# LOCALITY SENSITIVE CLUSTERING IN HIGH DIMENSIONAL SPACE

Haolin Gao, Bicheng Li, Gang Chen and Yongwei Zhao
Department of Data Processing Engineering, Zhengzhou Information Science and Technology Institute,
Zhengzhou, China
E-Mail: holygao@126.com

## ABSTRACT

In high dimension space, many conventional clustering algorithms do not work well in effectiveness and efficiency, especially for image data set. For example, k-means is widely used in image clustering especially visual clustering. But its drawback such as long clustering time and high memory cost seriously deteriorates feasibility in incremental large image set. To improve the feasibility, we proposed a Locality Sensitive Clustering method. Firstly, multiple hashing functions are generated. Secondly, data points are projected to get bucket indices. Thirdly, proper quantification interval is selected to merge the bucket indices, and the cluster labels are assigned for each point. Experimental results show that on synthetic data set this method performs almost as well as k-means, and on image data set it performs slightly worse than k-means algorithm about accuracy. But its advantage is in low memory cost, fast running speed and incremental clustering. So Locality Sensitive Clustering can be used to clustering data, especially in high dimensional space.

**Keywords:** Exact Euclidean Locality Sensitive Hashing, locality sensitive clustering, random projection, data clustering.

## INTRODUCTION

Many conventional clustering algorithms do not work well in effectiveness and efficiency for data sets in high dimensional spaces for several reasons. Firstly, the inherent sparsity of high dimensional data cumbers conventional cluster algorithm. Secondly, the distance between any two points becomes almost the same (Cao *et al*., 2002); therefore it is difficult to differentiate similar data points from dissimilar ones. Thirdly, clusters are embedded in the subspaces of the high dimensional space, and different clusters may exist in different subspaces of different dimensions (Agrawal *et al*., 1998).

Image clustering is a typical application of high dimensional clustering, for nearly all the dimensions of image features are high. So, the problem of high dimensional clustering also lies in image clustering. Though many clustering algorithms have developed for these years, most of them don't work well in image clustering, especially in visual dictionary construction. The common used clustering algorithm is still k-means, but the limitation of k-means seriously deteriorates its feasibility in incremental large image set.

Random projection is used in many areas including fast approximate nearest-neighbor (Indyk P *et al*., 1998; Sanjoy Dasgupta *et al*., 2013), clustering (Schulman, L. J, 2000), signal processing (Donoho, D. L, 2006), anomaly detection (James E. Fowler, 2012), dimension reduction (Alon Schclara *et al*., 2013) and so on. This is largely due to the fact that distances are preserved under such transformations in certain circumstances (Dasgupta, S. *et al*., 2002). Moreover, random projections have also been applied to classification for a variety of purposes (Balcan, M. F, 2006; Duarte, M.F, 2007; Shi, Q *et al*., 2009).

Exact Euclidean Locality Sensitive Hashing is a special case of random projection, and it is first introduced as approximate near neighbor algorithm (Andoni, 2008). It has attracted much attention in recently, and was mainly used in retrieval (Liang YingYu, 2009; Jegou H, 2010). In fact, the data points projected into a same buckets are much similar than those in different buckets. So, if we divide a data set according bucket indices into groups, the task of data clustering can also be achieved approximately. What's more, $E^2$LSH is a data independent method, so it can create a dynamic index for an incremental dataset. And if used in data clustering, it can be a dynamic clustering method. In fact, the introducer of LSH has pointed that LSH can serve as a fast clustering algorithm, but he didn't testify it. In fact, $E^2$LSH has applied for noun clustering by Ravichandran D (Ravichandran D. *et al*., 2005). However, clustering image data is more difficult than text word data for its complexity.

Therefore, we propose a Locality Sensitive Clustering method in high dimensional space based on distance and separability preservation property of random projection and the rationale of $E^2$LSH, which makes use of the advantage of Locality Sensitive Hashing. In this method data points are projected to bucket indices by hashing functions, and bucket indices are merged to cluster labels. It can cluster high dimension data at a high speed and low cost.

## RANDOM PROJECTION AND SEPERABILITY PRESERVATION

Dimension reduction is a main property of random projection, compared with classical dimension reduction algorithm PCA (Principal Component Analysis); Random projection offers many benefits (Sanjoy Dasgupta, 2000). PCA can't be used to reduce the dimension of a mixture of *n* Gaussians to below $\Omega(n)$ generally, whereas random projection can reduce the dimension to just $\Omega(\log n)$ . Moreover, PCA may not reduce the eccentricity of Gaussians. However, if the projected dimension is high enough, then a PCA-projected mixture could easily be far better separated than its randomly projected counterpart. For this reason PCA remains an important tool in the study of Gaussian clusters. Random Projection has another tremendous benefit, even if the original Gaussians are highly skewed, their projected counterparts will be more spherical. And it is much easier to design algorithms for spherical clusters than ellipsoidal ones.

As to data clustering, the Johnson-Lindenstrauss Lemma shows that the distance of data points are preserved after projection. This makes it capable for approximate nearest neighbor search in information retrieval. Only distance preservation is not sufficient for clustering, the preservation of the class margin of data set can be helpful in clustering.

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

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

This lemma says that all pairwise distances are preserved up to $1\pm\gamma$ with high probability 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 probability 1/2 on −1 and probability 1/2 on 1.

Let $u,v\in R^n$, $u'=\frac{1}{\sqrt{d}}uA$ and $v'=\frac{1}{\sqrt{d}}vA$ where $A$ is $n\times d$ random matrix, whose entries are chosen independently from either $N(0,1)$ or $U(-1,1)$. Then

$$\Pr_A\begin{bmatrix}(1-\gamma)\|u-v\|^2 \le \|u'-v'\|^2 \le \\ (1+\gamma)\|u-v\|^2\end{bmatrix} \ge 1-2e^{-(\gamma^2-\gamma^3)\frac{d}{4}} \qquad (2)$$

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 changed by at most $\pm\gamma/2$ (Avrim Blum, 2006). In particular, consider projecting all points in $S$ and the target vector $\omega$, if initially data was separable by margin $\gamma$, then after projection, since angles with $\omega$ have changed by at most $\gamma/2$, the data is still separable.

Shi *et al*. established the conditions under which margins are preserved after random projection, and to show that error free margins are preserved for both binary and multiclass problems if these conditions were satisfied (Qinfeng Shi *et al*., 2012) Balcan *et al*. studied the problem of margin preservation under random projection for binary classification, and provided a lower bound on the number of dimensions required if a random projection was to have a given probability of maintaining half of the original margin in the data (Balcan, M. F. *et al*., 2006).

But it demands infinite many projections in order to guarantee the preservation of an error free margin. They provided two margin definitions below:

Normalized Margin: A dataset $\mathcal{S}$ is linearly separated by margin $\gamma$ if there exists $\mathbf{u}\in\mathbb{R}^d$, such that for all $(x,y)\in S$,

$$y\frac{\langle \mathbf{u},x\rangle}{\|\mathbf{u}\|\|x\|}\ge\gamma \qquad (3)$$

Error-allowed Margin: A data distribution $D$ is linearly separated by margin $\gamma$ with error $\rho$, if there exists $\mathbf{u}\in\mathbb{R}^d$, such that

$$\Pr_{(x,y)\sim D}\left(y\frac{\langle \mathbf{u},x\rangle}{\|\mathbf{u}\|\|x\|}<\gamma\right)\le\rho \qquad (4)$$

If the original data has normalized margin $\gamma$ then as long as the number of projections

$$n\ge\frac{c}{\gamma^2}\ln\frac{1}{\rho\delta} \qquad (5)$$

for an appropriate constant c, the projected data has margin $\gamma/2$ with error $\rho$, with probability at least $1-\delta$. Equal (4) shows that a positive margin implies $\rho$ = 0, by which Equal (5) implies that $n=+\infty$. Thus in order to preserve a positive margin in the projected data infinitely many random projections are needed.

For binary margin preservation, given any random Gaussian matrix $\mathbf{R}\in\mathbb{R}^{n,d}$, if the dataset $S=\left\{\left(x_i\in\mathbb{R}^d, y_i\in\{-1,+1\}\right)\right\}_{i=1}^m$ is linear separable by margin $\gamma\in(0,1]$, then for any $\delta,\varepsilon\in(0,1)$ and any

$$n>\frac{12}{3\varepsilon^2-2\varepsilon^3}\ln\frac{6m}{\delta} \qquad (6)$$

with probability at least $1-\delta$, the dataset $S'=\left\{\left(\mathbf{R}x_i\in\mathbb{R}^n, y_i\in\{-1,+1\}\right)\right\}_{i=1}^m$ is linear separable by margin $\gamma-\frac{2\varepsilon}{1-\varepsilon}$.

The lower bound of the margin after random projection can become negative for certain values of $\varepsilon$. A negative margin implies that the projected data are not linearly separable. When the lower bound is positive, Equal (6) indicates that margin separability for binary classification is preserved with high probability under random projection.

For normalized multiclass margin, the multiclass dataset $S=\left\{\left(x_i \in \mathbb{R}^d, y_i \in Y=\{1,\cdots,L\}\right)\right\}_{i=1}^m$ is linear separable by margin $\gamma \in (0,1]$, if there exists $\left\{u_y \in \mathbb{R}^d\right\}_{y\in Y}$, such that for all $(x,y)\in S$

$$\frac{\langle \mathrm{u}_y, x\rangle}{\|\mathrm{u}_y\|\|x\|}-\max_{y'\neq y}\frac{\langle \mathrm{u}_{y'}, x\rangle}{\|\mathrm{u}_{y'}\|\|x\|}\geq\gamma \tag{7}$$

For any multiclass dataset $S$ and any Gaussian random matrix R, if $S$ is linearly separable by margin $\gamma\in(0,1]$, then for any $\delta,\varepsilon\in(0,1)$ and any

$$n>\frac{12}{3\varepsilon^2-2\varepsilon^3}\ln\frac{6Lm}{\delta}$$

with probability at least $1-\delta$, the dataset $S'=\left\{\mathbf{R}x_i\in\mathbb{R}^n, y_i\in Y\right\}_{i=1}^m$ is linear separable by margin $-\frac{1+3\varepsilon}{1-\varepsilon^2}+\frac{\sqrt{1-\varepsilon^2}}{1+\varepsilon}+\frac{1+\varepsilon}{1-\varepsilon}\gamma$.

For angle preservation, given any $w,x\in\mathbb{R}^d$ and any random Gaussian matrix $\mathbf{R}\in\mathbb{R}^{n,d}$, for any $\varepsilon\in(0,1)$, if $\langle w,x\rangle>0$, then with probability at least $1-6\exp\left(-\frac{n}{2}\left(\frac{\varepsilon^2}{2}-\frac{\varepsilon^3}{3}\right)\right)$ the following holds

$$\begin{aligned}&\frac{1+\varepsilon}{1-\varepsilon}\frac{\langle w,x\rangle}{\|w\|\|x\|}-\frac{2\varepsilon}{(1-\varepsilon)}\leq\frac{\langle\mathbf{R}\mathrm{w},\mathbf{R}\mathrm{x}\rangle}{\|\mathbf{R}\mathrm{w}\|\|\mathbf{R}\mathrm{x}\|}\\&\leq 1-\frac{\sqrt{1-\varepsilon^2}}{1+\varepsilon}+\frac{\varepsilon}{1+\varepsilon}+\frac{1+\varepsilon}{1-\varepsilon}\frac{\langle w,x\rangle}{\|w\|\|x\|}\end{aligned}\tag{8}$$

## 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 seen from the definition of hashing function. The *k* hashing functions are generated by random methods, and the inner-product perform the data projection. But it is different from general random projection. Each data point is projected by *k* hashing functions, and the results are *k* bucket indices, which were representation of a point. The *k* hashing functions also indicate difference from general random projection. The first is the projection itself, the projection were not performed on the whole axis of some direction, but on parts of the axis. The second is that general projection is done by a matrix operation, that is to say a data point multiplies a $n\times d$ random projection matrix A, but this converted to a data points multiplies a single hashing function. This means that the matrix operation is omitted in E$^2$LSH algorithm. This is of vital importance for large scale data processing, because matrix operation is nearly infeasible or high computation and memory cost.

In data clustering, E$^2$LSH could also come into application. Based on the former separability description, a data set could be distance, margin and angle preservation after projection. There properties make E$^2$LSH feasibility for data clustering. In fact, the bucket indices after projection can be used to group data points. It means that similar data are arranged into a single bucket or adjacent several buckets. So if we group them into a cluster, and further group all similar groups into corresponding clusters, the destination of data clustering is achieved. This is the main idea for Locality Sensitive Clustering.

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

$$h(v)=\left\lfloor\frac{a\cdot v+b}{w}\right\rfloor \tag{9}$$

where *a* is a n-dimension vector generated by $p$ - stable distribution function, and inner-product $(a\cdot v)$ work 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, projected points in $\mathbb{R}^n$ to $\mathbb{R}^k$:

$$\mathcal{G}=\{g:\mathbb{R}^d\to\mathbb{R}^k\},\ g(v)=(h_1(v),\cdots h_k(v)) \tag{10}$$

The Locality Sensitive Clustering (LSC) mainly includes several steps. The first is to compute optimal parameters *k* and *L*. Secondly, constructs random matrix *A* and random vector *b* and *w* to substitute *a*, *b* and *w* in hashing function $h=\left\lfloor\frac{a\cdot v+b}{w}\right\rfloor$. Thirdly, projects every point and stores the bucket index $(h_1,\cdots h_k)$ in but map chain. Fourthly, selects proper quantification interval and merge bucket indices, the last is to group data points according merged bucket indices. The procedure of Locality Sensitive Clustering (LSC) is showed as follows:

**Step-1:** for a data set *S*, optimal parameters *k* and *L*;

**Step-2:** generate $n\times k$ random matrix *A* from Gaussian distribution, and repeat *L* times, $A=(A_1,\cdots A_L)$, generate k dimension *b* and *w* according the definition of LSH function, $b=(b_1,\cdots b_k)$, $w=(w_1,\cdots w_L)$;

**Step-3:** perform random projection for all points $v_i\in S$, the results are bucket indices $B_i = (B_{i1}, B_{i2} \cdots B_{iL})$, each $B_{ij}$ is a $k$ dimension vector

$$B_{ij} = \left\lfloor \frac{v \cdot A_l + b}{w} \right\rfloor \quad (11)$$

where $l \in [1, L]$.

**Step-4:** select proper quantification interval to merge $m$ bucket indices $B_i$, $m = |S|$, $i \in [1, m]$.

$$B \overset{merge}{\rightarrow} B' \quad (12)$$

where $B = \{B_1, B_2, \cdots B_m\}$, $B' = \{B_1, B_2, \cdots B_n\}$, $n \le m$.

**Step-5:** assign points in $B_i'$ to class *i*.

$$classLabel = \begin{cases} 1, & v \in S \bigcap B_1'^{-1} \\ 2, & v \in S \bigcap B_2'^{-1} \\ \vdots & \\ n, & v \in S \bigcap B_n'^{-1} \end{cases} \quad (13)$$

where $B_i'^{-1}$ denotes the points whose bucket index after merging is $B_i'$.

## EXPERIMENTS

To intuitively show the clustering results, we first run clustering algorithms on two synthetic data sets. The synthetic data set 1 contains 30 points of 2 dimension belonged to 3 clusters orderly (called data set 1), that is the first 10 points belong to the first cluster; the last 10 points belong to the third cluster. Similarly, the data set 2 contains 30 points of 100 dimension belonged to 3 clusters (called data set 2) in order too. We also construct an image data 1, which comes from TRECVID image set, containing four categories and 75 images total (called image set 1), the 4 categories includes ‘compere’, ‘singer’, ‘rice’ and ‘sports’.

### The experiments on synthetic data sets

We first compare the clustering results of k-means with Locality Sensitive Clustering method on synthetic data set 1. The results of k-means on data set 1 are showed in Figure-1. The bucket indices on data set 1 are showed in Figure-2.

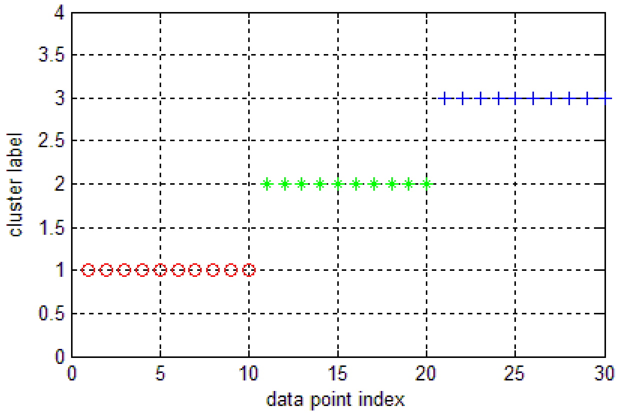

**Figure-1.** The clustering results of k-means on data set 1.

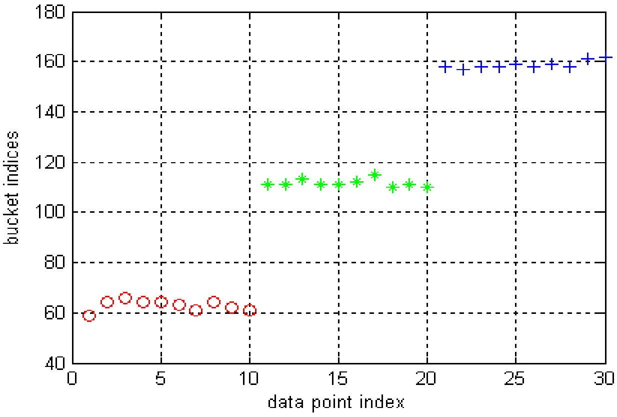

**Figure-2.** The bucket indices of LSC on data set 1.

From the Figure-1, we can see that all the 30 points are correctly grouped into 3 clusters, and the accuracy achieves 100 percent.

The Figure-2 indicates that the bucket indices of each point may be different in each cluster, but the difference among them in same cluster fluctuates in a small range, and the inner-class distance are smaller than the inter-class difference. So, the bucket indices can be used to decide cluster labels, but they need to be merged by specific quantification interval first.

The bucket indices generated from 5 LSC runs were shower in Table-1. From Table-1 we can also see that the bucket indices of same point in each cluster may also different. This comes from the randomization of hash functions.

**Table-1.** The bucket indices of LSC for the 1st cluster.

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

To verify the effect the new clustering method on high dimension data set, we also run new method for data set 2. The results of k-means are showed in Figure-3.

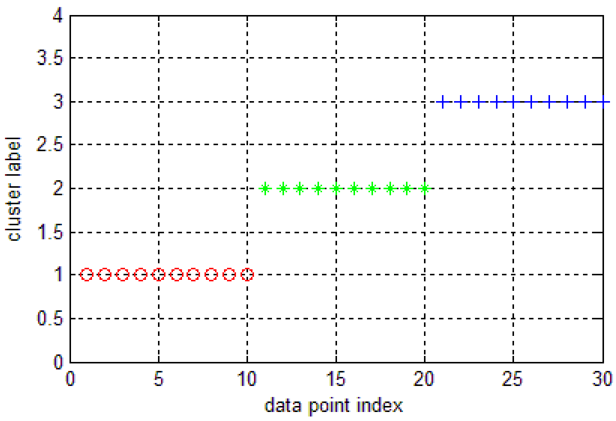

**Figure-3.** The clustering results of k-Means on data set 2.

We can see from Figure-3 that all the 30 points are correctly grouped into 3 clusters, and the accuracy achieves 100 percent. Repeat the experiment 5 times, the results are showed in Table-2. In this table, only the results of first cluster are showed, we can see that though the labels are different each time, the labels are also all correct, because they can be grouped into one cluster every time.

The result of LSC is also showed in Figure-4, the bucket indices also correctly indicate the origin cluster the corresponding points belonging to. The indices of each point in the first cluster were showed in Table-3 for 5 runs.

**Table-2.** The clustering results of k-means for the 1st cluster.

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

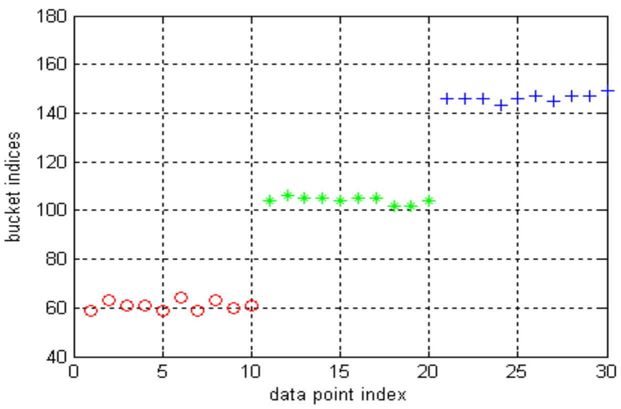

**Figure-4.** The bucket indices of LSC on data set 2.

The bucket indices of points in same cluster in Table-3 are also different. This comes from the randomization of random projection vectors. The quantification intervals of these bucket indices are different from that of data set 1. After merging bucket indices, right clustering results can be achieved.

**Table-3.** The clustering results of LSC for the 1st cluster.

| Data point | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| bucket index | 59 | 63 | 61 | 61 | 59 | 64 | 59 | 63 | 60 | 61 |
| bucket index | 69 | 76 | 73 | 70 | 72 | 73 | 71 | 72 | 70 | 73 |
| bucket index | 67 | 70 | 68 | 68 | 69 | 72 | 65 | 71 | 67 | 69 |
| bucket index | 71 | 77 | 74 | 73 | 76 | 76 | 71 | 74 | 70 | 75 |
| bucket index | 67 | 65 | 68 | 68 | 69 | 64 | 65 | 71 | 66 | 70 |

### The experiments on image set

To verify the effect the new clustering method on real data, we also run new method on image set 1. To compare the performance, we first run k-means on this image set. The results of k-means on image set 1 are showed in Figure-5.

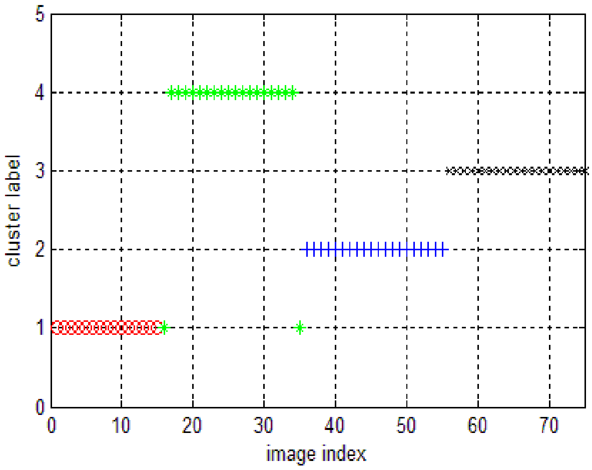

**Figure-5.** The clustering results of k-means for image set 1.

The result of LSC on image set 1 is showed in Figure-6, most of the cluster labels of cluster 2 and 4 are correct, clustering labels of cluster 3 are correct, and several cluster labels of cluster 1 are wrong. We can see that the accuracy of clustering labels are less than synthetic, this is because the distinctness of inter-cluster are less than synthetic data. But the results on real data are more meaningful.

Because 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 low computation cost, fast running speed and dynamic clustering which come from $E^2$LSH.

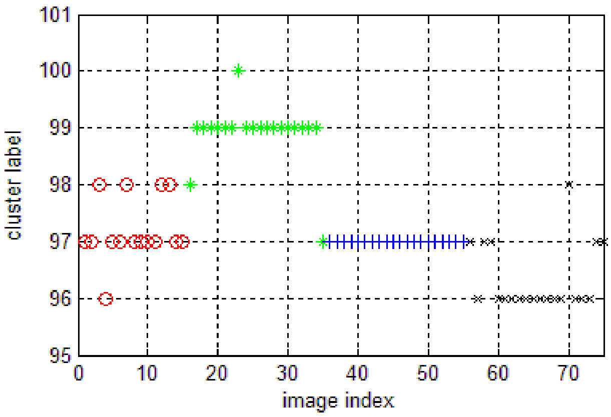

**Figure-6.** The clustering results of LSC for image set 1.

## CONCLUSIONS

To improve the feasibility of high dimensional data clustering especially image clustering, Locality Sensitive Clustering is presented based on $E^2$LSH. It first generates the multiple hashing functions, then projects each point by these hashing functions to get bucket indices, and then merge the bucket indices by proper quantification interval, at last decide the last labels for each data point according merged bucket indices. For the clustering accuracy, experiments show that on synthetic data set, LSC performs nearly as good as k-means algorithm, and on image set LSC performs slightly worse than k-means algorithm. But its advantage such as fast running speed, low memory cost and dynamic clustering are more urgent for large dataset clustering, especially in incremental dataset, and these elements are key components for the feasibility of large dataset clustering.

## ACKNOWLEDGEMENT

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

## REFERENCES

Cao Y. and Wu J. 2002. Projective ART for clustering data sets in high dimensional spaces. Neural Networks. 2002,15: 105-120.

Agrawal R., Gehrke J., Gunopulos D. and Raghavan P. 1998. Automatic subspace clustering of high dimensional data for data mining applications. In: Proceedings of SIGMOD Record ACM Special Interest Group on Management of Data. pp. 94-105.

Indyk P and Motwani R. 1998. Approximate nearest neighbors: towards removing the curse of dimensionality. In: Proceedings of the Symposium on Theory of Computing. Dallas, Texas, USA: ACM. pp. 604-613.

Sanjoy Dasgupta and Kaushik Sinha. 2013. Randomized partition trees for exact nearest neighbor search JMLR: Workshop and Conference Proceedings. 30: 1-21.

Schulman L. J. 2000. Clustering for edge-cost minimization. In: Proc. Annual ACM Symp. Theory of Computing. pp. 547-555.

Donoho D. L. 2006. Compressed sensing. IEEE Trans. Information Theory. 52(4): 1289-1306.

James E. Fowler and Qian Du. 2012. Anomaly Detection and Reconstruction from Random Projections. IEEE Transaction on Image Processing. 21(1): 184-195.

Alon Schclara, Lior Rokachb and Amir Amit. 2013. Ensembles of Classifiers based on Dimensionality Reduction. Pattern Analysis and Applications Journal, 16(5): 1305-1345.

Dasgupta S. and Gupta A. 2002. An elementary proof of a theorem of johnson and lindenstrauss. Random Structures and Algorithms. 22(1): 60-65.

Balcan M.F., Blum A. and Vempala S. 2006. Kernels as features: On kernels, margins, and low-dimensional mappings. Machine Learning. 65(1): 79-94.

Duarte M.F., Davenport M.A., Wakin M.B., Laska J.N., Takhar D., Kelly K.F. and Baraniuk R.G. 2007. Multiscale random projections for compressive classification. In: Proceedings of IEEE International Conference on Image Processing. 6: 161-164.

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.

Andoni and P. Indyk. 2008. Near-optimal hashing algorithms for approximate nearest neighbor in high dimensions. Communications of the ACM. 51(1): 117-122.

Liang YingYu, Li JianMin and Zhang Bo. 2009. Vocabulary-based Hashing for Image Search. In: Proceedings of International Conference on Multimedia. Beijing, China: ACM. pp. 589-592.

Jegou H, Douze M and Schmid C. 2010. Improving bag-of-features for large scale image search. International Journal of Computer Vision. 87(3): 316-336.

Ravichandran D, Pantel P and Hovy E. 2005. Randomized Algorithms and NLP: Using Locality Sensitive Hash Function for High Speed Noun Clustering. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, Stroudsburg, PA, USA: ACM. pp. 622-629.

Sanjoy Dasgupta. 2000. Experiments with Random Projection. In: Proceedings of the 16th Conference on Uncertainty in Artificial Intelligence. San Francisco, CA, USA. pp. 143-151.

Avrim Blum. 2006. Random Projection, Margins, Kernels, and Feature Selection. In: Proceedings of the 2005 International Conference on Subspace, Latent Structure and Feature Selection. LNCS 3940. pp. 52-68.

Qinfeng Shi, Chunhua Shen, Rhys Hill and Anton van den Hengel. 2012. Is margin preserved after random projection? International Conference on Machine Learning, Edinburgh, Scotland, UK.