基于超欧氏距离近邻传播的软件缺陷预测方法*

常瑞花[1]，沈晓卫[2]
(1. 武警工程大学 科研部，西安 710086; 2. 火箭军工程大学 核工程系，西安 710025)

摘 要： 为了进一步提高无标志软件缺陷数据预测的精度，提出了一种基于超欧氏距离近邻传播的软件缺陷预测方法。在近邻传播算法中引入密度思想，定义了密度因子和超欧氏距离测度概念，设计了密度敏感相似度度量元(即密集度量元)，解决了传统近邻传播算法采用欧氏距离表示数据相似度难以有效处理复杂结构数据的不足。该方法应用于无标志软件缺陷数据的预测，并通过三组航空航天软件数据仿真验证了该方法的有效性，提高了无标志软件缺陷数据预测的精度，为无标志软件缺陷预测提供了一种新的思路。

关键词： 密度；近邻传播；软件缺陷；超欧氏距离；预测
中图分类号： TP311.5 **文献标志码：** A **文章编号：** 1001-3695(2017)05-1384-04
doi:10.3969/j.issn.1001-3695.2017.05.024

Software defect prediction based on affinity propagation with hyper Euclidean distance

Chang Ruihua[1], Shen Xiaowei[2]
(1. *Dept. of Research, Armed Police Engineering University, Xi'an* 710086, *China*; 2. *Dept. of Nuclear Engineering, University of Rocket Engineering, Xi'an* 710025, *China*)

Abstract: In order to improve the accuracy of prediction for unlabeled software defect data, this paper proposed a novel software defect prediction method based on affinity propagation with hyper Euclidean distance. The traditional affinity propagation algorithm used Euclidean distance to represent data similarity, it was difficult to meet the characteristics of global data consistency and couldn't effectively deal with the complex data structure. In order to overcome the shortages, this paper introduced the idea of density, and defined density factors and hyper Euclidean distance. Meantime, it designed the density sensitive similarity metrics. The new method was used to deal with the unlabeled and complicated software defectdata, and three data sets were used to verify the effectiveness of the proposed method. The experimental results show that the proposed method is effective. It improves the prediction accuracy of unlabeled data and provides a practical way for unlabeled software defect prediction.

Key words: density; affinity propagation; software defect; hyper Euclidean distance; prediction

0 引言

随着航空航天系统中软件所占比例的不断增加，系统可靠性越来越依赖于软件的可靠性，而软件缺陷往往是导致系统出错、失效和崩溃的潜在根源[1]。软件测试和故障诊断是保证软件质量的重要手段，然而完全的软件测试需要耗费大量时间和资源，软件故障诊断则主要用在故障发生后。相比之下，软件缺陷预测技术可以在软件发布之前预测软件模块的缺陷，使得软件测试人员将有限的资源和精力用在含有缺陷的模块上，从而进一步提高武器系统的可靠性。

在过去的 30 多年中，国内外学者已经展开了软件缺陷预测的相关研究，提出了许多模型[1,2]。早期模型大部分利用统计的方法构建，如多元线性回归(multiple linear regression，MLR)[3]，然而存在预测准确性不高、不能获得满意置信度的问题。鉴于此，近年来许多研究人员利用机器学习技术建立模型，提出许多软件缺陷预测的方法[4-7]，如支持向量机(support vector machine，SVM)[4]、决策森林(decision forest，DF)[5]、朴素贝叶斯(naive Bayes，NN)[6,7]等，并取得了不错的预测效果。但是这些传统研究主要集中在有监督的学习方法上，该类方法获得较好的预测性能依赖具有完整标记的软件缺陷数据，但现实中存在大量标记不完整或无标志软件缺陷数据，另外完整标记样本需要耗费大量的人力和财力。因此，如何利用无标志数据进行准确的软件缺陷预测成为当前一个亟待解决的问题[8]。Zhong 等人[9]提出利用 K-means 和 Neural-Gas 的方法对无标志数据进行聚类，但该方法需要具有丰富机器学习和软件工程经验的人员对聚类模块进行类别标志，从而限制了其广泛应用。Seliya 等人[10]以 K-means 为基聚类算法，提出一种半监督学习方法，该方法取得了较好的效果，但预测的结果依赖聚类个数的确定；Catal 等人[11]提出一种基于度量边界和 X-means 的方法，然而该方法对聚类数敏感。上述方法主要利用 K-means 或其扩展算法进行聚类预测，该类算法主要建立在凸样本空间分布上，当样本空间不为凸时，算法易于陷入局部最

收稿日期：2016-07-13；修回日期：2016-08-30 基金项目：国家自然科学基金资助项目(51503224)；陕西省自然科学基金资助项目(2015JQ6224)；武警工程大学基础研究基金资助项目(WJY201602)；大学军事理论研究项目(JLX201680)

作者简介：常瑞花(1982-)，女，山西清徐人，工程师，博士，主要研究方向为机器学习、模式识别(sxwcrh@163.com)；沈晓卫(1982-)，男，江苏盐城人，讲师，博士，主要研究方向为智能信息处理.

优，且由于其是基于梯度的方法，对初始聚类中心的选择敏感，对于不同的初值也会收敛到不同的局部极值，导致算法极不稳定。近些年来，Zhang 等人[12]提出了一种基于连接关系的跨项目无标志软件缺陷预测方法；Nam 等人[13]针对跨项目缺陷预测方法鲁棒性差以及传统方法手工标注等不足，提出了一种基于梯度的 CLAMI（clustering instances & labeling instance & metric selection & instance selection）方法。总之，无标志软件缺陷预测方法的研究还处在起步阶段，相关的研究还不是很多。

2007 年，Frey 等人[14]在《Science》上发表的文章中提出一种近邻传播算法（affinity propagation，AP），该算法将所有数据点同时作为潜在的聚类中心，通过数据点之间的消息传递产生高质量的聚类中心，避免了聚类中心的选择，尤其对于规模很大的数据集，AP 算法是一种快速、有效的聚类方法，这是其他传统的聚类算法（如 K-中心聚类算法）所不能及的。但是，AP 算法对于一些聚类结构比较复杂的数据集，往往不能得到很好的聚类结果[15,16]。由于 AP 算法在数据形成的相似性矩阵基础上进行聚类，当前出现了一些相应改进的方法，如建立可变相似性度量[15]、半监督聚类改进近邻传播算法的相似度矩阵[16]等，然而前者依赖于存在标签数据的情况下，后者对于聚类的精度还有待于进一步提高。

考虑到 AP 算法处理大规模数据的优势，本文将 AP 算法引入到软件缺陷数据预测领域，并针对其处理聚类结构复杂的数据效果不理想的问题，引入了密度思想，定义了超欧氏距离，提出了新的相似度量元——密集度度量元，并构造了相应的相似度矩阵；然后在 AP 算法的框架下，给出一种基于超欧氏距离的近邻传播缺陷预测方法（affinity propagation with hyper Euclidean distance，APHE）；最后通过仿真实验验证了该方法的可行性。

1　基于超欧氏距离近邻传播的软件缺陷预测方法

AP 算法是一种基于近邻信息传播的聚类算法，其目的是找到最优的类代表点集合（类代表点对应实际数据集中的数据点，exemplar），使得所有数据点到最近的类代表点的相似度之和最大。算法框架如图 1 所示。

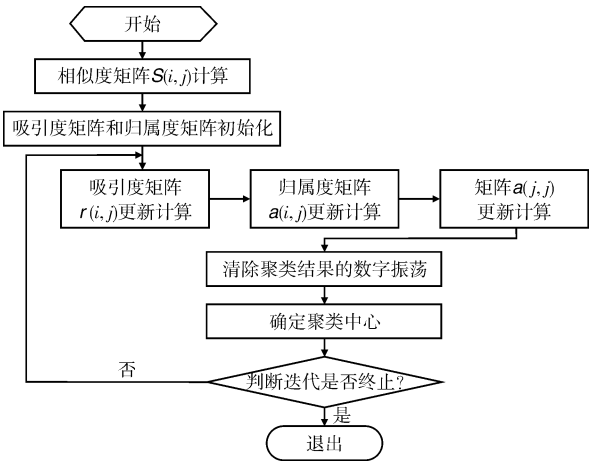

图 1　近邻传播算法的框架

从图 1 可以看出，AP 算法的输入是相似度矩阵 $S(i,j)$，作为一种信息传递算法，其在数据点间传递两种信息，分别为吸引度 $r(i,j)$ 和归属度 $a(i,j)$。$r(i,j)$ 表示点 i 传向候选聚类中心点 j，它反映点 j 适合作为点 i 聚类中心的程度，$r(i,j)$ 越大越适合；$a(i,j)$ 由候选聚类中心点 j 传向其所有潜在聚类成员点 i，它反映点 i 适合作为点 j 的聚类成员的程度，$a(i,j)$ 越大越适合，其初始值为 0。AP 算法不断迭代更新信息矩阵，当满足迭代结束条件时，返回结果。

传统的近邻传播聚类算法 AP 以 N 个数据点之间欧氏距离的相反数作为测度，构造 $N\times N$ 的相似度矩阵。当其处理结构复杂的软件度量元和缺陷数据时，欧氏距离仅反映了局部一致性性质，没有反映全局一致性性质[17]，因此，往往不能完全反映聚类结构，导致软件缺陷预测效果很不理想。鉴于此，将密度思想引入到近邻传播聚类算法中，定义了密度因子。

定义 1　密度因子。它是指分别以缺陷数据点 a、b 为圆心，γ 为半径的两个开球内，共同的数据点个数，具体如式（1）所示。

$$\mathrm{den}N(a,b)=|\{x|\rho(x,a)<\gamma\}\cap\{y|\rho(y,b)<\gamma\}| \tag{1}$$

其中：$\rho(x,a)<\gamma$ 表示以 a 为圆点、γ 为半径的超球体，x 表示该区域内数据点的个数。

软件缺陷预测中使用的软件度量元特征高达 20～40 维，传统的欧氏距离度量方法不能很好地反映高维数据间的相似度，文献[18]中设计的距离函数 hsim（ ）很好地克服了传统欧氏距离函数在高维空间中的不足，如式（2）所示。

$$\mathrm{hsim}(X,Y)=\frac{\left[\sum_{i=1}^{d}\frac{1}{1+|x_i-y_i|^p}\right]^{1/p}}{d} \tag{2}$$

其中：$X=(x_1,\cdots,x_d)$ 和 $Y=(y_1,\cdots,y_d)$ 是 d 维空间中的两个点。

受其启发，本文在数据的预处理阶段一方面利用非负矩阵分解的方法对高维数据进行维数的约简和降低；另一方面，为了进一步减小高维度对缺陷预测结果的影响，定义了超欧氏距离测度的概念。

定义 2　超欧氏距离。它表示两点之间的相似度（距离），x、y 两点之间的距离计算公式为

$$\mathrm{dist}(x,y)=\begin{cases}\sum_{i=1}^{d}|x_i-y_i|^d & d\leqslant 2\\ \dfrac{\sum_{i=1}^{d}|x_i-y_i|^2}{d} & d\geqslant 3\end{cases} \tag{3}$$

其中：d 为数据维数。可以看出式（3）在计算两点之间的距离时，根据维数进行了划分，当数据维数小于或等于 2 时，超欧氏距离测度和欧氏距离测度一致。

最后，给出密集度量元的计算公式，如式（4）所示。

$$SN(x,y)=\begin{cases}0 & x=y\\ -\mathrm{dist}(x,y)\,\mathrm{e}^{\mathrm{den}N(x,y)} & x\neq y\end{cases} \tag{4}$$

设计了一种基于密度敏感的相似性度量元（密集度量元）后，在 AP 算法的框架下，给出一种基于超欧氏距离的近邻传播缺陷预测方法 APHE。具体如下所示：

APHE 方法与传统的近邻传播算法 AP 最大的不同是相似度矩阵的构造。同 AP 算法类似，APHE 方法中利用阻尼因子 $\lambda\in[0,1)$ 避免振荡的产生，并设置阻尼因子 $\lambda=0.9$。

输入：软件度量元数据集 $D=\{x_1,x_2,\cdots,x_n\}$。

输出：聚类中心和指定评价指标下的软件缺陷预测结果。

a）计算任意两个数据点 x_i 和 x_k 之间的共同数据点数目 $\mathrm{den}N(x_i,x_k)$。

b）在 $\mathrm{den}N(x_i,x_k)$ 的基础上，根据式（2）和（3）计算超欧氏距离，构造相似度矩阵 $[s(x_i,x_k)]_{n\times n}$，n 为数据点个数。

c）初始化归属度矩阵和吸引度矩阵为 $a^0(i,k)=r^0(i,k)=0$。

d) 迭代更新。

(a) 吸引度 $[r(i,j)]_{n\times n}$ 矩阵和归属度矩阵 $[a(i,j)]_{n\times n}$ 按式(5) ~(7) 更新如下:

$$r(i,k)\leftarrow s(i,k)-\max_{k'\text{ s.t. }k'\neq k}\{a(i,k')+s(i,k')\} \tag{5}$$

$$\text{if } i\neq k, a(i,k)\leftarrow\min\{0, r(k,k)+\sum_{i'\text{ s.t. }i'\in\{i,k\}}\max\{0, r(i',k)\}\} \tag{6}$$

$$a(k,k)\leftarrow\sum_{i'\text{ s.t. }i'\neq k}\max(0, r(i',k)) \tag{7}$$

(b) 阻尼因子作为加权因子,将新计算得到的 r 和 a 分别与原矩阵的 r 和 a 加权求和。

(c) 对所有数据点求归属度和吸引度之和,根据 $\arg\max_k\{r(i,k)+a(i,k)\}$ 寻找每个数据点的类中心点。

(d) 判断迭代更新过程是否满足终止条件: 超过最大迭代次数; 信息改变量低于某一阈值; 类中心在连续迭代过程中保持稳定。满足条件之一即可返回退出,否则继续执行迭代更新。

e) 判断类中心点数是否满足要求,如果不满足,修正 λ,重复运行程序直至聚类的个数满足要求。

f) 在一定的评价指标下,计算并返回软件缺陷预测结果。

需要说明的是,运行完超欧氏距离近邻传播缺陷预测方法后,对于程序模块类别的标记主要依靠手工操作,结合软件缺陷数据不平衡的特性,对于样本数目多的聚类簇手工标记为无缺陷数据集,反之为缺陷数据集。

2 仿真实验与结果分析

2.1 数据描述

为了进一步验证基于超欧氏距离近邻传播聚类的软件缺陷预测方法的有效性,采用航空航天局 NASA MDP[19] 数据库中的三组数据集进行仿真实验,三组数据的描述如表 1 所示。同时表 2 给出了本仿真实验中使用的表示软件特征的度量元,主要包括 line of code 度量元(LineCount),McCabe 度量元(McCabe),Halstead 基本度量元(BHalstead) 及其扩展度量元(DHalstead)。

表 1 缺陷数据集

data	model	percent of defect	description
*CM*1	505	16.04%	spacecraft instrument
*PC*1	1 109	6.59%	flight software from an earth orbiting satellite
*PC*3	1 563	10.43%	flight software from an earth orbiting satellite

表 2 软件度量元

metrics	type	metrics	type
$V(g)$	McCabe	UniqOp	BHalstead
$EV(g)$	McCabe	UniqOpnd	BHalstead
$IV(g)$	McCabe	TotalOp	BHalstead
LOC	McCabe	TotalOpnd	BHalstead
N	DHalstead	UniqOp	BHalstead
V	DHalstead	LOCcode	LineCount
L	DHalstead	LOCComment	LineCount
D	DHalstead	LOCBlank	LineCount
I	DHalstead	LOCCodeAndComment	LineCount
E	DHalstead	…	…
B	DHalstead		
T	DHalstead		

2.2 模型评价

软件预测模型的评价一直是一个备受争议的课题[6,7]。模型评价方法大部分是基于二维混淆矩阵来表示[20],如表 3 和 4 所示。

表 3 二维混淆矩阵

		预测值	
		缺陷	无缺陷
真实值	缺陷	正确的肯定(*TP*)	错误的否定(*FN*)
	无缺陷	错误的肯定(*FP*)	正确的否定(*TN*)

表 4 不同的评估度量方法

不同的评价指标
$\text{accuracy}=\frac{TP+TN}{TP+TN+FP+FN}\times 100\%$
$\text{recall}=\frac{TP}{TP+FN}\times 100\%$, $\text{precision}=\frac{TP}{TP+FP}100\%$
$F\text{-measure}=\frac{2\times\text{recall}\times\text{precision}}{\text{recall}+\text{precision}}\times 100\%$

在传统的缺陷预测中,准确率(accuracy) 是最常用的评价指标,但其不适用于评估含有不平衡数据的缺陷预测模型,它导致分类结果严重的偏向多数类[21]。反馈率(recall) 和精确率(precision) 分别表示检测为缺陷模块的数目占实际缺陷数目的比率和预测为缺陷数目的比率,二者配合使用可以用于不平衡数据的模型评价,而 F 测量(F-measure) 正好实现了对 recall 和 precision 的折中。一个较高的 F-measure 值可以确保 recall 和 precision 均取得合理的值,软件缺陷预测领域中常用评价指标 F-measure 进行算法性能的衡量。

2.3 仿真结果与分析

实验环境为: Pentium 3.2 GHz CPU,1 GB DDR 内存,Windows XP 操作系统的 PC 机; MATLAB R2007a 和 Weka 3.6.2 以及 Java 集成开发工具 Eclipse 3.2。为了更准确地衡量算法的性能,给出了方法 10 次独立运行后预测结果的平均值。在 APHE 方法中需要提前设置半径 γ 参数,本文采用文献[22] 的方法,半径在区间 [1 ~50] 中进行选取。

下面选择 K-means 算法以及标准 AP 算法和本文 APHE 方法进行对比实验。选择 K-mean 聚类算法主要是考虑它是当前软件缺陷领域中处理无标志软件缺陷度量数据常用的方法[11,12],与其比较可以进一步验证 APHE 算法在预测软件缺陷时的有效性; 选择标准 AP 算法,主要是验证本文提出的基于密集度量元在处理复杂结构软件缺陷数据时的预测性能。另外,K-means 算法是随机初始化聚类中心。具体实验结果如图 2 和表 5 所示。

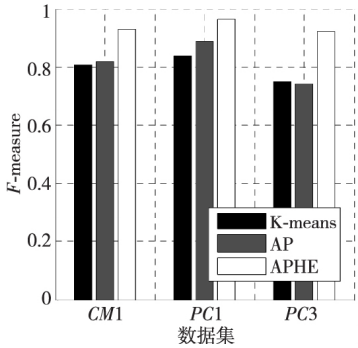

图 2 F-measure 指标下的对比结果

从图 2 可以看出,在 *CM*1,*PC*1 和 *PC*3 三组数据上,APHE

方法均具有最高的 F-measure，尤其对于数据 $PC3$，APHE 的 F-measure值远远高于 K-means 算法和标准近邻传播 AP 算法。同时在表 5 中，给出了三种算法在这三组数据上的总运行时间和总迭代次数。

表 5　三种算法的对比结果

类别	K-means		AP		APHE	
	迭代次数	总运行时间/s	迭代次数	总运行时间/s	迭代次数	总运行时间/s
*CM*1	500	45.832	71	5.726	49	3.235
*PC*1	500	89.141	133	11.958	96	8.268
*PC*3	500	113.563	150	15.622	112	11.413

从表 5 可以看出，K-means 聚类算法获得如图 2 所示的聚类预测值，迭代运行达到最大次数(500 次) 。K-means 的时间复杂度虽是线性的，然而 K-means 算法严重地依赖于初始条件且易于陷入局部最优，为获得最优解需要反复运行。相比之下，APHE 方法与 AP 算法更为快速有效，AP 算法和 APHE 方法的迭代次数大大降低，尤其是基于密集度量元近邻传播缺陷预测算法 APHE，迭代次数和迭代时间均得到了有效的降低。这说明基于密集度量元构造相似度矩阵很好地反映了软件缺陷数据的聚类分布，处理软件缺陷数据时，在更短的迭代次数内找到了聚类中心，实现了有效的划分，从而验证了利用超欧氏距离构建相似度矩阵的 APHE 方法在进行软件缺陷数据预测时是有效的。

2.4　APHE 方法性能分析

结合 APHE 方法的思想以及上述仿真实验数据的测试结果，简要分析 APHE 方法的性能。

a) APHE 方法在密集度量元上构建数据的相似性矩阵，很好地克服了欧氏距离的局部一致性，在复杂结构的数据集上得到很好的预测效果。

b) 时间复杂度分析。假设数据集的规模为 N，迭代次数为 k。在 APHE 方法中，时间复杂度主要来自两个方面: 基于密集度量元相似度的构建; 归属度矩阵和吸引度矩阵的迭代更新时间。前者的复杂度为 $O(N^2)$，后者的时间复杂度为 $O(kN^2)$，因此，总的时间复杂度大约为 $O(kN^2)$。类似地，标准近邻传播算法的时间复杂度也是 $O(TN^2)$，其中，T 为标准 AP 算法的迭代次数。然而由上述仿真实验可以看出，在处理同一个问题时，APHE 方法大大降低了迭代次数，因此，APHE 方法在一定程度上降低了总的运行时间。

3　结束语

本文提出了一种基于超欧氏距离近邻传播的软件缺陷预测方法: a) 将近邻传播算法引入到软件缺陷预测领域中; b) 针对其处理复杂数据的不足，定义了密度因子、超欧氏距离，并给出了一种基于密集度量元的相似度构造方法。实验结果表明，该方法具有更强的预测性能。下一步的工作将在更多的软件度量元数据上进行本方法性能的验证。

参考文献:

[1] 王青，伍书剑，李明树. 软件缺陷预测技术 [J] . 软件学报，2008，19(7) : 1565-1580.

[2] 陈翔，顾庆，刘望舒，等. 静态软件缺陷预测方法研究 [J] . 软件学报，2016，27(1) : 1-25.

[3] Ohlsson N，Zhao Ming，Helander M. Application of multivariate analysis for software fault prediction [J] . Software Quality Journal，1998，7(1) : 51-66.

[4] Teerawit C，Vateekul P. Software defect prediction in imbalanced data sets using unbiased support vector machine [J] . Information Science and Applications，2015，339(1) : 923-931.

[5] Siers M J，Islam M Z. Software defect prediction using a cost sensitive decision forest and voting，and a potential solution to the class imbalance problem [J] . Information Systems，2015，51(C) : 62-71.

[6] Menzies T，Greenwald J，Frank A. Data mining static code attributes to learn defect predictors [J] . IEEE Trans on Software Engineering，2007，33(1) : 2-13.

[7] Zhang Hongyu，Zhang Xiuzhen. Comments on "data mining static code attributes to learn defect predictors" [J] . IEEE Trans on Software Engineering，2007，33(9) : 635-637.

[8] Jiang Yue. Incremental development and cost-based evaluation of software fault prediction models [D] . Morgantown: West Virginia University，2009.

[9] Zhong Shi，Khoshgoftaar T，Seliya N. Unsupervised learning for expert-based software quality estimation [C] //Proc of the 8th IEEE International Conference on High Assurance Systems Engineering. New York: ACM Press，2004: 149-155.

[10] Seliya N，Khoshgoftaar T. Software quality estimation with limited fault data: a semi-supervised learning perspective [J] . Software Quality Journal，2007，15(3) : 327-344.

[11] Catal C，Sevim U，Diri B. Clustering and metrics thresholds based software fault prediction of unlabeled program modules [C] //Proc of the 6th International Conference on Information Technology: New Generations，Software Engineering Track. New York: ACM Press，2009: 199-204.

[12] Zhang Feng，Zheng Quan，Zou Ying，*et al*. Cross-project defect prediction using a connectivity-based unsupervised classifier [C] //Proc of International Conference on Software Engineering. New York: ACM Press，2016: 309-320.

[13] Nam J，Kim S. CLAMI: defect prediction on unlabeled datasets [C] // Proc of the 30th IEEE/ACM International Conference on Automated Software Engineering. Washington DC: IEEE Computer Society，2015: 452-463.

[14] Frey B J，Dueck D. Clustering by passing messages between data points [J] . Science，2007，315(5814) : 972-976.

[15] 董俊，王锁萍，熊范纶. 可变相似性度量的近邻传播聚类 [J] . 电子与信息学报，2010，32(3) : 509-514.

[16] 肖宇，于剑. 基于近邻传播算法的半监督聚类 [J] . 软件学报，2008，19(11) : 2803-2813.

[17] 王玲，薄列峰，焦李成. 密度敏感的半监督谱聚类 [J] . 软件学报，2007，18(10) : 2412-2422.

[18] 杨风召，朱扬勇. 一种有效的量化交易数据相似性搜索方法 [J] . 计算机研究与发展，2004，41(2) : 361-368.

[19] NASA. NASA metrics data program [DB/OL] . (2014-12-02) . http://mdp.ivv.nasa.gov/.

[20] Witten I H，Frank E. Data mining: practical machine learning tools and techniques [M] . San Francisco: Morgan Kaufmann Publisher Inc，2005: 201-227.

[21] Tan Pangning，Steinbach M，Kumar V. Introduction to data mining [M] . Boston: Addison Wesley，2006: 74-101.

[22] 李静伟. 基于共享近邻的自适应谱聚类算法 [D] . 大连: 大连理工大学，2010.