基于增量式鲁棒非负矩阵分解的短文本在线聚类

贺超波[1],汤 庸[2],张 琼[3],刘双印[1],刘 海[2]
(1. 仲恺农业工程学院信息科学与技术学院,广东广州 510225; 2. 华南师范大学计算机学院,广东广州 510631; 3. 中山大学数据科学与计算机学院,广东广州 510006)

摘 要: 对社会化媒体产生的大量短文本进行聚类分析具有重要的应用价值,但短文本往往具有噪音数据多、增长迅速且数据量大的特点,导致现有相关算法难于有效处理. 提出一种基于增量式鲁棒非负矩阵分解的短文本在线聚类算法 STOCIRNMF. STOCIRNMF 基于非负矩阵分解构建短文本聚类模型,通过 $l_{2,1}$ 范数设计模型的优化求解目标函数提高鲁棒性,同时应用增量式迭代更新规则实现短文本的在线聚类. 在搜狐新闻标题和微博短文本数据集上进行相关实验,结果表明 STOCIRNMF 不仅比现有代表性算法具有更好的聚类性能,而且能够有效对微博话题进行在线检测.

关键词: 短文本聚类; 鲁棒非负矩阵分解; 在线聚类; $l_{2,1}$ 范数; 增量式迭代更新规则
中图分类号: TP391 **文献标识码:** A **文章编号:** 0372-2112 (2019)05-1086-08
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2019.05.016

Short Text Online Clustering Based on Incremental Robust Nonnegative Matrix Factorization

HE Chao-bo[1], TANG Yong[2], ZHANG Qiong[3], LIU Shuang-yin[1], LIU Hai[2]
(1. *School of Information Science and Technology, Zhongkai University of Agriculture and Engineering, Guangzhou, Guangdong* 510225, *China*;
2. *School of Computer, South China Normal University, Guangzhou, Guangdong* 510631, *China*;
3. *School of Data and Computer Science, Sun Yat-sen University, Guangzhou, Guangdong* 510006, *China*)

Abstract: Clustering a large number of short texts in social media has great value in applications. However, short texts often have these characteristics: lots of noises, growing rapidly and massive data. Most existing short text clustering algorithms are not effectively enough to process such short texts. Aiming at this problem, we propose an algorithm of short text online clustering based on incremental robust nonnegative matrix factorization (STOCIRNMF). This algorithm uses NMF to build the short text clustering model and applies $l_{2,1}$ norm to devise its objective function for improving its robustness. Meanwhile, STOCIRNMF can cluster short texts incrementally by using incremental iterative update rules. We conduct extensive experiments on real Sohu news titles and Weibo datasets. The results show that STOCIRNMF not only has better performance of short text clustering than some representative algorithms, but also is very effective to detect micro blog's topics online.

Key words: short text clustering; robust nonnegative matrix factorization; online clustering; $l_{2,1}$ norm; incremental iterative update rules

1 引言

随着社会化媒体时代的到来以及移动智能终端的普及,广大用户可以比以往更加方便快捷地创造各种形式的互联网内容,而短文本(Short text)是其中最为常见的一种内容形式,例如门户网站的新闻标题、电子商

收稿日期:2018-05-31;修回日期:2018-08-19;责任编辑:蓝红杰
基金项目:国家自然科学基金(No. 61772211);广东省科技计划项目(No. 2017A040405057, No. 2017A030303074, No. 2016A030303058);广州市科技计划项目(No. 201807010043)

务平台的用户评论文本以及社交媒体的用户动态文本等都属于短文本类别. 对短文本进行聚类分析具有很强的应用价值,例如可以对用户评论进行观点挖掘[1],对社交媒体进行话题检测[2]以及情感分析[3]等. 由于短文本文字简短,表达灵活多样且增长迅速,往往具有特征难于提取、特征稀疏、噪音数据多以及数据量大等特点,传统的适合长文本以及小数据量的聚类算法,包括 K-means、LDA 以及 PLSA 等都难于处理,为此需要研究有效的短文本聚类算法. 目前国内外研究人员对短文本聚类算法已有相关研究,但大部分都集中在特征提取[4]、数据降维[5]以及噪声平滑[6]等问题的研究上,虽然都在不同程度上提高了短文本聚类质量,但都忽视了聚类算法的效率优化问题,往往难于快速处理增长迅速的短文本数据. 现实中的社交媒体短文本(如微博和微信的用户动态文本)增长迅速且规模巨大,常爆发许多具有时效性的热门话题,这迫切需要研究一种不仅具有聚类质量保证而且具有快速数据处理能力的短文本聚类算法.

非负矩阵分解(Nonnegative Matrix Factorization, NMF)是一种低秩矩阵近似分解模型,已被证明与 K-means 聚类算法密切相关,具有强大的数据聚类能力并且可以比 K-means、LDA 以及 PLSA 等算法获得更好的文本聚类质量[7,8]. 此外,NMF 还具有可进行增量式聚类的特点,可以应用于一些需要进行在线处理的数据分析任务[9,10]. NMF 具备的这些特点表明其可以应用于解决短文本聚类存在的问题,为此本文提出一种基于增量式鲁棒非负矩阵分解(Incremental Robust NMF, IRNMF)的短文本在线聚类算法:STOCIRNMF(Short text online clustering based on IRNMF),所做的主要工作包括以下三点. (1)基于 NMF 设计一种短文本聚类模型,采用 $l_{2,1}$ 范数设计优化求解目标函数提高模型的鲁棒性,并应用增量式迭代更新规则对该模型进行优化求解. (2)对增量式迭代更新规则的收敛性进行严格证明,并设计了可动态调整聚类数目且具有常数级时间复杂度的增量式短文本在线聚类算法 STOCIRNMF. (3)在搜狐新闻标题以及微博短文本数据集上进行了大量实验,结果表明 STOCIRNMF 不仅比现有代表性算法具有更好的聚类性能,而且能够有效对微博话题进行在线检测.

2　基于 IRNMF 的短文本在线聚类

2.1　短文本聚类模型

假设短文本集合 $\boldsymbol{S}=\{s_1,s_2,\cdots,s_n\}$,特征词词典 $T=\{t_1,t_2,\cdots,t_m\}$,那么文本和特征词可以构成一个词项-文本特征矩阵 $\boldsymbol{X}=[x_1,x_2,\cdots,x_n]\in\mathbb{R}_+^{m\times n}$. 对于 $\forall x_{ij}\in\boldsymbol{X}$ 的取值,虽然目前已有一些更为准确的度量方法,其中包括 TF-IDF、NCW (Normalized Cut Weight)[11] 以及 word2vec[12]等,但这些度量方法在短文本动态增长情况下需要重复计算,而且时间复杂度高,不能满足短文本在线聚类对于运行效率的要求. 为此,本文采用更为简单有效的 TF(Term Frequency)度量方法计算 x_{ij},即将 x_{ij} 表示为 t_i 在短文本 s_j 出现的次数. $\boldsymbol{X}$ 为非负值矩阵,假设短文本的聚类数目设置为 k,采用 $l_{2,1}$ 范数,可以设计如下基于 NMF 的短文本聚类模型 RNMF(Robust NMF):

$$\min_{\boldsymbol{P}\geq 0,\boldsymbol{Q}\geq 0}\{\boldsymbol{J}(\boldsymbol{X},\boldsymbol{PQ})=\|\boldsymbol{X}-\boldsymbol{PQ}\|_{2,1}\} \tag{1}$$

其中,$\boldsymbol{P}\in\mathbb{R}_+^{m\times k}$ 表示为特征词聚类指示矩阵,对于 $\forall p_{ai}\in\boldsymbol{P}$,其值越大则表示特征词项 t_a 隶属聚类 i 的强度越大;$\boldsymbol{Q}\in\mathbb{R}_+^{k\times n}$ 表示短文本聚类指示矩阵,对于 $\forall q_{ib}\in\boldsymbol{Q}$,其值越大则短文本 s_b 隶属聚类 i 的强度越大. 式(1)对于近似分解误差的计算可以表示为:

$$\|\boldsymbol{X}-\boldsymbol{PQ}\|_{2,1}=\sum_{i=1}^{n}\sqrt{\sum_{j=1}^{m}(\boldsymbol{X}-\boldsymbol{PQ})_{ji}^2}=\sum_{i=1}^{n}\|x_i-\boldsymbol{P}q_i\| \tag{2}$$

可以看出,式(2)对于误差没有进行平方计算,因此可以降低短文本噪声数据对于目标函数 $J(\boldsymbol{X},\boldsymbol{PQ})$ 的影响. 此外,根据式(2)可以判断,采用 $l_{2,1}$ 范数计算最小化误差可以令 q_i 的元素获得更多的 0 值,即可以提高 $\boldsymbol{Q}$ 的稀疏度. 而事实上,在大规模的短文本聚类任务中,$\boldsymbol{Q}$ 也是高度稀疏矩阵,因为即使考虑软聚类的可能性,对于任意一个短文本,都最多隶属少数几个类别. 综上分析可知,采用 $l_{2,1}$ 范数设计短文本聚类模型 RNMF,不仅在理论上可以提高模型的鲁棒性,而且还可以提高聚类指示矩阵 $\boldsymbol{Q}$ 的稀疏度,有利于减少模型求解的计算数据量并提高计算效率. 令 $\boldsymbol{D}=[d_{ii}]^{n\times n}$ 表示为一个对角矩阵,其中 $d_{ii}=1/\|x_i-\boldsymbol{P}q_i\|$,则 $J(\boldsymbol{X},\boldsymbol{PQ})$ 可以重写为:

$$J(\boldsymbol{X},\boldsymbol{PQ})=\mathrm{tr}(\boldsymbol{X}-\boldsymbol{PQ})\boldsymbol{D}(\boldsymbol{X}-\boldsymbol{PQ})^{\mathrm{T}} \tag{3}$$

$\min J(\boldsymbol{X},\boldsymbol{PQ})$ 可以转化为典型的受限约束求极值问题,根据文献[7]提出的优化求解策略可以获得 $\boldsymbol{P}$ 和 $\boldsymbol{Q}$ 的迭代更新规则为:

$$p_{ai}=p_{ai}\frac{(\boldsymbol{XDQ}^{\mathrm{T}})_{ai}}{(\boldsymbol{PQDQ}^{\mathrm{T}})_{ai}} \tag{4}$$

$$q_{ib}=q_{ib}\frac{(\boldsymbol{P}^{\mathrm{T}}\boldsymbol{XD})_{ib}}{(\boldsymbol{P}^{\mathrm{T}}\boldsymbol{PQD})_{ib}} \tag{5}$$

迭代应用式(4)和(5)使 $J(\boldsymbol{X},\boldsymbol{PQ})$ 收敛后可以获得 $\boldsymbol{P}$ 和 $\boldsymbol{Q}$ 的局部最优解,假设收敛需要的迭代次数为 t,则可以设计如下短文本聚类模型 RNMF 的迭代更新求解算法,如算法 1 所示.

算法 1　RNMF 迭代更新求解算法

输入:X,k,t

输出:$\boldsymbol{P},\boldsymbol{Q}$
步骤:
1. 随机初始化非负值矩阵 $\boldsymbol{P}$ 和 $\boldsymbol{Q}$;
2. $iter \leftarrow 0$;
3. While ($iter < t$)
4. 分别应用式(4)和(5)更新 $\forall p_{ai} \in \boldsymbol{P}$ 和 $\forall q_{ib} \in \boldsymbol{Q}$;
5. $iter = iter + 1$;
6. End While

2.2 增量式迭代更新规则

算法 1 的计算负载都集中在行 3 至行 5 所表示的迭代更新规则的计算上,其每次迭代所需要的时间复杂度为 $O(nmk)$,由于 m 值固定,该时间复杂度与短文本数量 n 成正比. 算法 1 虽然具有线性时间复杂度,但其并不适合在线处理动态增长的短文本聚类任务,原因在于当增加新的短文本时,算法 1 需要完全重新计算 $\boldsymbol{P}$ 和 $\boldsymbol{Q}$,每次迭代所需要的计算时间 $O(nmk)$ 也将由于 n 值的增加而不断增长. 当 n 值很大时,算法 1 运行效率将极其低下,甚至由于存储和计算资源的限制而不能正常运行. 为此,本文设计了一种增量式的迭代更新规则,可以利用 $\boldsymbol{P}$ 和 $\boldsymbol{Q}$ 的历史计算结果降低短文本数量增加时 $\boldsymbol{P}$ 和 $\boldsymbol{Q}$ 的迭代更新时间,从而可以提高算法 1 的运行效率. $\boldsymbol{P}$ 和 $\boldsymbol{Q}$ 的增量式迭代更新规则的推导过程介绍如下:

令 $\boldsymbol{X}_r$、$\boldsymbol{P}_r$ 和 $\boldsymbol{Q}_r$ 分别表示基于前 r 个短文本构建的词项-文本特征矩阵及其分解后的矩阵 $\boldsymbol{P}$ 和 $\boldsymbol{Q}$,J_r 为对应的目标函数,则有:

$$J_r = \|\boldsymbol{X}_r - \boldsymbol{P}_r\boldsymbol{Q}_r\|_{2,1} \tag{6}$$

同理,当第 $r+1$ 个短文本加入时,J_{r+1} 为:

$$J_{r+1} = \|\boldsymbol{X}_{r+1} - \boldsymbol{P}_{r+1}\boldsymbol{Q}_{r+1}\|_{2,1} \tag{7}$$

当短文本数量 r 增长到足够大时,作为特征词聚类指示矩阵的 $\boldsymbol{P}$ 将保持相对稳定,即新增加的短文本对 $\boldsymbol{P}$ 的影响较小,从而 J_r 可以做如下近似:

$$J_r \cong \|\boldsymbol{X}_r - \boldsymbol{P}_{r+1}\boldsymbol{Q}_r\|_{2,1} \tag{8}$$

综合式(7)和式(8)可以得到 J_{r+1} 与 J_r 的近似关系为:

$$J_{r+1} \cong J_r + \|x_{r+1} - \boldsymbol{P}_{r+1}q_{r+1}\|_{2,1} = J_r + f_{r+1} \tag{9}$$

其中 $f_{r+1} = \|x_{r+1} - \boldsymbol{P}_{r+1}q_{r+1}\|_{2,1}$ 为增量部分,令 $\boldsymbol{D}_r = [d_{ii}]^{r\times r}$ 表示为一个对角矩阵,其中 $d_{ii} = 1/\|x_i - \boldsymbol{P}_{r+1}q_i\|$,并令 $d = 1/\|x_{r+1} - \boldsymbol{P}_{r+1}q_{r+1}\|$,采用梯度下降法得到式(9)中 q_{r+1} 任一个元素 $(q_{r+1})_i$ 的增量式迭代更新规则为:

$$(q_{r+1})_i = (q_{r+1})_i - \mu_i \frac{\partial J_{r+1}}{\partial (q_{r+1})_i} \tag{10}$$

其中,$\frac{\partial J_{r+1}}{\partial (q_{r+1})_i} = \frac{\partial f_{r+1}}{\partial (q_{r+1})_i} = (-2\boldsymbol{P}_{r+1}^{\mathrm{T}}x_{r+1}d + 2\boldsymbol{P}_{r+1}^{\mathrm{T}}\boldsymbol{P}_{r+1}q_{r+1}d)_i$. 令 $\mu_i = \frac{(q_{r+1})_i}{(2\boldsymbol{P}_{r+1}^{\mathrm{T}}\boldsymbol{P}_{r+1}q_{r+1}d)_i}$,则式(10)可以重写为:

$$(q_{r+1})_i = (q_{r+1})_i \frac{(\boldsymbol{P}_{r+1}^{\mathrm{T}}x_{r+1})_i}{(\boldsymbol{P}_{r+1}^{\mathrm{T}}\boldsymbol{P}_{r+1}q_{r+1})_i} \tag{11}$$

同理,可以求得 $\boldsymbol{P}_{r+1}$ 任一个元素 $(p_{r+1})_{ai}$ 的增量式迭代更新规则为:

$$(p_{r+1})_{ai} = (p_{r+1})_{ai} \frac{(\boldsymbol{X}_r\boldsymbol{D}_r\boldsymbol{Q}_r^{\mathrm{T}} + x_{r+1}dq_{r+1}^{\mathrm{T}})_{ai}}{(\boldsymbol{P}_{r+1}\boldsymbol{Q}_r\boldsymbol{D}_r\boldsymbol{Q}^{\mathrm{T}}r + \boldsymbol{P}_{r+1}q_{r+1}dq_{r+1}^{\mathrm{T}})_{ai}} \tag{12}$$

可推导 $\boldsymbol{X}_{r+1}\boldsymbol{D}_{r+1}\boldsymbol{Q}_{r+1}^{\mathrm{T}} = \boldsymbol{X}_r\boldsymbol{D}_r\boldsymbol{Q}_r^{\mathrm{T}} + x_{r+1}dq_{r+1}^{\mathrm{T}}$ 和 $\boldsymbol{Q}_{r+1}\boldsymbol{D}_{r+1}\boldsymbol{Q}_{r+1}^{\mathrm{T}} = \boldsymbol{Q}_r\boldsymbol{D}_r\boldsymbol{Q}_r^{\mathrm{T}} + q_{r+1}dq_{r+1}^{\mathrm{T}}$,因此式(12)通过保存新增数据前的 $\boldsymbol{X}_r\boldsymbol{D}_r\boldsymbol{Q}_r^{\mathrm{T}}$ 和 $\boldsymbol{Q}_r\boldsymbol{D}_r\boldsymbol{Q}_r^{\mathrm{T}}$ 计算结果可以大幅降低计算时间复杂度.

2.3 在线聚类算法

短文本的在线聚类包含 2 个阶段:初始化阶段和增量式聚类阶段,其中初始化阶段取 r 个短文本进行基于 RNMF 的聚类,用于获取 $\boldsymbol{X}_r$、$\boldsymbol{P}_r$、$\boldsymbol{Q}_r$ 以及 k 的初始值,增量式聚类阶段则对后续动态增长的短文本进行增量聚类处理. 大规模短文本在动态增长过程中聚类数目 k 也会动态增加,例如,微博和微信等社交媒体总是在不断产生新的热门话题,因此增量式聚类阶段需要随着短文本数量的增多对 k 值进行动态调整. NMF 已被证明与 K-means 聚类算法紧密相关,$\boldsymbol{P}$ 的每一个列向量 p_i($i = 1,\cdots,k$)都可以作为对应聚类的质心. 对于新增的第 $r+1$ 个短文本特征向量 x_{r+1},如果 k 值不准确,那么 x_{r+1} 将同任何一个 p_i 都存在较大的距离,即 x_{r+1} 应该归入新的一个聚类并作为新聚类的初始质心,因此可以采取如下策略对 k 值进行动态设置:每次进行第 $r+1$ 个短文本的增量式聚类前,计算跟每个聚类质心的最小距离 $\Delta = \min\|x_{r+1} - p_i\|(i = 1,\cdots,k)$,当 Δ 大于阈值 ϕ 时,k 值增加 1,然后再执行基于 IRNMF 的在线聚类. 通过这种策略既可以简单有效动态调整 k 值,同时又可以保证在线聚类需要的时间效率,因为每次与各个聚类质心的距离计算仅需要 O(mk)的时间复杂度. 根据以上思路,设计了如算法 2 所示短文本在线聚类算法 STOCIRNMF.

算法 2 短文本在线聚类算法 STOCIRNMF

输入:$\boldsymbol{S},\boldsymbol{T},k,t,\phi$
输出:$C = \{c_1,c_2,\cdots\}$ //短文本聚类结果
步骤:
//初始化阶段
1. 取前 r 个短文本$\{s_1,s_2,\cdots,s_r\}$计算 $\boldsymbol{X}_r$;
2. 应用算法 1 计算 $\boldsymbol{P}_r$ 和 $\boldsymbol{Q}_r$;
3. 优化调整 k 值作为初始值;
//增量式聚类阶段
4. While $s_{r+1} \in \boldsymbol{S}$ do
5. 对 $\boldsymbol{X}_r$ 增加第 $r+1$ 列:$\boldsymbol{X}_{r+1} = [x_1,x_2,\cdots,x_r,x_{r+1}]$;

6. 对 $\boldsymbol{Q}_r$ 增加第 $r+1$ 列:$\boldsymbol{Q}_{r+1}=[q_1,q_2,\cdots,q_r,q_{r+1}]$,并随机初始化 q_{r+1};
7. $\boldsymbol{P}_{r+1}=\boldsymbol{P}_r$;
8. $\forall i=1,\cdots,k,\Delta=\min\|x_{r+1}-p_i\|$;
9. If $\Delta>\phi$ then
10. $k=k+1$;
11. 对 $\boldsymbol{P}_{r+1}$增加 1 列 $pk=x_{r+1}$,$\boldsymbol{Q}_{r+1}$则增加 1 个元素全为 0 的行;
12. End If
13. $iter=1$;
14. While ($iter<t$)
15. $\forall i=1,\cdots,k$,执行式(11);
16. $\forall a=1,\cdots,m$ and $\forall i=1,\cdots,k$,执行式(12);
17. $iter=iter+1$;
18. End While
19. $j=\underset{i}{\operatorname{argmax}}(q_{r+1})i$;//判断短文本 s_{r+1}的所属类别
20. $c_j=c_j\cup\{s_{r+1}\}$;
21. $\boldsymbol{X}_{r+1}\boldsymbol{D}_{r+1}\boldsymbol{Q}_{r+1}^{\mathrm{T}}=\boldsymbol{X}_r\boldsymbol{D}_r\boldsymbol{Q}_r{}^{\mathrm{T}}+x_{r+1}dq_{r+1}^{\mathrm{T}}$;//增量式计算并存储 $\boldsymbol{X}_{r+1}\boldsymbol{D}_{r+1}\boldsymbol{Q}_{r+1}^{\mathrm{T}}$
22. $\boldsymbol{Q}_{r+1}\boldsymbol{D}_{r+1}\boldsymbol{Q}_{r+1}^{\mathrm{T}}=\boldsymbol{Q}_r\boldsymbol{D}_r\boldsymbol{Q}_r{}^{\mathrm{T}}+q_{r+1}dq_{r+1}^{\mathrm{T}}$;//增量式计算并存储 $\boldsymbol{Q}_{r+1}\boldsymbol{D}_{r+1}\boldsymbol{Q}_{r+1}^{\mathrm{T}}$
23. $r=r+1$;
24. End While

当需要处理大规模短文本数据集时,STOCIRNMF 的计算复杂度都集中于增量式聚类阶段. 首先对于时间复杂度,可计算步骤 8 的时间复杂度为 $O(mk)$,步骤 15 的迭代更新时间复杂度为 $O(mk)$,步骤 16 由于在步骤 21 ~ 22 预先计算并存储 $\boldsymbol{X}_{r+1}\boldsymbol{D}_{r+1}\boldsymbol{Q}_{r+1}^{\mathrm{T}}$ 和 $\boldsymbol{Q}_{r+1}\boldsymbol{D}_{r+1}\boldsymbol{Q}_{r+1}^{\mathrm{T}}$,时间复杂度可以从 $O(rmk)$降为 $O(mk^2)$($k\ll r$). 综合计算可知 STOCIRNMF 每次处理一个新增短文本的聚类需要的时间复杂度为 $O(tmk^2)$,是一个常数级且与大规模短文本数量无关的时间复杂度. 事实上,STOCIRNMF 还适合于通过批处理方式,即每次同时对新增的一批短文本进行聚类进一步提高效率. 在空间存储开销方面,主要需要存储 $\boldsymbol{X}_r$、$\boldsymbol{P}_r$、$\boldsymbol{Q}_r$、$\boldsymbol{X}_r\boldsymbol{D}_r\boldsymbol{Q}_r^{\mathrm{T}}$ 以及 $\boldsymbol{Q}_r\boldsymbol{D}_r\boldsymbol{Q}_r^{\mathrm{T}}$,空间复杂度分别为 $O(m_r)$、$O(mk)$、$O(k_r)$、$O(mk)$以及 $O(k^2)$,因此 STOCIRNMF 的空间复杂度为 $O(2mk+mr+k_r+k^2)$. 由于 $\boldsymbol{X}_r$、$\boldsymbol{P}_r$、$\boldsymbol{D}_r$ 以及 $\boldsymbol{Q}_r$ 均为稀疏矩阵,在采用稀疏存储的情况下,空间复杂度还可以进一步降低. 需要指出的是,由于 k 可以动态增长,因此 STOCIRNMF 的实际时间和空间开销也是动态变化的.

3　实验分析

实验使用的短文本分词器为 IKAnalyzer 2012,编程语言为 Java,运行机器的配置为:操作系统 Win7 64 位、Inter Core i7-6700 3.4G CPU、32GB 内存以及 2TB 硬盘.

3.1　数据集与比较方法

实验采用了 3 个短文本数据集,一个来源于搜狗实验室的搜狐新闻数据(http://www.sogou.com/labs),另外两个来源于中文信息学会的 SMP2015 新浪微博数据集. 搜狐新闻数据包含了 20 个栏目 2008 年的新闻数据,实验随机选取了其中 10 个类别共计 42312 条新闻的标题作为 Sohu 新闻短文本数据集,各个分类名称及包含的短文本数量如表 1 所示. SMP2015 微博数据集包含约二十亿条微博内容,实验按预先设置的 10 个区分度高的主题词分别抽取相关微博共计 74318 条作为 WeiboA 短文本数据集. WeiboA 中的每一条微博关联的主题词作为类别标签,其各个分类名称及包含的短文本数量如表 2 所示. 此外,实验选取了 2011 年 7 月 23 日至 2011 年 7 月 25 日三天共 141340 条微博作为 WeiboB 短文本数据集. 由于 Sohu 和 WeiboA 数据集中的每一个短文本都已标注了类别标签,这 2 个数据集可用于验证 STOCIRNMF 的短文本聚类性能,而 WeiboB 每一条微博短文本的类别未知,可以用于测试 STOCIRNMF 进行微博话题在线检测的能力.

表 1　Sohu 新闻短文本数据集

类别名称	短文本数量	类别名称	短文本数量
House	6942	Learning	1636
Yule	6045	Business	7961
Health	1373	Travel	2055
IT	2753	Sports	9849
Women	1946	Auto	1752

表 2　WeiboA 短文本数据集

类别名称	短文本数量	类别名称	短文本数量
小米	11569	房价	8935
韩剧	7515	公务员	7572
恒大	8080	转基因	5965
雾霾	5955	反腐	6835
法网	6143	乔布斯	5749

为对比验证本文算法 STOCIRNMF 在 Sohu 新闻和 WeiboA 短文本数据集的聚类性能,选择了 5 种具有代表性的短文本聚类算法进行比较,分别是 K-means、LDA、NMF、RNMF 和 IncreSTS[13],其中 K-means、LDA、NMF、RNMF 均是完全聚类算法,而 IncreSTS 是增量式算法. NMF 采用 Frobenius 范数设计目标函数,RNMF 则采用 $l_{2,1}$范数设计目标函数,RNMF 实际上等同于非增量式的 STOCIRNMF 算法. 需要指出的是,以上五种算法均与 STOCIRNMF 一样采用 TF 作为短文本词项特征度量.

3.2　聚类性能比较

聚类性能评价方面包括聚类质量和运行时间,其中聚类质量则采用准确率(Accuracy)和 NMI(Normalized Mutual Information)两种常用评价准则. 由于 Sohu 新闻和 WeiboA 短文本数据集中的每一条短文本都具有分类标签,因此同时可以用于比较各种聚类算法的

聚类质量和运行时间. 考虑到 K-means、LDA、NMF、RNMF 和 IncreSTS 均需要预先指定聚类数目,为方便比较,这五种算法的聚类数目 k 均分别设置成与 Sohu 新闻和 WeiboA 短文本数据集的类别数目一致(即 k = 10),而 STOCIRNMF 的初始 k 值也设置为 10,ϕ 值均为 500. 对比实验中,各个算法均首先选取各数据集前 1 万条短文本进行聚类,然后每次新增 1 万条对剩余短文本进行聚类. 需要指出的是 K-means、LDA、NMF 以及 RNMF 每次都进行完全聚类,而 IncreSTS 和 STOCIRNMF 进行增量式聚类. 图 1 ~ 图 4 分别为各种算法在两个数据集不同短文本数量情况下的 Accuracy 和 NMI 对比情况,图 5 和图 6 则为相应的运行时间对比情况.

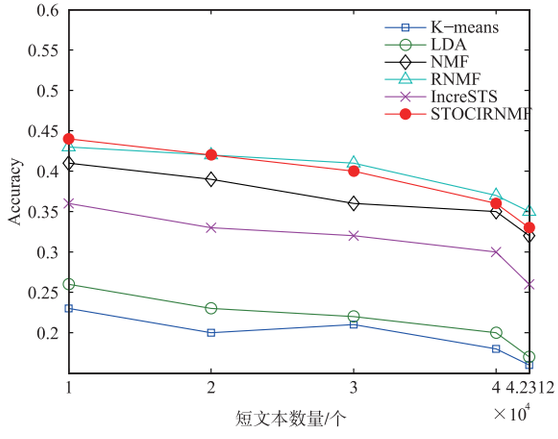

图1 Sohu的Accuracy对比

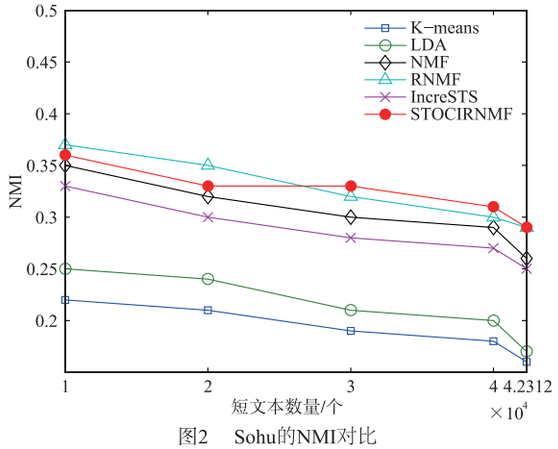

图2 Sohu的NMI对比

首先在聚类质量对比方面,从图 1 ~ 图 4 可以看出,在两个数据集的各个不同短文本数量下 STOCIRNMF、RNMF 以及 NMF 的 Accuracy 以及 NMI 均要优于 K-means 和 LDA,这表明基于 NMF 的短文本聚类算法比传统的聚类算法 K-means 和 LDA 更适合处理短文本. 此外,可以发现虽然在 Sohu 数据集上 STOCIRNMF、RNMF 以及 NMF 的聚类质量较为接近,但在 WeiboA 数

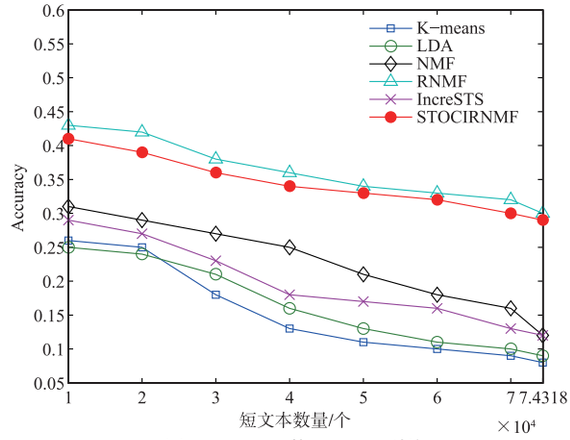

图3 WeiboA的Accuracy对比

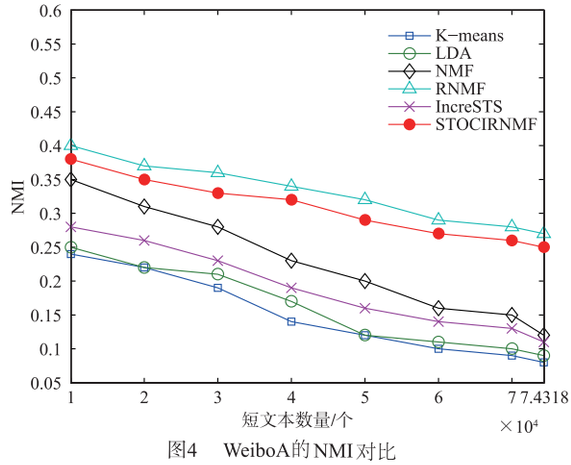

图4 WeiboA的NMI对比

据集上 STOCIRNMF、RNMF 要明显优于 NMF,其中 STOCIRNMF 的 Accuracy 和 NMI 的平均值分别比 NMF 提高了 53.1% 和 36.2%,而 RNMF 的 Accuracy 和 NMI 的平均值分别比 NMF 提高了 60.9% 和 46.1%. 原因在于微博内容的撰写较为随意而新闻标题比较严谨规范,因此 WeiboA 数据集比 Sohu 数据集存在更多的噪音数据,而采用 $l_{2,1}$ 范数的 STOCIRNMF 和 RNMF 比采用 Frobenius 范数的 NMF 具有更好的鲁棒性,可以降低噪声数据对于聚类质量的影响. 鲁棒性的强弱同样影响到同为增量式算法的 IncreSTS 的聚类质量,IncreSTS 并没有采用任何鲁棒性机制,其在 WeiboA 数据集上的 Accuracy 和 NMI 平均值也均远低于 STOCIRNMF 和 RNMF. 对于 STOCIRNMF 和 RNMF,RNMF 在各个数据集上的 Accuracy 和 NMI 的平均值均要略高于 STOCIRNMF,但差距均小于 0.02,因此两者在各个数据集上的聚类质量总体上都是比较接近的.

其次在运行时间对比方面,从图 5 ~ 图 6 可以看出,在两个数据集上 STOCIRNMF 均要明显优于采用完全聚类的 K-means、LDA、NMF 和 RNMF,并且也优于同

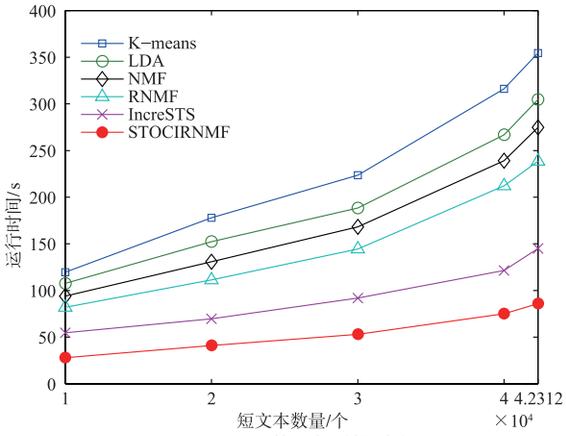

图5　Sohu的运行时间对比

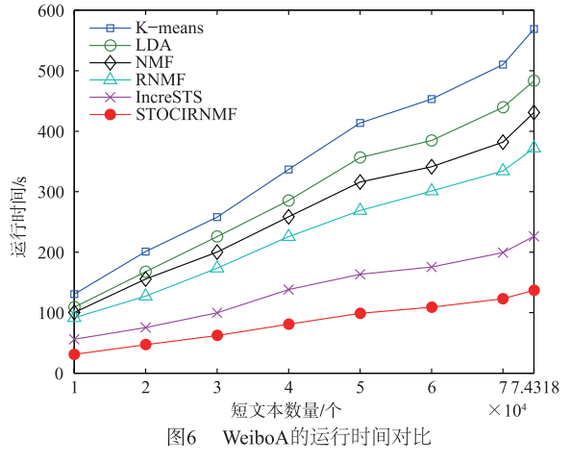

图6　WeiboA的运行时间对比

样采用增量式聚类的 IncreSTS. 例如,在 WeiboA 数据集中,当短文本数量达到 70000 条时,STOCIRNMF 仅需要 123s,而 IncreSTS、K-means、LDA、NMF 以及 RNMF 分别需要 202s、513s、441s、388s 和 337s. 虽然 IncreSTS 也采用增量式机制进行聚类,但由于其聚类过程需要计算大量短文本的两两相似度,从而导致其运行效率的提高并不明显. 此外,虽然在两个数据集上 RNMF 的聚类质量要略高于 STOCIRNMF,但 STOCIRNMF 运行效率的提升优势更为明显,例如,STOCIRNMF 在 WeiboA 数据集上的运行时间平均值比 RNMF 少了 151s,平均运行时间缩短了 63.7%. 由此可知,虽然 STOCIRNMF 比 RNMF 损失了一点聚类质量,但采用增量式聚类机制大幅提高了运行效率.

表 3 和表 4 分别为 STOCIRNMF 在 Sohu 和 WeiboA 数据集获取的各个聚类的典型主题词,主题词后面括号中的数值表示隶属强度,来自于矩阵 $\boldsymbol{P}$ 中的对应元素. 从表 3 和表 4 可以看出 STOCIRNMF 算法获取的主题词均是与相应聚类主题高度相关的,例如表 3 的 Sports 聚类中的主题词均与奥运相关,这是因为 Sohu 新闻短文本数据集为 2008 年的新闻数据,当时 Sports 栏目主要报道 2008 年北京奥运会.

表 3　Sohu 新闻短文本聚类的典型主题词

House	Yule	Health	IT	Women
楼市(0.33)	电影节(0.37)	医院(0.45)	网络(0.37)	时尚(0.32)
房价(0.26)	导演(0.35)	健康(0.41)	电信(0.35)	发型(0.31)
房地产(0.23)	电视剧(0.32)	治疗(0.39)	手机(0.29)	性感(0.28)
销售(0.19)	颁奖(0.29)	专家(0.35)	视频(0.25)	减肥(0.25)
开发商(0.17)	影片(0.27)	药品(0.29)	软件(0.21)	美容(0.23)
Learning	**Business**	**Travel**	**Sports**	**Auto**
高考(0.41)	董事会(0.43)	丽江(0.33)	奥运会(0.47)	车市(0.34)
答案(0.38)	股东大会(0.38)	香格里拉(0.28)	选手(0.42)	新车(0.32)
试题(0.34)	股份(0.36)	青海湖(0.23)	门票(0.41)	车展(0.28)
文科(0.29)	ST(0.31)	桂林(0.19)	志愿者(0.39)	品牌(0.25)
招生(0.26)	波动(0.29)	西宁(0.15)	直播(0.37)	油价(0.23)

表 4　WeiboA 短文本聚类的典型主题词

小米	韩剧	恒大	雾霾	法网
手机(0.33)	韩国(0.37)	足球(0.45)	浓度(0.37)	网球(0.41)
销售(0.26)	电视剧(0.35)	联赛(0.41)	颗粒(0.35)	法国(0.33)
内存(0.23)	文化(0.32)	亚洲(0.39)	北京(0.29)	巴黎(0.25)
屏幕(0.19)	韩流(0.29)	俱乐部(0.35)	预警(0.25)	决赛(0.21)
像素(0.17)	演员(0.27)	冠军(0.29)	污染(0.21)	冠军(0.18)
房价	**公务员**	**转基因**	**反腐**	**乔布斯**
房地产(0.41)	考试(0.43)	基因(0.33)	腐败(0.47)	苹果(0.46)
市场(0.38)	政府(0.38)	食品(0.28)	贪官(0.42)	公司(0.41)
成交(0.34)	部门(0.36)	DNA(0.23)	廉洁(0.41)	去世(0.37)
城市(0.29)	职位(0.31)	技术(0.19)	贪污(0.39)	电脑(0.28)
走势(0.26)	报考(0.29)	危害(0.15)	纪律(0.37)	悼念(0.21)

3.3　微博话题在线检测

使用 WeiboB 数据集测试 STOCIRNMF 进行微博话题在线检测的能力,实验首先选取前 1 万条微博短文本作为初始集(即 $r=10000$),然后依次加入剩余微博短文本进行在线聚类,该实验实际上模拟了一个进行微博短文本流在线聚类的应用场景,聚类结果即为微博话题. STOCIRNMF 的初始 k 值设置为 20,迭代次数 $t=50$,ϕ 值的设置对在线聚类具有重要影响,为方便进行比较分析,ϕ 值分别设置为 100,300 和 500,其对应的 k 值动态变化结果如图 7 所示,而相应的运行时间和采用稀疏存储的存储空间变化结果分别如图 8 和图 9 所示.

首先分析 ϕ 值对于在线聚类效率的影响,从图 7 可以看出,随着微博短文本数量的增加,ϕ 取值为 100 和 300 时 k 值都可以动态增加,但 $\phi=500$ 时 k 值没有变化. 此外,$\phi=100$ 时的 k 值增加数量比 $\phi=300$ 时大. 这些发现说明实际操作中通过灵活设置 ϕ 值可以对微博话题的检测数量进行控制,同时也证明 STOCIRNMF 可以较好地适应微博话题数量的动态变化. 图 8 中 $\phi=100$ 和 $\phi=300$ 的时间消耗变化曲线总体上都呈阶梯状

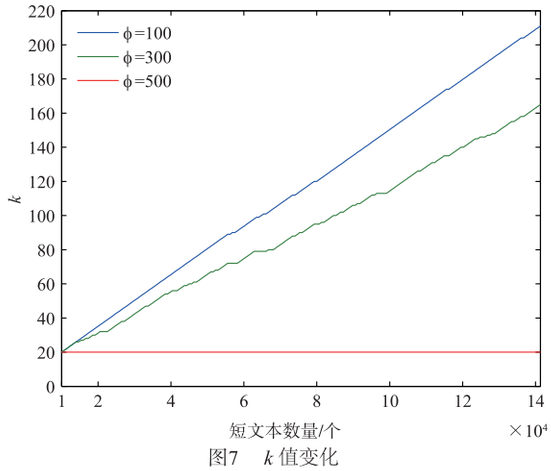

图7 k 值变化

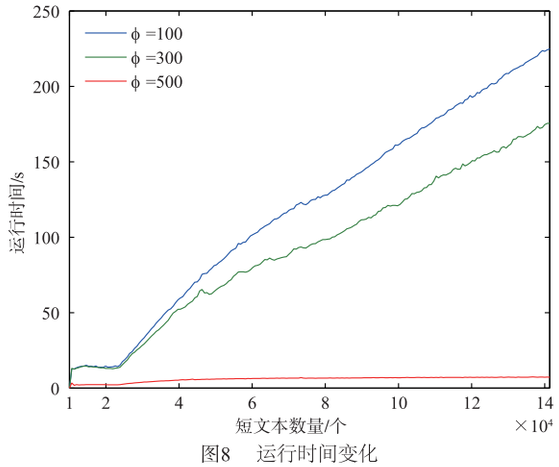

图8 运行时间变化

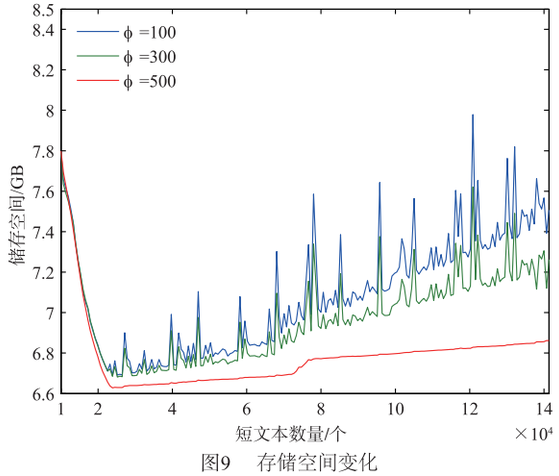

图9 存储空间变化

增长而非近似直线,这是因为 STOCIRNMF 的时间复杂度为 $O(tmk^2)$,而它们的 k 值会阶段性动态增加,因此所需时间也会阶段性增加,并且 k 值增加更多则需要更多的时间消耗. $\phi=500$ 时由于 k 值没有变化,其对应的时间消耗变化曲线呈近似直线,这也证明了 STOCIRNMF 的 k 值不变化时可获得常数级时间复杂度. 图9 展示的各个存储空间消耗变化曲线总体上是没有规律的,都存在数据量局部下降或者上升的情况,这是因为每次对新增数据处理后得到的相关矩阵稀疏度的高低是不固定的,尤其是 $\boldsymbol{P}$ 矩阵,其稀疏度的变化还受 k 值的影响. 但总体来说,k 值增加更多则需要消耗更多的存储空间,如 $\phi=100$ 时需要更多的存储空间且变化幅度大,而 $\phi=500$ 时由于 k 值不变,则需要相对较少的存储空间且变化较为平稳.

其次在运行效率方面,从图8 和图9 可以看出,$\phi=100$ 时,STOCIRNMF 在 n 值达到 14 万时,需要处理的时间约为 220s,占用的内存空间约为 7.5G. 此外,STOCIRNMF 在整个增量式聚类过程中需要占用的最大内存空间为 7.9G,因此其完全可以在单机环境下用于对固定时间段内不断增长的微博短文本进行在线聚类处理获得微博话题. 表5 为 $\phi=100$ 时对应获得的典型微博话题及强关联的主题词,从中可以看出 STOCIRNMF 具有较好的微博话题检测能力. 例如,表5 中列出的典型话题均为2011 年 7 月 23 日至2011 年 7 月 25 日微博爆发的关于“723 温州动车事故”的热门话题,这些话题具有明显的时效性,三天时间内微博用户一开始首先关注动车追尾的突发事件,然后是关注政府部门的现场救援工作以及通过微博寻找事故中的失联人员,最后是对救援工作以及高铁的安全性提出质疑. 以上分析结果表明,STOCIRNMF 不仅具有较好的在线数据处理效率,而且由于可以快速对具有一定数量规模的微博短文本进行话题检测,因此还非常适合用于在线检测具有时效性的微博话题.

表5 典型微博话题

话题	主题词	时间
动车追尾	温州(0.48),动车(0.43),追尾(0.38),事故(0.34),脱轨(0.31),新闻(0.27),高架桥(0.24),铁路(0.23),列车(0.21),坠桥(0.19)	2011 年 7 月 23 日
现场救援	事故现场(0.41),挖掘机(0.38),奇迹(0.37),祈祷(0.35),生命(0.33),遇难(0.32),幸存者(0.31),救援人员(0.28),政府部门(0.27),伤员(0.25)	2011 年 7 月 24 日
寻找失联人员	转发(0.37),亲人(0.32),帮忙(0.28),祈祷(0.23),寻找(0.21),平安(0.18),大家(0.16),团聚(0.15),儿童(0.13),小朋友(0.11)	2011 年 7 月 24 日
质疑救援工作	草率(0.45),愤怒(0.41),粗暴(0.36),工作(0.33),措施(0.31),人道(0.28),舆论(0.26),结束(0.21),方式(0.18),行动(0.15)	2011 年 7 月 25 日
质疑高铁安全性	责任(0.36),人祸(0.31),真相(0.26),安全(0.23),腐败(0.21),官员(0.19),垄断(0.16),质量(0.14),工程(0.13),设备(0.12)	2011 年 7 月 25 日

4 总结

对增长迅速的社会化媒体短文本进行有效聚类分析具有重要应用价值,而如何在保证聚类质量的基础上提高大规模短文本的聚类效率是其中的关键问题,本文为此提出了一种基于增量式鲁棒非负矩阵分解的短文本在线聚类算法 STOCIRNMF. STOCIRNMF 通过提高鲁棒性可以降低短文本噪声数据对于聚类质量的影响,同时可以通过增量式聚类实现大规模短文本的在线聚类. 通过使用真实的短文本数据集进行实验验证,结果表明 STOCIRNMF 具有较好的短文本聚类质量,同时具有较小的时间以及空间消耗,可以应用于在线检测具有时效性的微博话题. 由于 STOCIRNMF 进行矩阵迭代更新所需的相关数据需要驻留内存,因此其本质上属于基于内存计算的在线算法. 已有研究表明矩阵迭代更新计算也适合采用 Spark 分布式内存计算框架实现,因此在下一步工作中,将重点研究基于 Spark 的 STOCIRNMF 的实现方法,目的在于通过进一步提高其运算效率可以处理更大规模的短文本数据在线聚类问题.

参考文献

[1] Hu Y H, Chen Y L, Chou H L. Opinion mining from online hotel reviews-a text summarization approach[J]. Information Processing & Management, 2017, 53(2): 436 – 449.

[2] Cigarrán J, Castellanos Ángel, García-Serrano A. A step forward for topic detection in Twitter: an FCA-based approach[J]. Expert Systems with Applications, 2016, 57: 21 – 36.

[3] 黄发良,李超雄,元昌安,等. 基于 TSCM 模型的网络短文本情感挖掘[J]. 电子学报,2016,44(8):1887 – 1891. HUANG Fa-liang, LI Chao-xiong, YUAN Chang-an, et al. Mining sentiment for web short text based TSCM model [J]. Acta Electronica Sinica, 2016, 44(8): 1887 – 1891. (in Chinese)

[4] Zhang H, Zhong G Q. Improving short text classification by learning vector representations of both words and hidden topics[J]. Knowledge-Based Systems, 2016, 102(15): 76 – 86.

[5] Yin J H, Wang J Y. A dirichlet multinomial mixture model-based approach for short text clustering[A]. Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining[C]. USA: ACM, 2014. 233 – 242.

[6] Yu Z, Wang H X, Lin X M, et al. Understanding short texts through semantic enrichment and hashing[J]. IEEE Transactions on Knowledge and Data Engineering, 2016, 28(2): 566 – 579.

[7] Lee DD, Seung H S. Algorithms for non-negative matrix factorization[A]. Proceedings of 2000 Annual Conference on Neural Information Processing Systems[C]. USA: MIT Press, 2000. 556 – 562.

[8] Zhang X C, Zong L L, Liu X Y, et al. Constrained clustering with nonnegative matrix factorization[J]. IEEE Transactions on Neural Networks and Learning Systems, 2016, 27(7): 1514 – 1526.

[9] Bucak S S, Gunsel B. Incremental subspace learning via non-negative matrix factorization[J]. Pattern Recognition, 2009, 42(5): 788 – 797.

[10] Chen R G, Li H. Online algorithm for foreground detection based on incremental nonnegative matrix factorization [A]. Proceedings of the 2nd International Conference on Control, Automation and Robotics[C]. USA: IEEE, 2016. 312 – 317.

[11] Yan X H, Guo J F, Liu S H, et al. Clustering short text using Ncut-weighted non-negative matrix factorization[A]. Proceedings of the 21st ACM International Conference on Information and Knowledge Management [C]. USA: ACM, 2012. 2259 – 2262.

[12] Ganguly D, Roy D, Mitra M, Jones G J F. Word embedding based generalized language model for information retrieval[A]. Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval[C]. USA: ACM, 2015. 795 – 798.

[13] Liu C Y, Chen M Y, Tseng C Y. IncreSTS: towards real-time incremental short text summarization on comment streams from social network services[J]. IEEE Transactions on Knowledge and Data Engineering, 2015, 27(11): 2986 – 3000.

作者简介

贺超波　男,1981 年生于广东河源,现为仲恺农业工程学院副教授,主要研究方向为数据挖掘、机器学习与大数据技术.
E-mail: hechaobo@ foxmail. com

汤　庸(通信作者)　男,1964 年生于湖南张家界,现为华南师范大学计算机学院教授,主要研究方向为数据智能与云服务、学术社交网络与教育大数据.
E-mail: ytang@ m. scnu. edu. cn