结合指数函数改进的随机近邻嵌入式短文本聚类

汪晓晨[1] 宋叔尼[2*]
[1](东北大学理学院 辽宁 沈阳 110819)
[2](广东培正学院 广东 广州 510830)

摘 要 近年来深度学习在短文本聚类方面发挥巨大作用,最近提出的短文本聚类(Short Text Clustering,STC)算法在此方面取得不错的成效。为进一步提高聚类准确率并优化算法性能,基于指数函数提出改进的随机近邻嵌入算法。该算法用指数函数度量样本点与聚类中心差距,放大不同特征差别,并在后期使用 k-means ++ 算法预先确定聚类中心与聚类数目。在 Stackoverflow 数据集上的实验证明,随机指数嵌入聚类模型(e-STC)在准确率与标准互信息上均优于原 STC 模型,准确率相对提高 3.2%,互信息相对提高 2.9%。

关键词 短文本聚类 深度算法 随机近邻嵌入 特征提取

中图分类号 TP3 **文献标志码** A DOI:10.3969/j. issn. 1000-386x. 2024. 03. 035

STOCHASTIC NEIGHBOR EMBEDDING SHORT TEXT CLUSTERING IMPROVED BY EXPONENTIAL FUNCTION

Wang Xiaochen[1] Song Shuni[2*]
[1](*College of Sciences*, *Northeastern University*, *Shenyang 110819*, *Liaoning*, *China*)
[2](*Guangdong Peizheng College*, *Guangzhou 510830*, *Guangdong*, *China*)

Abstract In recent years, deep learning has played an important role on the short text clustering. The short text clustering algorithm (STC) proposed recently has achieved good results in this field. In order to further improve the clustering accuracy and optimize the performance of algorithm, an improved stochastic neighbor embedding algorithm based on exponential function (e-STC) is proposed. This algorithm magnified the difference between different features by using exponential function to calculate the gap between sample points and clustering center. In the later stage, K-Means ++ algorithm was used to determine the clustering center and clustering number in advance. The results of experiments on Stackoverflow dataset show that e-STC algorithm is superior to the original STC algorithm in terms of the accuracy and the normalized mutual information metric. The accuracy is improved by 3.2%, and the normalized mutual information is increased by 2.9% relatively.

Keywords Short text clustering Depth clustering Random neighbor embedding Feature extraction

0 引 言

在信息化时代,伴随着电子商务的兴起以及在线交流方式的大规模发展,互联网成为人与人之间表达、交流的重要平台。人们借助 Facebook、Twitter、新浪微博、腾讯微信等进行沟通,在此过程中产生大量文本数据,尤其以短文本数据为重,使得这些平台成为一个海量信息的熔炉。而将相似短文本聚集是建立用户群像、舆情分析和信息决策等工作的前提。因此,如何高效而准确地进行短文本数据聚类就显得格外重要。

聚类是把数据集中相似的样本进行分组的过程,它是探索无标签数据的重要手段,可以有效帮助寻找分析数据的结构。根据无标签数据种类的不同,数据聚类工作可以分为图像聚类和文本聚类。关于文本聚类,传统的聚类算法一般是输入给定的数据特征,然后

收稿日期:2021 - 01 - 06。国家自然科学基金项目(11801065)。**汪晓晨**,硕士生,主研领域:自然语言处理。**宋叔尼**,教授。

再使用不同的模型对其进行划分。最早用于文本聚类的是向量空间模型(Vector Space Model,VSM)[1],该模型将文本向量化并放置在同一向量空间中,用空间相似度表示语义相似度。TF-IDF 模型[2-3]是在 VSM 模型的基础上先计算每个文本的词频和权重向量,将其储存在向量空间模型中,抽取特征,对每个特征词重新加权处理,该模型具有很好的延展性。2002 年由 Blei 等[4]提出的主题模型(Latent Dirichlet Allocation,LDA)也在文本聚类方面发挥了巨大作用,一些学者使用这种模型将向量空间中的高维特征向量降维,有效地缓解向量空间高维且稀疏的压力[5-7]。基于外部词典的聚类方法同时在解决数据稀疏和忽略潜在语义的问题上发挥了巨大作用,该方法通过人为构造外部语料库发掘文本潜在语义,外部语料库的构成具有主观性且新词的迭出不穷需要外部词典的不断更新,这对聚类的效果有一定的负面影响[8-10]。另外,这些传统的聚类算法在对特征识别方面普遍精度不是很高,尤其是对于高维度的数据(比如图像、文本等),可能由于不能有效度量高维空间中的样本相似性而难以发现数据的内在聚类结构。近几年,为了进一步地挖掘相似样本之间的信息,考虑将神经网络与传统的文本聚类算法相结合,利用网络模型自动而高效地学习数据的潜在特征,进行数据挖掘与处理[11-12]。

目前的深度聚类算法一般使用自编码器(auto-encoder,AE)构造非线性嵌入进行无监督学习,根据特征提取与聚类的不同结合方式分为两种[13-15]。一种是先将高维数据嵌入低维数据空间,将自编码器与谱聚类结合,通过深度神经网络学习低维特征,最后采用聚类算法进行聚类[16-18]。但是这种将特征提取和聚类任务分开顺次进行的做法存在耦合度低,学习的特征与接下来的聚类任务不一定是正相关的问题。为了解决这个问题,Yang 等[19]使特征提取与聚类同时进行,在聚类过程中通过优化损失函数不断调整编码过程中的特征提取,从而提升聚类精度。Xie 等[20]在自编码器上堆叠一个聚类层,提出了堆叠自编码器的深度嵌入聚类(Deep Embedded Clustering,DEC)模型,该模型使用自编码器借助 t-分布随机邻域嵌入算法(t-distribution Stochastic Neighbor Embedding,t-SNE)[21]嵌入数据,接着进行特征的提取,通过不断降低 KL 损失函数调整模型参数,同时在 encoder 阶段使用 k-means 聚类进行最终的聚类,以此提高聚类精度。2017 年,Xu 等[22]在 DEC 模型的基础上先将原始文本特征嵌入到紧凑的二进制代码中,为神经网络结构提供辅助的学习信息,随之在聚类过程中拟合预先训练好的二进制码,从而提出了一种自督学的卷积神经网络(The Self-taught Convolutional Neural Network,STC2)。2019 年,Hadifar 等[23]在深度嵌入聚类模型(DEC)的基础上,借助 Xu 等的数据进行研究,提出了一种短文本聚类模型(STC)。不同于 DEC 模型的是,Hadifar 等除了将自编码器与聚类过程堆叠以外,还在该过程之前使用平滑逆频率算法[24]将高维的文本数据降为低维度的数据,缓解自编码器提取特征的压力,从而加快聚类速度,提升聚类性能。

本文结合指数函数改进了随机近邻嵌入算法,并将改进的随机近邻嵌入算法与 k-means ++ 算法应用到 STC 模型中,提出一种新的模型——随机指数嵌入聚类模型(e-STC)。一方面,本文结合指数函数对随机近邻嵌入算法进行改进,通过使用指数函数度量样本点与聚类中心差距,提升特征软分配度,放大不同特征差别,使得相似度低的文本在聚类时的区别更加明显,从而提升聚类性能。另一方面,使用 k-means ++ 算法代替原来的 k-means 算法进行聚类。实验结果表明,与 STC 模型相比,e-STC 模型聚类的准确率相对提高了 3.2%,标准互信息相对提高了 2.9%。

1　相关技术

1.1　平滑反频率嵌入算法

平滑反频率嵌入(Smooth Inverse Frequency,SIF)[24]是一种无监督的句子嵌入,又称 WR 嵌入,W (smooth inverse frequency weighting) 是权重,R (removing the common components) 是移除句中无关成分,该算法先计算句子中词向量的加权平均,然后删除第一个向量上的平均投影,将高维数据降维。对于无标签文本集中的每一个词 w,计算每一个词向量的加权平均值与其权重$\frac{a}{a+p(\omega)}$,其中 a 是一个不断调整的超参数,$p(w)$是待估词的词频。在对句子 S 的每个词 w 赋予权重后,借助主成分分析方法(Principal Component Analysis,PCA)或者奇异值分解(Singular Value Decomposition,SVD)移除句向量中的无关成分,从而用句子中最有代表意义的部分代表整体 $\boldsymbol{v}_s \leftarrow \boldsymbol{v}_s - \boldsymbol{u}\boldsymbol{u}^{\mathrm{T}}\boldsymbol{v}_s$,其中,$\boldsymbol{u}$ 为句向量构成的矩阵的第一个主成分。

算法 1　平滑反频率嵌入

输入:词嵌入 $\boldsymbol{v}_\omega$,ω 属于集合 V,句子集合 S,参数 a,待估计的词频 $p(\omega)$。

输出:句子嵌入 $\boldsymbol{v}_s$,s 属于 S。

步骤 1　对于句子集合中的每一个句子,计算:

$$v_s \leftarrow \frac{1}{|s|}\sum_{\omega \in s}\frac{a}{a+p(\omega)}v_\omega$$

步骤 2 将由步骤一计算所得的 v_s 按列组成矩阵，设 u 为矩阵 $\boldsymbol{X}$ 的第一个奇异向量。

步骤 3 对于句子集合中的每一个句子 s，计算 $v_s \leftarrow v_s - uu^{\mathrm{T}}v_s$。

1.2 自编码器

自编码器(AE)是一种无监督的特征学习算法，由输入层、隐藏层和输出层三层神经网络构成，它通过反向传播算法重构输入的无标签数据。如图 1 所示，AE 由编码器(encoder)和解码器(decoder)两个部分组成，其中：X 是输入层，$X_i(i=1,2,\cdots,n)$ 表示输入的未编码数据；H 是隐藏层，$h_i(i=1,2,\cdots,m)$ 表示编码后的数据；Y 是输出层，$Y_i(i=1,2,\cdots,n)$ 表示解码后的数据。自编码器的目标是使输入数据尽可能等于输出数据。编码器和解码器可以是单层也可以是多层，但应该是对称的，具有相等的层数。本文采用的是一般的三层自编码器。

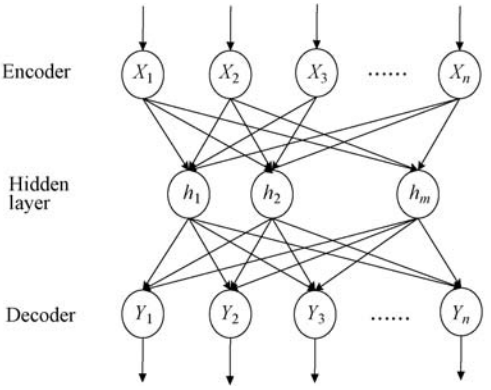

图 1 一般自编码器

1.3 k-means ++ 算法

k-means ++ 算法[25]是由 Arthur 等提出的，该算法在 k-means 算法的基础上增加了采用轮盘法选取初始聚类中心的步骤。首先随机选取一个初始聚类中心；然后计算数据集中的每个样本点与其对应聚类中心距样本的最短距离，求平方和，计算概率；接下来采用轮盘法直至得到 k 个聚类中心；之后采用 k-means 算法的步骤进行剩下的聚类过程。具体算法步骤如算法 2 所示。

算法 2 k-means ++ 算法

输入：数据集 S，聚类数目 k。

输出：k 个聚类结果。

步骤 1 数据集 S，随机选取点 s_1 作为第一个聚类的中心。

步骤 2 计算 $d(x_i)=\min|s_i-c_{ij}|, sum(d^2(x_i))$，其中，数据集 $s_i \in S, c_{ij}$ 是样本 s_i 对应的聚类中心。

步骤 3 计算数据集 S 中每个样本 s_i 被选择为下一个聚类中心的概率 $p(x_i)$，其中

$$p(x_i)=\frac{D(x)^2}{\sum_{x\in X}D(x)^2}$$

步骤 4 采用轮盘法选取下一个聚类中心，即在[0,1]范围中随机选取一个数记作 r，若 $r-p(x_1)>0$，则继续减 $p(x_2)$，直至 $r-p(x_j)<0$ 为止，此时 j 即为第二个聚类中心。

步骤 5 重复上述步骤 2 - 步骤 4 直至得到 k 个聚类中心。

步骤 6 采用 k-means 算法的步骤进行剩下的聚类过程。即对数据集中的每个元素，计算它到 k 个聚类中心的距离并将其分配到距离最近的中心所属的类；对每个类别，更新其聚类中心 $c_i=\frac{1}{|c_i|}\sum_{x\in c_i}x$；重复该步骤直至中心位置不再发生变化。

2 随机指数嵌入聚类模型

随机指数嵌入模型(e-STC)是将基于指数函数改进的随机近邻嵌入算法与 k-means ++ 算法应用在 STC 模型而构造的新模型。主要通过文本嵌入、特征提取、聚类三个阶段对短文本进行聚类，整体流程如图 2 所示。

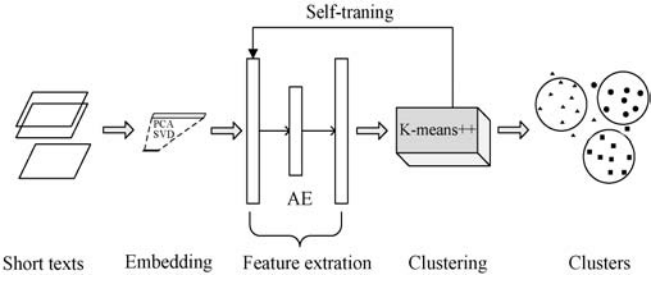

图 2 随机指数嵌入模型(e-STC)

第一阶段：使用平滑反频率嵌入(SIF)将高维文本数据降维，移除无关项，降低数据复杂度；第二阶段：构造一个三层的自编码器对降维后的数据编码再解码，提取特征，在该阶段，通过采用指数复合函数衡量嵌入点与聚类中心相似性的方式改进 t-SNE 算法，以达到放大不同特征差别、提升特征软分配度的目的；第三阶段：使用 k-means ++ 算法对解码后的数据进行文本聚类。

2.1 文本嵌入

在模型的第一阶段，模型使用 SIF 文本嵌入计算输入文本词向量的加权平均，移除无关项，达到对高维数据降维的目的。

2.2 特征提取

在模型的第二阶段，本文构造一个三层的自编码器，结构如图 3 所示，将第一阶段降维后的数据输入自编码器，依次经过 encoder 和 decoder 进行自训练(self-training)来提取特征。

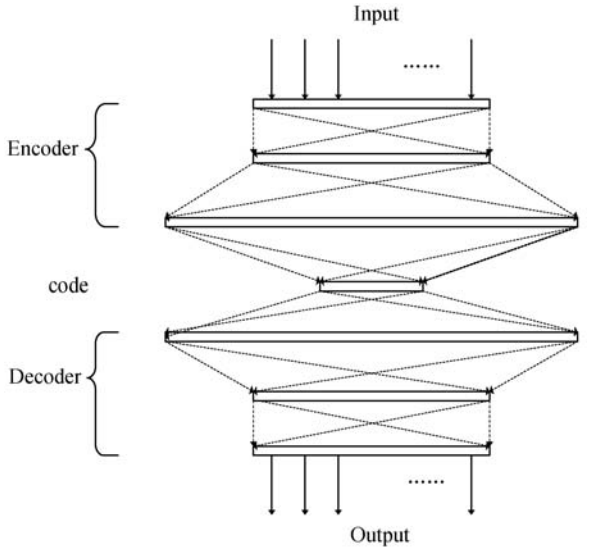

图 3　三层自编码器

在自训练阶段,Xu 等[22]借鉴 t-SNE 算法,用 t 分布为核来度量嵌入点 x 和中心点 μ 的距离,并对该距离进行规范化,提出一个新的概念——软分配度。本文在 Xu 等的基础上进行改进,为了进一步扩大不同特征的区别,提升软分配度,采用指数复合函数代替二次函数度量嵌入点 x 和中心点的距离,软分配度的计算公式如下:

$$q_{ij} = \frac{(1 + e^{\|z_i-\mu_j\|^2}/\alpha)^{-\frac{\alpha+1}{2}}}{\sum_j (1 + e^{\|z_i-\mu_j\|^2}/\alpha)^{-\frac{\alpha+1}{2}}} \tag{1}$$

式中:$z_i = f(x_i)$,$x_i \in X$,$z_i \in Z$,Z 是特征空间,函数 f 代表空间 X 到空间 Z 的映射;α 是学生 t 分布的自由度;q_{ij}为样本 i 对于聚类 j 的软分配度,取值范围为 0 ~ 1。因为自由度的取值对软分配度的大小存在影响,会影响到接下来的聚类结果,因此在本文的实验阶段对自由度进行了研究,最终选取最优自由度 $\alpha = 40$。

为了提高聚类的精确度,本文构造一个辅助的目标分布 P,通过提高软分配度 q_{ij}的幂次,将一次幂提高到二次幂,以此增加关键特征的比重,同时对每一个样本点的软分配度都进行归一化,从而降低隐藏的大型特征空间的复杂度。构造的辅助目标概率公式如下:

$$p_{ij} = \frac{q_{ij}^2/f_i}{\sum_j q_{ij}^2/f_i} \tag{2}$$

式中:$f_i = \sum_i q_{ij}$ 为聚类群频率。

编码器的重构损失正则约束使用 KL 散度可以避免深度聚类模型过拟合,因此算法的损失函数选取 KL 散度(Kullback-Leibler divergence)。本文模型通过对原始分布 q 和辅助分布 p 进行学习,使用 KL 散度来拉近这两个分布之间的距离,目标函数如下:

$$L = \mathrm{KL}(P \| Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}} \tag{3}$$

对于该损失函数 L,本文使用随机梯度下降法(Stochastic Gradient Descent,SGD)对其进行训练,以达到同时优化自编码器的网络参数 θ 和聚类中心 $\mu_j(j = 1,2,\cdots,K)$ 的目的。每次迭代,损失函数 L 关于嵌入点 z_i和聚类中心 μ_j的更新计算如下:

$$\frac{\partial L}{\partial z_i} = \frac{\alpha+1}{\alpha} \cdot \sum_j \left(1 + \frac{e^{\|z_i-\mu_j\|^2}}{\alpha}\right)^{-1} \cdot (p_{ij} - q_{ij}) \cdot (z_i - \mu_j) \cdot e^{\|z_i-\mu_j\|^2} \tag{4}$$

$$\frac{\partial L}{\partial \mu_j} = -\frac{\alpha+1}{\alpha} \cdot \sum_i \left(1 + \frac{e^{\|z_i-\mu_j\|^2}}{\alpha}\right)^{-1} \cdot (p_{ij} - q_{ij}) \cdot (z_i - \mu_j) \cdot e^{\|z_i-\mu_j\|^2} \tag{5}$$

在此框架下,e-STC 可以同时优化网络参数 θ 与聚类中心 $\mu_j(j = 1,2,\cdots,K)$。本文设置当聚类分配后改变的点小于 0.1% 时,训练停止。

2.3　聚　类

在模型的第三阶段,Xu 等采用 k-means 算法对经过自编码器处理后的数据进行聚类。k-means 算法的原理相对简单,且收敛速度很快,因此一般作为聚类的首选。同时,该算法的初始聚类中心是随机选取的,具有很大的偶然性,若初始中心选取不当,则聚类过程将变得繁琐且耗时,而且聚类结果可能会有较大误差。Arthur 等在 k-means 算法的基础上预先确定初始聚类中心以及聚类数目的步骤,从而提出了 k-means ++ 算法,该算法改进了 k-means 随机选取聚类中心的问题,其余流程与 k-means 算法基本一致。具体聚类流程如图 4 所示。

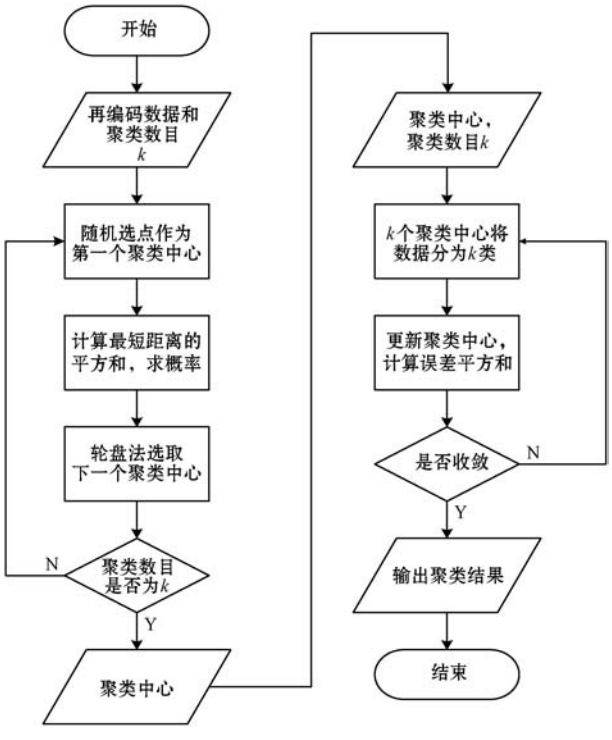

图 4　k-means ++ 流程

3 实验与结果分析

本节中,在 Python3 和 TensorFlow 的框架下,选取 StackOverflow 数据集,对 e-STC 模型与其他深度聚类算法进行对比分析。

3.1 数据集

StackOverflow,原始数据集包含了自 2012 年 7 月 31 日到 2012 年 8 月 14 日的 3 370 528 个样本。本文实验使用 Xu 等[22]的实验数据,选取了原始数据集的一部分,从 20 个不同的标签中随机选取 20 000 个问题,数据具体信息如表 1 所示。

表 1 StackOverflow 数据

数据集	类别数	短文本数	每个文本的平均标记数	词汇量
StackOverflow	2×10^4	2×10^4	8.3	2.3×10^4

3.2 参数设置

本文参数设置参照 STC 模型,实验采用 Word2vec 工具进行词嵌入,在 StackOverflow 整个数据集中进行题目和内容的训练。自编码器预训练数据时的初始权重由高斯分布(标准差为 0.01)随机生成,深度聚类层的设置为全连接层,隐层设置为 d,短文本数据的嵌入维度分别为 500、500、2 000、20,各层激活函数为 ReLU。在最小化 KL 散度时,使用 SGD 优化算法,并将其学习率设置为 0.01,动量设置为 0.9,收敛阈值设置为 0.1%。最后用 k-means ++ 算法对再编码后的数据进行聚类时,为了保证实验的普适性,降低偶然性,取 5 次实验的结果平均作为最终的实验结果。

3.3 评判标准

本文实验采取的两个评价标准是 2005 年由 Cai 等[26]提出,2014 年被 Huang 等[27]丰富完善的准确率(accuracy,ACC)与互信息(normalized mutual information metric,NMI)。这两个度量标准将从每个文本聚类的准确性与标签和聚类集之间的重合度两个角度分别来评估聚类性能。

给定文本 x_i,设 c_i 和 t_i 分别为聚类标签和语料库提供的标签。则精度的定义为:

$$A_{CC}=\frac{1}{n}\sum_{i=1}^{n}\delta(y_i=g(c_i)) \tag{6}$$

式中:n 是文本的总数;$\delta(x,y)$ 是一个指标函数;y_i 是文本的真实标签;c_i 是聚类算法得出的聚类结果;$g(c_i)$ 是置换映射函数,表示每个聚类标签 c_i 与真实标签的映射[28]。

标准化互信息(NMI)是信息论中的一种信息度量,代表一个随机变量中包含的另一个随机变量的信息量,可以用来度量两个聚类之间的相似程度。在本实验中,通过标签 T 和聚类集 C 的相似程度来度量聚类效果。计算公式如下:

$$NMI(T,C)=\frac{I(T,C)}{\sqrt{H(T)H(C)}} \tag{7}$$

式中:$I(\cdot)$ 是互信息;$H(\cdot)$ 是信息熵;$NMI(T,C)$ 是标签 T 和聚类集 C 之间的互信息;分母部分 $\sqrt{H(T)H(C)}$ 是为了归一化,将互信息缩至[0,1]范围内。

3.4 与其他聚类算法的对比

选取了传统聚类算法 TF-IDF[2]、SIF[24],深度聚类模型 STC2[22]、STC[23] 与本文的模型 e-STC 进行对比实验,实验结果如表 2 所示。可以看出:在同样的数据集 Stackoverflow 下,本文 e-STC 模型与深度聚类模型在准确率和标准互信息两方面效果均优于传统聚类,因为使用神经网络构造的自编码器对数据解码再编码,然后提取特征,该做法可对数据有效降维,从而缓解短文本数据稀疏的问题。另外 e-STC 模型聚类准确率与标准互信息均高于 STC2 模型与 STC 模型,原因如下:一方面,e-STC 模型采用了结合指数函数改进的随机近邻嵌入算法进行特征提取,有效地扩大了特征的软分配度,使得不同特征之间的差异度更加明显,有效降低特征被错误聚类的概率,从而提升聚类准确度;另一方面,使用 k-means ++ 算法对聚类中心与聚类数目进行了预选取,降低了偶然性。该实验使用的数据由 20 个不同类别的问题标签组成,聚类结果采用 ACC 和 NMI 进行衡量,e-STC 模型的 ACC 和 NMI 明显高于其他模型,且每一簇的聚类结果符合实际需要。

表 2 与其他聚类算法的对比(%)

模型	ACC	NMI
TF-IDF	22.43	17.82
SIF	30.61	29.03
STC2	54.04	50.50
STC	58.65	51.93
e-STC(本文模型)	60.55	53.42

3.5 拓展实验

(1) 在不同自由度 α 下的聚类性能。由统计知识可知,对于 t 分布而言,自由度 α 越大,该分布越接近高斯分布。如图 5 所示,可以观察到,在一定范围内,随着自由度的增大,t 分布越来越接近高斯分布,整个模型的聚类性能越来越好。但是由于我们在聚类初始化时便假定数据服从高斯分布,使得自由度超出某一

界定值后,随着自由度的增大聚类结果将不再有明显的变化,甚至有所下降。

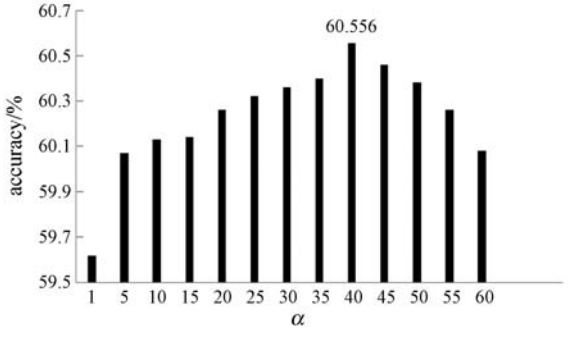

图 5　不同自由度下模型的聚类性能

(2) 不同改进算法下模型的聚类性能。为了验证基于指数函数改进的随机近邻嵌入算法与 k-means ++ 算法对 STC 模型的聚类性能均有效,本文在数据集 StackOverflow 上进行了拓展实验。使用基于指数函数改进的随机近邻嵌入算法与 k-means ++ 算法分别在原 STC 模型进行实验,实验结果如表 3 所示。可以发现,两种改进对原 STC 模型的聚类效果均有提升作用,其中仅使用基于指数函数改进的随机近邻嵌入算法可将模型相对精确率提高 2.1%,相对互信息提高 1.6%;仅使用 k-means ++ 算法也可将相对精确率提高 0.8%,相对互信息提高 0.3%。这表明两者对于模型性能提升均为有效的。

表 3　不同改进算法在数据集上的聚类准确率比较(%)

数据集	评价指标	STC	仅改进 t-SNE 算法	仅改进 k-means	e-STC
Stackover flow	ACC	58.65	59.91	59.13	60.55
	NMI	51.93	52.76	52.11	53.42

4　结　语

STC 是一种基于深度学习的无监督聚类算法,首先通过 SIF 算法将高维数据降维,然后自编码器进行特征的提取,最后采用 k-means ++ 算法进行最后的聚类。本文在 STC 模型的基础上提出随机指数嵌入模型(e-STC),一方面借助指数函数对 t-SNE 算法进行优化,使用指数函数度量样本点与聚类中心差距,放大不同特征差别,提升特征软分配度;另一方面又采用 k-means ++ 算法预先选取初始聚类中心,然后进行聚类。最后在 Stackoverflow 数据集进行实验,结果验证了本文模型相对于其他深度聚类算法有了显著提升,比 STC 模型准确率相对提高了 3.2%,标准互信息相对提高了 2.9%。今后将进一步研究基于指数函数改进的随机近邻嵌入算法对于长文本的聚类效果。

参 考 文 献

[1] Salton G, Wong A, Yang C. a vector space model for automatic indexing[J]. Communications of the ACM, 1975, 18(11): 613 – 620.

[2] Kuang Q, Xu X. Improvement and application of TF · IDF method based on text classification[C]//2010 International Conference on Internet Technology and Applications, 2010: 1 – 4.

[3] 戚后林. 基于词频和语义的文本聚类算法研究[D]. 南京:南京邮电大学,2017.

[4] Blei D M, Ng A Y, Jordan M I. Latent Dirichlet allocation [J]. The Journal of Machine Learning Research, 2003, 3: 993 – 1022.

[5] 邢长征,赵全颖,王伟,等. 基于优化密度的耦合空间 LDA 文本聚类算法研究[J]. 计算机应用研究,2017,34(7): 1966 – 1970.

[6] 李国,张春杰,张志远. 一种基于加权 LDA 模型的文本聚类方法[J]. 中国民航大学学报,2016,34(2): 46 – 51.

[7] Blei D, Ng A, Jordan M. Latent Dirichlet allocation[J]. Journal of Machine Learning Research, 2003, 3: 993 – 1022.

[8] 吴舜尧,邵峰晶,王金龙,等. 融合语义资源和关键词的文本聚类[J]. 计算机工程,2014,40(4): 223 – 227.

[9] Chen X, Zhang Y, Cao L, et al. An improved feature selection method for Chinese short texts clustering based on HowNet[C]//Computer Engineering and Networking, 2013: 635 – 642.

[10] 潘囿丞. 基于领域知识的自动答题方法研究[D]. 哈尔滨:哈尔滨工业大学,2016.

[11] 卢玲,杨武,杨有俊,等. 结合语义扩展和卷积神经网络的中文短文本分类方法[J]. 计算机应用,2017,37(12): 3498 – 3503.

[12] 张绮琦,张树群,雷兆宜. 基于改进的卷积神经网络的中文情感分类[J]. 计算机工程与应用,2017,53(22): 111 – 115.

[13] Masci J, Meier U, Ciresan D, et al. Stacked convolutional auto-encoders for hierarchical feature extraction[C]//International Conference on Artificial Neural Networks, 2011: 52 – 59.

[14] Baldi P. Autoencoders, unsupervised learning, and deep architectures[C]//2011 International Conference on Unsupervised and Transfer Learning Workshop, 2011: 37 – 50.

[15] Rifai S, Vincent P, Muller X, et al. Contractive auto-encoders: Explicit invariance during feature extraction[C]// 28th International Conference on International Conference on Machine Learning, 2011: 833 – 840.

[16] Tian F, Gao B, Cui Q, et al. Learning deep representations for graph clustering[C]//28th AAAI Conference on Artificial Intelligence, 2014: 1293 – 1299.

[17] 袁非牛,章琳,史劲亭,等. 自编码神经网络理论及应用综述[J]. 计算机学报,2019,42(1):203 - 230.
[18] 万静,吴凡,何云斌,等. 新的降维标准下的高维数据聚类算法[J]. 计算机科学与探索,2020,14(1):96 - 107.
[19] Yang J, Parikh D, Batra D. Joint unsupervised learning of deep representations and image clusters[C]//2016 IEEE Conference on Computer Vision and Pattern Recognition, 2016:5147 - 5156.
[20] Xie J, Girshick R, Farhadi A. Unsupervised deep embedding for clustering analysis[EB]. arXiv:1511.06335,2015.
[21] Maaten L. Learning a parametric embedding by preserving local structure[C]//12th International Conference on Artificial Intelligence and Statistics,2009:384 - 391.
[22] Xu J, Xu B, Wang P, et al. Self-taught convolutional neural networks for short text clustering[J]. Neural Networks, 2017,88:22 - 31.
[23] Hadifar A, Sterckx L, Demeester T, et al. A self-training approach for short text clustering[C]//4th Workshop on Representation Learning for NLP,2019:194 - 199.
[24] Arora S, Liang Y, Ma T. A simple but tough-to-beat baseline for sentence embeddings[C]//5th International Conference on Learning Representations,2017.
[25] Arthur D, Vassilvitskii S. K-means ++: The advantages of careful seeding[C]//18th Annual ACM-SIAM Symposium on Discrete Algorithms Stanford,2006:1027 - 1035.
[26] Cai D, He X, Han J. Document clustering using locality preserving indexing[J]. IEEE Transactions on Knowledge and Data Engineering,2005,17(12):1624 - 1637.
[27] Huang P, Huang Y, Wang W, et al. Deep embedding network for clustering[C]//2014 22nd International Conference on Pattern Recognition,2014:1532 - 1537.
[28] Papadimitriou C, Steiglitz K. Combinatorial optimization: Algorithms and complexity[M]. Courier Corporation,1998.

(上接第 206 页)

4 结 语

本文提出一种混合卷积模块用于改进 Mask R-CNN 的特征提取网络,并设计了几个简单而有效的组件来增强通道中的信息传播,通过可靠的信息传递丰富了每个阶段的建议特征。MixedMask 模型在检测精度上超过了 Mask R-CNN 模型,同时,设计的混合卷积模块可根据不同的实例对象修改内核尺寸和分组,适用于许多未来的对象检测和实例分割研究工作。尽管 MixedMask 是一个高效的实例分割模型,但在某些特定场景的应用中仍有改进的空间,在后面的工作中我们计划在检测效率上对其进行进一步的优化。

参 考 文 献

[1] Ren S, He K, Girshick R, et al. Faster R-CNN: Towards Real-Time object detection with region proposal networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2017,39(6):1137 - 1149.
[2] Pinheiro P O, Collobert R, Dollar P. Learning to segment object candidates[EB]. arXiv:1506.06204, 2015.
[3] Kirillov A, Levinkov E, Andres B, et al. InstanceCut: From edges to instances with MultiCut[C]//2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.
[4] He K, Gkioxari G, Dollar P, et al. Mask R-CNN[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2018,42(2):386 - 397.
[5] 温尧乐,李林燕,尚欣茹,等. 一种改进的 Mask RCNN 特征融合实例分割方法[J]. 计算机应用与软件,2019,36(10):130 - 133.
[6] Liu S, Qi L, Qin H, et al. Path aggregation network for instance segmentation[C]//2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition,2018.
[7] Huang Z, Huang L, Gong Y, et al. Mask Scoring R-CNN[C]//2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),2019.
[8] Cai Z, Vasconcelos N. Cascade R-CNN: High quality object detection and instance segmentation[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence,2021,43(5): 1483 - 1498.
[9] Lee Y, Park J. CenterMask : Real-Time Anchor-Free instance segmentation[C]//2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR),2020.
[10] Tan M, Le Q V. MixConv: Mixed depthwise convolutional kernels[EB]. arXiv:1907.09595,2019.
[11] Cai H, Zhu L, Han S. Proxylessnas: Direct neural architecture search on target task and hardware[EB]. arXiv:1812.00332,2018.
[12] Howard A G, Zhu M, Chen B, et al. MobileNets: Efficient convolutional neural networks for mobile vision applications[EB]. arXiv:1704.04861,2017.
[13] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions[C]//2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR),2015.
[14] 麻森权,周克. 基于注意力机制和特征融合改进的小目标检测算法[J]. 计算机应用与软件,2020,37(5):194 - 199.
[15] Hu J, Shen L, Albanie S, et al. Squeeze-and-Excitation networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence,2020,42(8):2011 - 2023.