文章编号:2095-476X(2012)06-0009-05

基于可调整邻域阈值的DBSCAN算法在应急预案分类管理中的应用

金保华, 林青, 赵家明

(郑州轻工业学院 计算机与通信工程学院, 河南 郑州 450002)

摘要:针对庞大的预案文本资源分类难的问题,将可调整的邻域阈值*Eps*取代原有的全局*Eps*,得到了改进的DBSCAN密度聚类算法.以预案文本间的相似度作为聚类基本的度量属性,将改进的DBSCAN算法应用于应急预案分类管理,以去除边界.仿真结果证明该方法不仅不影响预案本来的基础分类方式,而且更易于实现,在一定程度上能够缓解噪音点误识别问题,对提高预案文本的重用性和分类的准确率有一定的参考意义.

关键词:DBSCAN算法;文本相似度;应急预案文本管理;可调整邻域阈值

中图分类号:TP391 **文献标志码**:A DOI:10.3969/j.issn.2095-476X.2012.06.003

Application of DBSCAN algorithm based on adjustable threshold in the emergency plan classification management

JIN Bao-hua, LIN Qing, ZHAO Jia-ming

(*College of Computer and Communication Engineering, Zhengzhou University of Light Industry, Zhengzhou* 450002, *China*)

Abstract: Aiming at large plan texts resource classification problems, adjustable threshold Eps replaced the original global threshold Eps. An improved DBSCAN clustering algorithm based on density was put forward. The similarity between plan texts was taken as measurement attribute. Improved DBSCAN was applied in the field of plan classification to remove the boundary identification error. The simulation results showed that this method not only does not affect the result in basis classification way, but also have certain reference significance to improve accuracy and reusability of classification.

Key words: DBSCAN algorithm; text similarity; emergency plan text management; adjustable threshold

0 引言

应急预案需明确规定应急管理的责任主体、工作范围、运行机制等,因此,预案管理的工作效率是应急事件处置过程中的关键.但现存的预案信息系统只能单纯地实现文本上传与查看功能,检索效率低下.因此,与现代信息技术、文本挖掘技术相结合的数字化应急预案,是应急预案管理发展的方向.本文将改进的DBSCAN算法应用在应急预案分类管理方面,力求在面对突发情况之时,提供分类整合原有预案文本资源的功能,为应急指挥人员提供高层次有侧重的决策方案,以提高预案重用性.

收稿日期:2012-06-29

作者简介:金保华(1966—),男,河南省郑州市人,郑州轻工业学院副教授,主要研究方向为人工智能、计算机辅助决策系统.

1 预案文本间的相似度

应急预案作为一种规范性与结构性较好的文本,有一定的灵活性. 使用效果较好的向量空间模型[1]能对半结构化的预案文本信息做结构化处理,整个预案文本空间由经过范化的预案文本特征向量构成. 该特征向量可以表示为

$$\boldsymbol{V}(d) = \sum_{i=0}^{n} (t_i, w_i(d))$$

其中,t_i 为预案文本经过预处理后,利用分词工具提取出来的文本 d 中的关键词 i; $w_i(d)$ 为关键词 i 的权重,表示该词反映预案文本 d 主题的重要程度. 应用矢量空间模型的前提为:每篇预案文本是由一组相互独立的词条所构成的集合,表示为 $d = \{t_1, t_2, \cdots, t_n\}$.

设预案 $plan_x$ 可以表示为$(w_{x1}, w_{x2}, \cdots, w_{xn})$,预案 $plan_y$ 可以表示为$(w_{y1}, w_{y2}, \cdots, w_{yn})$,则两者之间的相似度为

$$\mathrm{Sim}(plan_x, plan_y) = \cos(plan_x, plan_y) = \frac{plan_x \times plan_y}{\| plan_x \| \times \| plan_y \|} = \frac{\sum_{i=1}^{n} (w_{xi} \times w_{yi})}{\sqrt{\sum_{i=1}^{n} w_{xi}^2} \cdot \sqrt{\sum_{i=1}^{n} w_{yi}^2}}$$

由此可以推断,设输入的预案文本集为$(plan_1, plan_2, \cdots, plan_i)$,则可以得到文本相似度倒排表

$$\mathrm{SimList}(plan_i, plan_j) = \sum_{\substack{i=1, j=2 \\ i<j}}^{n} [\mathrm{Sim}(plan_i, plan_j), plan_i - plan_j]$$

以及编号为 i 的文本与其他文本的相似度倒排表

$$\mathrm{SimList\_}i(plan_i, plan_j) = \sum_{j=1, i \neq j}^{n} [\mathrm{Sim}(plan_i, plan_j), plan_i - plan_j]$$

倒排表均按从大到小的相似度进行排列,输出形式均为

$$< \mathrm{Sim}(plan_i, plan_j), plan_i - plan_j >$$

其中,$\mathrm{Sim}(plan_i, plan_j)$ 为预案 $plan_i$ 与 $plan_j$ 之间的相似度,$plan_i - plan_j$ 为预案文本标识码.

2 DBSCAN 的改进及其在预案分类管理中的应用

2.1 DBSCAN 密度聚类算法原理

DBSCAN(density-based spatial clustering of application with noise) 是一种经典的基于密度的聚类算法,应用广泛. 其计算的时间复杂度为 $O(n^2)$,空间复杂度为 $O(n\log n)$,相关概念定义如下[2].

定义 1 类半径:数据集空间中的类簇半径,表示为 *Eps*.

定义 2 密度:空间中任意一点的密度是以该点为圆心、以 *Eps* 为半径的圆区域内包含的点数目.

定义 3 邻域:空间中任意一点的邻域是以该点为圆心、以 *Eps* 为半径的圆区域内包含的点集合.

定义 4 核心点:空间中某一点的密度如果大于某一给定阈值 *Minpts*,则称该点为核心点.

定义 5 直接密度可达:已知 *Eps* 和 *Minpts*,对于点 x 和点 y,如果 y 是核心点,而且 x 属于 y 的 *Eps* 邻域,则点 x 从点 y 直接密度可达.

定义 6 噪音:事先给定 *Eps* 和 *Minpts*, 基于密度聚类中的一个聚类就是可以密度连接所能包含的最多数据点的集合,不属于任何聚类的数据点的集合称为噪音.

假定输入参数为 *Eps* 和 *Minpts*,DBSCAN 算法的流程图如图 1 所示. 具体描述如下:

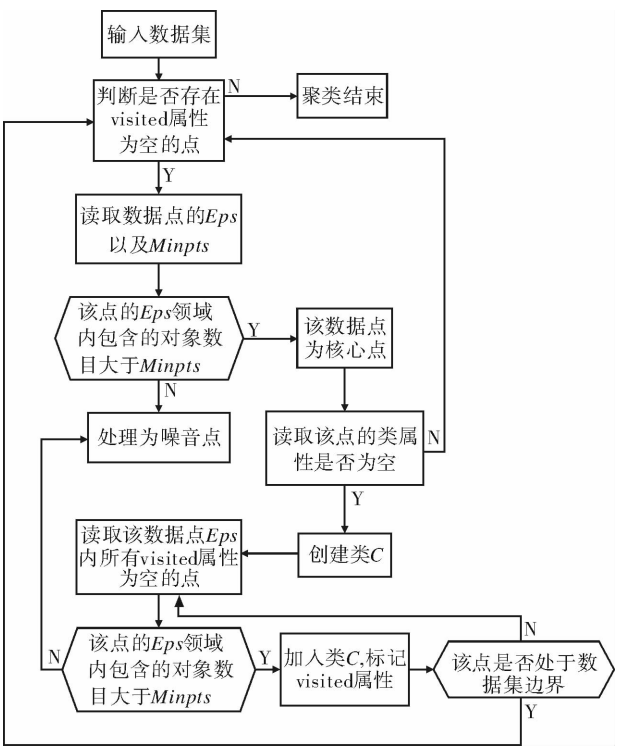

图 1 DBSCAN 算法的流程图

1)输入聚类数据,然后任意选取 1 个数据点 x,检查数据点 x 的 *Eps* 邻域.

2)如果 x 是核心点,而且没有被划分到某一个类,则创建一个类 C,找出所有从 x 密度可达的点并

加入 C.

3)依次检查 C 中未处理过的对象,如果该对象的 Eps 邻域内包含的对象数目大于 $Minpts$,就把该邻域中未包含于类 C 的对象加入 C 中.

4)如果 x 不是核心点,则被当作噪声处理.

5)转到第 1 步,重复执行算法;如果数据集合中所有的点都被处理过,则算法结束.

2.2　DBSCAN 算法的改进

DBSCAN 算法最大的优点是可以发现任意形状的类簇,而不受到噪音的影响. 聚类的结果会随着用户设定的全局 Eps 邻域值的改变而改变,为了确定 Eps 的初始值,DBSCAN 需要计算所有数据对象与其第 k 个(此处 $k = 6$)[3] 最邻近的对象之间的距离,并将结果按距离排序,由此得到 k-dist 图. 同时,DBSCAN 算法需要搜索从某个核心点出发到所有密度可达的点,这一步是经过反复进行区域查询实现的,因此需要建立 R^*-tree 来查询返回给定查询区域中的所有对象. 这 2 个点相应的代价都是系统的大量内存与 IO 开销[4].

原始 DBSCAN 算法建立 k-dist 图的时间复杂度为 $O(n^2)$, 建立 R^*-tree 的时间复杂度为 $O(n\log n)$, 基于聚类过程的大部分时间用在区域查询操作上,因此该算法平均时间复杂度为 $O(n\log n)$. 在明确了 SimList 表的本质即是 k-dist 图的基础上,笔者对该算法进行改进:DBSCAN 原始算法的第 2 步 R^*-tree 的建立过程被搜索 SimList 表种子队列 SeedList 中满足要求的记录所取代(在后面一节详细展开). 仿真试验时效图见图 2,经过改进的算法减少了原来搜索查询区域的时间,降低了系统的内耗,提高了效率.

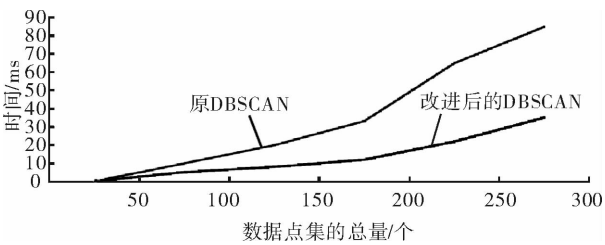

图 2　仿真试验时效图

原 DBSCAN 中,在预案文本密度与类之间的间距不均匀的情况下,较小的 Eps 值会将边界点处理成噪音点[5];反之,则可能将离得较近而密度较大的那些类合并为同一个类,使得聚类结果不够准确. 在改进的算法中,使用可调整的邻域阈值 Eps,可调整的 Eps 能够有效地针对密度不均的现状,与软聚类结合后使因数据过于稀疏而造成的错误识别边界点的情况得到一定的改善.

2.3　改进的 DBSCAN 算法在预案分类管理中的应用

随着各类突发公共事件的发生日益频繁,相关应急预案的框架体系逐步完善,最基本的分类已经无法满足应急决策者在进行应急决策时搜索与分类整合的需求. 利用文本聚类中 DBSCAN 密度聚类的方法,不仅能够对预案文本进行快速分类,方便决策者获取相关的预案信息,而且能够去除预案文本的边缘性信息,提高预案重用性.

一个预案要具备重用性,往往基于以下 2 个方面的原因:

1)从预案本身的功能来说,预案文本资源属于政府公文,而且都是依据相应的法律以及行政法规制定的,虽然地区间的实际情况不相同,但是既然法律存在通用性,预案的制定者就可以借鉴同级预案的经验进行编制,决策者也可以参考以往同类预案的方案进行应急指挥工作.

2)从聚类的目的来说,以文本相似度为属性度量依据,对预案文本使用软聚类后,即一个预案文本可以归入多个簇中,以达到预案文本可以在不同侧面体现参考价值的目的.

因此,结合预案文本的特性,可以对以预案文本相似度为度量依据形成的文本空间使用改进的 DBSCAN 算法. 改进后的算法的伪代码如下所示:

1)首先,定义预案文本对应的数据结构.

```
// 定义数据集数据点的类,并设定其属性 Minpts 为 int 类型,Eps 为 float 类型,visited 为枚举类型(0 代表未访问,1 代表已访问)
    public class MetaSet{
        private int Minpts,visited;
        private float Eps;
        public int getMinpts(){
        return Minpts;
        }
        public void setMinpts(){
        this.Minpts = Minpts;
        }
        public int getvisited(){
        return visited;
```

```
        }
        public void setvisited(){
        this. visited = visited;
        }
        public float getEps(){
        return Eps;
        }
        public void setEps(){
        this. Eps = Eps;
        }
    }
```

2)按序搜索 SimList 中未访问过的数据点形成种子队列,并对种子调用聚类方法.

```
public void SeedList(){
        < MetaSet > ls = new < ListArray > ();
        ls. add(数据点);
        for(MetaSet ms:ls){
        if(ms. visted = = 0&& ms. Eps 内 Minpts > a){
        Cluster(ms);
        ms. visited = 1;
        }
        else("类库生成完毕");
        }
        else("类库生成失败");
    }
```

3)生成以 MetaSet 类中的种子数据为核心点的类 C_0.

```
public void Cluster(MetaSet sm){
    CreatClusterClass(C0).
    < MetaSet > ls0 = new < ListArray > ();
    //以 Eps0 启发式的计算出当前需要扩展搜索的区域邻域阈值 Epsi
    for(int i = 0; ;i ++){
    Epsi = (Epsi - 1 + Min(Sim))/2 - 0.1
    for(MetaSet ms0:ls0){
    if(ms0. visted = = 0&& ms0. Epsi 内 Minpts > a){
    C0. add(ms0);
    }
    else{
    将 ms0 处理成为噪音点;
    }
  }
 }
}
```

3 仿真结果与分析

3.1 试验环境

使用 Java 编写仿真软件,在分词上采用的是中国科学院提供的 imdict-chinese-analyzer 开源分词工具. 本实验所使用的语料为郑州轻工业学院应急研发中心的预案语料库,其中预案文本严格按照预案的分类方法,即按预案类型主要分为自然灾害、事故灾难、公共卫生与社会安全 4 类. 从各类中选择一定数量的文档共 100 篇作为训练集,并随机选择自然灾害类 80 篇,事故灾难类 70 篇,公共卫生类 50 篇,社会安全类100篇作为测试的文本数据集合.

3.2 试验结果

取数据集合的 1/25 作为 *Minpts* 的值是一种行之有效的方法[6],即此处训练集与测试集的 *Minpts* 均取 4. 在此基础上,对固定 *Minpts* 参数的训练预案文本集选取不同的 *Eps* 值,并使用原始的 DBSCAN 对其聚类,精度(P) 与召回率(R) 分布情况分别如图 3,图 4 所示.

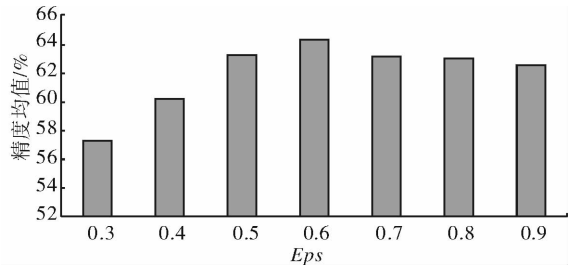

图 3 精度分布图

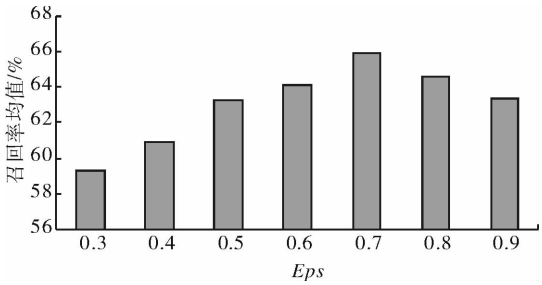

图 4 召回率分布图

由此可以得知,训练集的精度均值在 $Eps = 0.6$ 处取得峰值,召回率均值在 $Eps = 0.7$ 处取得峰值. 因此根据训练集的训练情况,设置测试集 *Minpts*, *Eps* 的初始值为 $Minpts = 4$, $Eps = 0.8$,并设定 *Eps* 的变化区间为[0.5,0.8]. 对测试集进行聚类试验,结果见表 1 和表 2.

表 1　原 DBSCAN 测试结果　　篇

大类	小类	随机抽取	系统分类	正确分类
自然灾害	2	80	99	62
事故灾害	2	70	80	46
公共卫生	2	50	39	26
社会安全	4	100	82	63

表 2　改进后的 DBSCAN 测试结果　　篇

大类	小类	随机抽取	系统分类	正确分类
自然灾害	11	80	96	68
事故灾害	6	70	60	50
公共卫生	4	50	54	32
社会安全	12	100	90	78

3.3　结果分析

模糊矩阵、熵、整体相似度、分类正确率、精度和召回率都是文本聚类的质量评价方法. 本文采用精度和召回率作为改进的 DBSCAN 算法性能的评价标准. 原始的 DBSCAN 算法与改进 DBSCAN 算法对 150 篇随机抽取的预案文本进行聚类的精度均值分别为 64.55% 与 75.02%,召回率均值为 65.75% 与 74.60%. 试验数据表明以下 4 点:

1)虽然测试集的文本被划分为 33 个子类,但是划分出来的子类之间均属于原来的大类,因此划分出小类对原来的数据文本的基本分类并无影响.

2)小类中预案文本之间在文本结构、内容上表示出一定的相似之处,较之原算法精度与召回率均得到了提高,符合为预案制定者提供参考的功能要求.

3)聚类效果明显,同一文本会被划分到不同的子类之中,预案管理系统用户可以根据需求将对决策者有利的子类保存,并进行个性化的调整. 而且可以看到,使用改进的 DBSCAN 算法划分出来的小类的数量明显比原始算法大,倘若从类中随机抽取预案文本来生成参考组,采用可调整的 *Eps* 能够更细致地得到更佳的参考组.

4)改进后的 DBSCAN 算法能够有效降低系统内存与 IO 的消耗,在预案文本的数据点个数持续增加的过程中,时间消耗仅是原算法的 1/3 左右.

4　结论

本文对原有的 DBSCAN 密度聚类算法进行改进,将可调整的 *Eps* 邻域阈值应用到预案文本聚类中,以取代原来的全局 *Eps*,提高了 DBSCAN 算法的准确性. 仿真结果表明,该方法不仅不影响预案本来的基础分类方式,而且更易于实现,在一定程度上能够缓解噪音点误识别问题,对提高预案文本的重用性和分类的准确率有一定的参考意义.

参考文献:

[1] 刘志勇,耿新青. 基于模糊聚类的文本挖掘算法[J]. 计算机工程,2009,35(5):44.

[2] Han J, Kamber M. 数据挖掘概念与技术[M]. 范明,孟小峰,译. 北京:机械工业出版社, 2008.

[3] 夏鲁宁,荆继武. SA-DBSCAN:一种自适应基于密度聚类算法[J]. 中国科学院研究生院学报,2009,26(4):530.

[4] Das S, Abraham A, Konar A. Automatic clustering an improved differential[J]. IEEE Transactions on Systems Man and Cybernetics (Part A): Systems and Humans, 2008,38(1):218.

[5] Sanjay C, Sun Pei. SLOM:a new measure for local spatial outliers[J]. Knowledge and Information Systems, 2006, 9(4):412.

[6] 于亚飞,周爱武. 一种改进的 DBSCAN 密度算法[J]. 计算机技术与发展,2011,21(2):30.