·人工智能专栏·

半监督语义动态文本聚类算法

钱志森[1,2]，黄瑞章[1,2]，魏 琴[2*]，秦永彬[1,2]，陈艳平[1,2]

(1. 贵州大学计算机科学与技术学院 贵阳 550025；2. 贵州大学贵州省公共大数据重点实验室 贵阳 550025)

【摘要】针对传统的动态文本聚类将描述方式不同的同类文本划分到不同组中；以及聚类类别个数与真实类别数之间差距明显等问题，该文提出了一种半监督语义动态文本聚类算法(SDCS)。该算法以语义表征文本的方式来捕获文本间的语义关系，在聚类过程中动态学习类别语义，让文本能根据语义准确聚类。同时该算法利用半监督聚类的方法对新类的产生进行监督，学习符合实际情况的聚类结果。实验结果表明该文提出的算法是有效可行的。

关 键 词 动态文本聚类；语义学习；半监督文本聚类；文本聚类

中图分类号 TP391.1 **文献标志码** A doi:10.3969/j.issn.1001-0548.2019.06.001

Semi-Supervised Semantic Dynamic Text Clustering Algorithm

QIAN Zhi-sen[1,2], HUANG Rui-zhang[1,2], WEI Qin[2*], QIN Yong-bin[1,2], and CHEN Yan-ping[1,2]

(1. School of Computer Science and Technology, Guizhou University Guiyang 550025;
2. Public Big Data Laboratory of Guizhou, Guizhou University Guiyang 550025)

Abstract In the traditional dynamic text clustering, the similar texts with different descriptions are divided into different groups; and the difference between the number of cluster categories and the number of real categories is obvious. Aiming at these problems, this paper proposes a semi-supervised semantic dynamic text clustering algorithm (SDCS). The algorithm captures the semantic relationship between texts by semantically representing the text, and dynamically learns the category semantics during the clustering process, so that the text can be accurately clustered according to semantics. At the same time, the algorithm uses the semi-supervised clustering algorithm to supervise the generation of new classes, and produces clustering results that are consistent with the actual situation. The experimental results show that the proposed algorithm is effective and feasible.

Key words dynamic text clustering; semantic learning; semi-supervised text clustering; text clustering

随着互联网的快速发展和移动设备的广泛使用，知乎、微博等应用平台产生了大量的数据流。这些数据具有传播速度快、随时间动态变化等特点。

不同于传统静态数据，这些数据流无法用传统的聚类算法直接处理，因此研究人员提出了动态文本聚类算法。动态文本聚类算法使用了在线和离线的两阶段处理方式，在在线阶段采用特定的数据结构来概述持续到来的数据，使用k-means等传统聚类算法为基准算法在离线阶段进行聚类。这种分而治之的处理模式成功解决了数据流的聚类问题。这些动态文本聚类算法多在假设文本特征的独立性而缺乏对语义的识别，使用组平均值为类心进行聚类，这种处理模式无法处理新到来的词典外的词特征，也无法解决同一事件不同风格的表达方式的问题，且无法识别事件、已有事件和新增文本间的关系。所以当数据流到来的时候文本会基于词共现特征去到错误的分组，无法利用语义信息进行正确聚类。同时，文本的聚类个数应随着数据流的发展而产生变化，这些传统的动态文本聚类算法普遍依赖于人为设定或自动学习的类别个数，导致无法准确生成符合实际情况的聚类个数，常常产生高于实际类别的聚类结果。针对这些问题，本文提出了一种半监督语义动态文本聚类算法(SDCS)。该算法融入文本的语义信息来表示文本，解决了传统动态文本聚类

收稿日期：2019－07－24；修回日期：2019－10－19

基金项目：国家自然科学基金联合基金重点项目(U1836205)；黔科合重大专项字([2018]3002)；国家自然科学基金重大研究计划(91746116)；贵州省重大应用基础研究项目(黔科合JZ字[2014]2001)；贵州省科技重大专项计划(黔科合重大专项字[2017]3002)；贵州省自然科学基金(黔科合基础[2018]1035)

作者简介：钱志森(1995－)，男，主要从事数据挖掘与机器学习方面的研究.

通讯作者：魏琴，E-mail：weiq@gzu.edu.cn

中因新词特征的出现而导致的文本表示困难问题，更好地捕获了文本间的关系；并且针对现有算法多聚焦于词级别的语义表示，缺乏对类别语义的描述提出在聚类过程中动态的学习类别语义，让只是因为描述，方式不同的同类文本能根据语义准确聚类。此外，该算法加入了半监督聚类的方法引入监督信息，利用这些监督信息对新类的产生进行监督，生成更符合实际情况的聚类结果。

1 相关工作

近年来研究人员提出了许多数据流聚类算法，大致可以分为基于层次、基于分区、基于密度及基于模型4类。

对于层次聚类算法，一旦决定组合两个簇就无法再撤消。经典处理流的算法birch[1]引入了簇特征向量CF[1]和高度平衡树Cftree[1]的概念，根据人工阈值把数据分配到树再合并或分离树。ODAC算法[2]以自顶向下的策略来维护簇的树状层次结构。适应流速度的算法CluStree[3]为微簇分配新数据的过程会随着流速度的变化而变化且自动调整微簇大小，可以独立于流速度在内存中保持相同数量的簇。文献[4]提出基于概率C-Means和高斯混合的一种参数增量更新的EROLSC聚类算法，能识别调整簇并检测异常数据。

基于分区的算法必须在开始前指定聚类的数量并对噪声和异常值敏感且都易于形成球状的簇，其中STREAM[5]使用k-means获得代表点并再次使用获得最终的聚类。之后基于k-means++[6]提出了著名StreamKM++[7]，它在内存中构造并维护一个表示数据流的核心集，之后再使用k-means++对其进行聚类。文献[8]提出了经典的两阶段CluStream算法于在线阶段以金字塔时间框架构建维护微簇，在离线阶段再聚类微簇，为用户在给定的时间范围提供过去微簇的信息。文献[9]提出基于边界点检测算法的BPIC算法，通过边界轮廓的识别来表示聚类结果。

而基于密度的算法则可以发现任意形状的簇并处理噪声，更细的密度网格算法将对象空间划分为有限数量的网格或单元格，然后对非空的网格执行聚类操作，能有效降低高维数据的计算复杂度。文献[10]提出DenStream，能有效支持任意形状簇并处理异常值，使用了衰减函数且不固定微簇数量。基于该算法，文献[11]提出了改进算法rDenStream，额外添加了一个回溯的第三阶段，让算法可以重新学习以前丢弃的数据，为被误判的簇提供了一个形成潜在微簇的机会。DStream[12]是基于网格的流聚类算法，在在线阶段将输入映射到一个网格中，而离线阶段尝试对密集网格进行聚类并删除稀疏网格。文献[13]针对CFSFDP无法自适应识别簇心的缺点提出了一种基于Max-min算法的快速搜索和密度峰值自适应聚类的算法ACFSFDP，前者确定簇数，后者获得簇心。

最后，基于模型的算法的基本思想是一个簇内的对象在统计中具有相同的分布，通常聚类效果好，但计算复杂度高且不适用于具有大量簇和少量对象的数据集。2016年文献[15]在GSDMM[14]的基础上提出了FGSDMM+，能自动检测簇的数量，文档选择一个非空簇或从Dirichlet多项式混合模型导出潜在新簇并降低后续文档选择潜在簇的可能性，最后采用Gibbs采样算法以获得最终的聚类结果。文献[16]提出了算法LLDStrea，该算法提出一种在局部无监督的情况下执行LDA的ULLDA的降维技术，最后将输入点分配给投影空间中的微簇。

但是，这些算法都没有考虑文本数据的语义信息，并使用假设文本特征独立计算的簇心来聚类，不能处理随事件发展而出现的词典外的新词特征，也不能正确聚类只是因为描述方式不同的文本数据。并且这些算法都过度依赖人工阈值，产生的聚类个数与实际差别较大。基于此，本文提出了一种半监督语义动态文本聚类算法。

2 半监督动态聚类算法

对每个时间窗口中的数据，SDCS算法的聚类过程分为半监督动态文本聚类和类别语义学习两个部分。其中半监督动态文本聚类模块主要涉及文本语义的更新、簇增长的监控、类别语义的动态调整等，在类别语义学习模块主要涉及类文本的构建、类别语义学习和增强3部分。模型会在这两部分迭代循环来相互促进，并将训练结果沿用到下一次循环或下一个时间窗。具体聚类框架如图1所示。

2.1 半监督动态文本聚类

对应图1所示的半监督动态文本聚类模块，假设 t 时刻窗口获取到 N 个数据 $\boldsymbol{X}=\{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N\}$，元素 $\boldsymbol{x}_j(j=1,2,\cdots,N)$、类别 $\boldsymbol{y}_k(k=1,2,\cdots)$ 是一个 m 维的向量。利用word2vec[17]模型来做词特征的语义嵌入，使用文本中所有词向量相加的策略来得到文本语义表示，之后根据聚类类别个数的变化使用文本与类别间的相关关系来更新文本语义向量，具体采用隐含狄利克雷分布主题模型(latent Dirichet

allocation, LDA)和潜在语义标记模型(latent semantic index, LSI)进行语义的增强。

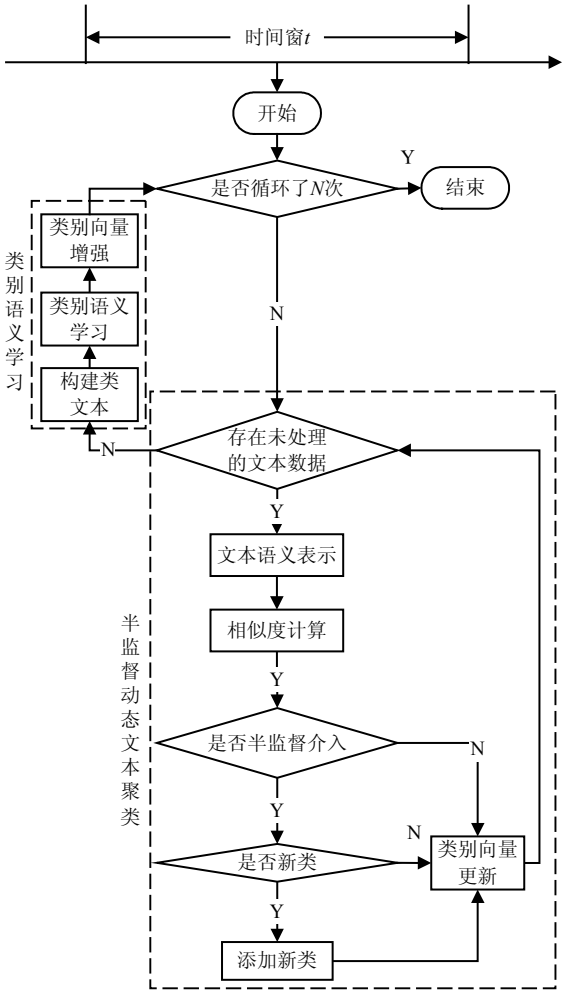

图1　SDCS聚类框架

为了搜索对应于每个输入的最佳聚类方案，本文使用取值为1或0的伯努利随机单元作为输出层的类别神经元，这种方式不容易陷入局部最小值。并且采用人工介入的方式监督新类的出现，该模块具体的模型框架如图2所示。第一列表示一个输入文本向量，第二列是二值伯努利神经元组成的聚类类别集合，$\boldsymbol{W}^t$ 为在 t 时刻类别的向量集合，模型最后的输出表示文本聚到各类的概率分布：

$$p_k = f(d_k) = 2(1-\sigma(d_k)) \quad k=1,2,\cdots \tag{1}$$

式中，d_k 为文本与类别间的欧氏距离，使用sigmoid函数 $\sigma(\cdot)$ 来转化为概率，其中距离 d_k 恒大于0，故 $0.5<\sigma(d_k)<1$，因此 $f(d_k)\in(0,1)$ 成立。使用该归一化准则能让文本以较大概率去到距离最短、最相似的类别，反之以较低概率去到距离最大、差异最明显的类别。

聚类过程中若得到文本的 $p_{k^*}=\max(p_k)$ 大于给定的阈值 η，则去到第 k^* 类，否则若 p_{k^*} 在 η 的左 δ 邻域中或者文本同时属于多个类的概率 p_k 差异不大时，则认为输入文本具有足够的新知识。当后者发生时模型会以半监督的方式介入，以人工监督的方式来判断新知识是否足够形成一个新类，以此更准确地捕获聚类个数。若确实是一个新类 $k+1$，此时类别向量更新涉及到4个主要操作：1) 更新文本中隐含主题的类别分布LDA和文本与类别的相关度LSI；2) 增加新类，将当前输入的文本向量作为新类别的类向量的初始化值；3) 类别语义向量和输入文本语义向量的扩展，由于新增了一个类别，因此必须将向量增加至少两个维度以适应LDA和LSI中新增类别的语义维；4) 增加阈值 $\eta=\eta+\alpha p_{k^*}$。新类的产生证明当前阈值属于合理范围，在小范围内增加也同样适用。

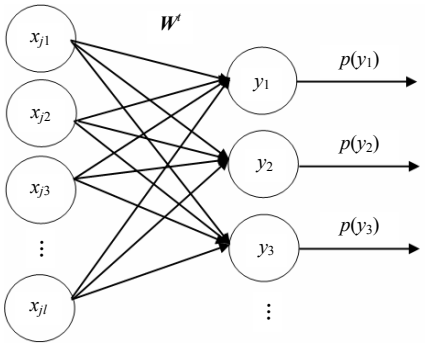

图2　半监督动态文本聚类模型

若监督结果不需要新增类别则文本直接聚到第 k^* 类，说明当前阈值设置过高引入了不必要的监督信息，应适当降低 $\eta=\eta-\alpha p_{k^*}$，让模型自适应更新阈值。

无论是 p_k 大于给定的阈值时文本聚到第 k^* 类还是最后监督结果认为聚到 k^* 类的情况，都通过一个伯努利实验来试触：

$$p(y_k)=p_k^{y_k}(1-p_k)^{1-y_k} \quad y_k\in(0,1), k=1,2,\cdots \tag{2}$$

当获胜单元 k^* 按概率取值1时给予奖励信号 $r_{k^*}>0$，相反则给予惩罚信号 $r_{k^*}<0$，对其他单元给予信号0，使得类别向量更新以梯度 Δw_{k^*j} 反向调节，完成整个半监督动态文本聚类过程。

$$\Delta w_{k^*j}=\begin{cases}\alpha\,|\,y_{k^*}-p_{k^*}\,|\,(x_j-w_{k^*j}) & k=k^*\\ 0, & k=k^*\end{cases} \tag{3}$$

本文使用REINFORCE算法簇[18]，获胜单元 k^* 的参数 w_{k^*j} 按梯度更新：

$$\Delta w_{k^*j}=\alpha(r-b_{k^*j})\frac{\partial \log g_{k^*}(y_{k^*},p_{k^*})}{\partial w_{k^*j}} \tag{4}$$

式中，α 为学习率；r 是某种度量标准给出的强化值；b_{k^*j} 称为强化基准，对不同的模型 $g_{k^*}(y_{k^*},p_{k^*})$ 不一样，这里指分布函数(2)。令强化基准为0，则：

$$\Delta w_{k^*j}=\alpha r\frac{\partial \log g_{k^*}(y_{k^*},p_{k^*})}{\partial w_{k^*j}}= \alpha r\frac{\partial \log g_{k^*}(y_{k^*},p_{k^*})}{\partial p_{k^*}}\frac{\partial p_{k^*}}{\partial d_k}\frac{\partial d_k}{\partial w_{k^*j}} \tag{5}$$

式中

$$\frac{\partial \log g_{k^*}(y_{k^*},p_{k^*})}{\partial p_{k^*}}=\begin{cases}\dfrac{1}{p_{k^*}} & y_{k^*}=1\\ -\dfrac{1}{1-p_{k^*}} & y_{k^*}=0\end{cases}=\frac{y_{k^*}-p_{k^*}}{p_{k^*}(1-p_{k^*})} \tag{6}$$

且

$$\frac{\partial p_{k^*}}{\partial d_k}=2(1-[f(1-f)])= 2(1-f)((1-f)-1)=p_{k^*}\left[\frac{p_{k^*}}{2}-1\right] \tag{7}$$

又

$$\frac{\partial d_{k^*}}{\partial w_{k^*j}}=\frac{\partial\sum_{j=1}^{L}(x_j-w_{k^*j})^2}{\partial w_{k^*j}}=-2(x_j-w_{k^*j}) \tag{8}$$

为了简便，选择以下比较合适的信号：

$$r=\begin{cases}\dfrac{1-p_{k^*}}{2-p_{k^*}} & 当k=k^*且y_{k^*}=1\\ -\dfrac{1-p_{k^*}}{2-p_{k^*}} & 当k=k^*且y_{k^*}=0\\ 0 & k\neq k^*\end{cases} \tag{9}$$

根据以上的推导可以得到参数更新的梯度式(3)，以此完成对 $\boldsymbol{W}^t$ 的更新调整。

2.2 类别语义学习

经过上一节半监督动态文本聚类模块，能得到一个初步的类别语义向量和聚类结果。为进一步提高类别语义的准确性，本文认为每个类中的文本应讨论相同的主题，同一个类中的词包含了类的描述信息，可基于这些信息来进一步学习类别的语义向量。在这样的思想启发下，本文使用了图3所示的两层Skip-gram模型[17]来进一步学习。

其中 $\boldsymbol{c}$ 代表类别的嵌入向量，由一个类文本输入模型训练得到，这里简单地将类中所有的文本拼接起来作为一个类文本。层与层之间是基于某个指定窗口大小的Skip-gram模型，它可以捕获单词之间的语义关系，非常契合本文的想法。

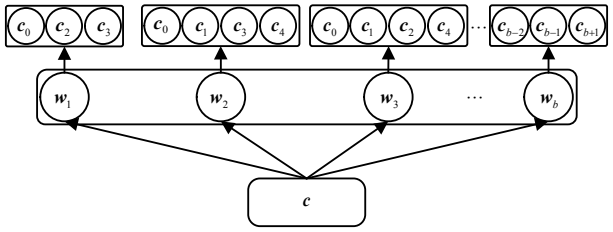

图3 类别语义学习模型

第一层Skip-gram模型能捕获类级别的单词共现模式，可以捕获类中单词之间的语义关系，最后使用负采样技术[19]最大化对数概率：

$$L_1=\sum_{\boldsymbol{c}_n\in\boldsymbol{T}}\sum_{\boldsymbol{w}_i^n\in\boldsymbol{V}_{c_n}}\log\left(\frac{\exp(\boldsymbol{w}_i^n\boldsymbol{c}_n)}{\sum_{\boldsymbol{w}_j^n\in\boldsymbol{V}}\exp(\boldsymbol{w}_i^n\boldsymbol{c}_n)}\right)= \sum_{\boldsymbol{c}_n\in\boldsymbol{T}}\sum_{\boldsymbol{w}_i^n\in\boldsymbol{V}_{c_n}}\left\{\log[\sigma(\boldsymbol{c}_n\boldsymbol{w}_i^n)]+\sum_{\boldsymbol{u}\in\mathrm{NEC}(\boldsymbol{w}_i^n)}\log[\sigma(-\boldsymbol{c}_n\boldsymbol{u})]\right\} \tag{10}$$

式中，$\boldsymbol{T}$ 是类文本向量集合，其中的第 n 个为 $\boldsymbol{c}_n$；$\boldsymbol{V}_{c_n}$ 为第 n 个类文本中包含的词向量集合，$\boldsymbol{w}_i^n$ 是第 i 个词向量；$\boldsymbol{V}$ 表示语料库词典中的词向量集合。第二层用来捕捉词的语义、语法，同样的：

$$L_2=\log\prod_{\boldsymbol{w}_i^n\in\boldsymbol{V}_{c_n}}\prod_{z\in\mathrm{Context}(\boldsymbol{w}_i^n)}\prod_{\boldsymbol{u}\in\{z\}\cup\mathrm{NEC}(z)}p(\boldsymbol{u}\mid\boldsymbol{w}_i^n)= \sum_{\boldsymbol{w}_i^n\in\boldsymbol{V}_{c_n}}\sum_{z\in\mathrm{Context}(\boldsymbol{w}_i^n)}\left\{\log[\sigma(\boldsymbol{w}_i^n\boldsymbol{z})]+\sum_{\boldsymbol{u}\in\mathrm{NEC}(z)}\log[\sigma(-\boldsymbol{w}_i^n\boldsymbol{u})]\right\} \tag{11}$$

最后使用梯度上升优化获得类别语义 $\boldsymbol{C}^t=\{\boldsymbol{c}_1,\boldsymbol{c}_2,\cdots\}$，再使用加法原则整合上一节半监督动态文本聚类模块得到的语义 $\boldsymbol{W}^t$ 来增强类别的语义，使用这种方式完成对聚类类别的语义更新。在该时间窗此后的迭代和下一个时间窗中用以初始化类别向量，即

$$\boldsymbol{W}^{t+1}=\boldsymbol{W}^t+\boldsymbol{C}^t \tag{12}$$

3 实验对比

本文在3个真实文本数据集上运行本文模型，对实验结果使用标准文档聚类评价指标，特别是归一化互信息NMI来评估聚类性能[20]。

3.1 真实数据集

首先使用3个数据集验证模型的有效性，第一个数据集TweetSet是从包括不同主题类的twitter语料

库中随机提取的5 450个数据。第二个数据集PaperSet来自Aminer-Paper语料库[21]。使用Kajjle比赛数据为第3个真实数据集。具体信息如表1所示(L：数据集样本总数，V：数据集词典大小，K：数据集包含的类别总数)。

表1　数据集信息

数据集	L	V	K
TweetSet	5 450	4 383	3
PaperSet	890	2 400	3
Kajjle	12 122	15 440	20

3.2　实验结果与分析

通过其他4种流聚类算法和标准k-means算法作为实验的对比基线。包括：1) 常用于新闻热点事件检测的Single-Pass数据流聚类算法——一种基于余弦相似度聚类数据的不可逆算法；2) 最先进的聚类算法FGSDMM+[15]——基于Dirichlet多项式混合模型的数据流动态聚类算法，是传统GSDMM[14]模型的推广，采用吉布斯采样算法进行学习推论；3) 最近提出的实时聚类算法RCADF[22]——基于数据特征匹配总数的实时聚类算法；4) 经典的CluStream算法。对每个模型运行后取NMI、F值进行比较。

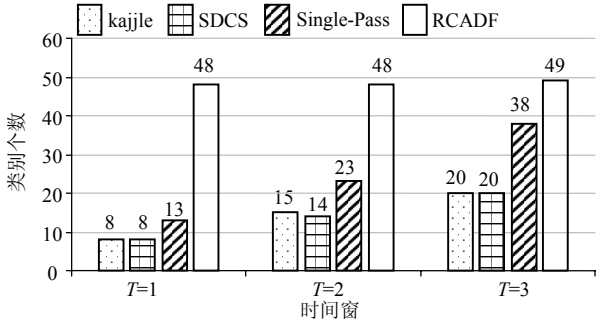

图4　模型聚类过程中聚类个数的变化

表2　kajjle数据集上动态聚类模型的NMI值

时间窗	SDCS	Single-Pass	RCADF
T=1	0.363	0.209	0.344
T=2	0.598	0.117	0.438
T=3	0.627	0.276	0.402

表3　kajjle数据集上动态聚类模型的F值

时间窗	SDCS	Single-Pass	RCADF
T=1	0.372	0.348	0.239
T=2	0.448	0.302	0.336
T=3	0.416	0.409	0.327

对真实数据集Kajjle进行3个时间窗的平均划分，时间窗中的超参数γ=0.3，α=0.05，iters=5，对比γ=0.03，iters=5，cluster_strictness=25流式聚类算法Single-Pass、RCADF。跟踪算法的聚类过程，由图4表示各算法聚类过程中簇的变化趋势，由图可知对比在3个时间窗中簇分别为8、7、5的数据而言SDCS模型表现更好，能自动获取各时间窗内8、6、6的新簇，表现出比其他模型更合理可靠的新类别发现能力，表明本文的半监督新类发现模式十分有效。表2、表3对应了各时间窗上各模型的NMI、F值，可见提出的算法优于其他模型，能力出众。

对3个数据集，除了在TweetSet、PaperSet中设置γ=0.1外其余保持不变，模型k-means和Clutream中设置与真实类别个数一致的k值。运行模型10次后各指标取平均值进行比较，结果如表4、表5所示。从实验结果看不管是对比经典的静态聚类算法k-means还是其他几种动态聚类算法，本文提出的模型表现更好。特别的它将属于“BUSINESS”类的文本“Fast-Food Chains With The Most Unhealthy Customers”、“The Most Damaged Brands In America”、“America's Disappearing Restaurant Chains”都正确聚到了一起，而Single-Pass、RCADF等算法错误地聚到了两个类。实验结果表明本文结合类别语义学习的算法对数据流进行的动态聚类是有效的，得到的聚类结果更符合实际情况。

表4　模型聚类NMI值

模型	TweetSet	PaperSet	Kajjle
Single-Pass	0.481	0.863	0.229
k-means	0.757	0.856	0.472
FGSDMM+	0.702	0.497	0.467
RCADF	0.612	0.757	0.332
Clustream	0.402	0.339	0.312
SDCS	0.851	0.945	0.535

表5　模型聚类F值

模型	TweetSet	PaperSet	Kajjle
Single-Pass	0.713	0.925	0.206
k-means	0.818	0.890	0.332
FGSDMM+	0.546	0.832	0.354
RCADF	0.693	0.807	0.256
Clustream	0.705	0.558	0.205
SDCS	0.925	0.980	0.415

综上所述，本文所提出的SDCS模型在文本动态聚类过程中，不仅引入了语义信息还允许人工的介入以半监督的方式生成更准确的类别个数。实验表明不管是在文本表征还是后面的类别语义学习，模型都更好地捕获了数据的信息而提高了聚类的准确度，并且模型结合半监督方法一起得到的聚类结果

更符合实际情况。实验对比发现模型整体表现是最优的，具有先进性和有效性。

4 结 束 语

本文提出一种基于语义处理数据流的动态聚类算法，让新到来的数据能根据学习到的类别语义直接聚类，提出结合半监督的方式追踪数据的动态变化来生成更准确的类别个数。模型算法既弥补了传统数据流聚类算法中缺乏语义描述的问题，又增强学习了类别的语义，使用语义向量的方法表示文本，不仅克服传统维度爆炸等问题，并且能在聚类过程中学习到准确的类别语义，对聚类性能的提高有积极影响。模型使用的半监督聚类方式也成功解决传统聚类算法聚类个数不准确的问题，得到更符合实际情况的聚类结果。最后和其他算法的对比，表现出良好的结果，表明本文算法是有效的、鲁棒的。

参 考 文 献

[1] TIAN Z, RAMAKRISHNAN R, LIVNY M. BIRCH: An efficient data clustering method for very large databases [C]//ACM SIGMOD International Conference on Management of Data. Montreal, Canada: ACM, 1996: 103-114.

[2] RODRIGUES P P, GAMA J, PEDROSO J P. ODAC: Hierarchical clustering of time series data streams[C]//Proceedings of the 6th SIAM International Conference on Data Mining. Bethesda, MD, USA: SIAM, 2006: 615-627.

[3] KRANEN P, ASSENT I, BALDAUF C, et al. The ClusTree: Indexing micro-clusters for anytime stream mining[J]. Knowledge and Information Systems Journal, 2011, 29(2): 249-272.

[4] IIBRAHIM O A, DU Y, KELLER J. Robust on-line streaming clustering[C]//International Conference on Information Processing and Management of Uncertainty in Knowledge-based Systems. Cádiz, Spain: Springer, 2018: 467-478.

[5] GUHA S, MEYERSON A, MISHRA N, et al. Clustering data streams: Theory and practice[J]. IEEE Transactions on Knowledge and Data Engineering, 2003, 15(3): 515-528.

[6] ARTHUR D. VASSILVITSKII S. K-means++: The advantages of careful seeding[C]//Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms. Philadelphia: Society for Industrial and Applied Mathematics, 2007: 1027-1035.

[7] ACKERMANN M R, LAMMERSEN C, MÄRTENS M, et al. StreamKM++: A clustering algorithm for data streams[J]. Journal of Experimental Algorithmics, 2012, 17(1): 173-187.

[8] AGGARWAL C C, HAN J, WANG J, et al. A framework for clustering evolving data streams[C]//Proceedings of the 29th International Conference on Very Large Data Bases. Berlin: VLDB Endowment, 2003: 81-92.

[9] BAO J P, WANG W Q, YANG T S, et al. An incremental clustering method based on the boundary profile[J]. PLOS ONE, 2018, 13(4): e0196108.

[10] CAO F, ESTERT M, QIAN W, et al. Density-based clustering over an evolving data stream with noise[C]// Proceedings of the 2006 SIAM International Conference on Data Mining. Bethesda, MD: SIAM, 2006: 328-339.

[11] LIU L X, GUO Y F, KANG J, et al. A three-step clustering algorithm over an evolving data stream[C]//IEEE International Conference on Intelligent Computing and Intelligent Systems. Shanghai, China: IEEE, 2009: 160-164.

[12] CHEN Y, TU L. Density-based clustering for real-time stream data[C]//Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA: ACM, 2007: 133-142.

[13] YANG F, CAO J, ZHOU K, et al. An adaptive clustering algorithm based on CFSFDP[C]//The 33rd Youth Academic Annual Conference of Chinese Association of Automation. Nanjing, China: IEEE, 2018: 404-408.

[14] YIN J, WANG J. A dirichlet multinomial mixture model-based approach for short text clustering[C]// Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. New York, USA: ACM, 2014: 233-242.

[15] YIN J, WANG J. A text clustering algorithm using an online clustering scheme for initialization[C]//ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. San Francisco, USA: ACM, 2016: 1995-2004.

[16] LAOHAKIAT S, PHIMOLTARES S, LURSINSAP C. A clustering algorithm for stream data with lda-based unsupervised localized dimension reduction[J]. Information Sciences, 2017, 381: 104-123.

[17] MIKOLOV T, SUTSKEVER I, CHEN K, et al. Distributed representations of words and phrases and their compositionality[J]. Advances in Neural Information Processing Systems, 2013, 26: 3111-3119.

[18] WILLIAMS R J. Simple statistical gradient-following algorithms for connectionist reinforcement learning[J]. Machine Learning, 1992, 8(3-4): 229-256.

[19] MIKOLOV T, CHEN K, CORRADO G, et al. Efficient estimation of word representations in vector space[C]// Proceedings of the International Conference on Learning Representations. Scottsdale, AZ, USA: ICLR, 2013:1-12.

[20] ZHONG S. Semi-supervised model-based document clustering: a comparative study[J]. Machine Learning, 2006, 65(1): 3-29.

[21] TANG J, ZHANG J, YAO L, et al. Arnetminer: Extraction and mining of academic social networks[C]//ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Las Vegas, USA: ACM, 2008: 990-998.

[22] LAMSAL R, MALCOMBER I. Real-time clustering al gorithm based on predefined level-of-similarity[EB/OL]. (2018-10-03). https://pdfs.semanticscholar.org/35fd/1eea45b0a54d28624771d8745f78226370d1.pdf.

编 辑 蒋 晓