

基于 K-Means 的文本层次聚类算法研究

尉景辉,何丕廉,孙越恒

(天津大学 电子信息工程学院,天津 300072)

(Yjh1120@126.com)

摘要:提出了一种基于 K-Means 的文本层次聚类算法。它结合凝聚层次聚类和 K-Means 算法的特点,减少凝聚层次法在凝聚过程中的错误,提高了聚类质量。实验结果表明,该算法的聚类质量优于层次聚类法。

关键词:文本聚类;向量空间模型;K-Means 算法;层次聚类

中图分类号: TP391 **文献标识码:** A

Research on text hierarchical clustering algorithm based on K-Means

YU Jing-hui, HE Pi-lian, SUN Yue-heng

(School of Electronic Information Engineering, Tianjin University, Tianjin 300072, China)

Abstract: A new text hierarchical clustering algorithm based on K-Means was presented, which combined features from both K-Means and agglomerative approach that allowed them to reduce the early-stage errors made by agglomerative method and hence improved the quality of clustering solutions. The experimental evaluation shows that, our algorithm leads to better solutions than agglomerative methods.

Key words: text clustering; vector space model; K-Means; hierarchical clustering

凝聚层次聚类法和 K-Means 算法是目前应用较为广泛的文本聚类方法。凝聚层次聚类法可能达到较高的精度^[1~4],但时间复杂度较高,而 K-Means 则与之相反,具有较快的聚类速度,但是精度却较低。为此结合凝聚层次法和 K-Means 的特点,本文提出了一个新的基于 K-Means 的文本层次聚类算法。实验结果表明,该方法优于凝聚层次算法方法。且它的时间复杂度介于凝聚层次法和 K-Means 之间,有较好的实用性。

1 文本的表示

本文中使用的向量空间模型(Vector Space Model, VSM)^[5]来表示每个文本。在该模型中,每个文本 d 被表示为向量空间中的一个向量 d 。一般地,本文使用 $tf \times idf$ 来衡量文本中每个词的权重,因而一个文本可以表示为:

$$(tf_1 \log(n/df_1 + 0.01), tf_2 \log(n/df_2 + 0.01), \dots, tf_m \log(n/df_m + 0.01))$$

其中, tf_i 为词 t_i 在文本中出现的频率, df_i 为文本集中包含词 t_i 的文本数, n 为文本总数。为了减小不同长度的文本对于计算文本相似度的影响,每个文本向量都被归一化到单位长度(即 $\|d_{tfidf}\| = 1$)。给定文本集 A , 定义 A 的复合向量 $D_A = \sum_{d \in A} d$, A 的质心 $C_A = D_A / |A|$ 。

在向量空间模型中,一般使用余弦相似度来度量文本之间的相似程度:

$$\text{sim}(d_i, d_j) = \cos(d_i, d_j) = \frac{d_i \cdot d_j}{\|d_i\| \|d_j\|}$$

其中 \cdot 表示两个向量的点积, $\|d\|$ 表示向量 d 的长度。

由于文本向量的长度被归一化为单位向量,因而在计算文本相似度是这个公式可以简化为 $\cos(d_i, d_j) = d_i \cdot d_j$ 。

2 基于 K-Means 的层次聚类算法

2.1 K-Means 及凝聚层次法的一般描述

K-Means 方法是基于划分的聚类方法。其基本思想为:对于给定的聚类数目 K , 首先随机选择 K 个文本作初始的类质心,然后根据每个文本与各个类质心的相似度,将它赋给最相似的类。然后重新计算每个类的质心。不断迭代以上过程,直到准则函数收敛。在文本聚类中,一般采用的准则函数^[6]是:

$$\text{maximize} \sum_{r=1}^k \sum_{d \in S_r} \cos(d_i, C_r) = \sum_{r=1}^k \|D_r\|$$

而且文献[6]中提出采用动态改变各类质心(即在每次有新的文档加入一个类时,动态地修正类的质心,而不是在迭代的最后才修正)的方法可以大大提高 K-Means 算法的精度和效率。K-Means 算法效率很高,它的复杂度是 $O(nkt)$, 其中 n 是文本总数, k 是聚类数目, t 是迭代次数。

由于采用了启发式方法,在降低计算的复杂性,提高运算速度的同时,也使 K-Means 不一定会得到全局最优的结果,而通常因为初始值的影响,以局部最优结束。为了最大限度地消除这种影响,本文采用的方法是:重复运行算法 N 次,选择一个使准则函数最优的结果作为最终聚类结果。在本文的实验中,令 $N = 10$ 。

与 K-Means 所采取的划分策略不同,凝聚层次法是一种自底向上的方法。它首先将每个文档作为一个类(原子类),然后合并这些类成为较大的类,直到所有的文档都在一个类别中,或者用户指定的某个终止条件被满足。不同层次凝聚法的主要区别在于类间相似度的定义方法不同。一般来说,有三种常用的方法,即: single-link、complete-link 和 UPGMA(也被称为组平均 Group Average)^[4]。对于两个类 S_i 和 S_j , 它们间的相似度分别定义为:

$$\text{sim}_{\text{single-link}}(S_i, S_j) = \max_{d_i \in S_i, d_j \in S_j} \{\cos(d_i, d_j)\}$$

收稿日期:2005-04-19;修订日期:2005-06-26

作者简介:尉景辉(1982-),男,山西襄汾人,硕士研究生,主要研究方向:信息检索、文本聚类;何丕廉(1942-),男,博士生导师,主要研究方向:自然语言处理、多媒体教学系统、地理信息系统;孙越恒(1974-),男,山东烟台人,博士研究生,主要研究方向:自然语言处理。

$$\text{sim}_{\text{complete-link}}(S_i, S_j) = \max_{d_i \in S_i, d_j \in S_j} \{\cos(d_i, d_j)\}$$

$$\begin{aligned} \text{sim}_{\text{UPGMA}}(S_i, S_j) &= \frac{1}{n_i n_j} \sum_{d_i \in S_i, d_j \in S_j} \cos(d_i, d_j) \\ &= \frac{D_i \cdot D_j}{n_i n_j} \end{aligned}$$

由于 single-link 只依据极少的信息, complete-link 则依赖于类内文本之间有着非常高的相似度这样一个假定, 因而 single-link 和 complete-link 在实际使用的效果较差。而 UPGMA, 它采用的度量两个子类内文本的两两相似度的均值来确定合并的子类, 可以较好的解决上述两种方法的问题。所以在本文的实验中, 就使用 UPGMA 方法。本文所指的凝聚层次法, 就是采用 UPGMA 的凝聚层次法。它的时间复杂度为 $O(n^2 \log n)$ 。

2.2 基于 K-Means 的层次文本聚类算法

对于 K-Means 这样的划分的聚类算法, 它的优点是当使用算法进行聚类的时候, 可以利用整个文本集的全局信息。但是凝聚层次法要聚类的时候, 则更多的利用的是文本集的局部特征。使用文本集的局部特征既有优点也有缺点。优点是人们可以很方便地将文本聚合成一个较小且足够内聚的类; 而 K-Means 之类的划分算法要想完成这项任务, 则可能因为在划分聚类过程中算法越过类的界限使这些文本分散而失败。缺点是, 如果一旦某一步做出了错误的合并决定, 由于凝聚层次法每步所做的处理不能被撤消, 类与类之间也不能交换对象, 因而这些错误会在以后的凝聚过程叠加, 会导致低质量的聚类结果。这样的情形在有大量相当的替代选择时候尤为明显。

基于以上的考虑, 结合传统凝聚层次聚类算法和 K-Means 算法的思想, 我们提出了一个基于 K-Means 的层次聚类算法, 即: 使用 K-Means 方法所产生的类来约束凝聚层次法的凝聚空间, 也就是说, 仅仅允许约束空间 (即 K-Means 算法所产生的类) 内的文档凝聚在一起。算法流程如下:

1) 使用上面所介绍的 K-Means 算法生成 K 个类, 我们称这 K 个类为约束类;

2) 对每一个约束类, 将其看作一个文本集, 应用凝聚层次聚类法生成一颗聚类树;

3) 将这 K 颗聚类树看作凝聚过程所产生的中间类, 再用凝聚层次聚类法将这 K 棵树合并为一颗完整的聚类树。

该算法的优点在于, 它既可以因使用 K-Means 算法而从文本集的全局特征得益, 又可以从凝聚层次法所使用的局部特征得益。此外, 本算法的计算复杂度为 $O(k(n/k)^2 \log(n/k) + k^2 \log k)$, 其中 k 是约束类的数目。如果 k 足够大的话, 比如取 k 为 \sqrt{n} , 则凝聚层次法的 $O(n^2 \log n)$ 的时间复杂度就会减为 $O(n^{2/3} \log n)$, 聚类效率会大大提高。

3 实验结果及评价

3.1 文本聚类的评价方法

我们使用 F-测量值 (F-Measure)^[6] 来评价聚类结果的质量。给定一个有 n_l 个文本的真实类 L 及一个有 n_s 个文本的聚成类 S , 假定 S 中包含 L 中的 n_b 个文本, 定义 L 和 S 的 F-值为:

$$F(L, S) = \frac{2 * R(L, S) * P(L, S)}{R(L, S) + P(L, S)}$$

其中, $R(L, S) = n_b / n_l$ 为查全率 (Recall), $P(L, S) = n_b / n_s$ 为准确率 (Precision)。L 的 F-测量值就是在聚类层次树 T 中 L 所得到最大的 F-值, 即:

$$FMeasure(L) = \max_{S \in T} F(L, S)$$

整个聚类层次树的 F-测量值定义为单个类的 F-测量值之和:

$$FMeasure = \sum_{L \in C} \frac{n_l}{n} FMeasure(L)$$

C 为真实类别的集合。完美的聚类结果是对于每一个真实类在所生成的层次树都有一个节点与之对应, 在这种情况下, F-测量值等于 1。一般说来, 如果 F-测量值越大, 聚类结果的质量越好。

3.2 实验结果及其评价

对实验所采用的数据集, 我们去停用词, 进行词干抽取后, 并且去除那些在不少于两个文本中出现的词后, 进行词频统计并处理。各个数据集的细节如表 1 所示。

表 1 数据集细节

| 文本集 | 来源 | 文本数 | 类数 | 词数 |
|-----|----------------|-------|----|--------|
| la1 | LA Times(TREC) | 3 204 | 6 | 21 604 |
| re0 | Reuters-21578 | 1 504 | 13 | 2 886 |
| re1 | Reuters-21578 | 1 657 | 25 | 3 758 |
| Wap | WebAce | 1 560 | 20 | 8 460 |

其中 la1 来源于用于 TREC-5 的 Los Angeles Times 的文章, re0、re1 来源于 Reuters-21578 数据集^[7], Wap 则来自 WebACE 项目^[8]。

在实验中, 分别实现了本文所提出的算法及凝聚层次法, 并分别取约束类的数目为: 20、40、 $n/20$ 、 \sqrt{n} , 运行本文的算法, 所得结果如表 2 所示, 其中加下划线的列表示该数据集上算法所取得的最好结果。

表 2 层次凝聚法同本文算法 F-测量值的比较

| 数据集 | 层次凝聚法 | 20 | 40 | \sqrt{n} | $n/20$ |
|-----|-------|--------------|-------|------------|--------------|
| La1 | 0.654 | <u>0.806</u> | 0.798 | 0.783 | 0.768 |
| Re0 | 0.584 | 0.629 | 0.632 | 0.635 | <u>0.645</u> |
| Re1 | 0.695 | 0.731 | 0.724 | 0.719 | <u>0.739</u> |
| Wap | 0.640 | 0.658 | 0.672 | 0.696 | <u>0.700</u> |

从上面的结果可以看出, 即使约束类的数目很小的时, 本文的方法较单纯的凝聚层次法也有了极大地改进。

参考文献:

- [1] GALE LD. A sequential algorithm for training text classifiers [J]. In Proceedings of ACM SIGIR Conference, 1994.
- [2] CRAVEN M, FREITAG D, *et al.* Learning to extract symbolic knowledge from the World Wide Web. Technical Report[R], School of Computer Science, CMU, 1998.
- [3] PAZZANI MJ, MURAMATSU J, *et al.* Syskill and Webert: Identifying interesting Web sites [J]. In AAAI-96. 1996.
- [4] DUBES RC, JAIN AK. Algorithms for Clustering Data [M]. Prentice Hall, 1988.
- [5] SALTON G, WONG A, YANG CS. A Vector Space Model for Automatic Indexing [J]. Communication of the ACM, 1975, 18(5): 613-620.
- [6] LARSEN B, AONE C. Fast and effective text mining using linear-time document clustering [A]. In Proc. of the Fifth ACM SIGKDD Int'l Conference on Knowledge Discovery and Data Mining, 1999. 16-22.
- [7] LEWIS DD. Reuters-21578 text categorization text collection 1. 0 [DB/OL]. <http://www.daviddlewis.com/resources/testcollections/reuters21578/>
- [8] HAN S, BOLEY D, GINI D, *et al.* WebAce: A Web Agent for Document Categorization and Exploration [J]. Proceedings of the 2nd International Conference on Autonomous Agents (Agents'98).