化搜索(准确率)
Net works	0.3	0.85
癌症	0.35	0.91
Office	0.4	0.6

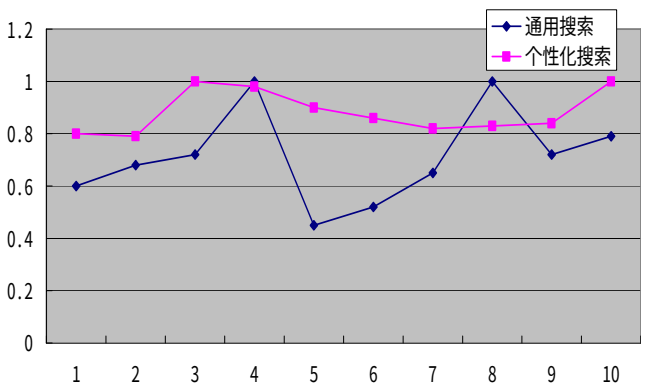

图 2　每个用户的平均准确率

通过实验数据分析，基于密度聚类的个性化搜索结果趋于一个相对稳定的状态，而通用搜索的结果相对用户的需求波动比较大，其中有些用户搜索结果不相关的程度比较高。对于在改进的方法中，也有些点的准确率不高比如 user1 和 user2，造成这种结果的原因有两种：①用户对一些搜索词的随意性，并没有体现出用户的真正意图；②用户兴趣的频繁改变。

6 结论

综上所述，本文通过对现有个性化搜索引擎和聚类算法中密度聚类特点的研究，在传统密度聚类的基础上提出了个性化搜索中数据聚类算法，结合用户兴趣实现了个性化信息搜索，提高了用户的搜索效率和准确率，并通过实验验证了算法的有效性。

参考文献

1 Speretta M, Gauch S.Personalized Search Based on User Search Histories//Multimedia and Ubiquitous Engineering,2005:229–232.

2 Pretschner A,GauchS.Ontology Based Personalized Search,1999.

3 Palleti P, Karnick H, Mitra P. Personalized Web Search using Probabilistic Query Expansion//web intelligence and intelligent agent technology, 2007:83–86.

4 Desai M, Spink A.An algorithm to cluster documents based on relevance//Information Porcessing and management, 2005,41:1035–1049.

5 EsterM, KriegelH-P, SanderJ, Xu X. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise.International Conference on Knowledge Discovery in Databases and Data Mining(KDD-96), AAAI Press, Portland Oregon, 1996:226–231.

6 蒙祖强,蔡自兴.个性化数据聚类的研究.计算机工程与应用,2003(33).