减聚类的模糊 C-均值算法在文本分类中的应用①

王 月 柴瑞敏 (辽宁工程技术大学 辽宁 葫芦岛 125105)

摘 要: 首先，选择合适的文本集合，并且对文本进行分词处理，然后，进行文档内部特征词的提取，通过采用词频统计的方法对文本向量进行降维处理，从而选择最佳的特征向量。最后，将非数值的文本数据进行量化处理后,利用减聚类优化的模糊 C-均值算法对文本集合进行聚类,从而提高文本聚类的效果。

关键词: 模糊聚类；文本分类；特征选取；VSM；减聚类

Application of Subtractive Clustering's Fuzzy C-Means Categorization to Text Categorization

WANG Yue, CHAI Rui-Min

(Liaoning Technical University, Huludao 125105, China)

Abstract: In this paper, fuzzy C-means categorization optimized by Subtractive clustering is applied to text clustering. First of all, the paper chooses a suitable text collection and deals with word segmentation of the text. Then, it extracts the internal idiocratic words of the documents, and uses word frequency statistics for the text dimensionality reduction processing, to choose the best eigenvector. Finally, after quantifying the text of the non-numerical data, it clusters the collections of text with fuzzy C-means algorithm which is optimized by Subtractive clustering, so as to enhance the effectiveness of text clustering.

Keywords: fuzy clustering; text categorization; feature selection; VAM; subtractive clustering

现在我们已经生活在一个数字化的时代中，数字正极大地影响着整个人类社会。然而，我们不得不承认随着大量信息给人们带来方便的同时，也带来了很多的问题，在日常生活中，文本已经成为重要的信息表达形式。据研究资料表明大约有 90%以上的信息包含在文本文档中。因此对文本文档的处理和分析成为当今数据挖掘和信息检索技术的热点之一。

处理和研究文本文档的技术有很多，其中最为重要的一个技术就是文本聚类。可以将文本数据按相似性聚类成簇，以供进一步挖掘和分析。而在本文主要研究中文文本聚类的问题。在文本聚类方面很多关于 FCM 的方法也被提了出来。由于该算法其本质上是一种局部最优搜索寻优技术，在进行聚类以前要求知道 C 值，这个是比较困难知道的。初始聚类中心的选择对于最后的聚类结果有很大的影响，如果初始聚类中心选择不当，目标函数有可能得不到全局最优值，而导致陷入局部极小值。这极大的影响了文本聚类的效果，所以本文将一种减法聚类和模糊 c-均值相结合的模糊聚类[1]应用到文本分类中，来提高对文本聚类的效果。

1 文本的预处理

1.1 分词

因为中文本身的特点所致，词成为文本的主要成分，是最能够反应文本语义的基本单位，一般情况下选择词作为特征项就能够充分表示中文的语义。所谓分词，就是把一个句子按照其中词的含义进行拆分。中文文本在书面表达或计算机内部表示时，字和字之间、词和词之间并没有明显的拆分标志，因而在对中文文本进行处理之前首先要进行分词处理。

1.2 特征项选取与降维处理

因为表示文本特征的词汇数量会随着文本数量增多而增多，造成相应词频构成的矩阵维数很高。利用高维词频矩阵计算文本相似程度会大大增加运算量，

① 收稿时间:2009-06-30

使得信息处理的效率降低。因此需要对提取出来的特征向量集进行降维处理，以便提高文本聚类的效率和运行速度。一般来说，以所有的词为初始特征项，其中存在一些对文本内容识别意义不大的词。例如：在文本集中出现的频率很低，甚至不出现的，称为稀有词，这种稀有词语一般也不适合作为文本的特征项，对于这一类词语，应该进行筛选。在各类文本中有一些词汇出现的频率很高，并且常影响真正有分类作用的实词，称之为禁用词，可以去掉，例如“的”，“他们”，“了”等。这些都是通过对文本集进行词汇频率统计，并选择一个合适的阈值来实现。根据文本集的大小阈值设置是不同的。为了不误去一些虽然频率出现的较低但是重要的特征词汇阈值通常不设的太高。如果某个词语在所有文档中出现次数大于阈值的保留，否则就去掉。

目前，已经有很多关于特征项选取的方法，例如信息增益、期望交叉熵、互信息等，不过这些方法都是需要已知分类信息才能进行。然而文本聚类是一种无指导的文本分类，是不可能在聚类前给出已知的分类信息的，所以在本文采用基于词频的特征选择。TF-IDF 方法是依据某个词的词频和其出现过的文本的频率来计算该词在整个文本中的权重，依据权重来进行特征选取的。权重越高则该词对文本的区分能力越强，否则相反。对于一个词语 t_i，权重的计算公式为：

$$W_{ik}(t_i) = f_{ik}(t_i)\log\left(\frac{N}{n_k}+c\right) \quad (1)$$

其中：

$W_{ik}(t_i)$---第 k 个分词 t_k 对第 i 篇文本 T_i 所起的作用

$f_{ik}(t_i)$---第 k 个分词 tk 在文本 T_i 中出现的频率

N---所有文本的数目

n_k---文本集 T 中出现第 k 个分词 tk 的文本数

c---常数。为了使当 $n_k=N$ 时，$\log\left(\frac{N}{n_k}+c\right)>0$，一般 c 取一个较小的正数，这里取 c=0.01。

由于文本长度对权重的影响，做词频均衡归一化处理，将各项权规范到〔0，1〕之间，计算公式如下：

$$W_{ik}(t_i) = \frac{\sqrt{f_{ik}(t_i)\log\left(\frac{N}{n_k}+c\right)}}{\sqrt{\sum_{k=1}^{m} f_{ik}(t_i)\log\left(\frac{N}{n_k}+c\right)}} \quad (2)$$

则文本 t_i 可用向量表示为：$w_i=(w_{i1}, w_{i2}, \ldots, w_{im})$。

1.3 文本表示模型

由于文本所使用的是人类的自然语言，它属于非结构化的数据，因此要对文本进行聚类之前，首先就要将文本表示成可以进行分析处理的形式，这也是文本聚类的首要步骤。目前，关于文本常用的语言模型有以下几种：

① 布尔模型(Boolean Approach)：是基于集合论与布尔代数的一种简单模型。由于权重的二值性，所以布尔模型只能用于信息检索中计算用户查询与文档的相关性，而无法利用该模型计算两个文档更深层面的相似度。

② 概率模型(Probabilitic Model)：是一系列模型的简称，它综合考虑了词频、文档频率和文档长度等因素，把文档和查询按照一定的概率关系融合，并在概率测度空间通过概率来衡量两个文本的语义相似度。

③ 向量空间模型[2](Veetor Space Model，VSM)：是 20 世纪 60 年代末由 Gerard Salton 等人提出的，其中最为著名的应用该模型的检索系统是 Smart 系统。它采用简洁的特征向量来表示文档，在进行特征选择时，不使用大量的句法语法信息，也无需对文档进行复杂的语义处理。因此，本文采用 VSM 模型表示文本及其特征。

下面介绍 VSM 模型的几个基本概念：

文本(Document)：泛指一般的文本或者文本中的片断(段落、句群或句子)，一般指一篇文章。

项(Term)：文档的内容特征常常被它含有的基本语言单位(字、词、词组和短语等)来表示，这些基本的语言单位统称为项，即文档可以用项集(Term List)表示为 $T(t_1, t_2, \cdots, t_m)$，其中 t_k 是项，$1\leqslant k\leqslant m$。

项的权值(Term Weight)：对于含有 m 个项的文本 $T(t_1, t_2, \cdots, t_m)$，项 t_k 常被赋予一定的权值 W_{ik}，表示它们在文本 Ti 中的重要程度，即 $T_i=T(t_1, W_1; t_2, W_2, \cdots, t_m, W_m)$，简记为 $T=T(W_{11}, W_{12}, \cdots, W_{1m})$。这时我们说项 t_k 的权重为 W_{ik}，$1\leqslant k\leqslant m$。

向量空间模型(VSM)：给定一文档 $Ti=T(t_1, W_1, t_2, W_2, \cdots, t_m, W_m)$如果 tk 在文本中既可以重复出现又应该有先后次序的关系，分析起来仍有一定的难度。为了简化分析，可以暂时不考虑 t_k 在文档中的先

后顺序并要求 t_k 互异，这时可以把 t_1，t_2，…，t_m 看成一个 m 维的坐标系，而 W_1，W_2，…，W_m 为相应的坐标值，因而 $T=T(W_1, W_1, \cdots, W_m)$被看成是 m 维空间中的一个向量，称 $T=T(W_{11}, W_{12}, \cdots, W_{1m})$ 为文本 T 的向量表示。

文本特征向量(Feature Vector)：在 VSM 模型中，每一个文档都可以用一个向量来表示。向量的元素是由项(词条)及其权重组成，该向量就称之为此文本的特征向量。特向量是文档的一个特征表示，在某种意义上可以完全代表文档的特性。

通过了对文本的分词，降维和权重计算以后，文本集可表示为：

$$T = \begin{array}{c} \\ T_1 \\ T_2 \\ T_3 \\ \vdots \\ T_n \end{array} \begin{array}{c} \begin{array}{cccccc} t_1 & t_2 & t_3 & \cdots & \cdots & t_m \end{array} \\ \begin{pmatrix} W_{11} & W_{12} & W_{13} & \cdots & & W_{1m} \\ W_{21} & W_{22} & W_{23} & \cdots & & W_{2m} \\ & & & & & \\ & & & \vdots & & \\ W_{n1} & W_{n2} & W_{n3} & \cdots & & W_{nm} \end{pmatrix} \end{array} \quad (3)$$

完成文本的预处理后，将文本内容简化为特征项及权重的向量表示，同时保证这种结构化的形式能够充分表现文本对象自己的特征，为进行聚类分析做准备。

2　文本聚类算法

1973 年 Dunn[3]首先提出了第一个模糊 C-均值(Fuzzy C-Means,FCM)聚类算法，它是对 Ball 和 Hall[4]提出的硬 c-均值(HCM)聚类算法的推广。模糊 C-均值聚类算法是通过优化模糊目标函数得到每个样本点对类中心的隶属度，从而决定样本点的归属，它的迭代过程采用了一种所谓的爬山技术来寻找最优解。Chiu[5]提出了改进的山峰聚类(由 Yager 和 Filev 在 1994 年提出)的方法，被称为减法聚类法。减法聚类根据数据密度的原理，大幅度减少训练样本个数。也就是说，如果一个数据点有多个临近的数据点，密度值大，那么就用该点代替周围的其余各点。另一方面，稀疏的数据点，它们也可以作为各自的聚类中心。在此基础上肖春景等人提出一种基于减法聚类与模糊 c-均值的模糊聚类的思想，而本文则是在此基础上进行修改应用到文本聚类中去，实验证明该方法在文本聚类应用中是有效的。

2.1 FCM 算法

FCM 算法的基本思想：假设 $X=\{x_1,x_2,\cdots,x_n\}$是 n 维空间的一个特征向量集，根据某种相似度度量，该集合被聚合成 c 个子集: $v_1,v_2,\cdots,v_c$ ，$2\leqslant c\leqslant n$。这 c 个子集组成特征向量集 X 的一个模糊划分：用模糊隶属度矩阵 $U = [u_{ij}] \in R^{c\times n}$ 表示，U 中的元素 u_{ik} 表示 X 中的任意样本 x_k 对第 i 类的隶属度，且 u_{ij} 应满足以下条件：

$$u_{ij} \in [0,1] \quad (4)$$

$$\sum_{i=1}^{c} u_{ij} = 1 \quad (5)$$

令 $V=\{v_1,v_2,\cdots,v_c\}$是聚类中心，其中：v_i 是子集 X_i $(1\leqslant i\leqslant n)$的中心。$d_{ik}$ 表示为第 k 个样本到第 i 类距离，$m\in[1,\infty)$是权重指数。则 FCM 的目标函数表示为：

$$J_m(u,v) = \sum_{k=1}^{n}\sum_{i=1}^{c} u_{ik}^m d_{ik}^2 \quad (6)$$

显然，FCM 算法是基于对目标函数的优化，即对类内加权平均误差和函数 $J_m(U,V)$求最优值，在这里表示为：$\{J_m(U,V)\}_{min}$。由于矩阵 U 中各个列都是独立的，因此

$$\{J_m(U,V)\}_{min} = \left\{\sum_{k=1}^{n}\sum_{i=1}^{c} u_{ik}^m d_{ik}^2\right\}_{min} = \sum_{k=1}^{n}\left[\left\{\sum_{i=1}^{c} u_{ik}^m d_{ik}^2\right\}_{min}\right] \quad (7)$$

用拉格朗日乘数法来求解，求极值的约束条件为等式(5)，则：

$$F = \sum_{i=1}^{c} u_{ik}^m d_{ik}^2 - \lambda\left(\sum_{i=1}^{c} u_{ik}^m d_{ik}^2\right) \quad (8)$$

最优化的一阶必要条件为：

$$\begin{aligned} &\sum_{i=1}^{c} u_{ik} - 1 = 0 \\ &m(u_{ik})^{m-1} d_{ik}^2 - \lambda = 0 \end{aligned} \quad (9)$$

通过求解式子(9)可以得到，当 $J_m(U,V)$为最优值时：

$$u_{ik} = \sum_{j=1}^{c}\left(\frac{d_{ik}}{d_{jk}}\right)^{\frac{2}{1-m}} \quad (10)$$

找到最佳的中心矢量，使得每个样本数据到最佳中心矢量的加权平方和达到最小。最佳的类中心矢量 v_i 可由下式获得：

$$v_i = \frac{\sum_{k=1}^{n} u_{ik}^m x_k}{\sum_{k=1}^{n} u_{ik}^m} \quad (i=1, 2, \cdots, c) \quad (11)$$

下面给出 FCM 算法的算法步骤如下：

① 初始化：给定聚类数 c，且 0≤c≤n，设定迭代停止阈值 ε，然后初始化中心向量矩阵 V(0)，设置迭代计数器 b=0；

② 初始化模糊聚类中心 V_i,(1≤i≤c)

③ 计算划分隶属度矩阵 $U^{(0)}$。

④ 更新模糊聚类中心为 V*,如果 $\|U^{(b)}-U^{(b+1)}\| \le \varepsilon$，迭代结束，否则返回到第(3)步。

2.2 减聚类模糊 C-均值在文本聚类中的应用

① 文本经过分词、特征表示、降维预处理后，就将包含 n 个文本的文本集转化成为 n 个 m 维文本向量空间，记为：(T_1，T_2，…，T_n)，不失一般性，根据具体的情况设定邻域的半径 r_a　r_b 和误差允许值 ε。将数据点归一化到一个超立方体中。由于每个文本都是聚类中心的侯选者，因此，文本 T_i 的密度指标可定义如下：

$$D_i = \sum_{j=1}^{n} \exp\left(\frac{-\|T_i - T_j\|^2}{(0.5 r_a)^2}\right) \quad (12)$$

其中：r_a>0 表示在 r_a 的距离内避免出现下一个聚类中心点。特别地，如果一个数据点有多个邻近的数据点,则该数据点具有高密度值，半径 r_a 定义了该点的一个邻域，半径以外的数据点对该点的密度指标影响很小。

利用公式(12)计算所有文本集合中 n 个文本的密度指标。

② 在计算出每个文本的密度指标后，选择具有最高密度指标的文本为第一个聚类中心，令 T_{c1} 为选中的点，D_{ck} 为密度指标,其余 n-1 个文本的密度指标可有下面公式进行修正，其中常数 r_b 通常是大于 r_a,为了避免聚类中心点之间的距离太近，通常一般取：r_b= 1.5 r_a。在本论文中认为选取 r_b= 1.25 r_a 比较合适。

$$D_i = D_i - D_{ck} \exp\left(\frac{-\|T_i - T_j\|^2}{(0.5 r_b)^2}\right) \quad (13)$$

再找出最高的作为第 2 个聚类中心 T_{c2}，依此类推，一直找到 q 个聚类中心，这里的 q 是根据每个文本的各维对选取中心影响的大小自己确定的，不需要事先确定。

距离已选定的聚类中心越近的文本经过这样的修正后，密度指标值迅速下降。当 D_i≤0 时，该文本 T_i 将再也不可能成为聚类中心点。这样，就达到了减少数据点的目的。

③ 初始化模糊分区矩阵 U(0)；

④ 对每一步用公式(12)计算 q 个中心 v i；

⑤ 更新划分矩阵 $U^{(b)}$：如果对于任意的 j，k 存在 $d_{jk}(b)>0$，那么得：

$$u_{ik}^{(b+1)} = \frac{1}{\sum_{j=1}^{q}\left(\frac{d_{jk}^{(b)}}{d_{jk}^{(b)}}\right)^{\frac{2}{(m-1)}}}$$

其中：$d_{jk}(b))$表示文本 T_k 与第 i 类中心向量 vi 之间的距离。

⑥ 若 $\|u^{(b+1)} - u^{(b)}\| \le \varepsilon$，则停止；否则，置 r =r+1 并返回步骤④。

3　实验结果与分析

在我们的试验中，选取了五类文章，其中 95 篇 A 类的文章，88 篇 B 类的文章，120 篇 C 类的文章，115 篇 D 类的文章，185 篇 E 类的文章。基于这样的样本集合，我们进行试验。

对这个文本集中的所有文本进行分词处理，词频统计，在进行过滤掉某些禁用词和稀有词之后，生成词频统计表。然后，通过采用基于词频的截词法，选取特征词表。为计算的速度和方便，在本文我们选取在词频表中出现频率为前 150 的，组成特征词表，这样对于每个文本,所生成的特征向量的维数就是 150。最后，根据特征词表中的词项，通过词频统计的方法利用公式(1)和公式(2)为每一个文本生成特征向量，并完成选取所有的样本集合。经过这样的预处理，将非结构化的文本最终表示成为一种结构化的形式，然后，运用算法将文本进行聚类。为了说明聚类的性能，采用 F_1 测度值作为聚类质量的评价指标：$F_1 = \frac{2pr}{p+r}$ 其中：p---表示准确率；r---表示查全率

表 1　两种算法的性能比较

算法	Fcm	优化后
F1 测度值	0.611	0.738

从表 1 我们可以看出，和传统的 FCM 比较，F_1 测度值从 0.611 提升到了 0.738 上升了 20.79 %，这是因为优化后的 FCM 应用到文本分类中，可以有效的收敛于全局最优，因此，聚类的性能比 FCM 更好。为了进一步能够说明该算法在文本聚类中应用的效果，我们分别用模糊 C-均值算法和减聚类优化的模

(下转第 209 页)

(上接第 174 页)

糊 C-均值算法应用到文本聚类中进行对比试验。试验结果如下:

表 2 两种算法在文本聚类中应用正确率的对比

类别	应有数目	正确聚类数		正确率	
		Fcm	优化后	Fcm	优化后
A	95	81	82	85.2	86.3
B	88	80	80	90.9	90.9
C	120	107	111	89.2	92.5
D	115	103	104	89.6	90.4
E	185	157	161	84.9	87.0

通过试验，我们可以从表 2 中看出，经过减聚类优化的模糊 C-均值算法在文本聚类结果的准确性上要高于模糊 C-均值算法。这是因为减聚类优化后的模糊 C-均值对初始的聚类中心不过于敏感，从不同的初值开始，都可以收敛于全局最优，因此，文本聚类的准确率要好。

参考文献

1 肖春景,张敏.基于减法聚类与模糊 c 均值的模糊聚类的研究.计算机工程, 2005,31:1－2.

2 姚清耘.基于向量空间模型的中文文本聚类方法的研究[硕士学位论文].上海:上海交通大学, 2008.

3 Dunn JC. A Fuzzy Relative of the ISODATA Processand Its Use in Detecting Compact Well-Scparatcd Cluster.J.Cybernct. 1974.1－8.

4 G Balland D H all. A Clustering Techniquc for Summarizing Multivariate Data, Bchav1. Sci.1967,12:153－155.

5 Chiu SL. Fuzzy model identification based on cluster estimation. Journal of Intelligent and Fuzzy Systems, 1994,2(3):54－58.