基于核函数的改进 k-means 文本聚类

张国锋 吴国文
(东华大学计算机科学与技术学院 上海 200050)

摘 要 通过对传统 k-means 算法优缺点的研究分析,提出一种改进的 k-means 聚类算法。随机初始化 $k/2$ 个簇心,划分最大的簇并删除空簇,在更新簇心的同时判断簇心位置的合理性;及时对簇心做出修改,使得最后聚类出的 k 个簇中不会出现空簇;使用高斯核函数作为测量向量之间距离的方法,提高聚类的准确性。基于此改进的 k-means 算法,使用在不同网站上采集的文章作为数据源,并利用 TF-IDF 以及 Word2Vec 技术对文本进行向量化处理,进而完成对文本的聚类任务。与传统的 k-means 文本聚类相比,不仅提高了聚类的准确性,而且改善了传统 k-means 算法结果可能会出现空簇的缺陷。

关键词 k-means 高斯核函数 TF-IDF 文本聚类

中图分类号 TP391.1 **文献标识码** A DOI:10.3969/j.issn.1000-386x.2019.09.049

IMPROVED K-MEANS TEXT CLUSTERING BASED ON KERNEL FUNCTION

Zhang Guofeng Wu Guowen
(*College of Computer Science and Technology, Donghua University, Shanghai 200050, China*)

Abstract Through the research and analysis of the advantages and disadvantages of the traditional k-means algorithm, we proposed an improved k-means clustering algorithm. We randomly initialized $k/2$ cluster cores, and divided the largest cluster and deleted the empty clusters. The cluster core was updated to determine the rationality of the cluster center position. The cluster core was modified in time to make the empty clusters would not appear in the last k clusters. The Gaussian kernel function was used as the method to measure the distance between vectors, which greatly improved the accuracy of clustering. Based on this improved k-means algorithm, articles collected on different websites were used as data sources, and we used TF-IDF and Word2Vec technologies to preprocess the text, and completed the task of clustering text. Compared with traditional k-means text clustering, it not only improves the accuracy, but also corrects the defect of empty clusters in the results of traditional k-means algorithm.

Keywords k-means Gaussian kernel function TF-IDF Text clustering

0 引 言

在计算机技术全面发展的当代,人工智能在生活当中的作用越来越重要,应用的范围也十分广泛,各行各业都对人工智能进行了大量的钻研。人们都希望在有限的时间内做足够多的事,这其中就包括阅读。就像在图书馆中阅读一样,人们在手机以及计算机上查阅资料或者阅读文章时,希望能够大大减少寻找同类型文章的时间,希望这些文章能够存放在一起而不是错综复杂的排列。但如果人工对文章进行分类需要花费大量的时间和精力,因此,让计算机来为人们提供最便捷的服务是大势所趋,机器学习中的聚类算法就得到了用武之地。

聚类算法属于“无监督学习”,而且是其中被人们研究、使用最多的算法。在聚类分析之前,每一个数据或样本属性的归类是不确定的,属性能被分成多少类一般也是需要预测的,只能依靠元数据进行分析,不像分类算法可以参考相关类别的信息。聚类分析方法主要在探索研究方面应用较多,最终的结果可能包含多种有价值的答案,如何进行筛选要依靠研究人员的实际需求和具体分析。无论实际数据能否真地被分成不

收稿日期:2019-01-13。**张国锋**,硕士生,主研领域:自然语言处理。**吴国文**,副教授。

同种类,使用聚类分析都可以将数据划分成特定数量的类别。聚类可以单独使用来获取数据的具体分布情况,通过研究聚类出的各个簇中数据的特征,找出特征显著的簇进行更加具体详细的分析。

1 文本预处理

若要对文章进行聚类,需要对文章进行一定的处理,这些操作包括对文章进行分词、去除停用词、使用 TF-IDF 找出每篇文章的关键词以及使用 Word2Vec 将关键词向量化。

1.1 分词处理

首先使用 jieba 算法对文章进行分词。jieba 分词有三种模式:精确模式、全模式以及搜索引擎模式。这里使用精确模式,此模式的目的是把语句最精确地切分开,适用于文本分析,分词之后的文本存入 txt 文件中。

其次需要去除停用词。分词之后的文本现在以若干词语集合的形式呈现。其中很多字词并没有实际的意义,例如"的"、"是"等,这些词会影响之后提取关键字的准确性,因此需要把这些没有实际意义的字词除去。由此构造了一个停用词字典,对词语集合进行筛选,若集合中的字词出现在停用词字典中,则删除。

1.2 特征选取

本文使用 TF-IDF 算法找出文本中的关键词,将权值最大的 20 个关键词作为特征代表文本进行聚类。TF-IDF 的原理可以通俗易懂的解释为:如果一个词语或者短语在某篇文章中以很高的频率出现,然而在其他的文章中几乎没有,那么就认为这个词语或者短语具有良好的代表性,适合用来做区分。具体计算公式如下:

$$TF\text{-}IDF = \frac{T_i}{N_t} \times \log\left(\frac{D_n}{D_{t+1}}\right) \tag{1}$$

式中:i 代表文本中的词语;T_i 表示该词语出现的次数;N_t 表示文章的总词数;D_n 表示文本总数;D_t 表示包含词语 i 的文本数。

返回的结果是一个列表和一个矩阵。列表中存放的是所有文本的词语汇总,每个词语只存储一次。矩阵每行存储的是每个词语在一个文本中的权值数据,排列顺序与列表中词语的排列顺序一致,若某个词并未出现在某个文本中,则权值为 0。

1.3 文本向量化

最后的处理工作是把文本向量化。把分词后的所有文本当作语料库,使用 Word2Vec 模型进行词向量化。Word2Vec 根据词义把词语映射到距离接近的空间中,词向量能够表达出一定的语义信息。此次选择 CBOW 训练模式,这种模式通过前后文预测目标词。属性 windows 意思是目标词与预测词的距离,此次大小设为 5,通过目标词前后文共 10 个词得到当前词的向量,维度 size 设定为 20。在得到的向量库中匹配出各个特征的向量 $\boldsymbol{V}_i$,将特征向量相加得到最终的文本向量 $\boldsymbol{V}_d$:

$$\boldsymbol{V}_d = \sum_{i=1}^{20} \boldsymbol{V}_i (len\boldsymbol{V}_i = 20) \tag{2}$$

这样就可以使用向量 $\boldsymbol{V}_d$ 来进行聚类工作。

2 相似度定义

本文使用高斯核函数计算文本之间的相似度。高斯核函数的公式如下:

$$k(x,y) = \exp\left(-\frac{\|x-y\|^2}{2\sigma^2}\right) \tag{3}$$

一种等价且更为简单的定义公式为:

$$k(x,y) = \exp(-\gamma\|x-y\|^2) \tag{4}$$

式中:$\gamma = 1/2\sigma^2$。

高斯核函数对于数据中的噪音有着很不错的抗干扰能力,函数中的 σ 参数决定了函数的有效区域,超过了此范围,数据的影响就会基本忽略。由于噪音对 k-means 算法的影响很大,所以使用高斯核函数来降低噪音的影响。此外,高斯核函数能够利用高维空间向量之间的内积得出两个点之间的距离,降低了计算难度。

高斯核函数对自身的参数 σ 比较敏感,本次实验通过交叉验证法确定参数 σ。使用距离协方差作为参数,因为协方差能很好地反映各维数据的离散状况,很符合核函数参数的性质。协方差公式如下:

$$S^2 = \frac{\sum_{i=1}^{n}(x_i - \hat{x})^2}{n-1} \tag{5}$$

其中:n 代表数据的总数;x_i 为第 i 个具体的数据。

从原始数据中随机选择 400 个数据,平均分为 4 组,其中三组记为 A、B、C,当作训练集,最后一组记为 D,作为最终的验证集,用验证集来选出最合适的一个当作参数。

具体做法是:使用传统 k-means 算法对三个训练集进行聚类,使用欧氏距离作为距离公式,k 值定为 3。每组聚类后会有 3 个簇,把各簇误差的协方差(精确到小数点后五位)分别记为 $s1$、$s2$、$s3$,把它们的平均值记作 S,这里的误差是指当前点到簇质心的距离。随后将它们分别代入高斯核函数中,使用测试集 D 的数据

进行聚类。测试集使用误差平方和(SSE)来评估聚类效果,SSE 的值越小,表示数据点离它们的质心越近,聚类效果也就越好。测试集每个簇的误差平方和分别用 $d1$、$d2$、$d3$ 表示,从中选出平均误差平方和最小的一组,将 S 值作为 σ。训练集具体数据如表 1 所示。

表 1　训练集数据

协方差	A	B	C
$s1$	0.002 15	0.001 49	0.003 45
$s2$	0.002 37	0.001 20	0.001 34
$s3$	0.001 69	0.003 23	0.001 73
S	0.002 07	0.001 97	0.002 17

测试集数据如表 2 所示。

表 2　测试集结果

性能指标	$s1$	$s2$	$s3$
$d1$	39.999 996 7	13.999 996 7	16.999 994 9
$d2$	45.999 995 6	49.999 995 6	35.999 996 9
$d3$	13.999 996 4	35.999 997 4	46.999 995 7
d	33.333 329 6	33.333 329 9	33.333 329 2

由测试集数据可了解到,当 S 为 0.002 17 的时候,效果最好,这表明,可以确定 σ 取值为 0.002 17。

3　k-means 算法改进

3.1　传统 k-means 算法

k-means 是划分方法中较经典的聚类算法之一。由于该算法的效率高,所以普遍应用于对大规模数据进行聚类。目前,许多算法均围绕着该算法进行扩展和改进。

k-means 算法的逻辑如下:确定聚类的个数 k,随机确定初始质心的坐标,选择合适的距离公式计算每个数据与每个质心的距离,并将其聚类到距离最近的簇中。在所有数据完成聚类后,更新每个簇的质心坐标并重新计算每个点与质心的距离,将数据点重新聚类到距离最近的簇中。重复以上步骤直到质心不再变化,聚类完成。

k-means 算法的优点是简单、快速,当数据很密集时,效果较好。缺点是要事先确定准备生成的簇的数目 k,对于初始质心坐标和噪声很敏感,不同的初始值结果可能会不一样,当 k 值预估过大时,可能出现空簇。

3.2　使用改进的 k-means 算法聚类

由于初始质心的随机性对 k-means 的结果影响很大,数据很可能收敛到局部最小值,并且会产生空簇,所以此次实验对传统 k-means 做出了改进。

用 k'表示距离指定 k 值的差,Δk 表示当前将要增加的质心数。改进后的算法先随机生成 $k/2$ 个初始质心,初始 $k' = k - k/2$。将所有数据聚类到 $k/2$ 个簇中,如果有 a 个簇中没有数据,则直接删除这些质心并更新 $k' = k' + a$。每次增加 $\Delta k = k'/2$ 个质心,利用误差平方和来判断聚类过程中的效果,找出聚类过程中 SSE 值最大的 Δk 个簇分别进行 k 为 2 的局部聚类,将原先的簇划分成两个,再重新计算每个簇的 SSE,重复以上步骤,直到簇的数目达到 k 为止。

具体步骤为:

(1) 初始化 $k/2$ 个质心,质心坐标随机确定:

$$k' = k - \frac{k}{2} \tag{6}$$

(2) 将所有的数据聚类到这些簇中,判断是否有空簇并记录个数 a,若有即刻删除,并更新 k'和 Δk:

$$k' = k' + a \tag{7}$$

$$\Delta k = k'/2 \tag{8}$$

(3) 比较每个簇的 SSE 值,找出值最大的 Δk 个簇,进行局部二分聚类。

(4) 更新 k'以及 Δk 的值并重复第 3 步的操作:

$$k' = k' - \Delta k \tag{9}$$

(5) 当簇的数量达到 k 即 k'为 0 时,聚类结束。

经过这一改进后,可以有效降低 SSE 的值,使数据最大化地收敛到全局最小值,还可以避免出现空簇,改善了有效簇未达到期望值的缺陷。

4　结果分析

4.1　实验一

为了对比,分别使用传统 k-means 算法和改进后的 k-means 算法做了两组聚类对比实验。选取的文本字数在 500 到 1 500 字之间,具有普遍性和一般性。数据文本共选取 2 000 篇,分为三组,第一组 300 篇,第二组 700 篇,第三组 1 000 篇。

由于计算距离公式不同,所以使用平均误差平方和(AvgSSE)与最大误差平方和(MaxSSE)的比值(pSSE)作为评估标准之一,计算公式如下:

$$pSSE = \frac{\mathrm{Avg}(\sum_{i=1}^{k} SSE_i)}{\mathrm{Max}(SSE)} \tag{10}$$

比值保留到小数点后 5 位。结果分别从聚类的迭

代次数(t)、pSSE 以及空簇的数量(Ne)进行对比。结果如表 3 所示。

表 3 实验一结果对比

数据属性	k	传统 k-means			改进 k-means		
		t	$pSSE$	Ne	t	$pSSE$	Ne
300	5	456	0.526 32	0	389	0.622 41	0
700	10	1 139	0.430 86	1	892	0.544 83	0
	15	1 151	0.404 75	5	1 072	0.463 45	0
1 000	15	1 948	0.370 37	4	1 734	0.473 69	0
	20	2 107	0.364 67	7	1 964	0.501 28	0

由最后结果对比可知,在迭代次数上,改进后的算法较传统算法所用次数明显减少。原因有两点:第一是初始的质心只为 k 值的一半,基础的迭代次数必然减少;第二是因为算法后期的局部聚类相当于二分聚类,每次增长的幅度相对较小。

在 $pSSE$ 方面,改进后的算法要大于传统算法。$pSSE$ 较小说明最大的误差平方和相对较大,传统的聚类方法得出的簇很可能出现某一簇的范围很大而其他簇的位置很集中,这就说明聚类效果不是很好。而改进后的算法聚类出的每个簇的数据更集中,平均每个数据距离质心更近,从而说明改进后的算法聚类效果要优于传统算法。

此外,较大数据量的两组分别聚类了两次进行对比。可以看到改进后的算法并没有出现空簇,而传统算法虽数据量没有变,但是由于初始的质心发生变化,导致出现的空簇数量不定,或多或少,随着文本的增多,出现空簇的可能性也增大。改进后的算法彻底修正了传统算法的这个缺陷,保证了聚类结果能达到数量上的基本要求。

4.2 实验二

实验二选取了金融类、汽车类、体育类、天气类以及食品类文本各 100 篇混合成源数据进行聚类。分别从准确率(precision)、召回率(recall)以及 F 值进行对比。准确率是指聚类后,各类文本数量 N_t 与该簇全部文本数量 N_T 的比值。公式为:

$$precision = N_t / N_T \tag{11}$$

召回率是指 N_t 占相对应类型文本 N_m 的比值:

$$recall = N_t / N_m \tag{12}$$

F 值计算公式为:

$$F = \frac{2 \times precision \times recall}{precision + recall} \tag{13}$$

实验结果如表 4 所示。

表 4 实验二结果对比 %

算法	文本类型	准确率	召回率	F 值
传统 k-means	金融类	49.49	49.00	49.25
	汽车类	56.67	51.00	53.68
	体育类	54.55	48.00	51.06
	天气类	49.14	57.00	52.78
	食品类	49.53	53.00	51.21
改进 k-means	金融类	51.52	51.00	51.26
	汽车类	61.29	57.00	59.07
	体育类	53.54	53.00	53.27
	天气类	54.46	61.00	57.55
	食品类	57.73	56.00	56.85

由实验二数据可以看出,改进后的算法整体在准确率和召回率上都有了明显提升,F 值也因此提高。只有食品类准确率少许降低,原因可能是食品类文本中的词语与其他类重复得过多,但召回率的提高使得 F 值并没有降低。由此从整体上来看,改进后算法的聚类效果比传统算法的聚类效果要好。

5 结 语

本文结合 TF-IDF 与 Word2Vec 对文本实现向量化,并针对传统 k-means 算法易收敛到局部最小值,对噪声敏感等不足之处做出改进,以高斯核函数作为距离公式,并在聚类过程中降低误差平方和,提升聚类效果,对文本进行了高效聚类。实验结果表明,本文算法在效果上优于传统 k-means 聚类,并消除了聚类结果出现空簇的可能性。

参 考 文 献

[1] Harrington P. 机器学习实战[M] 李锐, 李鹏, 曲亚东, 等译. 人民邮电出版社, 2013.

[2] 周丽杰,于伟海,郭成. 基于改进的 TF-IDF 方法的文本相似度算法研究[J]. 泰山学院学报,2015,37(3):18 - 22.

[3] 徐金宝. 核函数在划分聚类中的应用与实现[J]. 电脑知识与技术, 2013,9(27):6185 - 6188.

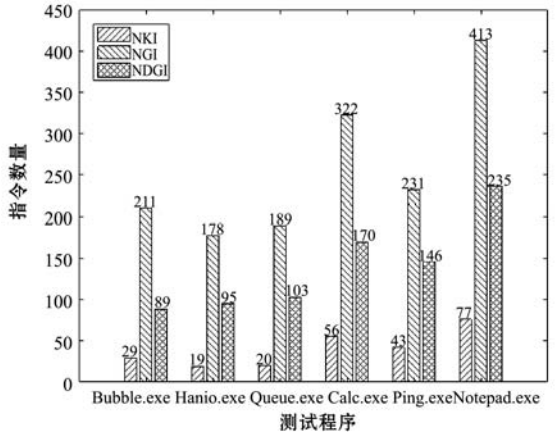
[4] 陈磊磊. 不同距离测度的 K-means 文本聚类研究[J]. 软件, 2015,36(1):56 - 61.

[5] 索红光, 王玉伟. 一种用于文本聚类的改进 K-means 算法[J]. 山东大学学报:理学版, 2008, 43(1):60 - 64.

[6] 陈雅芳. 中文文本分类方法研究[D]. 杭州:浙江大学, 2010.

[7] 张建萍, 刘希玉. 基于聚类分析的 K-means 算法研究及应用[J]. 计算机应用研究, 2007, 24(5):166 - 168.

(下转第 301 页)



(b)

图 10　混淆前后代码块数量和指令数量对比

由表 5 的实验结果可以看到,*RB* 的取值范围为 14.4 ~ 18.4,平均值为 16.9,意味着攻击者需要分析的混淆后代码块数量是混淆前的 16.9 倍。*RI* 的取值范围为 5.36 ~ 9.45,平均值为 7.09,意味着攻击者需要分析的混淆后指令数量是混淆前的 7.09 倍。*RD* 的取值范围为 0.42 ~ 0.63,平均值为 0.54,意味着通过静态反汇编只能正确得到 54% 的混淆后 gadget 指令。实验结果证明,混淆方法能够有效增加静动态获取和分析核心代码的难度,在一定程度实现代码隐藏的保护效果。

6　结　语

本文针对程序核心代码容易暴露给逆向攻击的问题,提出一种基于 ROP 技术的代码混淆方法。方法借鉴 ROP 攻击技术的代码组织和调用方式,通过在栈空间预设指令地址和数据,利用程序内存空间随机分布的 gadget 指令序列动态组合执行,实现目标代码的等价功能。同时方法采取搜索和构造两种方法获取混淆所需要的 gadget 指令序列,进一步实现执行代码和路径的随机多样化。实验和分析证明,方法能够有效增加攻击者获取和分析核心代码的难度,在静态和动态两方面实现较好的保护效果,具有较好的时间和空间开销性能。

参　考　文　献

[1] Collberg C, Thomborson C, Low D. A taxonomy of obfuscating transformations[R]. Department of Computer Science, The University of Auckland, New Zealand, 1997.

[2] Kulkarni A, Metta R. A new code obfuscation scheme for software protection[C]//Proceedings of the 2014 IEEE 8th International Symposium on Service Oriented System Engineering. IEEE, 2014: 409 - 414.

[3] Xie X, Liu F, Lu B, et al. Mixed obfuscation of overlapping instruction and self-modify code based on hyper-chaotic opaque predicates[C]//2014 Tenth International Conference on Computational Intelligence and Security. IEEE, 2014: 524 - 528.

[4] Behera C K, Bhaskari D L. Self-Modifying Code: A Provable Technique for Enhancing Program Obfuscation[J]. International Journal of Secure Software Engineering (IJSSE), 2017, 8(3): 24 - 41.

[5] 陈喆, 贾春福, 宗楠, 等. 随机森林在程序分支混淆中的应用[J]. 电子学报, 2018, 46(10): 2458 - 2466.

[6] 苏庆, 孙金田. 基于混沌不透明表达式的不透明谓词混淆技术研究[J]. 计算机科学, 2017,44 (12): 114 - 119.

[7] Falcarin P, Di Carlo S, Cabutto A, et al. Exploiting code mobility for dynamic binary obfuscation[C]//2011 World Congress on Internet Security (WorldCIS-2011). IEEE, 2011: 114 - 120.

[8] Ma H, Lu K, Ma X, et al. Software Watermarking using Return-Oriented Programming[C]//Proceedings of the 10th ACM Symposium on Information, Computer and Communications Security. ACM, 2015: 369 - 380.

[9] Mu D, Guo J, Ding W, et al. ROPOB: Obfuscating Binary Code via Return Oriented Programming[C]//International Conference on Security and Privacy in Communication Systems. Springer, Cham, 2017: 721 - 737.

[10] Lu K, Xiong S, Gao D. Ropsteg: program steganography with return oriented programming[C]//Proceedings of the 4th ACM conference on Data and application security and privacy. ACM, 2014: 265 - 272.

(上接第 284 页)

[8] 张建辉. K-means 聚类算法研究及应用[D]. 武汉:武汉理工大学, 2007.

[9] 周世兵, 徐振源, 唐旭清. K-means 算法最佳聚类数确定方法[J]. 计算机应用, 2010, 30(8):1995 - 1998.

[10] 吴夙慧, 成颖, 郑彦宁, 等. K-means 算法研究综述[J]. 现代图书情报技术, 2011, 27(5):28 - 35.

(上接第 292 页)

[18] Preeja V, Shahana A H. A binary Krill Herd approach based feature selection for high dimensional data[C]//International Conference on Inventive Computation Technologies. 2017.

[19] Algamal Z Y, Lee M H. A two-stage sparse logistic regression for optimal gene selection in high-dimensional microarray data classification[J]. Advances in Data Analysis & Classification, 2018(4):1 - 19.