一种大规模高维数据快速聚类算法

刘 铭 [1] 王晓龙 [1] 刘远超 [1]

摘 要 提出了一种面向大规模高维数据的自组织映射聚类算法. 算法通过压缩神经元的特征集合, 仅选择与神经元代表的文档类相关的特征构造神经元的特征向量, 从而减少了聚类时间. 同时由于选取的特征能够将映射到不同神经元的文档类进行有效区分, 避免了无关特征的干扰, 因而提升了聚类的精度. 实验结果表明该方法能够有效加快聚类的速度, 提升聚类的准确度, 达到比较理想的聚类效果.

关键词 向量压缩, 神经元合并, 类内相似度, 类间区分度
中图分类号 TP18

A Fast Clustering Algorithm for Large-scale and High Dimensional Data

LIU Ming[1] WANG Xiao-Long[1] LIU Yuan-Chao[1]

Abstract A novel self-organizing-mapping algorithm for large-scale and high dimensional data is proposed in this paper. By compressing neurons′ feature sets and only selecting relative features to construct neurons′ feature vectors, the clustering time can be dramatically decreased. Simultaneously, because the selected features can effectively distinguish different documents which are mapped to different neurons, the algorithm can avoid interferences of irrelative features and improve clustering precision. Experiments results demonstrate that this methodology can accelerate clustering speed and improve clustering precision significantly and can reach relatively ideal clustering effect.

Key words Vector compression, neuron combination, intra-cluster similarity, inter-cluster distinctness

聚类作为一种自动化程度较高的无监督机器学习方法, 近年来在信息检索、数据挖掘等领域获得了广泛的应用[1−2]. 在众多聚类算法中, 自组织映射聚类 (Self organization mapping, SOM) 是一种比较有效的方法, 它是由 Kohonen 首先提出[3], 并被随后加以研究的一种无导师的自组织和自学习网络. SOM 算法将高维空间的数据转化为二维空间, 并且在二维空间中很好地保持了输入数据之间的相似性, 其能够根据数据的分布逐步收敛到最佳的类别划分. 与其他聚类方法相比, SOM 聚类的优点在于: 可以实现实时学习, 算法具有自稳定性和自学习性, 无需外界给出评价函数, 抗噪音能力强.

基于上述优点, 现实应用中已经提出了许多基于 SOM 的文本聚类算法[4−5], 这些算法以神经元 (Neuron) 作为文档类的代表, 神经元特征向量中的每个维度对应于特征空间中的每个特征, 特征的权值相当于该特征在映射到此神经元的文档类中权值的平均分布[3]. SOM 文本聚类算法的实质就是发现能够有效划分文档集合的神经元集合. 观察发现, 能够将一个文档类与其他文档类进行有效区分的特征在整个特征空间中占有很小的范围. 例如, 作为 “社会科学” 类文档的代表特征只需类似于 “政治”、“文化” 等与该类别描述的信息相关的特征, 而对于像 “经济”、“体育” 这样的特征, 显然不能将该类别与其他类别进行有效区分, 而且上述这些不相关特征的存在还会使某些不属于该类别的文档由于含有上述不相关特征而被错误地划分到 “社会科学” 类中, 使得聚类结果较差.

统计发现, 对于 10 万篇以上的超大规模文档集, 其特征空间也在万的数量级上, 过大的特征空间显然会造成过大的神经元特征向量, 大大增加了聚类的时间. 而如前所述, 一个类别的代表特征在特征空间中仅占有很小的范围. 实验发现, 对 10 万篇左右的文档集合进行类别划分后, 作为每个文档类所描述信息的代表的特征只需大约 200 ∼ 300 个特征, 仅占整个特征空间的 1/50. 可以看出, 如果我们为每个文档类仅选择上述能够将此文档类与其他文档类进行有效区分的特征构造神经元的特征向量, 则可将神经元的特征向量压缩到 1/50, 从而有效地缩短聚类时间. 由后续的实验结果可以看出, 如果我们按照上述方法构造神经元的特征向量, 不仅能够大大降低聚类时间, 而且使得聚类后各类别的内部更加紧凑, 类别间的区分度也有所提高.

收稿日期 2008-07-01 收修改稿日期 2008-12-03
Received July 1, 2008; in revised form December 3, 2008
国家高技术研究发展计划 (863 计划) (2006AA01Z197, 2007AA01Z172), 国家自然科学基金 (60435020) 资助
Supported by National High Technology Research and Development Program of China (863 Program) (2006AA01Z197, 2007AA01Z172), and National Natural Science Foundation of China (60435020)
1. 哈尔滨工业大学计算机科学与技术学院 哈尔滨 150001
1. School of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001
DOI: 10.3724/SP.J.1004.2009.00859

1 基于向量压缩的 SOM 快速聚类算法 (VPSOM)

本文将神经元组织为扇形结构, 如图 1 所示, 扇形结构简单并且可以避免矩形结构的边缘问题, 同时在调整神经元结构时只需要插入少量的神经元, 这样可以避免矩形结构中由于插入过多的神经元而造成神经元训练不足, 形成如图 2 所示的 "欠利用".

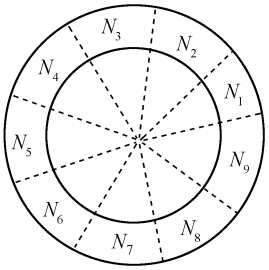


图 1 神经元扇形结构

Fig. 1 Round neuron structure

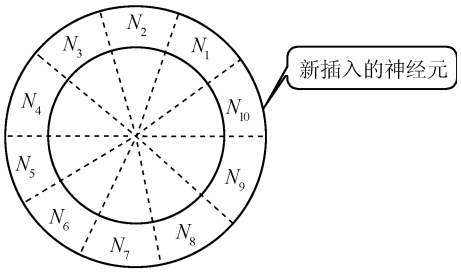


图 2 扇形结构插入神经元

Fig. 2 Neuron insertion in round structure

1.1 神经元初始化

大多数自组织映射算法以随机方式初始化神经元的特征向量[3−5]. 虽然自组织映射算法对神经元特征向量的初始化方法并不十分敏感, 但是由于随机初始化没有利用任何文本分布的先验知识, 初始化对聚类没有任何指导作用, 这样很可能会将许多相似的文档划分到不同的文档类中, 使得自组织映射算法的初期需要进行大量的训练来调整神经元的特征向量. 针对上述问题, 本文采用前向后向 (Forward-backward) 初始化, 该初始化方法能够将文档集合内相似的文档大致合并成一个文档类, 并形成代表每个文档类的神经元, 对聚类提供了初始类别分布的指导, 其详细步骤如下:

步骤 1. 设待聚类的文档集合为 DOC, 大小为 N, 其中第 k 篇文档为 D_k, 设神经元集合为 $NEURON$, 其中第 i 个神经元为 N_i;

步骤 2. 顺序扫描 DOC 初始化神经元, 设此时正在扫描 DOC 中的第 k 篇文档 D_k;

步骤 3. 按式 (1) (见下页) 计算 D_k 与 $NEURON$ 中每个神经元的相似度, 如果 D_k 与 $NEURON$ 中所有神经元的相似度均为 0, 则转步骤 4, 否则转步骤 5;

步骤 4. 以 D_k 作为一个新的神经元, 并将其插入到 $NEURON$ 中, 转步骤 6;

步骤 5. 取与 D_k 具有最大相似度的神经元, 设此神经元为 N_m, 按式 (2) (见下页) 调整 N_m 的特征集合 $FS(N_m)$ 中的特征及特征的权值;

步骤 6. 如果 N_m 为 DOC 的末尾, 则将 DOC 倒排, 转步骤 7, 否则扫描第 $k+1$ 篇文档, 转步骤 3;

步骤 7. 循环步骤 2 ∼ 6, 处理倒排的 DOC 直至末尾;

步骤 8. 将 $NEURON$ 中的神经元首尾相连组成如图 1 所示的扇形结构.

本文采用前向后向初始化神经元的原因如下:

实验发现, 如果仅采用前向 (Forward) 扫描初始化神经元, 会使文档集合中的大部分文档集中于前几个神经元代表的文档类中. 这是由于我们按照一定顺序 (从前向后) 扫描文档集合, 这样最先建立的神经元随着其特征集合的扩大会使更多的文档映射到这些神经元中, 而随着映射到最先建立的神经元的文档数的增多, 神经元的特征集合也会迅速扩大, 因此最先建立的神经元占有大量的文档. 以 10 万篇文档做实验, 前向扫描初始化后生成了 372 个神经元, 图 3 描述了映射到前 5 个神经元、中间 5 个神经元、最后 5 个神经元的文档数, 可见映射到前几个神经元的文档数过多, 而映射到后续神经元的文档数越来越少. 为了解决上述文档在神经元间分布不平衡的问题, 本文在前向 (Forward) 扫描后又加入了后向 (Backward) 扫描, 即按照由后向前的顺序再次扫描文档集合初始化神经元. 图 4 (见下页) 描述了采用前向后向扫描初始化后文档的分布情况, 可见此时文档在神经元间的分布较为平衡.

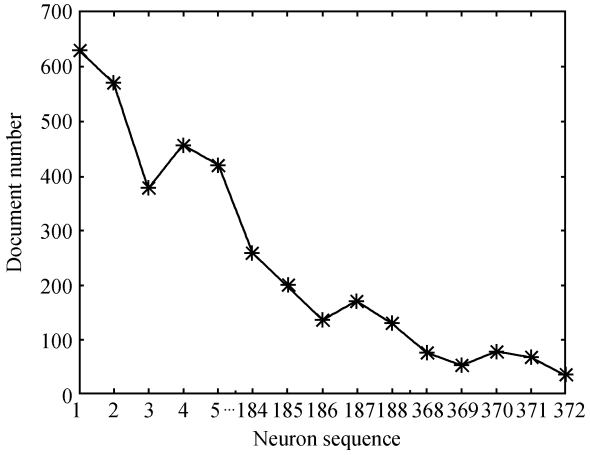


图 3 前向扫描初始化

Fig. 3 Initialization by forward scanning

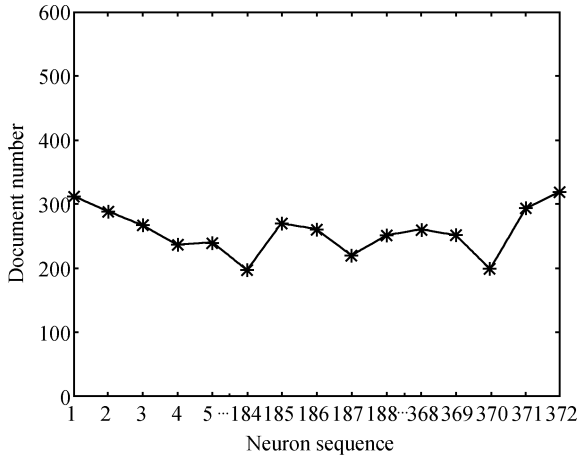


图 4　前向后向扫描初始化

Fig. 4　Initialization by forward-backward scanning

设文档 D_k 的特征集合和神经元 N_i 的特征集合的交集为 ISF_{ki}, 大小为 $|ISF_{ki}|$, 假设该交集中的特征 inf 在文档 D_k 的特征集合 $FS(D_k)$ 中的权值为 $DW_k(z)$, 在神经元 N_i 的特征集合 $FS(N_i)$ 中的权值为 $NW_i(y)$, 则 D_k 与 N_i 的相似度函数为

$$\mathrm{Sim}(D_k,N_i)=\begin{cases}0, & 若|ISF_{ki}|<\dfrac{|FS(D_k)|}{3}\\ \displaystyle\sum_{inf\in ISF_{ki}}\frac{DW_k(z)\times NW_i(y)}{DW_k(z)+NW_i(y)}\times\log_2(|ISF_{ki}|), & 否则\end{cases} \tag{1}$$

由式 (1) 可见, 如果文档的特征集合和神经元的特征集合的交集过小, 则认为文档反映的信息和神经元代表的文档类所反映的信息是不相关的. 反之, 则从两方面衡量文档与神经元之间的相似度, 首先如果特征 inf 在文档 D_k 中的权值很大, 并且其在神经元 N_i 中的权值也很大, 说明 inf 反映的信息在文档和神经元中均位于重要地位, 则 D_k 与 N_i 的相似度应较大. 其次如果 D_k 和 N_i 的特征集合的交集很大, 说明文档描述的信息和神经元代表的文档类所反映的信息大致相似, 则 D_k 与 N_i 的相似度应较大.

式 (1) 改变了传统自组织映射算法中文档与神经元的相似度计算方法. 传统的自组织映射算法通过欧式距离或余弦相似度来计算文档与神经元的相似度[3−5]. 而本文采用式 (1) 的原因是: 由于我们压缩了神经元的特征集合, 因此文档特征集合中的每个特征并不一定在神经元的特征集合中出现, 而欧式距离或余弦相似度需要计算文档特征集合中的每个特征的权值与该特征在神经元的特征集合中的权值的相差情况, 需要文档的每个特征均出现于神经元的特征集合中.

$$NW_m(y)(t+1)=\begin{cases}NW_m(y)(t)+0.1\times(DW_k(z)-NW_m(y)(t)), & 若\ df\in FS(N_m)\\ 0.1, & 若\ df\notin FS(N_m)\end{cases} \tag{2}$$

式 (2) 描述了如何根据文档 D_k 中的特征 df 调整神经元 N_m 中相应特征的权值. 设 df 在 D_k 中的权值为 $DW_k(z)$, df 在 N_m 中的权值为 $NW_m(y)$. 如果 df 在 N_m 的特征集合 $FS(N_m)$ 中出现, 则按照式 (2) 的第一部分调整 df 的权值. 如果 df 不在 N_m 的特征集合中出现, 则将 df 插入到 N_m 的特征集合中, 并对该特征赋予一个初始权值 0.1, 目的是将特征 df 反映的信息包含到神经元 N_m 中.

1.2　神经元权值调整

经过以上的初始化后可获得神经元的初始扇形结构, 之后即进入算法的训练阶段, 通过逐步调整神经元的特征集合及特征的权值, 使映射到同一神经元的文档类的内聚度越来越大, 映射到不同神经元的文档类间的区分度越来越大. 训练阶段每次从文档集合 DOC 中随机选取一篇文档, 同时按照式 (1) 计算此文档与神经元集合 $NEURON$ 中每个神经元的相似度, 并选取具有最大相似度的神经元, 设其为 N_m. 按照式 (5) (见下页) 调整 N_m 中特征的权值, 同时调整与 N_m 相邻的神经元中特征的权值, 并通过学习速率 LR 控制调整幅度, 通过临域 NH 控制调整范围.

学习速率 LR 的计算方法为

$$LR(t)=(-1)\times\frac{(ILR-MLR)}{EC}\times(CC\,\%(N+1))+ILR \tag{3}$$

其中, t 表示时间, 即算法迭代调整的次数, ILR 表示初始学习速率, MLR 表示学习速率的最小值, CC 表示当前的循环次数, EC 表示算法最多循环文档集合大小的 EC 倍, N 表示文档集合的大小.

领域 NH 的计算方法为

$$NH(t+1)=[NH(t)\times LR(t)] \tag{4}$$

可见随着循环次数 CC 的增大, 学习速率 LR 和邻域 NH 均呈递减的趋势, 其满足自组织映射算法中关于竞争学习的要求[3−5].

$$NW_b(h)(t+1)=$$
$$\begin{cases} NW_b(h)(t)+LR(t)\times\dfrac{(NH(t)-Dist(b,m))}{NH(t)}\times \\ \quad (DW_k(z)-NW_b(h)(t)), & \text{若 } df\in FS(N_b) \\ 0.1, & \text{若 } df\notin FS(N_b) \end{cases} \tag{5}$$

式 (5) 介绍了如何根据文档 D_k 中的特征 df 调整位于 N_m 的临域 NH 内的某个神经元中相应特征的权值, 设此神经元为 N_b. 其中 $Dist(b,m)$ 记录了神经元 N_b 和 N_m 在扇形结构中间隔的神经元数, 其他符号的含义参见式 (2).

本文以最小误差 (J_e) 作为基于向量压缩的自组织映射算法 (Vector pressing self organizing mapping, VPSOM) 的目标函数或收敛判别条件. J_e 以神经元的特征向量作为神经元代表的文档类的中心, 并通过计算文档类中各文档的特征向量和神经元的特征向量的欧式距离之和反映文档类的凝聚程度.

$$J_e=\sum_{i=1}^{c}\sum_{D_k\in C_i}||D_k-N_i||^2 \tag{6}$$

其中 c 代表类别数, N_i 代表神经元集合中的第 i 个神经元, C_i 代表映射到 N_i 的文档类.

传统的 SOM 算法使用梯度下降公式调整神经元的特征向量以使最小误差沿其最陡方向下降, 即

$$\begin{aligned} NW_i(h)(t+1)=&NW_i(h)(t)+LR(t)\times \\ &\frac{(NH(t)-Dist(i,m))}{NH(t)}\times \\ &(DW_k(z)-NW_i(h)(t)) \end{aligned} \tag{7}$$

对比式 (5) 和式 (7) 可知, 算法 VPSOM 的神经元调整方法和传统自组织算法中的神经元调整方法大致相同. 但是由于 VPSOM 压缩了神经元的特征集合, 因此文档特征集合中的特征并不一定在神经元的特征集合中出现, 则此部分不在神经元的特征集合中出现的特征在式 (5) 的第二部分中予以调整.

由本节的神经元训练过程中可以看出, 本文介绍的算法 VPSOM 以传统自组织算法为基础, 其改变了传统自组织算法中的相似度计算方法. 由式 (1) 可知, 算法 VPSOM 在相似度计算时仅仅考虑了文档与神经元的特征集合的交集中的特征, 而忽略了神经元的特征集合中不能作为文档类的代表的特征, 将这些特征在神经元特征向量中的权值视为 0. 而后算法 VPSOM 仍然按照传统自组织算法进行神经元权值调整, 其满足传统自组织算法的收敛性要求[3]. 因此算法能够保证随着迭代次数的增多, 其最小误差在逐渐减小直至收敛.

2 神经元内聚度及区分度分析

本文在对神经元的特征集合进行压缩时可能会存在以下两个问题:

1) 可能某个神经元的特征集合中包含错误特征, 将本不属于该类别的文档也纳入到神经元代表的文档类中, 造成文档类的内聚度很差.

2) 可能某个文档类的代表特征被多个神经元的特征集合所包含, 即将一个文档类拆分为多个类别, 造成文档类间的区分度很差.

针对上述问题, 算法必须能够衡量神经元是否已经很好地代表了一个类别, 还要判断出神经元包含的特征是否已经完备. 本文以文档集合的大小 N 为阈值, 每经过 N 次神经元权值调整后, 即进行神经元的内聚度及区分度分析.

2.1 神经元内聚度分析

本文采用文献 [6] 介绍的基于局部密度的聚类方法来判断映射到每个神经元的文档类中的离群点. 基于局部密度的聚类方法是基于密度的空间聚类算法 (Density based spatial clustering of applications with noise, DBSCAN) 的改进算法, 其只需要扫描一遍文档集合即可将文档集合划分为多个类别, 同时这种基于局部密度的聚类方法改变了传统密度聚类算法中对于邻域半径 E 和 E 邻域内的最少对象数 $MinPts$ 的参数依赖性, 但其缺点是算法的精度不高. 本文即采用上述方法, 以映射到同一神经元的文档类作为待聚类的文档集合, 以文档代表密度聚类中的点, 以文档间特征向量的欧式距离代表两点间的距离. 设映射到神经元 N_i 的文档类为 $DOC(N_i)$, 经局部密度聚类后生成的子类别为 $SC_1, SC_2, \cdots, SC_p$, p 为聚类后生成的子类别数. 设 SC_m 为具有最多文档数的子类别, 并以 SC_m 作为 N_i 代表的文档类. 清空 N_i 的特征集合, 取 SC_m 中每篇文档的特征集合内大于平均权重的特征插入到 N_i 的特征集合中. 经过如上的处理后, 神经元 N_i 的特征集合中仅保留了与此神经元代表的文档类 (即经局部密度聚类后具有最多文档数的子类) 相关的特征. 在采用局部密度聚类对映射到神经元的文档类进行类别划分时并不需要精确的划分结果, 因为我们可以通过算法的训练过程逐步调节神经元的特征集合, 使得神经元代表的文档类越来越精确.

2.2 神经元区分度分析

本文采用自动组合和随机组合相结合的方法搜索过于细分的神经元, 并将这些过于细分的神经元予以合并以提高类别间的区分度, 以式 (3) 介绍的学习速率调整两种方法对应的比例.

$$RC(t)+AC(t)=5 \tag{8}$$

$$AC(t) = [5 \times (1 - LR(t))] \tag{9}$$

式 (8) 中, $RC(t)$ 代表时刻 t 随机组合的神经元对数, $AC(t)$ 代表时刻 t 自动组合的神经元对数. 在神经元训练刚开始时, 由于训练次数较少, 类别的分布较为零散, 因此随机组合的比重大一些. 随着训练的进行以及神经元内聚度的改变, 类别划分已经逐步接近收敛状态, 此时逐渐增大自动组合的比重.

1) 随机组合合并过度细分的神经元: 按式 (8) 和式 (9) 从神经元集合中随机选择 $RC(t)$ 对神经元, 将每对神经元代表的文档类进行合并, 然后按照第 2.1 节介绍的局部密度方法建立一个新的神经元. 这样如果某个文档类被两个神经元所分割, 则可通过如上方法将这个被分割的文档类进行合并, 同时构造出反映这个合并的文档类的神经元.

2) 自动组合合并过度细分的神经元: 按式 (1) 计算文档集合中的每篇文档与神经元集合中的每个神经元之间的相似度, 并为每个神经元建立一个 map 向量, 向量的长度为文档集合的大小, 其中第 i 个维度的权值代表第 i 篇文档与此神经元的相似度. 以余弦相似度计算任两个神经元对应的 map 向量之间的相似度, 取相似度最大的 $AC(t)$ 对神经元予以合并, 其合并方式与上述随机组合类似. 其原因是, 如果两个神经元对应的 map 向量非常相似, 说明这两个神经元代表的文档类具有很大的相关性, 很可能属于同一个类别, 因此可以将这两个神经元代表的文档类合并.

对于上述自动组合和随机组合生成的神经元, 我们将其插入到被合并的两个神经元的中间位置, 即如果合并神经元集合中的第 i 个位置和第 j 个位置的神经元, 则将合并后生成的神经元插入到 $(i + j)/2$ 的位置上.

3　实验及分析

本文采用 Yahoo 网站的 2007 年新闻作为测试语料, 该语料包含了体育、医疗、社会、财经、教育等多个种类的新闻, 是一种平衡语料, 其包含大约 10 万篇文档共 298 M. 本文对比了是否采用前向后向初始化的 VPSOM 算法对于不同文档数的聚类时间, 及其在十万篇文档上的最小误差 (J_e) 随时间的变化情况. 实验结果如图 5 和图 6 所示.

图 5 显示了是否采用前向后向扫描初始化后, 算法 VPSOM 对于不同文档数的聚类时间. 可以看出, 在采用前向后向初始化后, 算法的聚类时间明显小于不采用该方法, 且随着文档数的增多, 这种情况越来越明显. 这是因为前向后向初始化提供了一个比较理想的初始类别划分, 在该初始划分下只需要进行少量调整算法即能较快收敛, 且随着文档数的增多, 前向后向初始化能够将大量的相似文档凝聚到一起, 加快了算法的收敛速度.

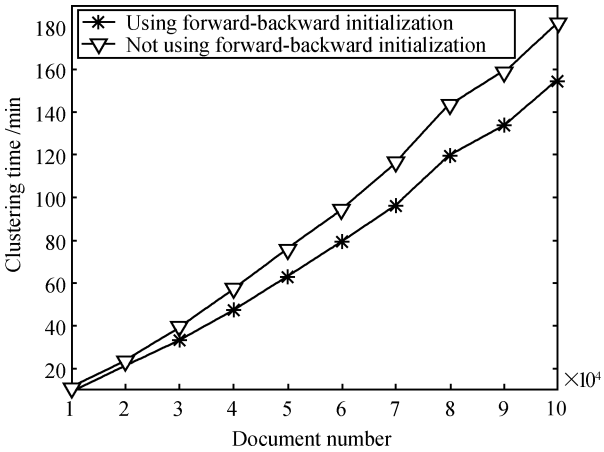


图 5　是否采用前向后向初始化的聚类时间

Fig. 5　Clustering time with forward-backward initialization or without

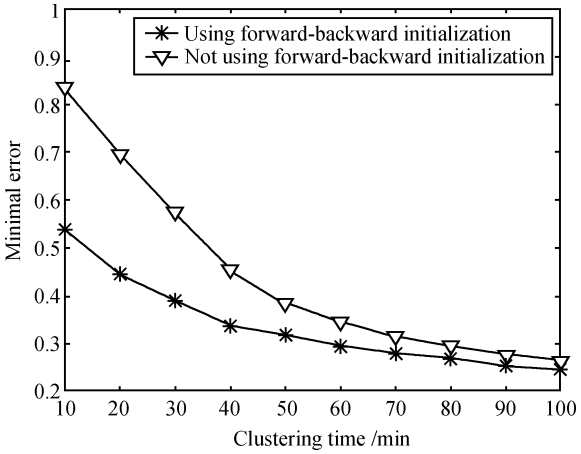


图 6　是否采用前向后向初始化的最小误差

Fig. 6　Minimal errors with forward-backward initialization or without

图 6 显示了是否采用前向后向扫描初始化后, 算法 VPSOM 的最小误差随时间的变化情况. 可以看出, 在采用前向后向初始化后, 训练初始时的最小误差和算法收敛时的最小误差的差距不是很大. 如文献 [3] 所述, 最小误差代表了类别的凝聚程度, 采用前向后向初始化后神经元初始结构的最小误差很小, 说明该初始结构已经是一个比较好的类别划分, 而如果不采用前向后向初始化, 算法的最小误差在训练初始时非常高, 即这种随机初始化产生的类别划分非常混乱, 算法需要经过大量的调整才能收敛.

本文依次将 VPSOM 与 K-means[1]、层次聚类[2]、GSOM (Growing SOM)[4]、GHSOM (Growing hierarchical SOM)[5] 对于十万篇文档的聚类时间进行了对比. 由于 K-means 需要预先设定类别

数, 因此本文采用文献 [7] 介绍的极大极小值方式确定 K-means 的类别数, 实验结果如图 7 所示.

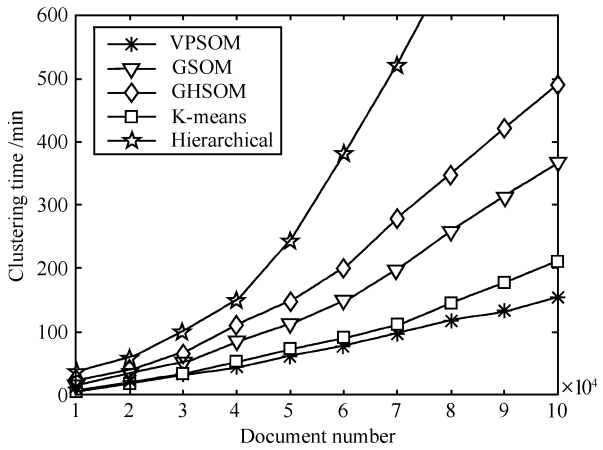


图 7 不同算法的聚类时间

Fig. 7 Clustering times of different algorithms

由图 7 的时间曲线中可以看出, 当文档数较少时 K-means 的性能最好, VPSOM 和 K-means 相差不多. 但是随着文档数的增多, VPSOM 的聚类时间的增长幅度却很小, 当文档数达到 3 万篇左右时, VPSOM 的聚类时间逐渐少于 K-means, 而 GSOM、GHSOM 的聚类时间相对比 K-means 和 VPSOM 要多一些. 其原因在于 K-means 是线性时间复杂度算法, 其时间复杂度为 O(kln), 其中 k 为类别数, l 为循环次数, n 为文档数, 而 GSOM 的时间复杂度为 O($k'mn$), 其中 k' 为神经元数即类别数, l 为循环次数, n 为文档数. 通常, GSOM 中的 k' 和 m 均比 K-means 中的 k 和 l 大, 因此图 7 中 GSOM 的时间复杂度比 K-means 要高. 算法 GHSOM 由于在 GSOM 的基础上又进行了层次分裂, 因此时间复杂度要高于 GSOM. 本文介绍的算法 VPSOM 首先通过前向后向初始化使得初始神经元结构与算法收敛时的神经元结构相差不多, 减少了迭代次数, 同时由于压缩了神经元的特征向量, 大大减少了相似度计算以及权值调整时需要计算的维度从而减少了聚类时间. 当文档数较少时, 这种特征向量压缩的程度很小, 因此聚类时间比 K-means 稍长, 但是随着文档数的增多, 向量压缩的效果逐渐显示出来, 当文档数达到 3 万篇左右时, VPSOM 将神经元的特征向量压缩了近 1/30, 这大大降低了相似度计算的时间, 而 K-means 没有对文档的特征向量进行任何处理, 因此其在大规模文档集上的聚类时间比 VPSOM 要长. 层次聚类的时间复杂度最高为 O(n^2), 聚类速度最慢.

由于本文的测试文档集规模巨大, 因此无法采用人工方法将测试文档集划分为多个类别, 然后再计算算法的准确率、召回率等值. 针对上述问题, 本文设计了两种新的测试方法.

方法 1. 运行算法直到收敛, 设算法生成的类别集合为 C, 其中第 i 个文档类为 C_i, 将 C_i 人工划分为多个子类别, 设其集合为 $IntSet(C_i)$, 以人工划分的具有最多文档数的子类别作为 C_i 的代表, 设此子类别为 $IntC_{\max}$, 将 $IntSet(C_i)$ 中不同于 $IntC_{\max}$ 的其他子类别包含的文档视为 C_i 的不相关文档, 则文档类 C_i 的准确率为 $|IntC_{\max}|/|C_i|$, 其中 $|C_i|$ 代表类别 C_i 包含的文档数, 取所有类别的准确率的平均值作为算法的准确率.

方法 2. 运行算法对测试文档集进行聚类, 当文档聚合为 500 个类别后即停止类别的增加, 然后仅进行神经元调整直至算法收敛. 随机查看聚类后每个类别中的 15 篇文档, 如果有一篇文档与该类的主题不符, 则视为错误归类. 设错误归类的文档数为 $ErrorNum$, 则可得每个类别的准确率为: $(15 - ErrorNum)/15$, 取所有类别的准确率的平均值作为算法的准确率.

采用上述两种方法分别计算 K-means、层次聚类、GSOM、GHSOM、VPSOM 相对于不同文档数的聚类准确率, 实验结果如图 8 和图 9 (见下页) 所示.

由图 8 和图 9 的准确率结果可以看出, 同一种算法对于上述两种计算方法得到的准确率曲线相差不大. 其中几种自组织映射算法均有较高的准确率, 且随着文档数的增多, 其准确率变化的幅度也不是很大, 即自组织映射算法可以应用于大规模文档聚类而不造成聚类结果的失真. 这是由于自组织映射算法将高维数据投影到二维空间上, 能够在一定程度上避免高维特征空间对聚类结果的影响, 同时通过逐步调节神经元使得算法能够达到一个比较理想的类别划分结果. 在这三种自组织映射算法中 VPSOM 的准确率最高, 这是由于该算法只选取能

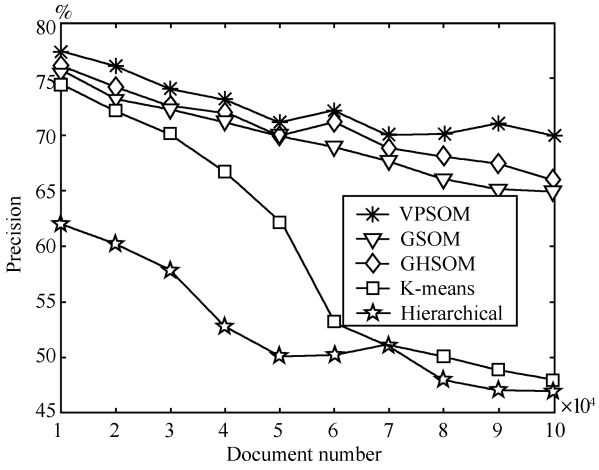


图 8 第一种方法的聚类准确率结果

Fig. 8 Precision results of the first method

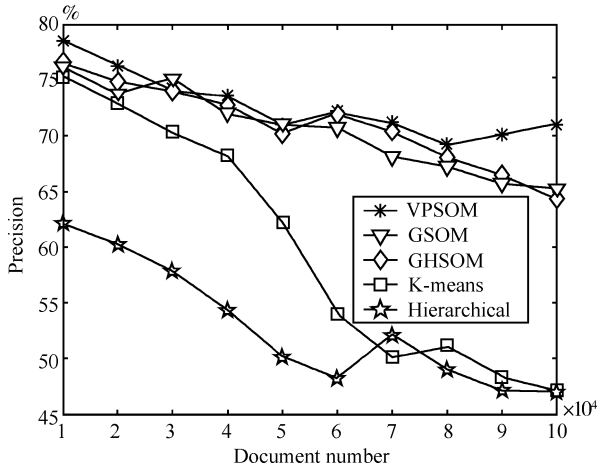


图 9　第二种方法的聚类准确率结果

Fig. 9　Precision results of the second method

够有效代表每个类别的特征构造神经元的特征集合, 可以消除无关特征对聚类结果的干扰使得算法具有很高的准确率, 同时通过判断神经元的内聚度以及发现并组合被割裂的文档类使得算法具有更高的准确率. 层次聚类、K-means 随文档数的增多性能下降很快, 这是因为随着特征空间的增大, 文档包含的特征仅占特征空间的很小一部分, 这样大部分文档的相似度趋近于 0. K-means 是在原始特征空间中进行聚类, 这样大量相似度趋近于 0 的文档会严重降低 K-means 对文档的划分能力. 层次聚类不仅是在原始特征空间中进行聚类, 且一旦文档在聚类某阶段归属于一个类别后则不能再对文档归属的类别进行调整, 因此上述两种算法在大规模文档集上的聚类准确率较低.

算法 VPSOM 仅选择能够有效代表文档类的特征构造神经元的特征向量, 从而压缩了神经元的特征向量. 显而易见, 这种特征向量的压缩方法不同于传统的特征降维算法. 其区别在于, 传统的特征降维技术或是通过衡量特征与类别之间的相互关系[8−9], 或是通过最小化高维特征空间与低维特征空间表示能力的误差[10−11], 在全局空间内选择有效特征以重构特征空间, 然而上述方法无法根据选择的特征对文档集合进行聚类, 也就是说上述算法还需要结合特定的聚类算法以完成文档类别划分. 而算法 VPSOM 的向量压缩方法则是在全局空间内选择能够有效代表文档类的特征构造神经元的特征集合或特征向量, 即是将特征空间划分为多个子集, 然后根据不同的特征子集将文档划分为多个类别. 目前比较常见的特征降维技术有: 基于关联分析的特征选择 (Feature selection based on associative rule, FAR)[8]、基于神经网络的特征选择 (Feature selection based on neural networks, FNN)[9]、基于主分量分析的特征选择 (Feature selection based on principal components analysis, FPCA)[10]、基于流形学习的特征选择 (Feature selection based on manifold learning, FML)[11]. 本文将上述算法应用于十万篇新闻语料的文本聚类中, 首先应用上述算法进行特征空间降维, 然后采用 SOM 算法进行文档聚类, 并与 VPSOM 算法进行对比, 比较了不同算法的聚类时间及聚类准确率, 结果如表 1 和表 2 所示.

表 1　应用特征降维算法的聚类时间

Table 1　Clustering times after using feature reduction

聚类时间 (min)	FAR	FNN	FPCA	FML	VPSOM
特征空间降维	135	192	115	143	–
文本聚类	166	137	158	175	154

表 2　应用特征降维算法的聚类准确率

Table 2　Clustering precisions after using feature reduction

准确率 (%)	FAR	FNN	FPCA	FML	VPSOM
第一种方法	63.57	61.08	63.12	65.44	69.98
第二种方法	64.03	62.15	62.77	64.78	71.01

由表 1 可见, 当我们结合特征降维算法和 SOM 算法进行文本聚类时, 聚类时间明显多于 VPSOM. 这是因为应用特征降维算法进行文本聚类均是分为两步执行的: 第一步进行特征降维, 第二步进行文本聚类, 因此聚类时间较长. 而算法 VPSOM 在聚类的同时进行特征选择, 通过选择能够有效进行类别区分的特征划分文档集合, 将聚类和特征选择结合在一起从而节省了聚类时间.

由表 2 可见, 算法 VPSOM 的聚类准确率显然要好于将特征降维技术应用于 SOM 聚类中. 其原因在于 VPSOM 中的特征选择以聚类为直接目标, 算法通过选择特征空间中能够有效区分文档类的特征构造神经元的特征向量, 可以有效避免无关特征对聚类结果的干扰. 而上述特征降维算法仅仅是在特征空间中选择有益于聚类的特征, 其并没有考虑选择的特征对类别化分结果的影响, 文档类的具体划分结果依赖于 SOM 聚类算法. 而传统的 SOM 聚类算法以特征空间内的所有特征构造神经元的特征向量, 显然会引入与类别不相关的特征, 因此采用特征降维算法的聚类准确率要低于 VPSOM.

4　结论

本文提出了一种用于处理大规模高维数据的聚类算法 (VPSOM). 算法通过压缩神经元的特征集合, 仅选择高维特征空间中能够将某一文档类与其他文档类进行有效区分的特征构造神经元的特征向

量, 此方法能够有效避免无关特征对聚类结果的干扰. 同时算法通过局部密度分析判断文档类的凝聚程度, 并依此过滤离群文档对聚类结果的干扰, 提升了算法的类内凝聚度, 通过合并过于相似的神经元, 使得算法的类间区分度也有了一定的提升. 由于算法 VPSOM 压缩了神经元的特征向量, 从而加快了相似度计算以及权值调整的速度, 降低了聚类的时间复杂度. 实验结果表明, 相比于传统的聚类算法, VPSOM 对于大规模文档集的聚类速度和聚类准确度均有大幅度的改善, 聚类效果比较理想.

References

1 Tsai C Y, Chiu C C. Developing a feature weight selfadjustment mechanism for a K-means clustering algorithm. *Computational Statistics and Data Analysis*, 2008, **52**(10): 4658−4672

2 Qu J, Jiang Q S, Weng F F, Hong Z L. A hierarchical clustering based on overlap similarity measure. In: Proceedings of the 8th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing. Qingdao, China: IEEE, 2007. 905−910

3 Kohonen T, Kaski S, Lagus K, Salojarvi J, Honkela J, Paatero V. Self organization of a massive document collection. *IEEE Transactions on Neural Networks*, 2000, **11**(3): 574−585

4 Arthur H, Saman K H. Class structure visualization with semi-supervised growing self-organizing maps. *Neurocomputing*, 2008, **71**(16-18): 3124−3130

5 Chan A, Pampalk E. Growing hierarchical self organising map (GHSOM) toolbox: visualisations and enhancements. In: Proceedings of the 9th International Conference on Neural Information Processing. Singapore, Singapore: IEEE, 2002. 2537−2541

6 Ni Wei-Wei, Sun Zhi-Hui, Lu Jie-Ping. K-LDCHD — a local density based K-neighborhood clustering algorithm for high dimensional space. *Journal of Computer Research and Development*, 2005, **42**(5): 784−791
(倪巍伟, 孙志挥, 陆介平. K-LDCHD — 高维空间 K 邻域局部密度聚类算法. 计算机研究与发展, 2005, **42**(5): 784−791)

7 Liu Yuan-Chao, Wang Xiao-Long, Liu Bing-Quan. An adapted algorithm of choosing initial values for K-means document clustering. *Chinese High Technology Letters*, 2006, **16**(1): 11−15
(刘远超, 王晓龙, 刘秉权. 一种改进的 K-means 文档聚类初值选择算法. 高技术通讯, 2006, **16**(1): 11−15)

8 Tien D D, Siu C H, Alvis C M F. Associative feature selection for text mining. *International Journal of Information Technology*, 2006, **12**(4): 59−68

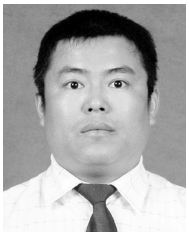
9 Verikas A, Bacauskiene M. Feature selection with neural networks. *Pattern Recognition Letters*, 2002, **23**(11): 1323−1335

10 Malhi A, Gao R X. PCA-based feature selection scheme for machine defect classification. *IEEE Transactions on Instrumentation and Measurement*, 2004, **53**(6): 1517−1525

11 Lin T, Zha H B. Riemannian manifold learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2008, **30**(5): 796−809

刘 铭 哈尔滨工业大学计算机学院博士研究生. 主要研究方向为文本聚类. 本文通信作者.
E-mail: mliu@insun.hit.edu.cn
(**LIU Ming** Ph. D. candidate at Harbin Institute of Technology. His main research interest is text clustering. Corresponding author of this paper.)

王晓龙 哈尔滨工业大学计算机学院教授. 主要研究方向为自然语言处理.
E-mail: wangxl@insun.hit.edu.cn
(**WANG Xiao-Long** Professor at Harbin Institute of Technology. His main research interest is natural language processing.)



刘远超 哈尔滨工业大学计算机学院副教授. 主要研究方向为聚类分析.
E-mail: lyc@insun.hit.edu.cn
(**LIU Yuan-Chao** Associate professor at Harbin Institute of Technology. His main research interest is clustering analysis.)