TRANSACTIONS ON ELECTRICAL AND ELECTRONIC MATERIALS
Vol. 15, No. 3, pp. 125-129, June 25, 2014

Regular Paper

pISSN: 1229-7607 eISSN: 2092-7592
DOI: http://dx.doi.org/10.4313/TEEM.2014.15.3.125

Enhanced Locality Sensitive Clustering in High Dimensional Space

Gang Chen†, Hao-Lin Gao, Bi-Cheng Li, and Guo-En Hu
Department of Data Processing Engineering, Zhengzhou Information Science and Technology Institute, Zhengzhou 450001, China

Received January 13, 2014; Revised February 3, 2014; Accepted March 24, 2014

A dataset can be clustered by merging the bucket indices that come from the random projection of locality sensitive hashing functions. It should be noted that for this to work the merging interval must be calculated first. To improve the feasibility of large scale data clustering in high dimensional space we propose an enhanced Locality Sensitive Hashing Clustering Method. Firstly, multiple hashing functions are generated. Secondly, data points are projected to bucket indices. Thirdly, bucket indices are clustered to get class labels. Experimental results showed that on synthetic datasets this method achieves high accuracy at much improved cluster speeds. These attributes make it well suited to clustering data in high dimensional space.

Keywords: Enhanced locality sensitive clustering, Bucket indices, Random projection, Data clustering

1. INTRODUCTION

Data clustering is an important task in many areas, as such many clustering algorithms have been developed. However, many of these are neither effective nor efficient for data points in high dimensional spaces. There are several main reasons for this. Firstly, the inherent sparsity of high dimensional data hinders conventional clustering algorithms. Secondly, the distance between any two points becomes almost the same in high dimensional space [1], therefore it is difficult to differentiate similar data points from dissimilar ones. Thirdly, clusters are often embedded in subspaces of high dimensional space, and different clusters may exist in different subspaces [2].

Image clustering is a typical application of high dimensional clustering algorithms, this is because for image features nearly all the dimensions are high. So, the problem of high dimensional clustering is very apparent in image clustering applications. What's more, the scale of an image dataset is generally large, and this scale may also change in some online applications. The performance of conventional clustering algorithms deteriorates significantly when used on this kind of dataset. For example, k-means is the mainstream cluster method in the image visual bag of words model for visual dictionary construction. However, when the number of images is large, the time taken to cluster is very long and may even be unacceptable for practical purposes. On top of this, when the image data is ever increasing, re-clustering is needed for k-means as it is not a dynamic cluster method. These limitations of k-means seriously damage its feasibility for use with large incremental image datasets. Some improvements to k-means, such as hierarchal k-means [3] and approximate k-means [4], have been presented, however, these have been shown not to support dynamic clustering [5]. Currently, widely used clustering methods such as spectrum clustering [6] and Affinity Propagation [7] methods incur high memory and computation costs due to the matrix factorization that they require. Therefore, a more efficient, new cluster method is needed.

Random projection is used in many areas including fast approximate nearest-neighbor applications [8,9], clustering [10], signal processing [11], anomaly detection [12] and dimension reduction [13]. This is largely due to the fact that distances are preserved under such transformations in certain circumstances [14]. Moreover, random projections have also been applied to classifications for a variety of purposes [15,16].

E^2LSH(Exact Euclidean Locality Sensitive Hashing) is a spe-

† Author to whom all correspondence should be addressed:
E-mail: maplechen111@gmail.com

cial case of random projection, and it was first introduced as approximate near neighbor algorithm [17]. E^{2}LSH has attracted much attention recently, and has mainly been used in retrieval [18,19]. In fact, the data points projected into the same buckets were found to be much more similar than those projected into different buckets. So, if we take a dataset and divide into groups according to its bucket indices, the task of data clustering has already be achieved to a certain approximation. What's more, E^{2}LSH is a data independent method, so it can create a dynamic index for an incremental dataset. In this way, E^{2}LSH can be used as a dynamic clustering method. In fact, the introducer of LSH has pointed out that LSH can serve as a fast clustering algorithm, but he did this without going on to validate the claim. In fact, E^{2}LSH has even been applied to noun clustering by Ravichandran D [20]. However, clustering image data is more difficult than text word data because of its added complexity.

Therefore, we proposed an enhanced Locality Sensitive Clustering (LSC) method for use in high dimensional space based on E^{2}LSH. This method takes advantage of Locality Sensitive Hashing (LSH) and can cluster high dimension data at a high speed.

2. ENHANCED LOCALITY SENSITIVE CLUSTERING BASED ON RANDOM PROJECTION

The main property of random projection is dimension reduction. Compared with the classical dimension reduction algorithm PCA (Principal Component Analysis), random projection offers many benefits [21]. For example, generally PCA can't be used to reduce the dimension of a mixture of n Gaussians to below (Ωn), whereas random projection can reduce the dimension to just $\Omega(\log n)$. Random Projection has another tremendous benefit, even if the original Gaussians are highly skewed, their projected counterparts will be more spherical. This is a great advantage as it is much easier to design algorithms for spherical clusters than ellipsoidal ones.

As for data clustering, the Johnson-Lindenstrauss Lemma shows that the distances to data points are preserved after projection. This makes it capable for use in approximate nearest neighbor search when performing information retrieval.

2.1 The distance preservation of random projection

The Johnson-Lindenstrauss Lemma is famous for its distance preservation property. It can be described like this: Given $0<\varepsilon<1$ and any set S in $\mathbb{R}^n$, for a positive integer $d=O(\frac{1}{\gamma^2}\log|S|)$, there exists a map $f:\mathbb{R}^n\to\mathbb{R}^d$, such that for all $u,v\in S$,

$$(1-\gamma)\|u-v\|^2 \le \|f(u)-f(v)\|^2 \le (1+\gamma)\|u-v\|^2 \quad (1)$$

This lemma says that all pairwise distances are preserved, with high probability, up to 1±γ after mapping.

Let $N(0,1)$ denote the standard normal distribution with mean 0 and variance 1, and U(-1,1) denote the distribution that has the probability 1/2 on -1 and probability 1/2 on 1.

Let $u,v\in\mathrm{R}^n$, $u'=\frac{1}{\sqrt{d}}uA$ and $v'=\frac{1}{\sqrt{d}}vA$ where A is the random matrix, whose entries are chosen independently from either or . Then

$$(1-\gamma)\|u-v\|^2 \le \|f(u)-f(v)\|^2 \le (1+\gamma)\|u-v\|^2 \quad (2)$$

Imagine a set S of data in some high-dimensional space R^n, and suppose that we randomly project the data down to R^d. By the Johnson-Lindenstrauss Lemma, $d=O\left(\gamma^{-2}\log|S|\right)$ is sufficient so that, with high probability, all angles between points change by at most ±γ/2 [22]. In particular, consider projecting all points in S and the target vector ω, if initially data was separable by margin γ, then after projection, since angles with ω have changed by at most γ/2, the data will still be separable.

2.2 Locality sensitive clustering

E^{2}LSH is a special case of LSH (Locality Sensitive Hashing), and it is a random projection based method. This can be illustrated by the definition of the hashing function. The k hashing functions are generated by random methods, and their inner-product performs the data projection. This, however, is different from general random projection, because each data point is projected by k hashing functions, and as such results in k bucket indices that represent an original point. The k hashing functions also indicate a difference from general random projection in two ways. The first is the projection itself, the projection was not performed on the whole axis in one direction, but on parts of the axis. The second is that general projection is done by a matrix operation (a data point multiplies a $n\times d$ random projection matrix A), but in E^{2}LSH, a data point multiplies a single hashing vector, as such the matrix operation is omitted in E^{2}LSH. This is of vital importance for large-scale data processing, because the multiple matrix operations needed by traditional algorithms are impractical due to high computation and memory costs.

E^{2}LSH can also be used for data clustering. Based on the separability description above, we can say a dataset's distance, margin and angle could be preserved after projection. These properties make E^{2}LSH feasible for data clustering. In fact, the bucket indices found after projection can be used to group data points. This comes from the fact that similar data are arranged into a single bucket or the adjacent several buckets. So if we group these into a cluster, and further group all similar groups into corresponding clusters, the goal of data clustering can be achieved. This is the main idea behind Locality Sensitive Clustering.

E^{2}LSH is based on the p - stable distribution function, its single hashing function is defined as:

$$h(v)=\left\lfloor\frac{a\cdot v+b}{w}\right\rfloor \quad (3)$$

where a is a n-dimension vector generated by the p - stable distribution function, and the inner-product $(a\cdot v)$ works as a single channel random projection, b is the offset added to the random projection, and the module operation ensure the projected value (bucket index) is in a specific range.

The projection function is similar to LSH and projects points in $\mathbb{R}^n$ to $\mathbb{R}^k$:

$$\mathcal{G}=\{g:\mathbb{R}^d\to\mathbb{R}^k\},\ g(v)=(h_1(v),\cdots h_k(v)) \quad (4)$$

The enhanced Locality Sensitive Clustering (LSC) includes several main steps. Firstly, optimal parameters k and L are com-

puted. Secondly, the hashing function for point v $h = \left\lfloor \frac{a \cdot v + b}{w} \right\rfloor$ is constructed. Thirdly, all points are projected to bucket index (h_1,$\cdots h_k$), and the bucket indices of all points are clustered to get the cluster labels. The procedure for enhanced Locality Sensitive Clustering (LSC) is shown below:

Step 1, the optimal parameters k and L are calculated for a dataset S.

Step 2, k dimension vector A is generated from the Gaussian distribution, b and w are also generated according the definition of the LSH function.

Step 3, all the points $v_i \in S$ are projected to a k dimension bucket index B_i, and these bucket indices constitute a matrix B.

Step 4, randomly selected one column B_j from matrix B, and cluster bucket indices B_j to get n clusters where $j \in [1,m]$, $m=|S|$ and n is the number of clusters.

$$B_j \overset{cluster}{\rightarrow} l_k \tag{5}$$

where l_k denotes the cluster labels, $l_k = k$, $k \in [1,n]$, $n<m$.

Step 5, assign points in B_j to class k

$$classLabel(v) = k, \quad v \in S \bigcap B_j^{-1} \tag{6}$$

where B_j^{-1} denotes the points whose bucket index is B_j.

3. EXPERIMENTS

3.1 Experiments on an image dataset

To verify the effect of the new clustering method on real data, we constructed an image set from the TRECVID image dataset, the set contains four categories and 75 images total. The 4 categories are 'compere', 'singer', 'rice' and 'sports'.

To compare the performance of our new algorithm, we first ran k-means on this image set. The results of k-means on the image dataset are showed in Fig. 1.

The result of LSC on the image dataset is shown in Fig. 2. Most of the cluster labels in cluster 2 and 4 are correct, the clustering labels of cluster 3 are correct while several cluster labels of cluster 1 are wrong. We can see that the accuracy of clustering labels are less than in the synthetic data, this is because the distinctness of inter-clusters are less than in the synthetic data. However, results on real data are more meaningful.

As LSC is a randomized algorithm, it is understandable that its clustering results are less accurate than k-means. On the other hand, the advantage of LSC lies in its low computation cost, fast running speed and the dynamic clustering which comes from E^2LSH.

To compare the accuracy of the two methods, we computed a MAP of the clustering results for the 4 classes using LSC, k-means, Affinity Propagation (AP) cluster and Spectrum Cluster (SC) methods. The results are showed in Fig. 3. It can be seen that the accuracy of LSC is about 0.9, which is less than k-means, AP and SC.

To compare clustering times, we ran the four cluster methods (LSC, k-means, AP cluster and SC) on the image dataset. The results are showed in table 1. It can be seen that LSC consumes the least time while AP and SC cost significantly more time than LSC and k-means.

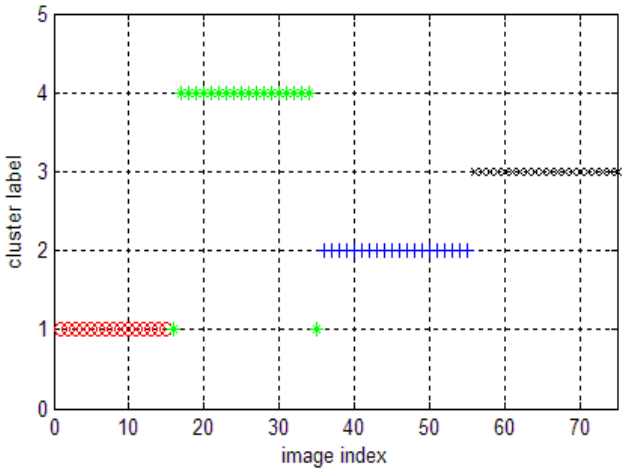

Fig. 1. Clustering results of k-means for image dataset.

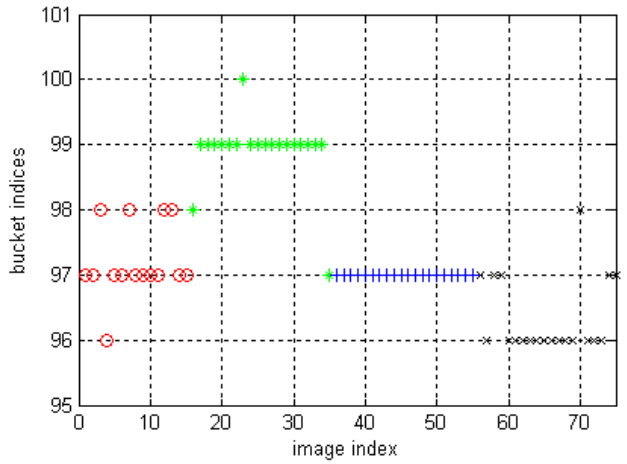

Fig. 2. The clustering results of LSC for image dataset.

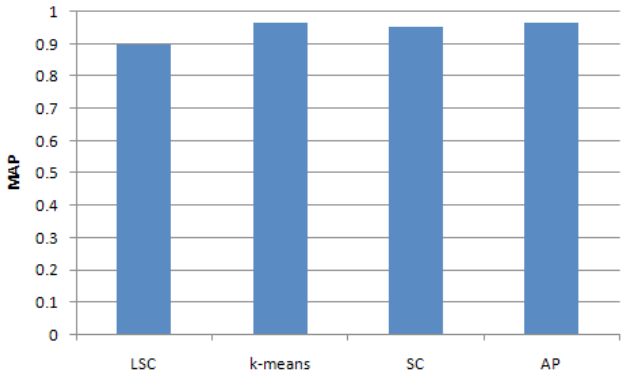

Fig. 3. Accuracy of clustering results for four clustering methods on the image dataset.

Table. 1. Clustering times of the four clustering methods on the image dataset

k-means	SC	AP	LSC
0.0156	0.094	4.3335	0.0127

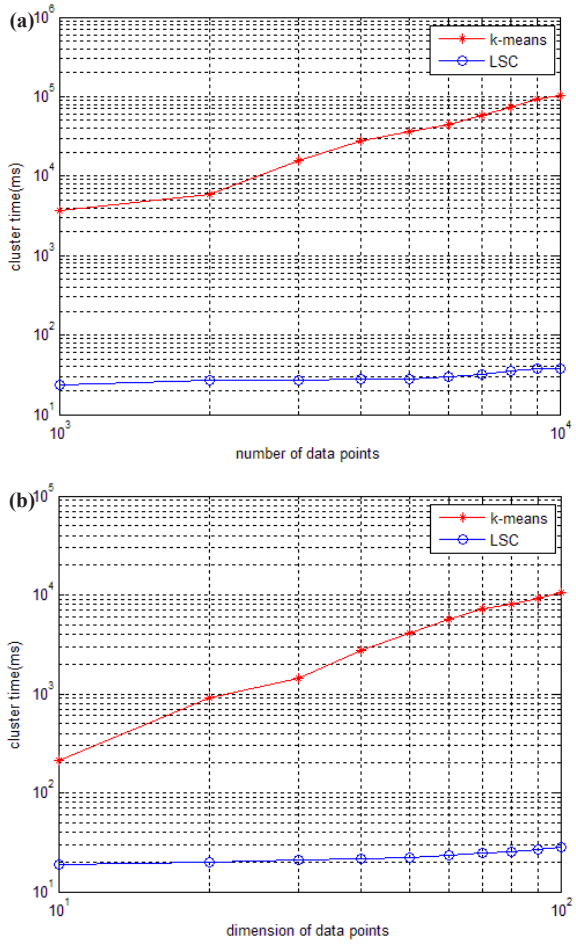

Fig. 4. Running times of the two methods for the two datasets. (a) running times for synthetic dataset 1 and (b) running times for synthetic dataset 2.

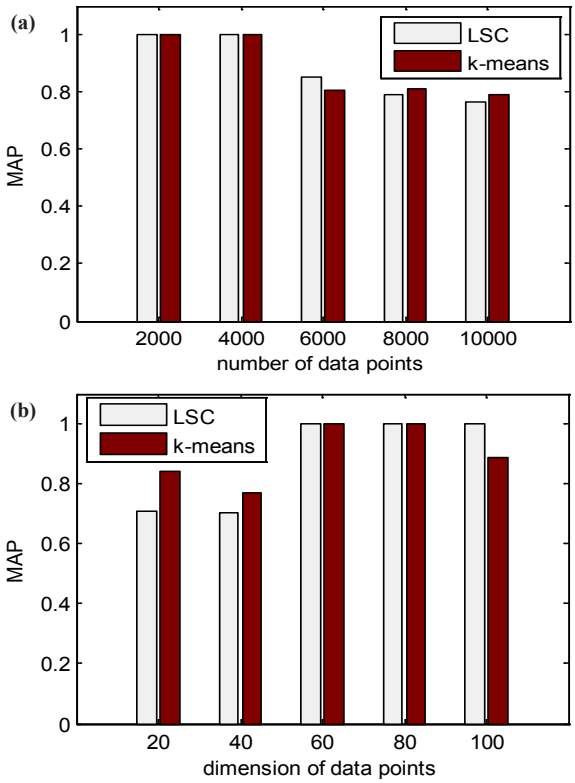

Fig. 5. MAP of the two methods for the two datasets. (a) MAP of the two methods for synthetic dataset 1 and (b) MAP of the two methods for synthetic dataset 2.

3.2 Experiments on an incremental dataset

To conveniently see the running time of our new clustering method, we construct a synthetic dataset 1 and synthetic dataset 2. Synthetic dataset 1 is an incremental dataset in scale, which contains 5 clusters of 100 dimension pieces of data, the number of data points increases from 1,000 to 10,000. Synthetic dataset 2 is an incremental dataset in dimension, which contains 5 clusters of 1,000 data points, the dimension of the points increases from 10 dimension data to 100 dimension data.

The running time of LSC and k-means for these two datasets is shown in Fig. 4. Figure 4 (a) indicates that the running time of k-means increases much more quickly than LSC when the number of data points increases from 1,000 to 10,000. When the number of data points is at 1,000, the time cost of k-means is at least 100 times more than that of LSC, and when the number of data points increases to 1,000, the time cost of k-means is at least 1,000 times more than that of LSC. Therefore, the advantage of LSC becomes more notable in larger scale datasets. In Fig. 4 (b), we can see the higher the dimensions of the data used is, the more notable the advantage of LSC is. Even in with low dimension data of 10, the time cost of k-means is at least 10 times more than LSC. To compare the cluster accuracy, MAPs of the two methods were calculated for these two datasets. The results are shown in Fig. 5. The figure indicates that the cluster accuracy of LSC is similar to k-means, and may be better than k-means on some datasets. Therefore, LSC is more suitable for incremental datasets, and it can be a good cluster method for both high dimension and large-scale datasets.

4 CONCLUSIONS

To improve the feasibility of high dimensional data clustering, especially for use in image clustering, an enhanced Locality Sensitive Clustering method based on E^{2}LSH is presented. This method first generates multiple hashing functions, then it projects each point using these hashing functions to get bucket indices. The bucket indices are then clustered to get class labels. In terms of clustering accuracy, experimental results show that LSC's performance is close to k-means, AP and SC. The advantage of LSC is its fast running speed that makes it suitable for incremental clustering. This property makes it a very valuable method for large dataset clustering and especially for incremental dataset clustering.

ACKNOWLEDGMENT

This work was supported by Nature Science Foundation of China No. 60872142.

REFERENCES

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

[2] R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan, *Proc. of SIGMOD Record ACM Special Interest Group on Management of*

Data, 94 (1998).
[3] D. Nister and H. Stewenius, *Proc. of IEEE Conference on Computer Vision and Pattern Recognition[C]* (New York, USA, 2006) p. 2161.
[4] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*(Minneapolis, USA, 2007) p.1-8.
[5] R. Marée, P. Denis, and L. Wehenkel, *Proc. of ACM SIGMM International Conference on Multimedia Information Retrieval Philadelphia* (Pennsylvania, USA:ACM, 2010) p. 29-38.
[6] J. Shi and J. Malik, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**, 888 (2000) [DOI: http://dx.doi.org/10.1109/34.868688].
[7] B. J. Frey and D. Dueck, *Science*, **315**, 972 (2007) [DOI: http://dx.doi.org/10.1126/science.1136800].
[8] P. Indyk , R. Motwani, *Proc. of the Symposium on Theory of Computing* (Dallas, USA:ACM, 1998) p. 604.
[9] S. Dasgupta and K. Sinha, Randomized Partition Trees for Exact Nearest Neighbor Search JMLR: Workshop and Conference Proc., **30**, 1 (2013).
[10] Schulman, L. J. Clustering for edge-cost minimization. In Proc. Annual ACM Symp. Theory of Computing, 2000: 547-55.
[11] D. L. Donoho, Compressed Sensing. *IEEE Trans. Information Theory*, **52**, 1289 (2006).
[12] J. E. Fowler and Q. Du, *IEEE Transaction on Image Processing*, **21**, 184 (2012).
[13] A. Schclara, L. Rokachb, and A. Amit, *Ensembles of Classifiers Based on Dimensionality Reduction* (2013).
[14] S. Dasgupta and A. Gupta, *Random Structures & Algorithms*, **22**, 60 (2002). http://dx.doi.org/10.1002/rsa.10073
[15] M. F. Balcan, A. Blum, and S. Vempala, *Machine Learning*, **65**, 79 (2006) [DOI: http://dx.doi.org/10.1007/s10994-006-7550-1].
[16] Q. Shi, J. Petterson, G. Dror, J. Langford, A. J. Smola, and S.V.N. Vishwanathan, *J. Mach. Learn. Res.*, **10**, 2615 (2009).
[17] Andoni and P. Indyk, *Communications of the ACM*, **51**, 117 (2008).
[18] Y. Y. Liang, J. M. Li, and B. Zhang, *Proc. of International Conference on Multimedia* (Beijing, China, ACM, 2009) p. 589.
[19] H. Jegou, M. Douze, and C. Schmid, *International Journal of Computer Vision*, **87**, 316 (2010) [DOI: http://dx.doi.org/10.1007/s11263-009-0285-2].
[20] D. Ravichandran, P. Pantel, and E. Hovy, *Proc. of the 43rd Annual Meeting on Association for Computational Linguistics* (Stroudsburg, USA, ACM, 2005) p. 622.
[21] S. Dasgupta, *Experiments with Random Projection. In Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence* (San Francisco, USA, 2000) p. 143.
[22] A. Blum, *Proc. of the 2005 International Conference on Subspace, Latent Structure and Feature Selection* (LNCS 3940, 2006) p.52.
[23] Q. Shi, C. Shen, and R. Hill, *International Conference on Machine Learning* (Edinburgh, Scotland, UK, 2012).