初始化 K-means 的谱方法

钱 线 [1] 黄萱菁 [1] 吴立德 [1]

摘 要 众所周知, K-means (以下简称 KM) 对初始点十分敏感. 本文提出了一种新的初始化 KM 的方法, 它先估计出 k 个类的特征中心的位置, 然后用估计出的特征中心来初始化 KM. 在人工数据集和真实数据集上的实验表明, 本文的方法所得到的结果要好于其他一些初始化 KM 的方法.

关键词 聚类, K-means 算法, 特征中心
中图分类号 TP391

A Spectral Method of K-means Initialization

QIAN Xian[1] HUANG Xuan-Jing[1] WU Li-De[1]

Abstract It is well known that K-means algorithm (KM) is very sensitive to the initial conditions. In this paper, we propose a new method to initialize KM. It estimates the eigencenters of the k clusters, and initializes KM with these estimated eigencenters. Experiments on the artificial data set and the real data set show that our method significantly outperforms other initialization methods.

Key words Clustering, K-means algorithm, eigencenter

1 引言

K-means(KM) 是一种局部搜索的聚类算法, 正如许多论文所指出的, 它的结果依赖于初始点的选择. 为了解决这个问题, 人们提出了许多初始化的方法. Millan[1] 指出, KM 的初始值可以由 Ward[2] 的层次聚类的方法得到. Ward 方法认为开始时每个数据点都是一个单独的类, 每次从已有的类中挑出最相似的两个类进行合并, 直至合并到只有一个类为止. 在初始化 KM 前, 先用 Ward 方法做一次聚类, 当类的个数合并为 k 时, 停止聚类, 求出这 k 个类的中心作为 KM 的初始值. Higgs[3] 和 Snarey[4] 等人分别提出先用 MaxMin 算法对数据点聚类, 并将聚类结果的类中心作为 KM 的初始化值. 这两种初始化 KM 方法本身就是一种聚类算法, 因此它们自身聚类结果的好坏将影响到 KM 算法. Kaufman 和 Rousseeuw[5] 提出了一种初始化 KM 的方法: Kaufman approach(KA). 该方法逐个地从数据集中选出代表点, 直至取到 k 个点为止. 第一个代表点是离数据集中心最近的那个点, 其余的代表点根据一个启发式的规则从剩余的点中选出.

以上所提到的方法都是启发式的算法, 我们不知道这些方法在哪些场合下有效. 本文提出了一种新的初始化 KM 的方法, 在已知类的个数 k 的条件下, 首先估计出 k 个类的特征中心, 然后用这些特征中心来初始化 KM.

我们提出的这一方法基于一个新的概念: 类的特征中心. 实际上它是一个带权的类中心. 和一般类中心不同的是, 各个类的特征中心可以比较准确地从多类的数据集中估计出来. 在估计完每个类的特征中心之后, 就可以用这些估计出来的中心初始化 KM.

下面文章的内容安排如下: 首先在第 2 节中给出特征中心的概念, 接着在第 3 节给出估计各个类的特征中心的算法. 实验结果将在第 4 节给出. 最后在第 5 节给出总结.

2 特征中心

本节给出类的特征中心的概念. 现在假设只有一个类 C, $\boldsymbol{x}_i$ 是该类中的第 i 个数据点, n 为数据点的个数, 则该类的特征中心被定义为

$$\boldsymbol{m} = \sum_{\boldsymbol{x}_i \in C} w_i \boldsymbol{x}_i \tag{1}$$

$\boldsymbol{m}$ 就是该类的特征中心. 其中 w_i 为 $\boldsymbol{x}_i$ 的权重, 是一个与 $\boldsymbol{x}_i$ 有关的量, 它代表了 $\boldsymbol{x}_i$ 与其他数据点相邻程度的总和. 该权重越大, 表明 $\boldsymbol{x}_i$ 落在类 C 中越密集的区域. 可见和一般意义上的类中心相比, 特征中心偏向数据点密集的区域. 其中 w_i 服从约束

$$\sum_{i=1}^{n} w_i = 1 \qquad w_i \geq 0 \quad i = 1, 2, \ldots, n \tag{2}$$

收稿日期 2006-3-3 收修改稿日期 2006-7-2
Received March 3, 2006; in revised form July 2, 2006
国家自然科学基金 (60435020) 资助
Supported by National Natural Science Foundation of P. R. China (60435020)
1. 复旦大学计算机科学与工程系 上海 200433
1. Department of Computer Science and Engineering, Fudan University, Shanghai 200433
DOI: 10.1360/aas-007-0342

我们认为, $\boldsymbol{x}_i$ 与其他数据点相邻的程度越高, w_i 就越高; $\boldsymbol{x}_i$ 与 w_j 越高的数据点 $\boldsymbol{x}_j$ 相邻, 那么 w_i 也就越高. 综合这两点, 我们可以得到

$$w_i = c \sum_{\boldsymbol{x}_j \in C} S_{ij} w_j \tag{3}$$

其中 c 是一个与 $\boldsymbol{x}_i$ 无关的常数, 它的目的是使得最后结果中的 w_i 满足 (2) 的约束. S 为该类的邻接矩阵, 定义为

$$S_{ij} = \mathrm{sim}(\boldsymbol{x}_i, \boldsymbol{x}_j) = \exp(-\frac{|\boldsymbol{x}_i - \boldsymbol{x}_j|^2}{\sigma^2}) \tag{4}$$

这个定义和谱聚类[6] 中邻接矩阵的定义一样, 其中 σ 是一个参数.

由此, 得到了计算 C 中每个数据权重的叠代算法. 记 $\boldsymbol{W} = [w_1, w_2, ..., w_n]^{\mathrm{T}}$, $\epsilon > 0$ 是一个预先给定的任意小的正数, $\boldsymbol{e}$ 为所有分量为 1 的向量. 则 $\boldsymbol{W}$ 的计算方法如表 1 所示.

表 1　权重叠代算法

Table 1　Iterative weighted algorithm

权重叠代算法
1. 初始化所有 x_i 权重, $\boldsymbol{W}(0) = [\frac{1}{n}, \frac{1}{n}, ..., \frac{1}{n}]^{\mathrm{T}}$
2. 计算邻接矩阵 S
3. $\boldsymbol{W}(t+1) = \boldsymbol{W}(t)S$
4. 标准化 $\boldsymbol{W}(t+1)$, 即 $\boldsymbol{W}(t+1) = \frac{\boldsymbol{W}(t+1)}{\boldsymbol{e}^{\mathrm{T}}\boldsymbol{W}(t+1)}$
5. 如果 $\lvert\boldsymbol{W}(t+1) - \boldsymbol{W}(t)\rvert < \epsilon$, 停止; 否则转到第 3 步

根据文献 [7] , $\boldsymbol{W} = \frac{\boldsymbol{\xi}}{\boldsymbol{e}^{\mathrm{T}}\boldsymbol{\xi}}$, 其中 $\boldsymbol{\xi}$ 是 S 最大特征值所对应的特征向量, 分母的作用是使得最后求得的 $\boldsymbol{W}$ 满足约束 (2).

3　多类特征中心估计

本节讨论如何从多类的数据集中估计出各个类的特征中心.

记 X 为数据集矩阵, X 的每一列代表了一个数据点, 那么多类特征中心的估计算法如表 2 所示.

表 2　多类特征中心估计算法

Table 2　Multi-class eigencenter estimation algorithm

多类特征中心估计算法
1. 计算 X 的邻接矩阵 S
2. 计算 S 的前 k 大特征值所对应的特征向量: $\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \ldots, \boldsymbol{\xi}_k$
3. 对所有的 $\boldsymbol{\xi}_i$, 如果 $\boldsymbol{\xi}_{i_j} < 0$, 则令 $\boldsymbol{\xi}_{i_j} = 0$ $(j = 1, 2, \ldots, n)$
4. 标准化 $\boldsymbol{\xi}_i$, 即 $\boldsymbol{\xi}_i = \frac{\boldsymbol{\xi}_i}{\boldsymbol{e}^{\mathrm{T}}\boldsymbol{\xi}_i}$
5. 根据 $\boldsymbol{m}_i = X\boldsymbol{\xi}_i$ 估计出 k 个类的特征中心

下面给出这一算法的理论解释. 考虑两个类的情况. 假设现在有 C_1, C_2 两类, 记 S 为所有数据点的邻接矩阵

$$S = \begin{pmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{pmatrix} \tag{5}$$

其中, S_{11}, S_{22} 分别为 C_1, C_2 的邻接矩阵, S_{12}, S_{21} 为这两个类的类间邻接矩阵, 分别对 S_{11}, S_{22} 作特征值分解

$$S_{11} = (\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \ldots) \begin{pmatrix} \lambda_1 & 0 & \ldots \\ 0 & \lambda_2 & \ldots \\ \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} \boldsymbol{\xi}_1^{\mathrm{T}} \\ \boldsymbol{\xi}_2^{\mathrm{T}} \\ \vdots \end{pmatrix} \tag{6}$$

$$S_{22} = (\boldsymbol{\zeta}_1, \boldsymbol{\zeta}_2, \ldots) \begin{pmatrix} \mu_1 & 0 & \ldots \\ 0 & \mu_2 & \ldots \\ \vdots & \vdots & \ddots \end{pmatrix} \begin{pmatrix} \boldsymbol{\zeta}_1^{\mathrm{T}} \\ \boldsymbol{\zeta}_2^{\mathrm{T}} \\ \vdots \end{pmatrix} \tag{7}$$

这里 λ_i, μ_i 分别是 S_{11}, S_{22} 的第 i 大特征值, $\boldsymbol{\xi}_i, \boldsymbol{\zeta}_i$ 是相应的特征向量. 设 X 为数据集矩阵, 每一列代表一个数据点. $X_1 = \{\boldsymbol{x} | \boldsymbol{x} \in C_1\}$, $X_2 = \{\boldsymbol{x} | \boldsymbol{x} \in C_2\}$, 那么 C_1, C_2 的特征中心分别为

$$\boldsymbol{m}_1 = X_1 \frac{\boldsymbol{\xi}_1}{\boldsymbol{e}^{\mathrm{T}}\boldsymbol{\xi}_1} \qquad \boldsymbol{m}_2 = X_2 \frac{\boldsymbol{\zeta}_1}{\boldsymbol{e}^{\mathrm{T}}\boldsymbol{\zeta}_1} \tag{8}$$

由于类之间的交界区域, 数据分布十分稀疏, 换句话说, 类之间数据点的相邻程度很低, 即 $S_{12} \approx 0, S_{21} \approx 0$, 这样我们有

$$S \approx \begin{pmatrix} S_{11} & 0 \\ 0 & S_{22} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\xi}_1 & \boldsymbol{\xi}_2 & \ldots & 0 & 0 & \ldots \\ 0 & 0 & \ldots & \boldsymbol{\zeta}_1 & \boldsymbol{\zeta}_2 & \ldots \end{pmatrix} \cdot \begin{pmatrix} \lambda_1 & 0 & \ldots & 0 & 0 & \ldots \\ 0 & \lambda_2 & \ldots & 0 & 0 & \ldots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ldots \\ 0 & 0 & \ldots & \mu_1 & 0 & \ldots \\ 0 & 0 & \ldots & 0 & \mu_2 & \ldots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \cdot \begin{pmatrix} \boldsymbol{\xi}_1^T & 0 \\ \boldsymbol{\xi}_2^T & 0 \\ \vdots & \vdots \\ 0 & \boldsymbol{\zeta}_1^T \\ 0 & \boldsymbol{\zeta}_2^T \\ \vdots & \vdots \end{pmatrix} \tag{9}$$

不失一般性, 假设 $\lambda_1 \geq \mu_1$, 如果又有 $\mu_1 > \lambda_2$, 那么根据 S 最大的两个特征值对应的特征向量求出的两个特征中心将约等于两个类的真实的特征中心

$$\boldsymbol{m}_i \approx X \frac{\boldsymbol{\eta}_i}{\boldsymbol{e}^{\mathrm{T}}\boldsymbol{\eta}_i} \qquad i = 1, 2 \tag{10}$$

其中 $\boldsymbol{\eta}_i$ 是 S 的第 i 大特征值对应的特征向量. 幸运的是只要各个类所含的数据点数目相差不是很大, $\mu_1 > \lambda_2$ 往往成立. 下面来说明这一点.

根据 [8], λ_1 随着 S_{11} 的任意一个元素的增加而增加, 即 λ_1 随着 $S_{11_{ij}}$ 的增大而增大. 并且由 [9], S_{11} 是一个半正定矩阵, 也就是说它的所有特征值非

负：$\lambda_i \geq 0$. 注意到 $\sum \lambda_i = \mathrm{tr}(S_{11}) = n_1$, n_1 为 C_1 中数据点的个数. 于是 $\lambda_1 - \sum_{i\geq 2}\lambda_i$ 将随着 $S_{11_{ij}}$ 的增大而增大. 由于 C_1 类内的样本点分布密集, 也就是说 S_{11} 中有很多元素的值很高, 因此有 $\lambda_1 \gg \lambda_2$, 且由于 n_1, n_2 相差不是很大, 这样 $\sum \lambda_i$ 和 $\sum \mu_i$ 相差也不大, 因此我们有很大的把握相信 $\mu_1 > \lambda_2$.

由于 $S_{12}, S_{21} \neq 0$, 所以 S 的第二大特征向量可能有负的分量, 这和 $w_i \geq 0, \forall \boldsymbol{x}_i$ 相矛盾. 在本文中, 我们采用了一种比较粗糙的处理方法, 即忽略所有的负分量, 并将它们置为 0.

尽管上面只讨论了两个类的情况, 但不难类推到 k 个类的情况中去. 这样便得到了先前给出的多类特征中心估计的算法. 一旦估计出了各个类的特征中心, 我们就可以将用这些特征中心作为初始点, 初始化 KM. 这种改进后的 KM 算法称为特征 KM(Eigen K means) 算法, 简写为 EKM.

4 实验

我们分别在人工数据集和真实数据集上做实验来证实这一方法的有效性, 同时作为比较, 我们也给出前面提到的 Ward 方法, MaxMin 方法, KA 和随机初始化的 KM 聚类结果.

这里我们用纯度 (Purity)[10] 来衡量聚类结果. 限于篇幅, 在此不作详细阐述. 简言之, 纯度值在 0 到 1 之间, 纯度越高, 聚类的结果越好.

首先, 我们用 EKM 对 2 维的人工数据集作聚类, 这样便于直接观察到聚类结果.

图 1 是一个 2 维的人造数据集, 有三个类, 还有一些噪音数据点. 5 种初始化 KM 方法得到的聚类结果分别用图 2~6 表示出. 可以看到 EKM 和 KA 的初始点选择得非常好, EKM 估计的类中心十分准确. 而其他方法的聚类效果不太理想, 其中 MaxMin 偏好选那些偏远的点, 显然这是不合理的.

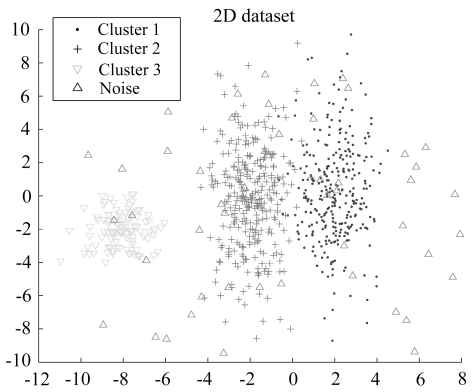


图 1 2D 人工数据集

Fig. 1 2D synthetic dataset

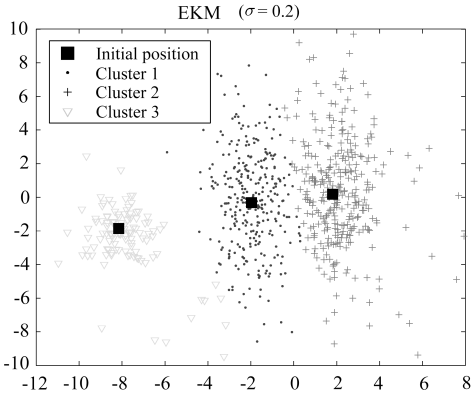


图 2 特征中心初始化 ($\sigma = 0.2$)

Fig. 2 Initialization with eigencenters($\sigma = 0.2$)

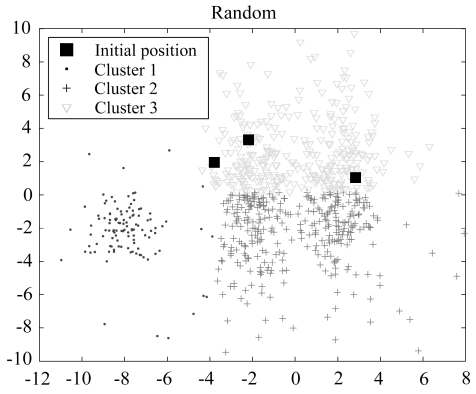


图 3 随机初始化

Fig. 3 Random initialization

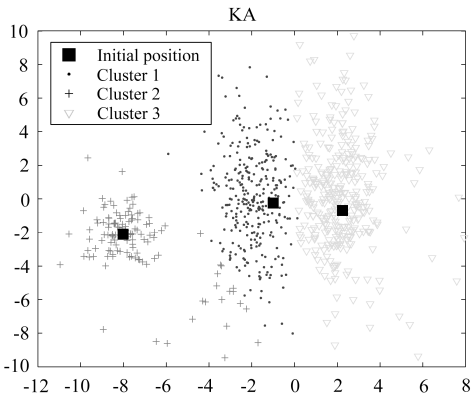


图 4 KA 初始化

Fig. 4 Initialization by KA algorithm

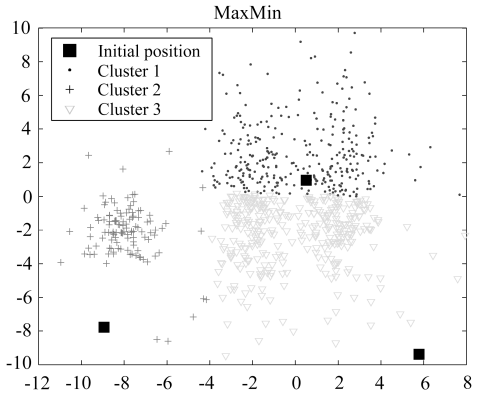


图 5　MaxMin 初始化

Fig. 5　Initialization by MaxMin algorithm

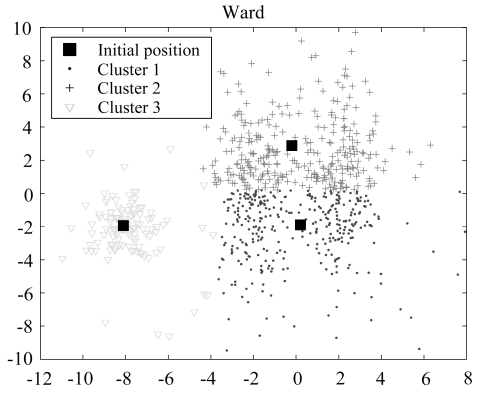


图 6　Ward 初始化

Fig. 6　Initialization by Ward algorithm

接下来, 在真实数据上做实验. 我们用到了 4 个数据集: iris 数据集[11], image segmentation 数据集[11], texture 数据集[11] 和 diabetes 数据集[12]. 表 3 给出了这些数据集的描述.

表 3　真实数据集

Table 3　Real datasets

数据集	维数 d	类的个数 k	数据点总数 N
iris	4	3	150
diabetes	3	3	145
segmentation	18	7	2310
texture	19	11	5500

对每个数据集, 假设类的个数已知, 分别用 5 种算法初始化 KM, 对于随机初始化 KM 算法, 我们做 100 次实验, 每次都随机抽取初始点, 再对这 100 次 KM 结果纯度求一个平均纯度. 由于 segmentation 数据集和 texture 数据集较大, 所以实验中对这两个数据集进行了采样. 一共采样 50 次, 每次随机抽取 7 个类, 每个类选 50 个子样组成一个子样集, 并在这个子样集上做实验, 最后取这 50 次采样实验结果纯度的均值作为该数据集上聚类结果的纯度. 实验结果如表 4 所示, 其中括号前的数字表示结果的纯度, 括号内的数字表示运行所耗费的时间, 单位为秒. EKM 中相应的参数 σ 分别为 iris: $\sigma = 1$, diabetes: $\sigma = 0.5$, segmentation: $\sigma = 1$, texture: $\sigma = 3$. 粗体字标明了较高的纯度值. 可以看出, 除了在第四个数据集上的结果不如 KA 之外, EKM 的纯度总是最高的. 由于在 segmentation 数据集和 texture 数据集上进行了采样, 因此我们对这两个数据集上的实验结果纯度进行了置信水平 α=0.02 的 t 检验, 结果表明, 在 segmentation 数据集上, EKM 算法的结果要明显好于其它算法. 而在 texture 数据集上, EKM 的结果与 KA 的结果无显著差异. 从运行时间上来看, EKM 要比随机算法和 MaxMin 算法长, 但比 KA 和 Ward 短.

前面的实验基于一个假设, 就是 k 是已知的, 但一般情况下, k 是未知的. 因此我们对于 k 未知的情况也做了实验, 并作比较. 从 segmentation 数据集中随机抽取了 5 个类, 每个类有 50 个样本点, 然后分别让 k=2,3,4,5,6,7, 聚类, 并与真实的类别作比较, 得到其纯度值. 反复这样的实验 50 次, 取平均纯度作为最后的实验结果, 如下页表 5 所示. 可见, 即使是在 k 不等于真实类别个数的情况下, EKM 算法仍旧能够得到很好的结果. 相比之下, KA 的结果与 EKM 十分接近, 也比较满意. Ward 方法和 Random 方法的结果居中, 而 MaxMin 的方法的实验结果却十分不理想.

表 4　真实数据集上的实验结果

Table 4　Clustering results of real datasets

	Random	EKM	MaxMin	KA	Ward
iris	0.81(0.009)	**0.893**(0.097)	**0.893**(0.034)	0.887(0.332)	**0.893**(0.242)
diabetes	0.792(0.008)	**0.841**(0.09)	0.821(0.033)	0.641(0.31)	**0.841**(0.239)
segmentation	0.585(1.58)	**0.660**(33.6)	0.407(2.97)	0.631(45.0)	0.598(101.4)
texture	0.771(2.3)	0.878(45.1)	0.5022(3.3)	**0.883**(49.8)	0.838(99.0)

表 5 不同 k 的取值下 segmentation 据集上的实验结果

Table 5 Clustering results of segmentation datasets with different k

k	Random	EKM	MaxMin	KA	Ward
2	0.379(0.54)	**0.398**(3.84)	0.214(0.69)	0.391(3.98)	0.392(3.33)
3	0.514(0.73)	**0.583**(7.10)	0.227(0.94)	0.579(8.59)	0.526(8.13)
4	0.616(0.91)	**0.715**(11.57)	0.383(1.18)	0.691(14.98)	0.661(17.03)
5	0.672(1.09)	**0.733**(17.7)	0.448(1.43)	0.726(23.54)	0.699(31.83)
6	0.695(1.27)	0.697(25.32)	0.518(1.68)	**0.701**(33.88)	0.695(57.55)
7	0.650(1.43)	**0.690**(34.62)	0.538(1.90)	0.685(45.99)	0.676(95.39)

5 总结

本文提出了一种新的初始化 KM 的算法, 该算法先估计出各个类的特征中心, 再用这些中心点初始化 KM. 实验证明, 估计出的类的特征中心与真实的类中心十分接近, 从而使得 KM 的性能大大地提高. 此外, 这个估计类的特征中心的方法不仅可以和 KM 结合, 还可以和高斯混合模型等其他局部搜索的聚类算法相结合.

References

1 Milligan G W. An examination of the effect of six types of error perturbation on fifteen clustering algorithms. *Psychometrika*, 1980, **45**(3): 325∼342

2 Ward J H. Hierarchical grouping to optimize an objective function. *Journal of American Statistical Association*, 1963, **58**: 236∼244

3 Higgs R E, Bemis K G, Watson I A, Wikel J H. Experimental designs for selecting molecules from large chemical databases. *Journal of Chemical Information and Computer Sciences*, 1997, **37**(5): 861∼870

4 Snarey M, Terrett N K, Willet P, Wilton D J. Comparison of algorithms for dissimilarity-based compound selection. *Journal of Molecular Graphics & Modelling*, 1997, **15**(6): 372∼385

5 Kaufman L, Rousseeuw P J. *Finding Groups in Data. An Introduction to Cluster Analysis*. Canada: John Wiley & Sons, Inc., 1990

6 Ng A Y, Jordan M I, Weiss Y. On spectral clustering: analysis and an algorithm. In: Proceedings of Neural Information Processing Systems Conference. 2001

7 Golub Gene H, Van Loan Charles F. *Matrix Computations*, 3rd edition. London: The Johns Hopkins University Press, 1996, 405∼414

8 Rao C R, Rao M B. *Matrix Algebra and Its Applications to Statistics*. World Scientific, 1998. 471

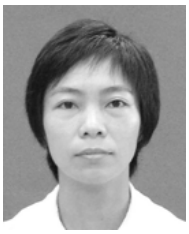
9 Bapat R B, Rachava T E S. *Nonnegative Matirces and Applications*. Cambridge University Press, 1997. 163∼164

10 Zhao Y, Karypis G. Criterion Functions for Document Clustering: Experiments and Analysis (Technical Report). 2001

11 Blake C L, Merz C J. UCI Machine Learning Repository[Online], available: http://www.ics.uci.edu/mlearn/MLRepository.html, 1998

12 Magidson J, Vermunt J K. Latent class analysis. Kaplan Ded., Handbook of Quantitative Methodology for the Social Sciences. Sage Publications, 2004

钱 线 复旦大学计算机系博士生. 2004 年获得复旦大学信息学院计算机系学士学位, 主要研究方向为自然语言处理, 机器学习. 本文通信作者.
E-mail: qianxian@fudan.edu.cn
(**QIAN Xian** Ph. D. candidate at Fudan University. He received his bachelor degree from Fudan University in 2000. His research interest covers natural language processing and machine learning. Corresponding author of this paper.)



黄萱菁 复旦大学计算机系教授. 1998 年获得复旦大学信息学院计算机系博士学位. 研究领域为自然语言处理. E-mail: xjhuang@fudan.edu.cn
(**HUANG Xuan-Jing** Professor at Fudan University. She received her Ph. D. degree in 1998 in Computer Science Department from Fudan University. Her research interest covers natural language processing.)

吴立德 复旦大学计算机系教授, 1958 年毕业于复旦大学数学系数学专业. 研究领域为自然语言处理, 图像处理. E-mail: ldwu@fudan.edu.cn
(**WU Li-De** Professor at Fudan University. He graduated in 1958 from Mathematic Department of Fudan University. His research interest covers natural language processing and image processing.)