Artificial Intelligence and Robotics Research 人工智能与机器人研究, 2016, 5(2), 23-34
Published Online May 2016 in Hans. http://www.hanspub.org/journal/airr
http://dx.doi.org/10.12677/airr.2016.52003

Hans汉斯

Image Cluster Method Based on Ensemble Locality Sensitive Clustering

Tianqiang Peng[1], Haolin Gao[2]

[1]Department of Computer Science and Engineering, Henan Institute of Engineering, Zhengzhou Henan
[2]Information System Engineering Institute, Information Engineering University, Zhengzhou Henan
Email: ptq_drumboy@163.com

Received: Apr. 25th, 2016; accepted: May 15th, 2016; published: May 18th, 2016

Abstract

To overcome the weakness of k-means in image clustering especially visual image clustering, we proposed an Ensemble Locality Sensitive Clustering method. It first determined the number of clusters of dataset, then generated the multiple clustering resolutions based on Exact Euclidean Locality Sensitive Hashing algorithm, at last, cluster ensemble methods were applied to get final partition. The experiments on synthetic dataset and image dataset show that new method reaches the same level with k-means combined with cluster ensemble about clustering accuracy on synthetic data set, and slightly less accuracy on image dataset. But the advantage of new method is its clustering time is faster than k-means, and it is suitable for incremental clustering. Therefore, Ensemble Locality Sensitive Clustering is a promising clustering method for high dimension image data.

Keywords

Exact Euclidean Locality Sensitive Hashing, Random Projection, Image Clustering , Cluster Ensemble

集成式位置敏感聚类方法

彭天强[1]，高毫林[2]

[1]河南工程学院，计算机工程与科学系，河南 郑州
[2]信息工程大学，信息系统工程学院，河南 郑州
Email: ptq_drumboy@163.com

文章引用: 彭天强, 高毫林. 集成式位置敏感聚类方法[J]. 人工智能与机器人研究, 2016, 5(2): 23-34.
http://dx.doi.org/10.12677/airr.2016.52003

收稿日期：2016年4月25日；录用日期：2016年5月15日；发布日期：2016年5月18日

摘　要

针对常用图像聚类尤其是图像视觉聚类方法聚类速度慢、不支持增量聚类的局限，提出了集成式位置敏感聚类方法。该方法首先根据聚类有效性指标估计合适的聚类数目，然后生成多重哈希函数，并用它们对各数据点进行映射得出多重桶标记，再对数据集各桶标记进行聚类得出多个基划分，最后对多个基划分进行集成得出最终划分。实验结果表明，在准确率方面，集成式位置敏感聚类在人工数据上与k-means结合聚类集成的方法相当，在图像集上与k-means结合聚类集成的方法接近。但集成式位置敏感聚类的优点在于其聚类时间快、适合于增量聚类等。因此，集成式位置敏感聚类方法可以用于解决高维图像特征聚类问题。

关键词

精确欧式空间位置敏感哈希，随机映射，图像聚类，聚类集成

1. 引言

数据聚类是把给定的数据集划分为多个类的无监督的过程，使得同一类内的数据点比不同类间的数据点更相似。在对高维空间中的数据集聚类时，许多算法的有效性和效率会明显降低。主要有以下几个原因，一是高维数据的内部稀疏性阻碍了这些算法性能的发挥；二是高维空间中两点距离趋于一致[1]，因此从不相似数据中找出相似数据的难度更大；三是高维空间中的类可能存在于子空间中，不同的类可能存在于不同的子空间中[2]。图像聚类是典型的高维空间聚类，因为几乎所有图像特征的维数都很高。因此，高维聚类的难题也存在于图像聚类中。虽然近年来出现了不少聚类算法，但他们中的许多算法在对图像聚类尤其是图像视觉聚类中的效果并不好。在视觉聚类中经常使用的仍然是 k-means 算法，这或许是因为设计一个能替代 k-means 的通用聚类算法确实比较困难。但是，在增量数据集上，k-means 的缺点会比较明显，因为它的聚类速度很慢，而且不支持增量聚类，运行时间会随着数据集规模的增大而急剧增长。在用于视觉词典生成的聚类中存在着多义性和同义性等问题，这就需要找出新算法。

随机映射是近年来研究较多的算法，它可以降低维数同时在一定程度上保持原有结构不变。例如，欧式空间的 n 维的点可以映射到 d 维并且 d 远小于 n。这样的随机映射主要应用在降维[3]和快速近似最近邻搜索[4]，它的应用范围还包括信号处理[5]、异常检测[6]、聚类[7]和分类等[8] [9]。

E^2LSH (Exact Euclidean Locality Sensitive Hashing) [10]是随机映射的一个特例，近年来也引起了人们更多的注意，被应用在不少领域如检索[11]、复制检测[12]等。它在用于近似最近邻检索时，在保持较高准确率的同时有着很低的运行时间，是当前近似最近邻检索的主流算法。它把高维向量映射到低维空间后，也完成了维数约减的任务。同时，相同的桶中的点比不同桶中的点更相似。因此，如果根据桶标志对数据点进行分组，也可以达到对数据点进行聚类的目的。而且，E^2LSH 是一个数据无关的方法，它在对一个点进行映射时与其它点无关，这意味着它可以方便地为一个动态数据库建立索引。类似地，用于聚类时，当数据集规模变化时不需要对整个数据集重新进行聚类，只需要对变化的数据点运算即可，也就是说它可以方便地起到动态聚类的作用，再加上它的时间优势使得它可以较好的用于增量大数据集聚类。事实上，LSH 算法的提出者已经指出 LSH 可以用于快速聚类，但他没有进一步进行理论证明和实验验证。不过，Ravichandran D 已经将 LSH 的欧式空间实现方案 E^2LSH 用于名词聚类[13]。然而，图像

数据比文本数据要复杂得多，因此，对图像进行聚类也比对名词进行聚类要困难的多。

为解决 k-means 算法在图像聚类中的问题，提出了集成式位置敏感聚类方法(ELSC, Ensemble based Locality Sensitive Clustering)，该方法基于 E^2LSH 算法和聚类集成算法。它同时利用了 E^2LSH 和聚类集成的优点，这使得它可能成为有效的聚类方法。其中基于 E^2LSH 的聚类称为位置敏感聚类，它的可行性来自于随即映射的距离保持特性和可分性保持特性。

2. 位置敏感聚类可行性

随机映射在模式识别和数据挖掘中的主要作用是降维。除了降维之外，随机映射还可用于聚类。聚类的基础在于它对数据集可分性的保持。这里的可分性保持主要涉及距离保持和边界保持。对于数据聚类，Johnson-Lindenstrauss 定理说明经过映射数据点间的距离以很高的概率得以保持。这是信息检索中近似最近邻搜索算法的基础。

2.1. 随即映射的距离保持和边界保持特性

Johnson-Lindenstrauss 定理说明了映射的距离保持性质。给定 $0<\varepsilon<1$ 和数据集 $S\subset\mathbb{R}^n$，对于正整数 $d\geq 4\left(\frac{\varepsilon^2}{2}-\frac{\varepsilon^3}{3}\right)^{-1}\log|S|=O\left(\frac{1}{\gamma^2}\log|S|\right)$，存在一个映射 $f:\mathbb{R}^n\to\mathbb{R}^d$，使得对于所有 $u,v\in S$，

$$(1-\varepsilon)\|u-v\|^2\leq\|f(u)-f(v)\|^2\leq(1+\varepsilon)\|u-v\|^2 \tag{1}$$

该定理说明，映射后成对数据点的距离有很大的可能性被以 $1\pm\gamma$ 为倍数保留下来。对于高维空间 R^n 中的数据集 S，假设随机地把数据映射到 R^d，映射后的维数 $d=O\left(\gamma^{-2}\log|S|\right)$ 足够以较高的概率使数据点间的角度最多改变 $\pm\gamma/2$ [14]。特殊地，如果把 S 中所有数据点映射到向量 ω，如果原始数据以边界 γ 可分，那么映射后，角度改变最大为 $\gamma/2$，数据点仍然可分。

分类中边界保持也可为聚类提供有效信息。文献[15]建立了随机映射后边界保持的条件，并说明如果该条件满足，无差错边界在二分类和多分类问题中能被保持。文献[8]研究了二分类随机映射的边界保持问题，并给出了随机映射以给定概率保持原始数据一半的边界所需要维数的下界，但它需要无限多维数才能保持无差错边界。如果原始数据有归一化边界 γ，那么只有映射数目

$$n\geq\frac{c}{\gamma^2}\ln\frac{1}{\rho\delta} \tag{2}$$

时，对任何合适的常量 c，映射后的数据以大于 $1-\delta$ 的概率以错误 ρ 有边界 $\gamma/2$。

二进制边界保持的内容为：给定任意的随机高斯矩阵 $\boldsymbol{R}\in\mathbb{R}^{n,d}$，如果数据集 $S=\left\{\left(x_i\in\mathbb{R}^n,y_i\in\{-1,+1\}\right)\right\}_{i=1}^m$ 是被边界 $\gamma\in(0,1]$ 线性可分的，那么对任何 $\delta,\varepsilon\in(0,1)$ 和任何

$$d>\frac{12}{3\varepsilon^2-2\varepsilon^3}\ln\frac{6m}{\delta} \tag{3}$$

情况，数据集 $S'=\left\{\left(\boldsymbol{R}x_i\in\mathbb{R}^d,y_i\in\{-1,+1\}\right)\right\}_{i=1}^m$ 以至少 $1-\delta$ 的概率被边界 $\gamma-\frac{2\varepsilon}{1-\varepsilon}$ 线性可分。

随机映射后边界的下界对某些 ε 可能变为负值。负边界意味着映射后的数据不是线性可分的。当边界为正时，上式说明二进制分类中的边界经过随机映射后以很高的概率得以保持。

多类无差错边界保持的内容为：对于归一化边界，如果存在 $\left\{u_y\in\mathbb{R}^n\right\}_{y\in Y}$ 对所有 $(x,y)\in S$，使得

$$\frac{\langle u_y,x\rangle}{\|u_y\|\|x\|}-\max_{y'\neq y}\frac{\langle u_{y'},x\rangle}{\|u_{y'}\|\|x\|}\geq\gamma \tag{4}$$

那么，多类数据集 $S=\left\{\left(x_i\in\mathbb{R}^d, y_i\in Y=\{1,\cdots,L\}\right)\right\}_{i=1}^{m}$ 是被边界 $\gamma\in(0,1]$ 线性可分的。

多类边界保持的问题如下：对任何多类数据集 S 和任何高斯随机矩阵 $\boldsymbol{R}$，如果 S 是被边界 $\gamma\in(0,1]$ 线性可分的，那么对任何 $\delta,\varepsilon\in(0,1)$ 和任何 $d>\frac{12}{3\varepsilon^2-2\varepsilon^3}\ln\frac{6Lm}{\delta}$ 情况，数据集 $S'=\left\{\boldsymbol{R}x_i\in\mathbb{R}^d, y_i\in Y\right\}_{i=1}^{m}$ 以至少 $1-\delta$ 的概率是被边界 $-\frac{1+3\varepsilon}{1-\varepsilon^2}+\frac{\sqrt{1-\varepsilon^2}}{1+\varepsilon}+\frac{1+\varepsilon}{1-\varepsilon}\gamma$ 线性可分的。

2.2. 位置敏感聚类

E^{2}LSH 是 LSH (Locality Sensitive Hashing)的欧式空间实现方式，是一种基于随机映射的方法。因此以它为基础设计的算法具有对数据进行聚类的能力。位置敏感聚类的基础就是 E^{2}LSH 算法的位置敏感哈希函数。该函数是基于 p-稳定分布函数的， $p\in(0,2]$。位置敏感哈希函数的定义为：

$$h(v)=\left\lfloor\frac{a\cdot v+b}{w}\right\rfloor \tag{1}$$

其中 a 是由 p-稳定分布函数产生的 n 维随机向量，内积 $(a\cdot v)$ 对数据点进行随机映射，b 对映射后的结果加上一个偏移，取模运算确保映射后的哈希值(桶标记)在一定范围内。可见，哈希函数的生成要用随机的方法，内积运算对数据点进行映射。但 E^{2}LSH 与一般的随机映射不同，它使得数据点映射后不在随机向量所在方向的全坐标轴上，而是映射到了坐标轴的一部分。另外一个不同在于映射运算，一般的随机映射需要构造随机映射矩阵并进行矩阵运算，也就是每个点都要与一个矩阵进行运算。但 E^{2}LSH 中，每个点只需要进行内积计算，这可以降低内存消耗和计算复杂度。

在随机映射可分性保持的基础上，一个数据集在经过位置敏感哈希函数映射后，相对距离和类边界能够得以保持。这为 E^{2}LSH 用于聚类提供了可行性经过映射后，桶标记相同的点比桶标记不同的点相似度高，而且桶标记相邻的点也比桶标记不相邻的点的相似度高。因此，可以利用桶标记对数据点进行分组，也就是把桶标记相同或相邻的数据点分为一类。这种以 E^{2}LSH 的位置敏感哈希函数为基础的聚类方法就是位置敏感聚类(LSC, Locality Sensitive Clustering)方法。

3. 集成式位置敏感聚类

集成式位置敏感聚类方法(Ensemble LSC)的基础是位置敏感聚类和聚类集成。聚类集成是集成学习的一部分，是近年机器学习领域的研究热点之一。聚类集成对改善聚类性能有着重要的作用，这是因为没有一个单一的聚类算法能够发现不同形状和不同分布的类，而不同数据集的内部结构是不同的。聚类集成同一数据集的多个基聚类方案结合起来，得出一个统一的聚类方案，该方案比大多数基聚类方案的聚类质量要好。聚类集成与集成学习的另一个研究方向——分类融合类似，分类融和可以将分类器用 bagging、boosting 融合起来，也可以将分类器的结果进行融合。然而，聚类集成更具挑战性。首先，不同基聚类器得出的聚类数目可能不同，其次，聚类标签只是一个标记符号，继承以前需要解决不同聚类方案的标签对应问题。第三，集成器无法获取原始数据，只能访问基聚类的类标签。使用聚类集成的原因和优势主要在于[16]：改善聚类质量，提供稳健聚类方法，提供模型选择问题新方法，知识复用，多角度聚类，分布计算。

3.1. 聚类集成框架

聚类集成方法主要由两步组成：生成多个基聚类划分，即用基聚类算法对数据集创建多个划分；生成最终划分，即对基聚类得出的划分进行合成得出一个新的划分。Topchy 等证明[17]，只要采用合适的

一致函数(Consensus Function)，弱聚类算法能够产生高质量的集成聚类。用于产生基聚类划分的一致函数有多种，这里采用如下的三种方法。

基于聚类的相似划分算法(Cluster-based Similarity Partitioning Algorithm, CSPA) [16]首先创建一个 $n\times n$ 相似矩阵，n 是数据集 X 的对象的个数，这可以看作全连接图的邻接矩阵，图的节点表示数据集的对象，两点间的边是一个相关权值，其值等于同一类中这两个对象出现在同一类中的次数。超图划分算法(Hyper-Graph Partitioning Algorithm, HGPA) [18]创建一个超图，节点代表对象，相同权值的超边代表类。然后用 HMETIS 将该超图划分为大致相等的 k 部分。元聚类算法(Meta-CLustering Algorithm, MCLA)生成的图表示类间的关系，节点对应于各个类，每个边的权值用二进制 Jaccard 相似度计算。然后使用 METIS 把该图划分为 k 个元类。最后通过将每个对象分配给关联最多的元类得出最终划分。

Strehl 和 Ghosh 把一致划分(即最终划分)定义为与各个基聚类器得出的划分共享信息最多的划分[19]，为测量两个划分共享的信息量，他们定义了基于信息论中互信息的归一化互信息(Normalized Mutual Information, NMI)，用来度量两个划分相似度。用 $n_h^{(a)}$ 表示划分 $\lambda^{(a)}$ 类 C_h 中数据点的个数，$n_l^{(b)}$ 表示划分 $\lambda^{(b)}$ 中类 C_l 中数据点的个数，$n_{h,l}$ 表示同时在这两个类中点的数目。那么，NMI 的定义为：

$$\phi^{(NMI)}\left(\lambda^{(a)},\lambda^{(b)}\right)=\frac{\sum_{h=1}^{k^{(a)}}\sum_{l=1}^{k^{(b)}}n_{h,l}\log\left(\frac{n\cdot n_{h,l}}{n_h^{(a)}n_l^{(b)}}\right)}{\sqrt{\left(\sum_{h=1}^{k^{(a)}}n_h^{(a)}\log\frac{n_h^{(a)}}{n}\right)\left(\sum_{l=1}^{k^{(b)}}n_l^{(b)}\log\frac{n_l^{(b)}}{n}\right)}} \tag{6}$$

其中 $k^{(a)}$ 和 $k^{(b)}$ 分别表示两个划分的类的个数。NMI 的最大值为 1 最小值为 0。

在此基础上，一致划分就是和所有基划分 $\lambda^{(q)}$ 有最大平均互信息的划分：

$$\lambda^{*}=\arg\max_{\hat{\lambda}}\sum_{q=1}^{m}\phi^{(NMI)}\left(\hat{\lambda},\lambda^{(q)}\right) \tag{7}$$

其中 $\hat{\lambda}$ 是所有可能的 m 个划分中的一个。这里也采用这种方法进行最终划分。

3.2. 集成式位置敏感聚类

由于位置敏感聚类存在一定的随机性，提出集成式位置敏感聚类的方法来改善聚类质量。集成式位置敏感聚类是位置敏感聚类和聚类集成的合成。主要包括三部分，确定聚类数目，生成多重位置敏感聚类方案，多个聚类方案集成。这里我们采用事先指定聚类数目的方式。有了聚类数目以后，就可以生成多个基聚类并进行集成。ELSC 的完整过程如下：

1) 为数据集 S 确定聚类数目 k；

2) 生成服从高斯分布的 $n\times d$ 的随机矩阵 $A=\left(A_1,A_2,\cdots,A_n\right)^{\mathrm{T}}$ 和 b、w，A_i 是一个 d 维向量，$n=|S|$；

3) 对所有数据点 $v_i\in S$ 进行随机映射，每个点 v_i 的映射的结果为 d 维桶标记 $B_i=\left(B_{i1},B_{i2},\cdots,B_{id}\right)$，

$$B_i=\left\lfloor\frac{v_i\cdot A+b}{w}\right\rfloor \tag{8}$$

这样，全部数据点的桶标记就是一个 $n\times d$ 维矩阵 B。B 的每一列都是对全部数据集的一次映射。

4) 用 PAM 算法对 B 的每一列聚成 k 类，得出的结果即对数据集的 d 个划分，用 $\{\lambda_1,\lambda_2,\cdots,\lambda_d\}$ 表示。

5) 使用聚类集成方法对 d 个基划分进行集成并得出最终划分。采用的聚类集成方法包括 CSPA、HGPA 和 MCLA，它们各自得出一个集成划分，记作 $\{\lambda_1',\lambda_2',\lambda_3'\}$，最终划分是和其它两个划分有最大归一化互信息的划分，即

$$\lambda^* = \arg\max_{\hat{\lambda}} \sum_{q=1}^{2} \phi^{(NMI)}\left(\hat{\lambda}, \lambda'_q\right) \tag{9}$$

4. 实验

为直观地看到聚类结果，首先在人工数据集上进行试验。人工数据集有两个，一个包括 30 个 2 维的数据点(记作“数据集 1”)，依次属于 3 个类，每个类 10 个点，前 10 个点属于第一类，最后 10 个点属于第 3 类。另外一个数据集包括 30 个 100 维的数据点(记作“数据集 2”)，同样依次属于 3 个类。此外，为验证在实际数据上的聚类结果，构造了一个图像集(记作“图像集 1”)，该图像集从 TRECVID 数据集选取，包括 4 类 75 幅图像，每类选取部分图像，这些图像也依次分别属于 4 个类，这 4 个类是“compere”、“singer”、“rice”和“sports”。实验测试环境为 8G DDR3 1600 RAM 和 AMD 速龙 II X2 255 双核主频 3.11 GHz CPU，操作系统为 Windows 7 64 位旗舰版，开发环境为 Matlab 2013a 和 Visual Studio 2010 及相关开发包。

4.1. 人工数据集实验

首先在人工数据集 1 上进行比较，由于 ELSC 是基于聚类集成的算法，将它与基于聚类集成的 k-means 算法(记作“Ek-means”)进行比较。K-means 算法的基本思想是：以空间中 k 个点为中心进行聚类，对最靠近他们的对象归类。通过迭代的方法，逐次更新各聚类中心的值，直至得到最好的聚类结果。k-means 聚类如图 1 所示。

由图 1 可见，所有 30 个点被正确地分为 3 个类。将该实验重复多次，其中 5 次聚类结果如表 1 所示，表中仅给出了第 1 类数据的类标签。可见，虽然每次得出的类标签不同，但这些标签都是正确的，因为它们都把这些点聚到了一类。在这些类标签上运行聚类集成算法后，Ek-means 得出完全正确的结果，最终类标签为[1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3]。

对数据集 1 运行 ELSC 算法，其桶标记计算结果见图 2 所示。桶标记是由位置敏感哈希函数对数据点映射后得出的。将该实验重复多次，其中 5 次的桶标记如表 2 所示。

从表 2 可见，每个聚类结果中同一个类的每个点的桶标记不同，但每个类类内桶标记都在较小的范围内波动。由图 2 可知，类内两点桶标记的距离小于类间两点桶标记的距离。不同聚类结果中同一个点的桶标记也可能不同，这与哈希函数中的随机性有关。

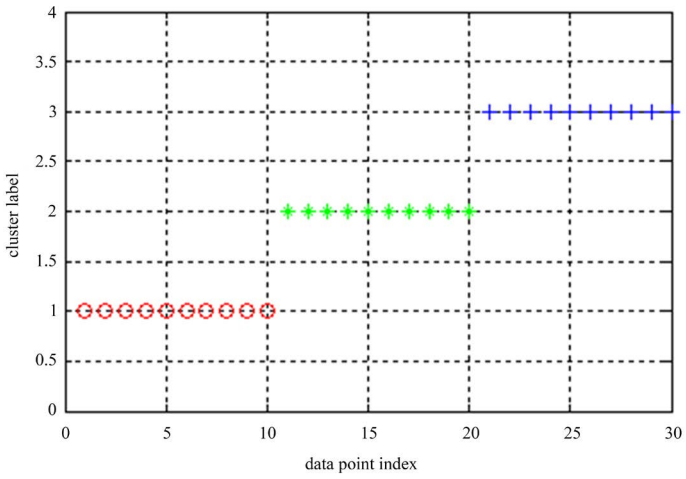

Figure 1. k-means clustering result in data set 1
图 1. 数据集 1 上 k-means 聚类结果

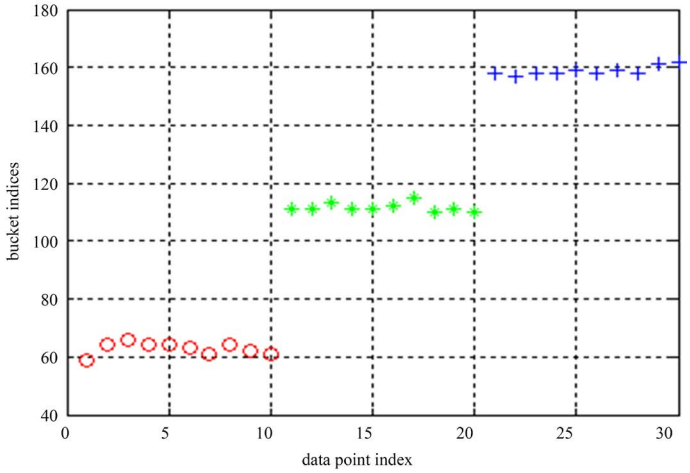

Figure 2. ELSC bucket number result in data set 1
图 2. 数据集 1 的 ELSC 桶标记计算结果

Table 1. The label of k-means clustering from the first type in data set 1
表 1. 数据集 1 第 1 类数据 k-means 聚类得出的类标签

Data point index	1	2	3	4	5	6	7	8	9	10
Label 1	1	1	1	1	1	1	1	1	1	1
Label 2	2	2	2	2	2	2	2	2	2	2
Label 3	1	1	1	1	1	1	1	1	1	1
Label 4	3	3	3	3	3	3	3	3	3	3
Label 5	3	3	3	3	3	3	3	3	3	3

Table 2. The bucket indices of ELSC algorithm from the first type in data set 1
表 2. 数据集 1 第 1 类数据的 ELSC 桶标记计算结果

Data point index	1	2	3	4	5	6	7	8	9	10
Bucket indices 1	59	64	66	64	64	63	61	64	62	61
Bucket indices 2	63	68	68	65	66	66	65	65	63	64
Bucket indices 3	68	74	71	71	71	72	67	71	69	71
Bucket indices 4	69	75	73	72	71	72	71	73	69	70
Bucket indices 5	57	63	60	59	59	62	61	61	57	59

对这些桶标记进行运算，可得出数据集 1 的多个基划分。采用 PAM 算法对基划分进行聚类，并运行聚类集成算法，得出的最终划分的类标签是[1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3]，这与 Ek-means 的结果相同。

为验证新的聚类算法在高维数据上的运行效果，在数据集 2 上也进行了实验。先运行了 k-means 算法，其聚类结果如图 3 所示，图中所有 30 个数据点被正确地分为 3 类。重复该实验多次，对这些聚类结果进行集成后，Ek-means 可得出完全正确的结果。最终的类标签是[1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3]。

在数据集 2 上运行 ELSC 算法，得出的桶标记如图 4 所示，桶标记正确地反映了对应点所属的类别信息。对这些桶标记进行 PAM 聚类以产生多个基划分。并运行聚类集成算法，最终划分的类标签为

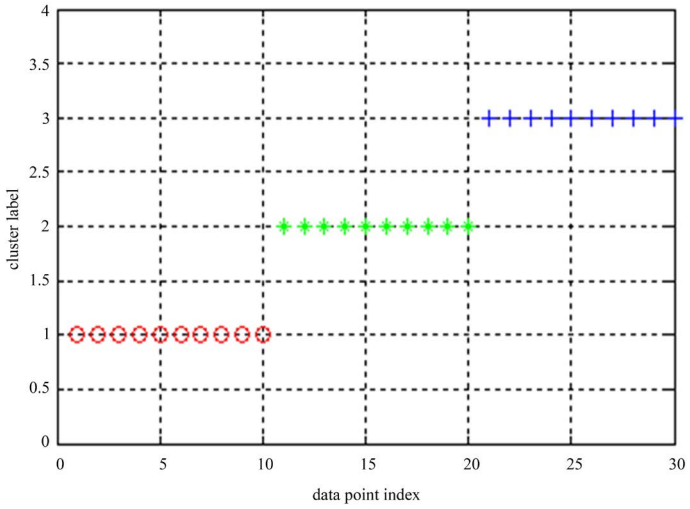

Figure 3. k-means clustering result in data set 2
图 3. 数据集 2 的 k-means 聚类结果

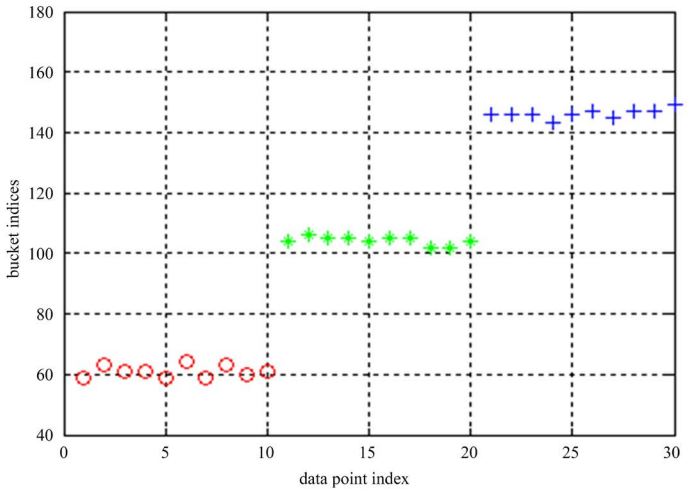

Figure 4. ELSC clustering result in data set 2
图 4. 数据集 2 的 ELSC 算法的桶标记

[1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3]。该结果也与 Ek-means 相同。

这些实验说明，在人工数据集上，ELSC可达到与Ek-means相同的准确率。

4.2. 图像集实验

在图像集 1 上运行 k-means 结合聚类集成(记作“EKM”)和 ELSC 算法进行比较。先运行 k-means 算法，其结果如图 5 所示。然后重复多次，得出多个基划分(或基聚类)，再在基划分的基础上运行聚类集成算法，最终划分的类标记为[1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4]。可见，虽然基划分存在一些错误的类标签，但 EKM 算法产生的最终划分是完全正确的。

图像集 1 的 ELSC 算法产生的桶标记如图 6 所示，第 1 类、第 2 类和第 4 类的桶标记中部分点对应的类别信息是错误的，第 3 类图像的结果完全正确。多次划分中第 2 类图像的桶标记如表 3 所示。

对这些桶标记分别运行 PAM 聚类算法，就可以得出图像集 1 的多个基划分，最后运行三种聚类集成

算法，得出三种集成划分，并通过互信息的衡量去最好的集成划分作为最终划分。最终划分的标签为[3 3 3 3 3 3 2 3 3 3 3 3 3 3 3 3 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 3 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3]。图像集 1 的最终划分显示在图 7 中，该次试验的聚类准确率为 93%。

Table 3. The bucket indices of the ELSC algorithm in the second type of images
表 3. 图像集 1 第 2 类图像 ELSC 算法的桶标记

Image index	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
bucket indices 1	42	40	39	36	37	49	39	39	40	40	40	39	41	40	34	42	47	39	40	52
bucket indices 2	15	8	8	10	10	8	9	9	13	8	8	10	9	9	9	9	9	8	8	13
bucket indices 3	24	26	26	26	25	28	27	25	25	26	26	25	26	26	25	26	27	27	26	19
bucket indices 4	15	7	7	9	9	5	7	11	10	9	9	15	9	9	10	8	7	8	8	17

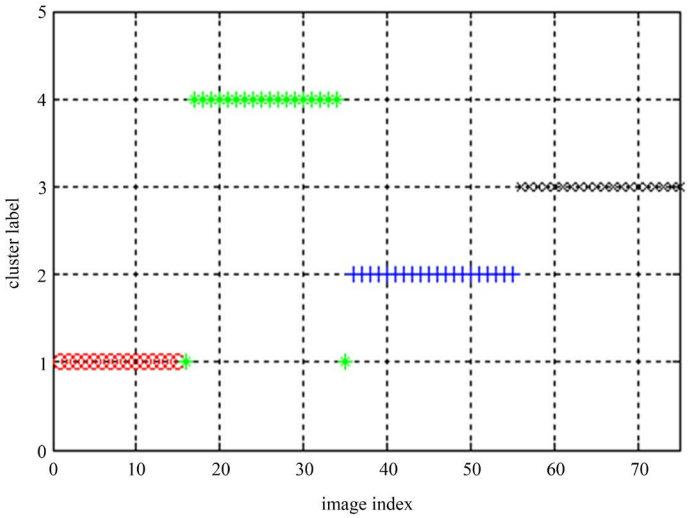

Figure 5. k-means clustering result in image set 1
图 5. 图像集 1 的 k-means 聚类结果

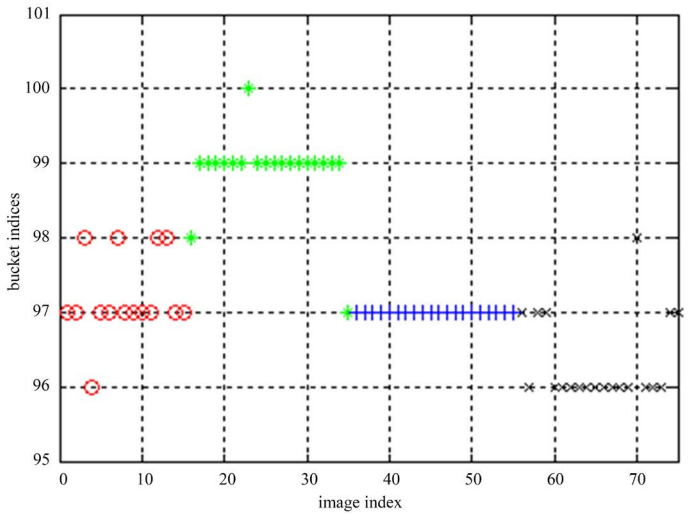

Figure 6. ELSC bucket indic result in Image 1
图 6. 图像集 1 的 ELSC 算法的桶标记

为验证算法的运行时间和准确率，在图像集 1 上多次运行两种算法。两种算法的基划分从 10 增加到 100，增加间隔为 10，在不同数量的基划分上分别进行聚类集成，得出 10 个最终划分。并将该过程重复 10 次，将聚类时间进行平均，最终得出的聚类时间如图 8 所示，ELSC 的运行时间要少于 EKM，而且基聚类数目越大，其聚类时间的优势越大。

在准确率上，采用 F 值和 MAP 值进行比较。MAP 将 4 个类作为整体进行计算，其结果如图 9 所示，可见在一些基划分上两种算法各有优势，但平均而言，它们的平均值分别为 0.9667 和 0.9662，即 ELSC 仍有一定优势。F 值对 4 个类分别计算，其值如表 4 所示。在各类别各划分上，两种算法各有优势，但差距不大。对不同的基划分方案进行平均，前 3 个类 EKM 略高，第 4 类 ELSC 略高。

图像集 1 的实验证明，ELSC 算法在真实数据集上能达与 EKM 相当甚至稍高的准确率。而且它优势在于继承了 E^2LSH 低内存消耗、低计算代价和短运行时间的特点，以及其自身的数据无关特性决定的对增量聚类的适应性。E^2LSH 的全数聚集运行时间低于该数据集规模的线性水平，数据集更新时只需要对更新数据进行计算，这都远远低于 k-means 的计算时间。尤其是数据更新时，k-means 需要重新对全数聚

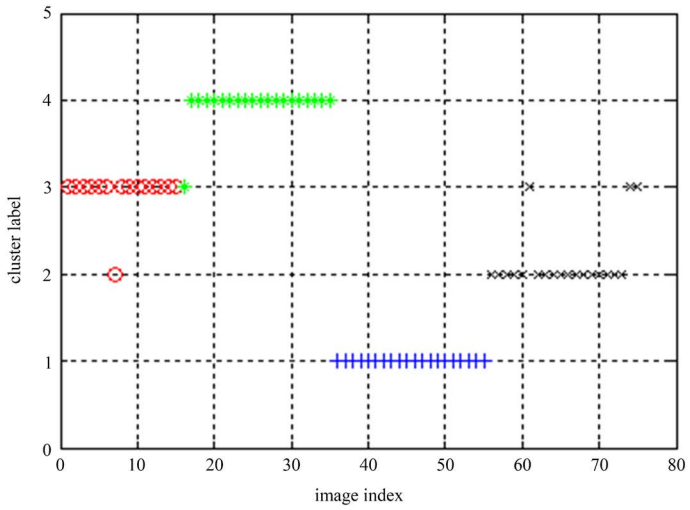

Figure 7. Class symbol of ELSC division in Image 1
图 7. 图像集 1 ELSC 算法最终划分的类标签

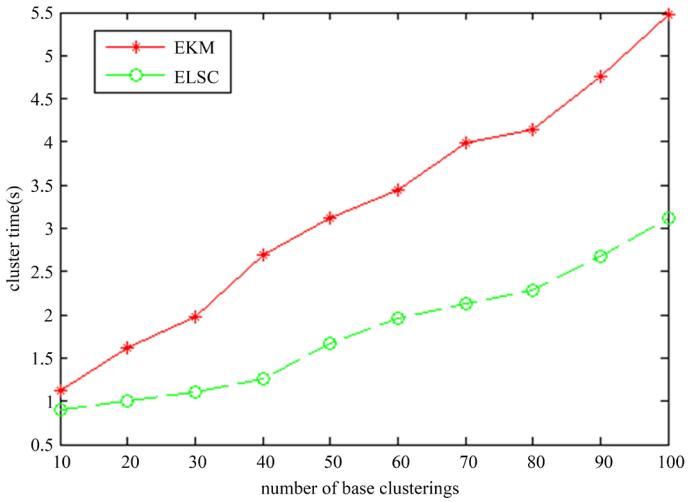

Figure 8. The clustering time of two algorithms in image set 1
图 8. 图像集 1 两种算法的聚类时间

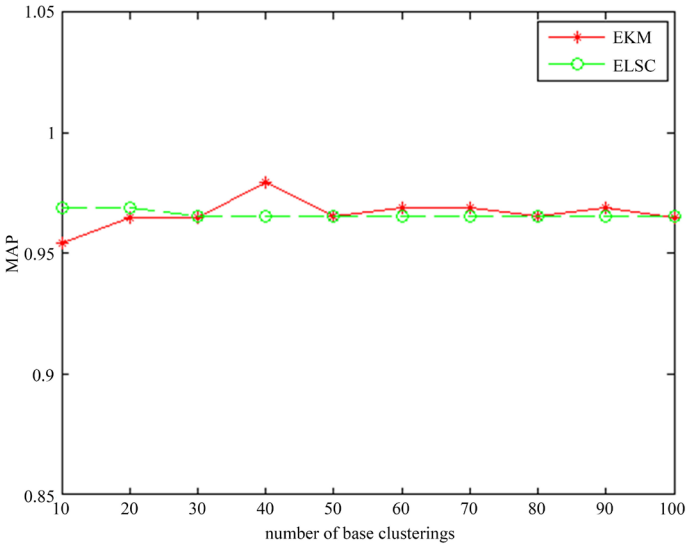

Figure 9. The MAP of ELSC in image set 1
图 9. 图像集 1 ELSC 算法的 MAP 值

Table 4. The F number of the algorithms in each type of image set 1
表 4. 图像集 1 各类别的 F 值

基聚类数目	类别 1		类别 2		类别 3		类别 4	
	EKM	ELSC	EKM	ELSC	EKM	ELSC	EKM	ELSC
10	0.938	0.947	0.976	0.938	0.919	1.000	1.000	1.000
20	0.882	1.000	0.947	1.000	1.000	0.947	0.947	0.938
30	0.882	0.833	0.947	0.947	1.000	1.000	0.947	0.889
40	0.968	0.889	0.974	1.000	1.000	0.833	1.000	0.947
50	0.889	0.833	1.000	0.947	0.833	0.889	0.947	1.000
60	0.938	0.833	1.000	0.947	0.947	0.889	1.000	1.000
70	0.947	0.833	0.938	0.947	1.000	0.889	1.000	1.000
80	1.000	1.000	0.889	0.889	0.947	0.833	0.833	0.947
90	0.947	1.000	1.000	0.889	0.938	0.833	1.000	0.947
100	0.882	0.947	0.947	0.833	1.000	0.889	0.947	1.000
平均	0.927	0.912	0.962	0.934	0.958	0.900	0.962	0.967

集聚类。综合准确率、时间复杂度和可增量运算几点来看，ELSC 是可以用于高维空间数据聚类的。尤其是增量聚类和快速聚类对当前大数据处理是非常重要的。

5. 总结

集成式位置敏感聚类(ELSC)方法是在聚类集成和 E^2LSH 的基础上提出的，它具有快速聚类和增量聚类的优点，随机性也比以 E^2LSH 为基础的位置敏感聚类低。它在人工数据集和图像数据集上都表现出了较好的性能，不但聚类时间大大缩短，聚类准确率也与 EKM 相当。另外，它还具有内存消耗低、计算代价小等对大规模数据聚类相当重要的优点。这些决定了 ELSC 在大规模高维数据集聚类上将能发挥重要的作用，尤其是在图像视觉词典生成过程中的聚类算法中，ELSC 将会是 k-means 的一个替代算法，我

们下一步将深入研究 ELSC 如何应用于视觉词典法，在图像分类、识别和检索等任务中展开应用。

基金项目

国家自科学基金项目(编号 61301232)资助。

参考文献 (References)

[1] Cao, Y. and Wu, J. (2002) Projective ART for Clustering Data Sets in High Dimensional Spaces. *Neural Networks*, **15**, 105-120.

[2] Agrawal, R., Gehrke, J., Gunopulos, D. and Raghavan, P. (1998) Automatic Subspace Clustering of High Dimensional Data for Data Mining Applications. *Proceedings of SIGMOD Record ACM Special Interest Group on Management of Data*, 94-105. http://dx.doi.org/10.1145/276304.276314

[3] Schclara, A., Rokachb, L. and Amit, A. (2013) Ensembles of Classifiers Based on Dimensionality Reduction. *Pattern Analysis and Applications Journal*, **5**, 1305-4345.

[4] Dasgupta, S. and Sinha, K. (2013) Randomized Partition Trees for Exact Nearest Neighbor Search. *Proceedings of Workshop and Conference Proceedings*, **30**, 1-21.

[5] Baraniuk, R.G., Davenport, M., De Vore, R. and Wakin, M.B. (2008) A Simple Proof of the Restricted Isometry Principle for Random Matrices. *Constructive Approximation*, **28**, 253-263.

[6] Fowler, J.E. and Du, Q. (2012) Anomaly Detection and Reconstruction from Random Projections. *IEEE Transaction on Image Processing*, **21**, 184-195.

[7] Schulman, L.J. (2000) Clustering for Edge-Cost Minimization. *Proceedings of Annual ACM Symposium Theory of Computing*, 547-555.

[8] Balcan, M.-F., Blum, A. and Vempala, S. (2006) Kernels as Features: On Kernels, Margins, and Low-Dimensional Mappings. *Machine Learning*, **65**, 79-94.

[9] Shi, Q., Petterson, J., Dror, G., Langford, J., Smola, A.J. and Vishwanathan, S.V.N. (2009) Hash Kernels for Structured Data. *Journal of Machine Learning Research*, **10**, 2615-2637.

[10] Andoni and Indyk, P. (2008) Near-Optimal Hashing Algorithms for Approximate Nearest Neighbor in High Dimensions. *Communications of the ACM*, **51**, 117-122.

[11] Jegou, H., Douze, M. and Schmid, C. (2010) Improving Bag-of-Features for Large Scale Image Search. *International Journal of Computer Vision*, **87**, 316-336. http://dx.doi.org/10.1007/s11263-009-0285-2

[12] Liu, Z., Liu, T. and David, G. (2010) Effective and Scalable Video Copy Detection. *Proceedings of the ACM SIGMM International Conference on Multimedia Information Retrieval*, ACM, New York, 119-128.

[13] Ravichandran, D., Pantel, P. and Hovy, E. (2005) Randomized Algorithms and NLP: Using Locality Sensitive Hash Function for High Speed Noun Clustering. *Proceedings of the* 43*rd Annual Meeting on Association for Computational Linguistics*, Stroudsburg, 622-629. http://dx.doi.org/10.3115/1219840.1219917

[14] Blum, A. (2006) Random Projection, Margins, Kernels, and Feature-Selection. *Proceedings of the* 2005 *International Conference on Subspace, Latent Structure and Feature Selection, LNCS*, 52-68.

[15] Shi, Q.F., Shen, C.H., Hill, R. and van den Hengel, A. (2012) Is Margin Preserved after Random Projection. *Proceedings of International Conference on Machine Learning*, Edinburgh.

[16] Pons, S.V. and Sulcloper, J.R. (2011) A Surver of Clustering Ensemble Algorithms. *International Journal of Pattern Recognition and Artificial Intelligence*, **25**, 337-372. http://dx.doi.org/10.1142/S0218001411008683

[17] Topchy, A.P., Jain, A.K. and Punch, W.F. (2005) Clustering Ensembles: Models of Consensus and Weak Partitions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **27**, 1866-1881. http://dx.doi.org/10.1109/TPAMI.2005.237

[18] Topchy, A., Minaei-Bidgoli, B., Jain, A.K. and Punch, W.F. (2004) Adaptive Clustering Ensembles. *Proceedings of the* 17*th International Conference*, Washington DC, 272 - 275. http://dx.doi.org/10.1109/icpr.2004.1334105

[19] Strehl and Ghosh, J. (2002) Cluster Ensembles: A Knowledge Reuse Framework Multiple Partitions. *Journal of Machine Learning Research*, 583-617.