基于正交非负矩阵分解的K-means聚类算法研究

李孟杰 谢 强 丁秋林
(南京航空航天大学计算机科学与技术学院 南京 210016)

摘 要 为提高K-means聚类算法在高维数据下的聚类效果,提出了一种基于正交非负矩阵分解的K-means聚类算法。该算法对原始数据进行非负矩阵分解,并分别通过改进的Gram-Schmidt正交化和Householder正交化加入了正交约束,以保证低维特征的非负性,增加数据原型矩阵的正交性,然后进行K-means聚类。实验结果表明,基于IGS-ONMF和H-ONMF的K-means聚类算法在处理高维数据上具有更好的聚类效果。

关键词 高维数据,非负矩阵分解,降维,正交NMF,K-means聚类
中图法分类号 TP274 **文献标识码** A **DOI** 10.11896/j.issn.1002-137X.2016.5.037

Orthogonal Non-negative Matrix Factorization for K-means Clustering

LI Meng-jie XIE Qiang DING Qiu-lin
(College of Computer Science and Technology, Nanjing University of Aeronautics and Astronautics, Nanjing 210016, China)

Abstract The orthogonal NMF K-means clustering algorithm based on basic theory of NMF was proposed to improve the quality of K-means clustering in high-dimensional data. We presented orthogonal NMF algorithm, added orthogonal restraint to data prototype matrix from factorization with improved Gram-Schmidt and Householder orthogonalization separately, which both ensure non-negative of low-dimensional feature and enhance the orthogonality of matrix, and then made K-means clustering. Experimental results show that K-means clustering based on H-ONMF has better clustering results on high-dimensional data.

Keywords High-dimensional data, NMF, Dimension reduction, Orthogonal NMF, K-means clustering

1 引言

聚类是根据"物以类聚"的思想将数据分到不同类或者簇的过程,所以同一个簇中的对象有很大的相似性,而不同簇间的对象有很大的相异性。目前,基于划分、层次、密度、网格、神经网络等的多种聚类算法已经成功运用于各项研究及应用中。这些聚类算法在处理小规模数据和低维数据时效果较好,但是随着数据规模增大、维度升高,其性能会急剧下降,因而不能满足大规模、高维数据集的聚类分析。K-means是一种典型的划分聚类算法,它用一个聚类的中心来代表一个簇,即在迭代过程中选择的聚点不一定是聚类中的一个点,在处理大数据集时,该算法具有可伸缩、高效且计算复杂度低等优点,因此在数据聚类领域得到了广泛应用。随着数据量和数据维数增多,受"维数灾难"效应的影响,在低维数据空间表现良好的K-means聚类方法运用到高维空间上时往往无法获得好的聚类效果,由于不断地进行迭代,不断地计算新的聚类中心,其算法效率就逐渐下降,在数据集中数据维数和个数非常庞大时,算法耗费巨大,而现实生活中的数据大部分又都属于规模比较大、维度比较高的数据集,由此对算法的时间复杂度进行分析和改进是很有必要的。为了使得K-means聚类算法在高维数据上具有更好的性能,本文以K-means聚类算法为切入点,引入了非负矩阵分解,对原始的高维数据进行非负矩阵分解,使得输入到K-means聚类算法中的高维数据集的噪声信息尽可能的小。

非负矩阵分解理论最早由D. D. Lee和H. S. Seung[1]在1999年于《Nature》上提出,最初用于人脸识别。非负矩阵分解(Non-negative Matrix Factorization, NMF)作为一种新型的矩阵分解方法,它克服了传统矩阵分解的很多问题,具有实现简便、分解形式和结果可解释以及占用存储空间少等优点,已经成为一种相对成熟的数据分析手段,被成功运用于图像处理[1,2]、模式识别[3]、数据挖掘[4,5]、文本聚类[6,7]、生物医学工程[8]等多个领域。D. D. Lee和H. S. Seung两人提出的NMF算法[9]应用于数据聚类时性能较差,为提高NMF在数据聚类上的效果,许多学者在NMF的正交性上做了相应的研究。2006年,Ding等[10]对NMF加入了正交约束,首次将其分为3类,分别是对称三因子分解、单正交非负矩阵分解和双正交非负矩阵分解;在此研究基础上,Li[11]和Cao[12]等人分别在各自的应用领域加入了正交约束,均获得了良好效果。2014年,Redko等[13]提出了将ONMF运用到聚类算法中,实验结果表明,ONMF具有较好的聚类纯度和较高的效率。

本文在传统的K-means聚类算法及Batcha和Redko等人研究的基础上,对原始数据NMF分解后的数据原型矩阵

到稿日期:2015-06-18 返修日期:2015-10-18 本文受江苏省产学研联合创新资金项目(SBY201320423)资助。
李孟杰(1991-),女,硕士生,主要研究方向为知识工程、数据挖掘、人机交互,E-mail:lmjiemaggie@163.com;谢 强(1973-),男,博士,副教授,硕士生导师,主要研究方向为知识工程、信息系统与信息安全、人机交互;丁秋林(1936-),男,博士,教授,博士生导师,主要研究方向为信息系统、企业信息化。

分别通过 Gram-Schmidt 和 Householder 正交化加入正交限制，提出了改进的 Gram-Schmidt 正交和 Householder 正交 K-means 聚类算法，基于真实数据集和模拟数据集上的实验表明，两个算法相比传统的 K-means 聚类算法提高了聚类的纯度和熵。

2 非负矩阵分解及其分类

2.1 非负矩阵分解

非负矩阵分解[1,14]（NMF）是一种分析非负数据的重要工具。NMF 是将高维的原矩阵分解为两个低维非负矩阵，分解的过程实际上是一个不断迭代优化的过程，基于一个合理的目标函数，利用迭代的方式交替迭代求得分解后的非负矩阵 W 和 H。形如：

$$V=WH+E\Rightarrow V\approx WH \tag{1}$$

其中，$V\in R_+^{n\times m}$ 为待分解矩阵，$W\in R_+^{n\times r}$ 和 $H\in R_+^{r\times m}$ 为分解后的两个非负矩阵，$x\in R_+$ 表示 $x\geqslant 0$，要求 $\|E\|$ 尽可能小并且快速收敛。r 的选取是根据实际情况设置的，要求：

$$r\ll\min\{n,m\} \tag{2}$$

由于分解前后的矩阵中仅包含非负元素，W 中的列向量可以解释为对 V 中所有基向量的加权和，而权重系数为右矩阵 H 中对应列向量中的元素。在数据聚类中可以将左矩阵 W 解释为数据原型矩阵，右矩阵 H 解释为聚类成员指示矩阵。有学者指出，非负矩阵分解是个 NP-hard 问题，可以划为优化问题，用迭代方法交替求解 W 和 H。其中目标函数有多种方式，在文献[1]中引入了两种目标函数，分别是欧氏距离（Euclidean distance）和 K-L（Kullback-Leibler）散度。以欧氏距离来度量，式(1)改写为：

$$\min_{W,H}\sum_{i,j}(V_{ij}-(WH)_{ij})^2 \tag{3}$$

如果以 K-L 散度为度量，则改写为：

$$\min_{W,H}\sum_{i,j}\left(V_{ij}\log\frac{V_{ij}}{(WH)_{ij}}-V_{ij}+(WH)_{ij}\right) \tag{4}$$

由于 W、H 是非凸的，对应的最优解应用迭代公式进行迭代，当 $\|V-WH\|$ 误差小于 δ 或者达到最大迭代次数时，则停止，此时得到最优解的 W 和 H。

2.2 非负矩阵分解的分类

非负矩阵分解根据 W 和 H 的正负性分为 3 种：半非负矩阵分解、单正交矩阵分解和双正交矩阵分解。其中如果只对式(1)中的 W 加入非负限制，而对其中 H 的正负性没有限制，则称之为半非负矩阵分解（Semi-NMF）。

单正交矩阵分解（Uni-Orthogonal Non-negative Matrix Factorization，UONMF）根据式(1)分为两种，一种是对矩阵 W 加入正交限制，如下所示：

$$V\approx WH,WW^{\mathrm{T}}=I \tag{5}$$

其中，W、H 均为正交矩阵，根据 NMF 的生成方式对 W、H 矩阵按照下面的表达式进行迭代更新：

$$W=W\otimes\frac{VH^{\mathrm{T}}}{WW^{\mathrm{T}}VH^{\mathrm{T}}} \tag{6}$$

$$H=H\otimes\frac{V^{\mathrm{T}}W}{H^{\mathrm{T}}HV^{\mathrm{T}}W} \tag{7}$$

这里的$\otimes$为矩阵的直积运算，不断进行递归迭代来产生最终的矩阵。另一种对矩阵加入正交限制的迭代方式与式(6)、式(7)类似。对 W 和 H 同时加入正交限制，直接分解后为：

$$V\approx WH,WW^{\mathrm{T}}=I,G^{\mathrm{T}}G=I \tag{8}$$

对原始矩阵 X 的行列进行 K-means 聚类时，H 为一个 V 的列 K-means 聚类的聚类指示符矩阵（cluster indicator matrix），W 为 V 的行 K-means 聚类的聚类指示符矩阵。

双正交具有很大的限制性，并且使得分解后的低秩矩阵稀疏性大。根据式(1)，引入了一个额外的矩阵 S 得到双正交矩阵分解（Bi-Orthogonal Non-negative Matrix Factorization，BONMF），定义如下：

$$V\approx WSH \tag{9}$$

$$V\in R_+^{n\times m},W\in R_+^{n\times k},S\in R_+^{k\times l},H\in R_+^{l\times m}$$

$$W^{\mathrm{T}}W=I,H^{\mathrm{T}}H=I,V,W,S,H\geqslant 0$$

其中，W、H 均为正交矩阵，根据 NMF 的生成方式对 W、S、H 矩阵按照下面的表达式进行迭代更新：

$$W=W\otimes\frac{VH^{\mathrm{T}}S^{\mathrm{T}}}{WW^{\mathrm{T}}VH^{\mathrm{T}}S^{\mathrm{T}}} \tag{10}$$

$$S=S\otimes\frac{W^{\mathrm{T}}VH^{\mathrm{T}}}{W^{\mathrm{T}}WSHH^{\mathrm{T}}} \tag{11}$$

$$H=H\otimes\frac{V^{\mathrm{T}}WS}{H^{\mathrm{T}}HV^{\mathrm{T}}WS} \tag{12}$$

从以上的公式可以看出单 UONMF 就是对 W 或者 H 进行正交限制，可以将 UONMF 理解为 BONMF 的一种特殊表现形式，即当 $S=I$ 时。

3 基于正交非负矩阵分解的 K-means 算法

矩阵的正交化有两种常用的方法：Householder 反射变换和 Gram-Schmidt 正交化。本文同时选用这两种正交方法设计了 Gram-Schmidt 正交化非负矩阵分解算法（Gram-Schmidt Orthogonal Non-negative Matrix，GS-ONMF）和 Householder 正交化非负矩阵分解算法（Householder Orthogonal Non-negative Matrix，H-ONMF）。在传统的 K-means 聚类算法的基础上，对原始数据首先进行 NMF 分解得到数据原型矩阵 W，分别通过 GS-ONMF 和 H-ONMF 算法对数据原型矩阵 W 加入正交限制，提出了基于 Gram-Schmidt 和 Householder 正交非负矩阵分解的 K-means 聚类算法。基于真实数据集和模拟数据集的实验表明，两个算法相比传统的 K-means 聚类算法提高了聚类的熵和纯度。

3.1 正交非负矩阵分解算法设计

3.1.1 Gram-Schmidt 正交非负矩阵分解

本节首先用到了 Gram-Schmidt 正交化，利用投影原理在已有正交基的基础上构造一个新的正交基，在 R^n 的 r 维子空间中设置一个线性无关向量组 $W=\{v_1,v_2,\cdots,v_r\}$，满足 $r\leqslant n$，生成正交线性无关向量组 $W'=\{u_1,u_2,\cdots,u_r\}$。用表达式(13)定义 Gram-Schmidt 正交化定义投射算子：

$$proj_u(v)=\frac{\langle u,v\rangle}{\langle u,u\rangle}u \tag{13}$$

其中，$\langle u,v\rangle$ 是向量 u 和 v 的内积，向量 v 通过 $proj_u(v)$ 投影到向量 u 的线性空间上，使得 v 达到正交。

利用 NMF 进行正交化，首先要计算基向量 W，由于内积的存在，直接利用 Gram-Schmidt 正交化并不能得到非负向量集，因此需要使用 Semi-NMF 使得 W 中具有正交基向量。另一方面，由于在 Semi-NMF 产生正交基向量的过程中忽略了另一个矩阵向量，因此需要一个近似正交性去均衡两个矩阵向量。对式(13)中的映射算子进行了改进，加入了常数 α_{GS}：

$$proj_u^*(v)=\alpha_{GS}\frac{\langle u,v\rangle}{\langle u,u\rangle}u,\alpha_{GS}\in[0,1] \tag{14}$$

当 $\alpha_{GS}=0$ 时对应的是初始矩阵，当 $\alpha_{GS}=1$ 时对应的是完全正交矩阵，因此用 $W_{\alpha_{GS}}$ 表示当 α_{GS} 在其取值范围内时所对应的基向量矩阵。那么对应 Semi-NMF 可以根据任意的 α_{GS} 值写成如下形式：

$$V\approx W_{\alpha_{GS}}H, H>0 \tag{15}$$

式中的 $W_{\alpha_{GS}}$ 在不同的 α_{GS} 时具有不同的正交性，对于不同的正交性，用式(16)中的 *Ortho* 来衡量：

$$Ortho=\| W_{\alpha_{GS}}^{T} * W \| \tag{16}$$

3.1.2 HouseHolder 正交非负矩阵分解

目前对非负矩阵的正交化使用传统的 Gram-Schmidt 正交化。Gram-Schmidt 正交化方法计算方便，但是要求各个基向量之间必须不相关，并且 Gram-Schmidt 正交化方法需要进行大量的内积和范数计算，整体的数值性能较差，随着分解后基向量矩阵秩的增加，算法所涉及到的舍入误差逐渐变大，正交化之后基向量矩阵的正交性能也会相应变差。基于此种情况，在以上所设计算法 WGS-ONMF 的基础上，采用了 HouseHolder 变换正交，首先对基向量矩阵 W 进行 QR 分解，并得到新的正交基向量矩阵，WH-ONMF 算法只是对 W 矩阵进行了酉变换，涉及到的内积和范数运算较少，因此所涉及到的舍入误差较小，算法的数值性能更加稳定可靠。

在进行 HouseHolder 变换之前，从式(1)得知基向量矩阵 W 为一个非负 $n\times r$ 矩阵，要先将分解的 W 用如下形式表示：

$$W=\{w_1, w_2, \cdots, w_r\} \tag{17}$$

其也可以用如下形式表示：

$$W=\begin{Bmatrix} w_{11} & w_{12} & \cdots & \cdots & w_{1r} \\ w_{21} & w_{22} & \cdots & \cdots & w_{2r} \\ w_{31} & w_{32} & \cdots & \cdots & w_{3r} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ w_{n1} & \cdots & \cdots & \cdots & w_{nr} \end{Bmatrix} \tag{18}$$

定理 1 $\forall A\in R^{m\times n}(m>n)$ 是列满秩矩阵，存在分解式 $A=QR$，其中 $Q=R^{m\times n}$ 为列法正交矩阵，$R\in R^{n\times n}$ 为非奇异上三角矩阵。

$$A_{m\times n}=QR=\begin{bmatrix} * & \cdots & * \\ \vdots & \ddots & \vdots \\ * & \cdots & * \end{bmatrix}_{m\times m}\begin{bmatrix} * & \cdots & * \\ \vdots & \ddots & \vdots \\ 0 & \cdots & * \\ 0 & \cdots & 0 \end{bmatrix}_{m\times n} \tag{19}$$

根据定理 1 对矩阵 W 中的对角线元素及其以下的元素进行 HouseHolder 变换，可将对角线以下的元素变换为 0，即：

$$\begin{aligned} H_i w_i(n)&=H_i[w_{i1}, w_{i2}, \cdots, w_{in}]^{T} \\ &=[w_{i1}', w_{i2}', \cdots, w_{in}', 0, \cdots, 0]^{T} \end{aligned} \tag{20}$$

其中，H_i 表示第 i 次 HouseHolder 变换，$w_{ij}'(a\leqslant j\leqslant i)$ 为变换后的元素。这样，依次经过 r 次 HouseHolder 变换后，可将分解后的基向量矩阵变换成一个上三角矩阵，即：

$$H_r H_{r-1}\cdots H_1 W=\begin{Bmatrix} w_{11}' & w_{12}' & \cdots & w_{1r}' \\ 0 & w_{22}' & \cdots & w_{2r}' \\ \cdots & 0 & \cdots & \cdots \\ \cdots & \cdots & \cdots & w_{rr}' \\ 0 & 0 & \cdots & 0 \end{Bmatrix} \tag{21}$$

令 $Q=H_1 H_2\cdots H_r$，则 $W=QR$，而 Q 就是矩阵 W 的一组正交基，其每一列为新的正交基向量 $w_i'(n)$。

由 HouseHolder 变换的性质可知 $HH^{T}=H^2=I$：

$$\begin{aligned} QTQ&=(H_1 H_2\cdots H_r)^{T}(H^1 H^2\cdots H^r) \\ &=H_r^{T}\cdots H_2^{T} H_1^{T} H_1 H_2\cdots H_r=I \end{aligned} \tag{22}$$

即：

$$\sum_{i=1}^{r} w_i'(n) w_j'(n)=\begin{cases}\delta, & i=j \\ 0, & i\neq j\end{cases} \tag{23}$$

这就证明了新生成的 w_i' 的正交性。经过 HouseHolder 正交变换，矩阵 W 即可分解成一个正交矩阵 W' 与上三角矩阵的乘积。基于 HouseHolder 变换的方法得到的 W 矩阵是相互正交的，从而实现了矩阵 W' 中 r 个向量的正交化，得到规范的正交基向量矩阵。上述基于 HouseHolder 变换的基向量矩阵正交化方法涉及到的内积和范数的运算比较少，因此其数据舍入的误差比较小，算法的数值性能较好；并且在进行 HouseHolder 变换时并没有要求矩阵 W 中各向量具有非线性，因此也不会受到 W 相关性的影响。

3.2 基于正交非负矩阵分解的 K-means 算法设计

通过 3.1 节中两种正交化方法得到了正交矩阵 W，接着在进行迭代求最优解时引入了参数 α、矩阵 W 和 H，加入了梯度下降和交替，形成了一个新的迭代规则，公式如下：

$$\min(\| V-WH \|_{W}^{2}+\| W^{T}W-\alpha I \|_{W}^{2}) \tag{24}$$

$$W=W\otimes\frac{VH+\alpha W}{WH^{T}H+WW^{T}W} \tag{25}$$

$$H^{T}=H\otimes\frac{W^{T}V}{W^{T}WH^{T}} \tag{26}$$

在使用以上梯度下降法进行迭代计算时采用加性迭代策略，以保证算法的一致性。叠加迭代规则如下：

$$\begin{aligned} W&=W+\eta_W(WH^{T}H+WW^{T}W-VH-\alpha W) \\ H&=W+\eta_{H^{T}}(W^{T}WH^{T}-W^{T}V) \end{aligned} \tag{27}$$

$$\begin{aligned} \eta_W&=\frac{W}{WHH+WW^{T}W} \\ \eta_H&=\frac{H^{T}}{W^{T}WH^{T}} \end{aligned} \tag{28}$$

从上式可以看出，随着 W 和 H 的迭代规则的逐步改进，由最初的式(10)纯乘性迭代规则，最终形成了本算法中的乘性加性交替的迭代更新规则，从而提高了算法的收敛性和一致性。将经过迭代形成的数据原型矩阵 W 作为 K-means 算法的原始数据样本，算法如下。

输入：数据集 X，预先设定的类数 r，迭代参数 α，正交化方式 K。

输出：聚类结果 C*。

1. 将数据集 X 用矩阵 $V_{m\times n}$ 表示，其中每一列为数据样本，由 m 个特征描述。
2. 利用非负矩阵分解得到 V≈WH，其中 W 的列可被认为是聚类中心，G^{T} 为聚类成员指示矩阵。
3. If K==1 then

   用 3.1.1 节中的 GS-ONMF 算法得到正交矩阵 W。

   ELSE

   用 3.1.2 节中的 H-ONMF 算法得到正交矩阵 W。
4. repeat

   $W=W+\eta_W(WH^{T}H+WW^{T}W-VH-\alpha W)$

   $H=W+\eta_{H^{T}}(W^{T}WH^{T}-W^{T}V)$

   until 满足目标函数式(24)。
5. 将得到的 W 中的每一列作为初始聚类中心($w_1, w_2, \cdots, w_r$)。
6. repeat

   若 H_{ik}^{T} 是行 H_i^{T} 中的最大值，则数据样本 i 属于对应的聚类 w_k。

   将样本 i 划分给离它最近的中心点 w_k 所表示的簇。

计算每个簇内的平均值，并将其作为该簇新的中心点。

Until 所有类中心不再改变或达到最大迭代次数。

7. RETURN $C^*=\{C_1,C_2,\cdots,C_r\}$

4 实验结果与分析

4.1 实验配置

本节通过实验来验证 IGS-ONMF 算法和 H-ONMF 算法的性能，在真实和模拟数据集上对聚类结果进行对比。实验的硬件环境是：Dual-Core CPU 3.20GHz，内存 4GB；软件环境是：操作系统为 32 位 Microsoft Windows7 Ultimate。所有实验在 Visual Studio 2010 和 Matlab 的 R2014a 版本中实现，数据集包括 2 个模拟和 4 个 UCI(http://archive.ics.uci.edu/ml/index.html)数据集，具体信息如表 1 所列。

表 1 数据集信息

Name	# Instances	# Attributes	# Class
DataSet1	200	3	3
DataSet2	300	10	2
Mushroom	200	9	2
Glass	214	10	7
Iris	150	4	3
Yeast	500	10	10

其中模拟数据集 DataSet1 和 DataSet2 均采用 Matlab R2014a 生成，满足多变量高斯分布。原始的 Mushroom 数据集有 8124 个样本(2 个类)，在实验中随机选取 200 个样本(每个类 100 个样本)。Yeast 数据集有 1484 个样本(10 个类)，随机为每个类选取 50 个样本用于实验。其余两个数据集均采用原始数据集。

4.2 实验结果比较

在本文 3.1 节 IGS-ONMF 算法设计中，对 Gram-Schmidt 正交化进行了改进，在式(13)中对投射因子加入了权重因子，权重因子取不同的值会影响到矩阵 W 的正交性，而 W 的正交性会直接影响到 IGS-ONMF 算法的整体性能，造成实验结果不公正。为了使接下来的实验结果具有一致性，首先对各个数据集求得其最佳的 α_{GS}，然后将最佳的 α_{GS} 代入到 IGS-ONMF 算法中，从而进行算法之间的比较。图 1—图 4 分别给出了在 4 种不同的数据集下 W 矩阵的正交性随着 α_{GS} 的改变的变化情况。

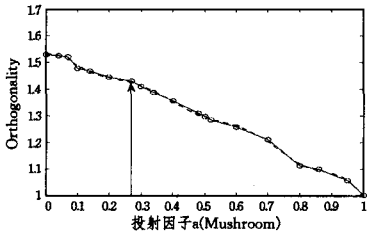

图 1 数据集 Mushroom 在不同投射因子下的正交性

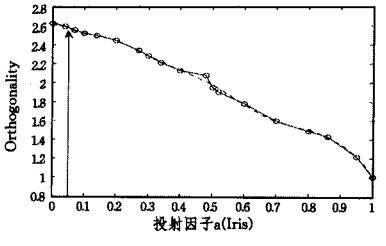

图 2 数据集 Iris 在不同投射因子下的正交性

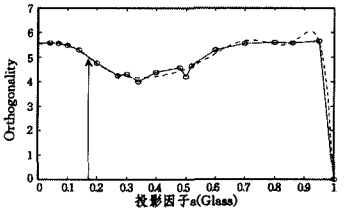

图 3 数据集 Glass 在不同投射因子下的正交性

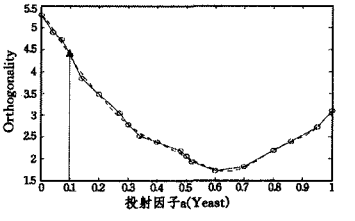

图 4 数据集 Yeast 在不同投射因子下的正交性

在本节中引用了 Ding 等[8]所提出的衡量聚类的两个评价指标 Purity 和 Entropy，其中评价指标 Purity：

$$Purity=\sum_{i=1}^{K}\frac{n_i}{n}P(S_i) \tag{29}$$

$$P(S_i)=\frac{1}{n_i}\max_j(n_i^j) \tag{30}$$

其中，S_i 是一个 n_i 大小的聚类，表示 j 类被放到 i 聚类中的数目。K 为聚类总数，n 为点的总数 n_i^j。根据式(29)，Purity 越高，其对应的聚类算法性能越好。另一个指标 Entropy：

$$Entropy=-\frac{1}{n\log_2 m}\sum_{i=1}^{K}\sum_{j=1}^{m}n_i^j\log_2\frac{n_i^j}{n_i} \tag{31}$$

其中，m 为源标签的数量，K 是聚类的数量。Entropy 的值越小，则其对应的聚类效果越好。根据以上两个指标，在 4 种不同维数、不同种类的数据上进行了算法比较。本实验将对传统的算法、IGS-ONMF 和 H-ONMF 下的算法分别独立运行 100 次，然后取其各指标的平均值。同时，在每次运行时设置最大迭代数为 100，以避免程序陷入局部最优或无限迭代，聚类数以实际样本标签的类数来预设，对于模拟数据集，样本 K 的取值为 3，如表 2 所列。

表 2 模拟数据集下的 Purity 值

数据集	指标	K-means	IGS-ONMF	H-ONMF
Dataset1	Purity	0.7876	0.8144	0.8553
	Entropy	0.1821	0.1752	0.1645
Dataset2	Purity	0.8047	0.8555	0.8773
	Entropy	0.1796	0.1752	0.1701

从表 2 可以看出，在两个满足多变量高斯分布的模拟数据集上，IGS-ONMF 和 H-ONMF 下的 K-means 算法在两种指标下(Purity 和 Entropy)都能取得较优的表现，并且与传统的 K-means 算法相比聚类效果有了很大的提升。为了进一步说明本文算法的有效性，实验设置不变，在真实数据集上比较传统的算法、IGS-ONMF 和 H-ONMF 下的算法，结果如图 5 和图 6 所示。

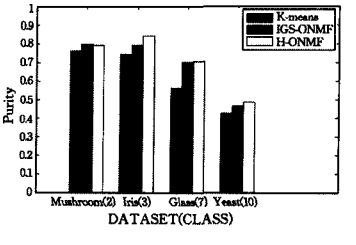

图 5 不同数据集下算法对应的 Purity 值

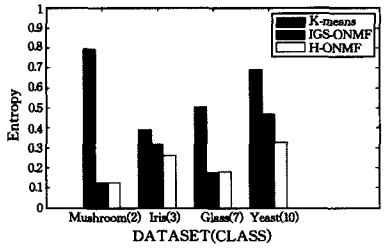

图 6　不同数据集下算法对应的 Entropy 值

图 5 和图 6 给出了基于 3 种算法在不确定程度和维度的 4 种数据集上进行聚类的结果。在数据维数较低的数据集 Mushroom 和 Iris 上，传统的 K-means 聚类算法在指标 Purity 上与本文中所提出的 IGS-ONMF 和 H-ONMF 下的 K-means 算法性能较为相似；但当数据的维数增加时，传统的 K-means 聚类算法的性能明显降低，而对于基于 IGS-ONMF 和 H-ONMF 的 K-means 算法来说，数据对应的维数越高，对应聚类算法的处理性能和鲁棒性就越高。H-ONMF 的 K-means 算法由于保证了数据的数值特征和完全正交性，能够在维数中等的数据集上有优越的聚类效果；而另一方面，IGS-ONMF 的 K-means 算法设置了一个权重系数用于控制正交性，在聚类过程中具有更强的适用性和鲁棒性，更适合维数较高的数据集。

结束语　本文在非负矩阵分解的基础上对高维数据进行处理，并将其数据原型矩阵分别通过改进的 Gram-Schmidt 正交化和 Householder 正交化加入了正交约束，由此提出了基于正交非负矩阵分解的高维数据 K-means 聚类算法。对 IGS-ONMF 和 H-ONMF 两种算法在真实和模拟数据集下进行了 K-means 聚类比较，实验结果表明，两种算法在一定程度上改善了高维数据下聚类的纯度和熵，具有较强的适应性和鲁棒性。

参 考 文 献

[1] Seung H S, Lee D D. Learning the parts of objects by non-negative matrix factorization[J]. Nature, 1999, 401(6755): 788-791

[2] Chu M, Plemmons R J. Nonnegative matrix factorization and applications[J]. Image, 2005, 34: 1-5

[3] Liu Wei-xiang, Zheng Nan-ning, You Qu-bo. Non negative matrix factorization and its application in pattern recognition[J]. Chinese Science Bulletin, 2006, 51(3)(in Chinese)
刘维湘，郑南宁，游屈波. 非负矩阵分解及其在模式识别中的应用[J]. 科学通报，2006，51(3)

[4] Tang J, Ceng X, Peng B. New Methods of Data Clustering and Classification Based on NMF[C]// 2011 International Conference on Business Computing and Global Informatization. IEEE Computer Society, 2011: 432-435

[5] Cichocki A, Amari S I, Zdunek R, et al. Extended SMART Algorithms for Non-negative Matrix Factorization[M]// Artificial Intelligence and Soft Computing(ICAISC 2006). Springer Berlin Heidelberg, 2006

[6] Ma Hui-fang, Zhao Wei-zhong, Tan Qing, et al. Orthogonal Nonnegative Matrix Tri-factorization for Semi-supervised Document Co-clustering[C]// Proceedings of the 14th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD2010). 2010: 189-200

[7] Chen Y H, Wang L J, Dong M. Semi-supervised Document Clustering with Simultaneous Text Representation and Categorization[C]// Buntine W, Grobelnik M, Mladeni D, eds. Machine Learning and Knowledge Discovery in Databases: Lecture Nates in Computer Science. Springer, Heidelberg. 2009: 211-226

[8] Stadlthanner K, Theis F J, Puntonet C G, et al. Extended sparse nonnegative matrix factorization[M]// Computational Intelligence and Bioinspired Systems. Springer Berlin Heidelberg, 2005: 249-256

[9] Lee D D, Seung H S. Algorithms for Non-negative Matrix Factorization[J]. Nips, 2000, 32(6): 556-562

[10] Ding C, Tao L I, Peng W, et al. Orthogonal Nonnegative Matrix Tri-Factorizations For Clustering[M]// Sigkdd. 2006: 126-135

[11] Li H, Adal T, Wang W, et al. Non-negative Matrix Factorization with Orthogonality Constraints and its Application to Raman Spectroscopy[J]. Journal of Vlsi Signal Processing Systems for Signal Image & Video Technology, 2007, 48(1/2): 83-97

[12] Cao B, Shen D, Sun J T, et al. Detect and Track Latent Factors with Online Nonnegative Matrix Factorization[C]// International Joint Conference on Artificial Intelligence. 2007: 2689-2694

[13] Redko I, Bennani Y, Redko I. Controlling orthogonality constraints for better NMF clustering[C]// 2014 International Joint Conference on Neural Networks (IJCNN). IEEE, 2014: 3894-3900

[14] Li Le, Zhang Yu-jin. A Survey on Algorithms of Non-Negative Matrix Factorization[J]. Acta Electronica Sinica, 2008, 36(4): 737-743(in Chinese)
李乐，章毓晋. 非负矩阵分解算法综述[J]. 电子学报，2008，36(4)：737-743

[15] Li B, Zhou G, Cichocki A. Two Efficient Algorithms for Approximately Orthogonal Nonnegative Matrix Factorization[J]. Signal Processing Letters IEEE, 2015, 22(7): 843-846

(上接第 192 页)

江敏，肖诗斌，王弘蔚，等. 一种改进的基于《知网》的词语语义相似度计算[J]. 中文信息学报，2008，22(5)：84-89

[20] 新浪军事新闻[EB/OL]. (2014-03)[2015-04-05]. http://mil.news.sina.com.cn

[21] NLPIR[EB/OL]. (2015-03-27)[2015-04-05]. http://www.nlpir.org

[22] 新浪国际军情新闻[EB/OL]. (2015-04)[2015-04]. http://roll.mil.news.sina.com.cn/col/gjjq/index.shtml

[23] 凤凰网新闻频道[EB/OL]. (2015) [2015-04]. http://news.ifeng.com

[24] Xia Tian. Study on Chinese Words Semantic Similarity Computation[J]. Computer Engineering, 2007, 33(6): 191-194(in Chinese)
夏天. 汉语词语语义相似度计算研究[J]. 计算机工程，2007，33(6)：191-194

[25] Xia Tian. X-Similarity[EB/OL]. (2014-08) [2015-04-07]. http://code.google.com/p/xsimilarity