

谱聚类算法综述^{*}

蔡晓妍 戴冠中 杨黎斌

(西北工业大学自动化学院 西安 710072)

摘要 谱聚类算法是近年来国际上机器学习领域的一个新的研究热点。谱聚类算法建立在谱图理论基础上,与传统的聚类算法相比,它具有能在任意形状的样本空间上聚类且收敛于全局最优解的优点。本文首先介绍了图论方法用于聚类的基本理论,然后根据图划分准则对谱聚类算法进行分类,着重阐述了各类中的典型算法,并对算法进行了比较分析,最后进行总结并提出了几个有价值的研究方向。

关键词 谱聚类,谱图理论,图划分

Survey on Spectral Clustering Algorithms

CAI Xiao-yan DAI Guan-zhong YANG Li-bin

(College of Automation, Northwestern Polytechnical University, Xi'an 710072, China)

Abstract Spectral clustering algorithms are newly developing technique in recent years. Unlike the traditional clustering algorithms, these apply spectral graph theory to solve the clustering of non-convex sphere of sample spaces, so that they can be converged to global optimal solution. In this paper, the clustering principle based on graph theory is first introduced, and then spectral clustering algorithms are categorized according to rules of graph partition, and typical algorithms are studied emphatically, as well as their advantages and disadvantages are presented in detail. Finally, some valuable directions for further research are proposed.

Keywords Spectral clustering, Spectral graph theory, Graph partition

1 引言

聚类分析是机器学习领域中的一个重要分支^[1],是人们认识和探索事物之间内在联系的有效手段。所谓聚类(clustering)就是将数据对象分组成为多个类或簇(cluster),使得在同一簇中的对象之间具有较高的相似度,而不同簇中的对象差别较大。传统的聚类算法,如 k-means 算法、EM 算法等都是建立在凸球形的样本空间上,但当样本空间不为凸时,算法会陷入局部最优。

为了能在任意形状的样本空间上聚类,且收敛于全局最优解,学者们开始研究一类新型的聚类算法,称为谱聚类算法(Spectral Clustering Algorithm)。该算法首先根据给定的样本数据集定义一个描述成对数据点相似度的亲和矩阵,并计算矩阵的特征值和特征向量,然后选择合适的特征向量聚类不同的数据点。谱聚类算法最初用于计算机视觉^[3]、VLSI 设计^[4]等领域,最近才开始用于机器学习中^[5],并迅速成为国际上机器学习领域的研究热点。

谱聚类算法建立在图论中的谱图理论基础上,其本质是将聚类问题转化为图的最优划分问题,是一种点对聚类算法,对数据聚类具有很好的应用前景。但由于其涉及的理论知识较多,应用还处于初级阶段,因此国内这方面的研究报道非常少。本文作为一篇综述性文章,旨在向读者介绍这种新型的聚类算法,使研究人员尽可能详细地总结该领域的研究现状。后续篇幅安排如下:第 2 部分介绍了图论方法用于聚类的基本理论;第 3 部分根据图划分准则,将谱聚类算法分为两

类,详细研究了各类中的代表性算法,并对算法进行了比较分析;最后对本文进行总结,提出了几个有价值的研究方向。

2 基本理论

2.1 图划分准则

谱聚类算法的思想来源于谱图划分理论^[2]。假定将每个数据样本看作图中的顶点 V ,根据样本间的相似度将顶点间的边 E 赋权重值 W ,这样就得到一个基于样本相似度的无向加权图 $G=(V,E)$ 。那么在图 G 中,就可将聚类问题转化为在图 G 上的图划分问题。基于图论的最优划分准则就是使划分成的两个子图内部相似度最大,子图之间的相似度最小^[5]。划分准则的好坏直接影响到聚类结果的优劣。常见的划分准则有 Minimum cut, Average cut, Normalized cut, Min-max cut, Ratio cut, MNCut 等。下面我们将分别介绍这几种准则。

2.1.1 最小割集准则^[6](Minimum cut)

谱图理论中,将图 G 划分为 A, B 两个子图(其中 $A \cup B = V, A \cap B = \phi$)的代价函数为:

$$cut(A, B) = \sum_{u \in A, v \in B} w(u, v) \quad (1)$$

Wu 和 Leahy 提出最小化上述剪切值来划分图 G ,这一划分准则被称为最小割集准则。他们用这个准则对一些图像进行分割,并产生了较好的效果,同时他们也注意到,该准则容易出现歪斜(即偏向小区域)分割。Shi 和 Malik 提出的规范割集准则及 Hagen 和 Kahng 提出的比例割集准则均可避免这种情况的发生。

^{*}基金项目:国家 863 计划资助项目(2005AA147030)。蔡晓妍 博士生,主要研究方向为智能信息处理、网络与信息安全;戴冠中 教授,博士生导师,主要研究领域为自动控制、信息安全;杨黎斌 博士生,研究方向为网络与信息安全、嵌入式系统。

2.1.2 规范割集准则^[5] (Normalized cut)

Shi 和 Malik 在 2000 年根据谱图理论建立了 2-way 划分的规范割目标函数(Ncut):

$$Ncut(A, B) = \frac{cut(A, B)}{assoc(A, V)} + \frac{cut(A, B)}{assoc(B, V)}$$

其中 $assoc(A, V) = \sum_{u \in A, t \in V} w(u, t)$ (2)

最小化 Ncut 函数被称为规范割集准则。该准则不仅能够衡量类内样本间的相似程度,也能衡量类间样本间的相异程度。

Shi 和 Malik 同时还提出了规范关联目标函数(Nassoc):

$$Nassoc(A, B) = \frac{assoc(A, A)}{assoc(A, V)} + \frac{assoc(B, B)}{assoc(B, V)} \quad (3)$$

$assoc(A, A)$ 与 $assoc(B, B)$ 分别是子图 A, B 内所有顶点之间的连接权值之和。这一无偏函数进一步反映了类内样本间互相连接的紧密程度。同时,可以看出 Ncut 函数与 Nassoc 函数是相关的,它们满足关系: $Ncut(A, B) = 2 - Nassoc(A, B)$ 。因此,最小化 Ncut 函数等价于最大化 Nassoc 函数,但通常情况下都是通过最小化 Ncut 函数获取图的最优划分。

2.1.3 比例割集准则^[7] (Ratio cut)

Hagen 和 Kahng 提出了比例割目标函数(Rcut):

$$Rcut = \frac{cut(A, B)}{\min(|A|, |B|)} \quad (4)$$

其中 $|A|, |B|$ 分别表示子图 A, B 中顶点的个数。最小化 Rcut 函数只考虑了类间相似性最小,减小了过分割的可能性,但运行速度较慢。

2.1.4 平均割集准则^[8] (Average cut)

平均割目标函数为:

$$Avcut(A, B) = \frac{cut(A, B)}{|A|} + \frac{cut(A, B)}{|B|} \quad (5)$$

可以看出 Avcut 和 Ncut 函数都表示无向图 G 中边界损失与分割区域相关性的比值之和,因此最小化 Avcut 与 Ncut 目标函数都能产生较准确的划分。其共同缺点是倾向于欠分割且易分割出只包含几个顶点的较小子图。文献[5]通过实验发现,当把 Normalized cut 和 Average cut 准则分别用于同一图像的分割问题时,Normalized cut 准则能够产生更好的划分结果。

2.1.5 最小最大割集准则^[9] (Min-max cut)

最小最大割集准则要求最小化 $cut(A, B)$ 的同时,最大化 $assoc(A, A)$ 与 $assoc(B, B)$ 。该准则可通过最小化下面的目标函数得以实现:

$$Mcut = \frac{cut(A, B)}{assoc(A, A)} + \frac{cut(A, B)}{assoc(B, B)} \quad (6)$$

我们将这个目标函数称为最小最大割函数,或简称为 Mcut 函数。最小化该函数可避免分割出仅包含几个顶点的较小子图,因此它倾向于产生平衡割集,但实现速度较慢。Mcut 与 Ncut 一样满足类间样本间的相似度小而类内样本间的相似度大的原则,与 Ncut 具有相似的行为,但当类间重叠较大时,Mcute 比 Ncut 更加高效。

2.1.6 多路规范割集准则^[10] (Multiway Normalized cut)

上述五种划分准则所使用的目标函数都是将图 G 划分为 2 个子图的 2-way 划分函数,Meila 提出一种可以将图 G 同时划分为 k 个子图的 k -way 规范割目标函数:

$$MNcut = \frac{cut(A_1, V - A_1)}{assoc(A_1, V)} + \frac{cut(A_2, V - A_2)}{assoc(A_2, V)} + \dots + \frac{cut(A_k, V - A_k)}{assoc(A_k, V)} \quad (7)$$

Melia 指出 Ncut 和 MNcut 的差异之处仅在于所使用的谱映射不同,并且当 $k=2$ 时, MNcut 与 Ncut 等价。多路规范割集准则在实际应用中合理有效,但其优化问题通常难以解决。

2.2 相似矩阵、度矩阵及 Laplacian 矩阵

由于图划分问题的本质,求图划分准则的最优解是一个 NP 难问题。一个很好的求解方法是考虑问题的连续放松形式,这样便可将原问题转换成求解相似矩阵或 Laplacian 矩阵的谱分解,因此将这类方法统称为谱聚类,可以认为谱聚类是对图划分准则的逼近^[11]。

相似矩阵通常用 W 或 A 表示,有时也称为亲和矩阵(Affinity Matrix)。该矩阵的定义为:

$$W_{ij} = \exp\left(-\frac{d(s_i, s_j)}{2\sigma^2}\right) \quad (8)$$

其中 s_i 表示每个数据样本点, $d(s_i, s_j)$ 一般取 $\|s_i - s_j\|^2$, σ 为事先指定的参数。

将相似矩阵的每行元素相加,即得到该顶点的度,以所有度值为对角元素构成的对角矩阵即为度矩阵,度矩阵常用 D 表示。

Laplacian 矩阵分为非规范 Laplacian 矩阵和规范 Laplacian 矩阵。非规范 Laplacian 矩阵表示为 $L = D - W$, 规范 Laplacian 矩阵有两种形式,分别是:

$$\begin{aligned} L_{sym} &= D^{-\frac{1}{2}} L D^{-\frac{1}{2}} = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}} \\ L_{rw} &= D^{-1} L = I - D^{-1} W \end{aligned} \quad (9)$$

2.3 势函数、Fiedler 向量及谱

势函数为表示某样本划分归属的指示向量(indicator vector),其定义为:

$$q_i = \begin{cases} 1 & \text{若 } i \in A \\ 0 & \text{若 } i \in B \end{cases} \quad (10)$$

若最终势函数中某样本对应的值为 1,则该样本属于集合 A ,若为 0 则属于集合 B 。但实际划分求解得到的结果 q_i 常为 0 到 1 之间的实数值,此时可用 k 均值聚类等方法进一步决定样本的归属。

许多谱聚类算法都将图划分问题转化为求解 Laplacian 矩阵的第二小特征向量问题。这里的第二小特征向量就是第二个最小特征值对应的特征向量,它代表了最佳图划分的一个解(即势函数),把这一特征向量称为 Fiedler 向量。与特征向量(不一定是 Fiedler 向量)对应的特征值称为谱。

3 谱聚类算法

根据不同的准则函数及谱映射方法,谱聚类算法发展了很多不同的具体实现方法,但是都可以归纳为下面三个主要步骤^[12]:

Step1 构建表示样本集的矩阵 Z ;

Step2 通过计算 Z 的前 k 个特征值与特征向量,构建特征向量空间;

Step3 利用 k -means 或其它经典聚类算法对特征向量空间中的特征向量进行聚类。

上述步骤是谱聚类算法的一个框架,在具体实现过程中,不同的算法在数据集矩阵 Z 的表示上存在着不同。例如根据 2-way cut 的目标函数, $Z=W$; 根据随机游动关系,则 $Z=D^{-1}W$ 等。划分准则一般分为 2-way 和 k -way, 本文根据所使用的划分准则,将算法分为迭代谱和多路谱两类,并分别讨论了各类中典型的谱聚类算法。

3.1 迭代谱聚类算法

3.1.1 PF 算法^[13]

Perona 和 Freeman 提出用相似矩阵 W 的第一个特征向量 x_1 进行聚类(本文中第一个特征向量指的是相应矩阵中最大特征值 λ_1 所对应的特征向量)。他们指出对于块对角相似矩阵,特征向量 x_1 中非零元素对应的点属于同一类,零元素对应的点属于另外一类。PF 算法作为最简单的谱聚类算法引起了学术界的广泛关注,在理论研究和应用方面都有很多论文发表。

3.1.2 SM 算法^[5]

SM 算法由美国学者 Shi 和 Malik 在 2000 年提出。他们指出在约束 $x^T W e = x^T D e = 0$ 条件下,

$$\min Ncut(A, B) = \min \frac{x^T (D-W)x}{x^T D x} \quad (11)$$

将 x 松弛到连续域 $[-1, 1]$, 求解 $\min Ncut(A, B)$ 的问题就可以转化为下式:

$$\arg \min_{x^T D = 0} \frac{x^T (D-W)x}{x^T D x} \quad (12)$$

根据 Rayleigh 商原理,式(12)的优化问题等于求解下列等式的第二小特征值问题:

$$(D-W)x = \lambda D x \quad (13)$$

与第二小特征值对应的特征向量(即 Fiedler 向量)就包含了图的划分信息。SM 算法描述如下:

Step1 通过样本集建立无向加权图 G , 根据图 G 构造矩阵 W 和 D ;

Step2 计算式(13)第二小特征值及对应的 Fiedler 向量;

Step3 根据启发式规则在 Fiedler 向量中寻找划分点 i , 使在该点上得到的 $Ncut(A, B)$ 值最小。将 Fiedler 向量中大于等于该值的点归为一类,小于该值的点归为另一类。

定义规范相似矩阵为:

$$N = D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$$

$$\text{即 } N(i, j) = \frac{W(i, j)}{\sqrt{D(i, i)} \sqrt{D(j, j)}} \quad (14)$$

根据正规化定理^[4], 可以得出矩阵 W 的 Fiedler 向量等于矩阵 N 的次大与最大特征向量之间的分量方式比值。因此, SM 算法与 PF 算法的不同之处在于: ① SM 算法用的是规范相似矩阵, 而 PF 算法仅用相似矩阵; ② SM 算法用的是前两个特征向量, 而 PF 算法只用了第一个特征向量。

Shi 和 Malik 同时还提出将 $\min Avcut$ 的离散形式松弛为连续形式进行聚类。但实验表明, SM 算法得出的聚类效果要明显好于用 PF 算法及最小化 $Avcut$ 标准得到的聚类效果。

3.1.3 SLH 算法^[14]

SLH 重定位算法将相似矩阵 W 和数字 k 作为输入, 并通过下面的步骤输出一个新的矩阵 Q :

Step1 求矩阵 W 的前 k 个特征向量 x_1, \dots, x_k , 构造矩阵 $X = [x_1, \dots, x_k] \in R^{n \times k}$;

Step2 将矩阵 X 的行向量规范化为单位长度并构造新矩阵 $Q = X X^T$;

Step3 根据 Q 中的元素聚类相应的点。理想情况下, $Q(i, j) = 1$ 表示 i, j 两点属于同一类, $Q(i, j) = 0$ 表示 i, j 两点属于不同类。一般情况下, 属于同一类的点对应的 $Q(i, j)$ 值均接近 1, 属于不同类的点对应的 $Q(i, j)$ 值接近 0。

Weiss 提出了一种将 SLH 算法和 SM 算法相结合的新型谱聚类算法^[4]。该算法将 SLH 算法中的原始相似矩阵 W 用规范相似矩阵 N 代替。实验表明, 此算法能够更加快速地产生产正确的聚类结果。

3.1.4 KVV 算法^[15]

KVV 算法与 SM 算法十分相似, 不同之处仅在于 SM 算法是寻找使 $Ncut(A, B)$ 值最小的划分点, 而 KVV 算法则是寻找使 $Rcut(A, B)$ 值最小的划分点。KVV 算法减少了过分割的可能性, 但算法的运行速度相对较慢。

3.1.5 Mcut 算法^[16]

Ding 根据谱图理论, 将(6)式重新写为:

$$Mcut = \frac{x^T (D-W)x}{x^T W x} + \frac{y^T (D-W)y}{y^T W y} \quad (15)$$

对于 2-way 划分, 令 q 为划分指示向量, 则:

$$q_i = \begin{cases} a & i \in A \\ -b & i \in B \end{cases} \quad (16)$$

最小化式(15)可得

$$\min_q Mcut(A, B) = \min_q \frac{J_N(A, B)}{1 - J_N(A, B)/2} \Rightarrow \min_q J_N(A, B)$$

$$J_N(A, B) \equiv J_N(q) = \frac{q^T (D-W)q}{q^T D q} \quad (17)$$

在约束 $x^T e = 0$ 条件下, 将 x 松弛到连续域 $[-1, 1]$, 根据 Rayleigh 商原理, (17)式的优化问题等于求解下式的第二小特征值问题:

$$(D-W)x = \frac{\lambda}{1+\lambda} D x \quad (18)$$

因此, 将图 G 划分为两个子图的 Mcut 算法可以简单描述如下:

Step1 计算(18)式的 Fiedler 向量。

Step2 在 Fiedler 向量中, 寻找使 Mcut 值最小的划分点。

Step3 用基于连接的聚类算法进行优化。

Kannan 将该算法与 SM 算法、KVV 算法进行了比较, 发现 Mcut 算法能够产生更加平衡的划分结果, 尤其当类间重叠较大时, 效果更为明显。

3.2 多路谱聚类算法

大部分谱聚类算法都是利用 2-way 划分准则迭代地对样本数据进行聚类。但是近几年研究发现, 若使用更多的特征向量并且直接计算 k 路分割将会得到更好的聚类效果^[17, 18]。

3.2.1 NJW 算法^[19]

Ng, Jordan 等人选取拉氏矩阵 L_{sym} 的前 k 个最大特征值对应的特征向量, 使其在 R^k 空间中构成与原数据一一对应的表述, 然后在 R^k 空间中进行聚类。NJW 算法描述如下:

Step1 计算矩阵 L_{sym} 的前 k 个最大特征值所对应的特征向量 x_1, \dots, x_k (必要时需作正交化处理), 构造矩阵 $X = [x_1, \dots, x_k] \in R^{n \times k}$;

Step2 将矩阵 X 的行向量转变为单位向量, 得到矩阵 Y , 即 $Y_{ij} = \frac{X_{ij}}{\sqrt{\sum_j X_{ij}^2}}$

Step3 将矩阵 Y 的每一行看作是 R^k 空间中的一个点, 对其使用 k 均值算法或任意其它经典算法, 得到 k 个聚类;

Step4 将数据点 y_i 划分到聚类 j 中, 当且仅当 Y 的第 i 行被划分到聚类 j 中。

选取矩阵 L_{sym} 的前 k 个最大特征值所对应特征向量的原因在于: 对于存在 k 个理想的彼此分离簇的有限数据集, 可以

证明矩阵 L_{sym} 的前 k 个最大特征值为 1, 第 $k+1$ 个特征值则严格小于 1, 二者之间的差距取决于这 k 个聚类的分布情况^[19]。当聚类内部分布得越密, 各聚类间分布得越开时, 第 $k+1$ 个特征值就越小, 此时, 以 Y 矩阵中的每行作为 k 维空间中的一个点所形成的 k 个聚类, 它们将彼此正交地分布于 k 维空间中的单位球上, 并且在单位球上形成的这 k 个聚类对应着原空间中所有点形成的 k 个聚类。

3.2.2 MS 算法^[20]

Meila 将相似性解释为 Markov 链中的随机游动, 分析了这种随机游动的概率转移矩阵 $P=D^{-1}W$ 的特征向量, 并根

据随机游动对 MNcut 进行了概率解释。在这个解释框架下提出了多个特征相似矩阵组合下的谱聚类算法。算法步骤与 NJW 算法相似, 但该算法是用随机游动矩阵 P 的前 k 个特征向量构造矩阵 X , 并直接将矩阵 X 中的各行看成 R^k 空间中的各点进行聚类。MS 算法在实际应用中取得了一定的效果, 但当度矩阵 D 中各对角线元素差别较大时, 聚类效果较差^[19]。

以上对典型的谱聚类算法进行了概述, 表 1 为这些算法的一个简单比较。

表 1 典型谱聚类算法的比较

算法	目标函数	所用方程	选取的特征向量	特点
PF 算法	cut(A, B)	$Wx = \lambda x$	最大特征向量	易产生仅包含几个顶点的较小簇
SM 算法	Ncut(A, B)	$(D-W)x = \lambda Dx$	Fiedler 向量	当类间重叠较大时易出现歪斜分割
SLH 算法	—	$Wx = \lambda x$	前 k 个特征向量	速度较慢
Mcut 算法	Mcut(A, B)	$(D-W)x = \frac{\lambda}{1+\lambda} Dx$	Fiedler 向量	能够产生更加平衡的划分结果, 但后置处理需要花费大量时间
NJW 算法	—	$D^{-\frac{1}{2}}WD^{-\frac{1}{2}}x = \lambda x$	前 k 个特征向量	需要进行正交初始化
MS 算法	MNcut	$D^{-1}Wx = \lambda x$	前 k 个特征向量	当度矩阵 D 中各对角线元素差别较大时, 聚类效果较差

4 当前的几个热点研究问题

尽管谱聚类相对于其它聚类方法具有许多优势, 并在实践中也取得了很好的效果, 但它仍有许多亟需研究和解决的问题, 尤其在下述几个方面:

1) 如何构造相似矩阵 W : 谱聚类算法中相似矩阵 W 的构造依赖于相似函数 W_{ij} 的构造。本文介绍的算法所使用的相似函数均由式(8)表示, 由于尺度参数 σ 是人为选取的, 使得该函数带有一定的局限性。文献^[19]通过反复运行 NJW 算法来自动确定 σ 的大小, 消除了人为因素, 却增加了运算时间。当前, 绝大部分文献中相似矩阵的构造都依赖于领域知识, 而没有给出通用的规则, 系统地研究谱聚类中相似矩阵的构造问题将是未来研究的一个热点。

2) 如何处理特征向量: 在应用谱方法进行聚类问题的研究中, 用多少个特征向量进行聚类, 如何选取、计算及使用这些特征向量等问题均没有得到很好的理论解释, 这些都是未来急需解决的问题。

3) 如何自动确定聚类数目: 聚类数目的多少直接影响聚类的质量。已有的自动确定聚类数目的谱聚类算法有: 基于 Rcut 的谱聚类算法^[21], 非线性降维(NLDR, nonlinear dimensionality reduction)算法^[22]和基于距离的启发式方法^[23]。但用这些算法在一些公共数据集上得出的聚类数和相应的聚类结果都不能令人满意^[23]。所以, 如何自动地确定聚类数目是一个关键的问题, 是未来研究的方向。

4) 如何选取 Laplacian 矩阵: 谱聚类算法所使用的 Laplacian 矩阵有三种形式(见本文 2.2 节), 但这三种形式的 Laplacian 矩阵所应用的具体环境还没有得到彻底解决。文献^[23]提出应首先分析度矩阵 D 中各对角线元素 d_{ii} , 若各 d_{ii} 大小大致相同, 则用这三种形式的 Laplacian 矩阵得出的聚类结果基本相同; 反之, 聚类结果也大相径庭。如何根据具体环境选择合适的 Laplacian 矩阵, 我们还需要进行大量的理论研究和实验工作。

5) 如何运用到大规模学习问题中: 由于谱聚类算法涉及求解特征值和特征向量问题, 计算复杂度较大, 不利于进行大规模的计算和扩展, 因此, 研究高效、可扩展、适宜大规模学习问题的谱聚类算法是我们未来的研究方向。

结束语 谱聚类算法是聚类分析中一个崭新的分支, 由于这种算法不用对数据的全局结构作假设, 并且具有识别非凸分布聚类的能力, 非常适合于许多实际问题, 因此在短短的几年时间内, 引起了国际学术界的广泛关注。谱聚类算法的研究将极大丰富聚类算法的研究内容, 给聚类问题的求解提供了新的思路, 具有巨大的科研价值和应用潜力。希望通过本文介绍, 使读者对该领域有一个初步的认识, 并能将此方法应用到科学和工程领域中的各种聚类问题中去。

参考文献

- [1] Jain A, Murty M, Flynn P. Data clustering; A Review[J]. ACM Computing Surveys, 1999, 31(3):264-323
- [2] Fiedler M. Algebraic connectivity of graphs. Czech, Math. J., 1973, 23:298-305
- [3] Malik J, Belongie S, Leung T, et al. Contour and texture analysis for image segmentation. In Perceptual Organization for Artificial Vision Systems. Kluwer, 2000
- [4] Weiss Y. Segmentation using eigenvectors; A unified view// International Conference on Computer Vision. 1999
- [5] Shi J, Malik J. Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000, 22(8):888-905
- [6] Wu Z, Leahy R. An optimal graph theoretic approach to data clustering; theory and its application to image segmentation[J]. IEEE Trans on PAMI, 1993, 15(11):1101-1113
- [7] Hagen L, Kahng A B. New spectral methods for ratio cut partitioning and clustering. IEEE Trans. Computer-Aided Design, 1992, 11(9):1074-1085
- [8] Sarkar S, Soundararajan P. Supervised learning of large percep-

- tual organization; Graph spectral partitioning and learning automata. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 2000, 22(5):504-525
- [9] Ding C, He X, Zha H, et al. Spectral Min-Max cut for Graph Partitioning and Data Clustering[C]//Proc. of the IEEE Intl Conf. on Data Mining, 2001, 107-114
- [10] Meila M, Xu L. Multiway cuts and spectral clustering. U. Washington Tech Report. 2003
- [11] Zhou D, Bousquet O, Lal T N, et al. Scholkopf: Learning with Local and Global Consistency. *Advances in Neural Information Processing Systems*//Thrun, S. L. Saul and B. Scholkopf, eds. MIT Press, Cambridge, MA, USA, 2004, 16:321-328
- [12] Verma D, Meila M. A comparison of spectral clustering algorithms. Technical report, 2003. UW CSE Technical report 03-05-01
- [13] Perona P, Freeman W T. A factorization approach to grouping //H. Burkardt and B. Neumann, eds. Proc. ECCV. 1998:655-670
- [14] Scott G L, Longuet-Higgins H C. Feature grouping by relocalization of eigenvectors of the proximity matrix// Proc. British Machine Vision Conference. 1990:103-108
- [15] Kannan R, Vempala S, Vetta A. On clusterings- good, bad and spectral. In *FOCS*, 2000:367-377
- [16] Ding C, He X F, Zha H Y, et al. A min-max cut algorithm for graph partitioning and data clustering[A]//The 2001 IEEE International Conference on Data mining. San Jose, USA, 2001
- [17] Alpert C, Kahn A, Yao S. Spectral partitioning: The more eigenvectors, the better. *Discrete Applied Math.*, 1999, 90:3-26
- [18] Malik J, Belongie S, Leung T, et al. Contour and texture analysis for image segmentation. In *Perceptual Organization for Artificial Vision Systems*. Kluwer, 2000
- [19] Ng A Y, Jordan M I, Weiss Y. On spectral clustering; Analysis and an algorithm//T. G. Dietterich, S. Becker, and Ghahramani, eds. *Advances in Neural Information Processing Systems*, Cambridge, MA, MIT Press, 2002, 14:849-856
- [20] Meila M, Shi J. Learning segmentation by random walks. In *NIPS*, 2000:873-879
- [21] Chan P K, Schlag D F, Zien J. Spectral k-way ratio-cut partitioning and clustering[J]. *IEEE Tran on Computer Aided Design of Integrated Circuits and Systems*, 1994, 13(9):1088-1096
- [22] Luxburg U. A Tutorial on Spectral Clustering. Technical report. 2006
- [23] Brand M, Kun H. A Unifying Theorem for Spectral Embedding and Clustering[C]// Proceeding of the 9th International Conference on Artificial Intelligence and Statistics. Key West, Florida, 2003
- [24] Dhillon S, Guan Y, Kulis B. A unified view of kernel k-means, spectral clustering and graph cuts. University of Texas at Austin UTCS Technical Report TR-04-25. 2004
- [25] Zhang T, Ando R. Analysis of spectral kernel design based semi-supervised learning// Y. Weiss, B. Scholkopf and J. Platt, eds. *Advances in neural information processing systems 18*. Cambridge, MA; MIT Press, 2006
- [26] David S, Luxburg U, Pal D. A sober look on clustering stability //G. Lugosi and H. Simon, eds. *Proceedings of the 19th Annual Conference on Learning Theory (COLT)*. Berlin, Springer, 2006:5-9
- [27] Bie T D, Cristianini N. Fast SD Prelaxations of graph cut clustering, transduction, and other combinatorial problems. *JMRL*, 2006(7):1409-1436
- [28] Lang K. Fixing two weaknesses of the spectral method//Weiss Y, Scholkopf B, Platt J, eds. *Advances in Neural Information Processing Systems*, Cambridge, MA; MIT Press, 2006, 18:715-722
- (上接第 13 页)
- [36] Doan A, Madhavan J, Domingos P, et al. Ontology matching: A machine learning approach[C]//S. Staab & R. Studer, eds. *Handbook on Ontologies in Information Systems*, Springer-Verlag, 2004:397-416
- [37] Schnurr H-P, Angele J. Do Not Use This gear with a Switching Lever! Automotive Industry Experience with Semantic Guides [C] // 4th International Semantic Web Conference (ISWC2005). LNCS 3729, 2005:1029-1040
- [38] 魏哲雄, 冯志勇. 基于字典技术的本体整合系统[J]. *计算机应用*, 2007, 27(2):428-430
- [39] Noy N F, Musen M A. An Algorithm for Merging and Aligning Ontologies; Automation and Tool Support[C]//Proceedings of the Workshop on Ontology Management at the Sixteenth National Conference on Artificial Intelligence (AAAI-99). Orlando, 1999
- [40] Noy N F, Musen M A. SMART: Automated Support for Ontology Merging and Alignment[C]//Proceedings of the Twelfth Workshop on Knowledge Acquisition Modeling and Management. Banff, Canada, 1999(7)
- [41] Horridge M, Knublauch H, et al. A practical guide to building OWL ontology using the protégé-OWL Plugin and CO-ODE tools[EB/OL]. The University of Manchester, 2004(8)
- [42] Noy N F, Klein M. Ontology evolution; Not the same as schema evolution[J]. *Knowledge and Information Systems*, 2004, 6(4):428-440
- [43] Klein M, Noy N F. A Component-Based Framework for Ontology Evolution[C]// Workshop on Ontologies and Distributed Systems at IJCAI-03. Acapulco, Mexico, 2003
- [44] Dou Dejiang, McDermott D, Qi Peishen. Ontology Translation on the Semantic Web[J]// Proceedings Int'l Conf. on Ontologies, Databases and Applications of Semantics (ODBASE2003). LNCS 2888, 2003:952-969
- [45] AIFB. University of Karlsruhe. KAON Demo Page[EB/OL]. <http://kaon.semanticweb.org/>. (Accessed on Oct. 6, 2006)
- [46] <http://www.daml.org/language/>. (Accessed on Mar. 20, 2004)
- [47] Ding Ying. Ontology Management System[EB/OL]. http://sw-portal.deri.at/papers/deliverables/d17_v01.pdf, 2004. 8. 31. (Accessed on Dec 21, 2006)
- [48] Harrison R, Chan C W. Distributed ontology management system[C]// Canadian Conference on Electrical and Computer Engineering. 2005(5)
- [49] Ontoprise. OntoStudio[EB/OL]. <http://ontoworld.org/wiki/OntoStudio>. (Accessed on Dec. 16, 2006)
- [50] Euzenat J. Evaluating ontology alignment methods[C]// Dagsstuhl Seminar Proceedings. Semantic Interoperability and Integration. 2005
- [51] Maedche A, Staab S. Discovering Conceptual Relations from Text[C]//Proceedings of the 14th European Conference on Artificial Intelligence (ECAI 2000). Amsterdam; IOS Press, 2000:321-325