Accelerated kmeans Clustering using Binary Random Projection

Yukyung Choi, Chaehoon Park, and In So Kweon

Robotics and Computer Vision Lab., KAIST, Korea

Abstract. Codebooks have been widely used for image retrieval and image indexing, which are the core elements of mobile visual searching. Building a vocabulary tree is carried out offline, because the clustering of a large amount of training data takes a long time. Recently proposed adaptive vocabulary trees do not require offline training, but suffer from the burden of online computation. The necessity for clustering high dimensional large data has arisen in offline and online training. In this paper, we present a novel clustering method to reduce the burden of computation without losing accuracy. Feature selection is used to reduce the computational complexity with high dimensional data, and an ensemble learning model is used to improve the efficiency with a large number of data. We demonstrate that the proposed method outperforms the-state of the art approaches in terms of computational complexity on various synthetic and real datasets.

1 Introduction

Image to image matching is one of the important tasks in mobile visual searching. A vocabulary tree based image search is commonly used due to its simplicity and high performance [1–6]. The original vocabulary tree method [1] cannot grow and adapt with new images and environments, and it takes a long time to build a vocabulary tree through clustering. An incremental vocabulary tree was introduced to overcome limitations such as adaptation in dynamic environments [4, 5]. It does not require heavy clustering for the offline training process due to the use of a distributed online process. The clustering time of the incremental vocabulary tree is the chief burden in the case of realtime application. Thus, an efficient clustering method is required.

Lloyd kmeans [7] is the standard method. However, this algorithm is not suitable for high dimensional large data, because the computational complexity is proportional to the number and the dimension of data. Various approaches have been proposed to accelerate the clustering and reduce the complexity. One widely used approach is applying geometric knowledge to avoid unnecessary computations. Elkans algorithm [8] is the representative example, and this method does not calculate unnecessary distances between points and centers. Two additional strategies for accelerating kmeans are refining initial data and finding good initial clusters. The P.S Bradley approach [9] refines initial clusters as data close to the modes of the joint probability density. If initial clusters are selected

by nearby modes, true clusters are found more often, and the algorithm iterates fewer times. Arthur kmeans [10] is a representative algorithm that chooses good initial clusters for fast convergence. This algorithm randomly selects the first center for each cluster, and then subsequent centers are determined with the probability proportional to the squared distance from the closest center.

The aforementioned approaches, however, are not relevant to high dimensional large data except for Elkans algorithm. This type of data contains a high degree of irrelevant and redundant information [11]. Also, owing to the sparsity of data, it is difficult to find the hidden structure in a high dimension space. Some researchers thus have recently solved the high dimensional problem by decreasing the dimensionality [12, 13]. Others have proposed clustering the original data in a low dimensional subspace rather than directly in a high dimensional space [14–16]. Two basic types of approaches to reduce the dimensionality have been investigated: feature selection [14] and feature transformation [15, 16]. One of the feature selection methods, random projection [17], has received attention due to the simplicity and efficiency of computation.

Ensemble learning is mainly used for classification and detection. Fred [18] first introduced ensemble learning to the clustering society in the form of an ensemble combination method. The ensemble approach for clustering is robust and efficient in dealing with high dimensional large data, because distributed processing is possible and diversity is preserved. In detail, the ensemble approach consists of the generation and combination steps. Robustness and efficiency of an algorithm can be obtained through various models in the generation step [19]. To produce a final model, multiple models are properly combined in the combination step [20, 21].

In this paper, we show that kmeans clustering can be formulated by feature selection and an ensemble learning approach. We propose a two-stage algorithm, following the coarse to fine strategy. In the first stage, we obtain the sub-optimal clusters, and then we obtain the optimal clusters in the second stage. We employ a proposed binary random matrix, which is learned by each ensemble model. Also, using this simple matrix, the computational complexity is reduced. Due to the first ensemble stage, our method chooses the initial points nearby sub-optimal clusters in the second stage. Refined data taken from an ensemble method can be sufficiently representative because they are sub-optimal. Also, our method can avoid unnecessary distance calculation by a triangle inequality and distance bounds. As will be seen in Sec. 3, we show good performance with a binary random matrix, thus demonstrating that the proposed random matrix is suitable for finding independent bases.

This paper is organized as follows. In Sec. 2, the proposed algorithm to solve the accelerated clustering problem with high dimensional large data is described. Sec. 3 presents various experimental results on object classification, image retrieval, and loop detection.

2 Proposed algorithm

The kmeans algorithm finds values for the binary membership indicator r_{ij} and the cluster center c_j so as to minimize errors in Eq. (1). If data point x_i is assigned to cluster c_j then $r_{ij} = 1$.

$$J = \sum_{i}^{N} \sum_{j}^{K} r_{ij} \|x_i - c_j\|_2 \tag{1}$$

We can do this through an iterative procedure in which each iteration involves two successive steps corresponding to successive optimizations with respect to r_{ij} and the c_j. The conventional kmeans algorithm is expensive for high dimensional large datasets, requiring $O(K * N * D)$ computation time, where K is the number of clusters, N is the number of input data and D is the maximum number of non-zero elements in any example vector. Large datasets of high dimensionality for kmeans clustering should be studied.

$$J \cong \sum_{i}^{\hat{N}} \sum_{j}^{K} r_{ij} \|\hat{x}_i - \hat{c}_j\|_2 \tag{2}$$

We proposed a novel framework for an accelerating algorithm in Eq. (2). The goal of our method is to find the refining data $\hat{x}$ that well represent the distribution of the original input x. Using the refining data obtained from Eq. (3), the final clustering process for K clusters is equal to the coarse to fine strategy for accelerated clustering. The number of $\hat{x}$, namely $\hat{N}$ is relatively small, but refined data $\hat{x}$ can sufficiently represent the original input data x. c and $\hat{c}$ represent the center of the clusters in each set of data.

$$J \cong \sum_{i}^{N} \sum_{j}^{\hat{N}} r_{ij} \|x_i - \hat{x}_j\|_2 \tag{3}$$

To obtain the refining data introduced in Eq. (2), this paper adopts a kmeans optimizer, as delineated in Eq. (3), because it affords simplicity and compatibility with variations of kmeans. The refining data $\hat{x}$ in Eq. (3) are used as the input data of Eq. (2) to calculate the K clusters. In the above, $\hat{x}$ denote refined data that have representativeness of the input x. $\hat{N}$ is the number of data $\hat{x}$. The $\hat{N}$ value is much smaller than N.

$$J \cong \sum_{i}^{N} \sum_{j}^{\hat{N}} r_{ij} \|A(x_i - \hat{x}_j)\|_2 \tag{4}$$

For estimating the refining data with the conventional kmeans optimizer, we propose a clustering framework that combines random selection for dimension reduction and the ensemble models. This paper proposes a way to minimize data loss using a feature selection method, binary random projection. Our approach can discover underlying hidden clusters in noisy data through dimension reduction. For these reasons, Eq. (3) is reformulated as Eq. (4). According to Eq. (4),

the proposed method chooses $\hat{x}$ that best represents x. In the above, matrix A is the selection matrix of features. This matrix is called a random matrix.

$$J \cong \sum_{m}^{T} \sum_{i}^{\widetilde{N}^m} \sum_{j}^{\widetilde{K}^m} r_{ij}^m \|A^m(\widetilde{x}_i^m - \hat{x}_j^m)\|_2 \tag{5}$$

Eq. (4) is rewritten as Eq. (5) using ensemble learning models. Ensemble learning based our method reduces the risk of unfortunate feature selection and splits the data into small subset. Our work can select more stable and robust refining data $\hat{x}$ comparable with the results of Eq. (4) In the above, $\widetilde{x}$ and $\widetilde{N}$ denote sampling data of input x and the number of sampling data, and T and m denote the number and the order of ensemble models, respectively.

$$J \cong \sum_{m}^{T} \sum_{i}^{\widetilde{N}^m} \sum_{j}^{\widetilde{K}^m} r_{ij}^m \|(\widetilde{x}_i^{'m} - \hat{x}_j^{'m})\|_2 \tag{6}$$

Eq. (6) can be derived from Eq. (5) by random selection instead of random projection. In the above, the prime symbol denotes that variables are projected by matrix A.

Finally, this paper approximates the kmeans clustering as both Eq. (6) and Eq. (2). This approach presents an efficient kmeans clustering method that capitalizes on the randomness and the sparseness of the projection matrix for dimension reduction in high dimensional large data.

As mentioned above, our algorithm is composed of two phases combining Eq. (3) and Eq. (2). In the first stage, our approach builds multiple models by small sub-samples of the dataset. Each separated dataset is applied to kmeans clustering, and it randomly selects arbitrary attribute-features in every iteration. As we compute the minimization error in every iteration, we only require sub-dimensional data. The approximated centroids can be obtained by smaller iterations than one phase clustering. These refined data from the first stage are used as the input of the next step. The second stage consists of a single kmeans optimizer to merge the distributed operations. Our algorithm adopts a coarse to fine strategy so that the product of the first stage is suitable to achieve fast convergence. The algorithm is delineated below in Algorithm 1.

2.1 Feature selection in single model

$$\sum_{i}^{\widetilde{N}^m} \sum_{j}^{\widetilde{K}^m} r_{ij}^m \|A^m(\widetilde{x}_i^m - \hat{x}_j^m)\|_2 \tag{7}$$

Eq. (7) indicates the m_{th} single model in the ensemble generation stage. In each model, our algorithm finds values for r_{ij}^m, $\hat{x}_j^m$ and the A^m so as to minimize errors in Eq. (7). This problem is considered as a clustering problem in the high dimensional subspace. In this chapter, we describe basic concepts of the dimension reduction approaches, and we analyze the proposed algorithm with in comparison with others [14, 22].

Random projection Principal component analysis (PCA) is a widely used method for reducing the dimensionality of data. Unfortunately, it is quite expensive to compute in high dimensional data. It is thus desirable to derive a dimension reduction method that is computationally simple without yielding significant distortion.

As an alternative method, random projection (RP) has been found to be computationally efficient yet sufficiently accurate for the dimension reduction of high dimensional data. In random projection, the d-dimensional data in original spaces is projected to d'-dimensional sub-spaces. This random projection uses the matrix $A_{d' \times d}$, where the columns have unit lengths, and it is calculated through the origin. Using matrix notation, the equation is given as follows: $X^{RP}_{d' \times N} = A_{d' \times d} X_{d \times N}$. If a projection matrix A is not orthogonal, it causes significant distortion in the dataset. Thus, we should consider the orthogonality of matrix A, when we design the matrix A.

We introduce the random projection approach into the proposed method to improve the computational efficiency. The recent literature shows that a group among a set of high-dimensional clusters lies on a low-dimensional subspace in many real-world applications. In this case, the underlying hidden subspace can be retrieved by solving a sparse optimization problem, which encourages selecting nearby points that approximately span a low dimensional affine subspace. Most previous approaches focus on finding the best low-dimensional representation of the data for which a single feature representation is sufficient for the clustering task [23, 24]. Our approach takes into account clustering of high-dimensional complex data. It has more than a single subspace due to the extensive attribute variations over the feature space. We model the complex data with multiple feature representations by incorporating binary random projection.

Random projection matrix Matrix A of Eq. (7) is generally called a random projection matrix. The choice of the random projection matrix is one of the key points of interest. According to [22], elements of A are Gaussian distributed (GRP). Achiloptas [14] has recently shown that the Gaussian distribution can be replaced by a simpler distribution such as a sparse matrix (SRP). In this paper, we propose the binary random projection (BRP) matrix, where the elements a_{ij} consist of zero or one value, as delineated in Eq. (8).

$$a_{ij} = \begin{cases} 1 & \text{with probability } \alpha \\ 0 & \text{with probability } 1 - \alpha \end{cases} \tag{8}$$

Given that a set of features from data is λ-sparse, we need at least λ-independent canonical bases to represent the features lying on the λ-dimensional subspace. Because BRP encourages the projection matrix to be λ-independent, the data are almost preserved to the extent of λ-dimensions even after the projection. If the projection vectors are randomly chosen regardless of the independence, it can be insufficient to accurately span the underlying subspace because of the rank deficiency of the projection matrix. This shows that SRP without imposing the independent constraint gives rise to representation errors when projecting onto a subspace.

Algorithm 1 Proposed accelerated kmeans algorithm

X : input data, K : the number of clusters, $\hat{C}$: final centers of clusters
R : the binary membership indicator, C : the center of clusters
A : proposed random matrix
T : the number of ensemble models
$\widetilde{X}$: sampling data, $\widetilde{C}$: the center of clusters in single ensemble
$\widetilde{X}'$: sampling data in lower dimensional space
$\widetilde{C}'$: the center of clusters in lower dimensional space
N : the total number of sampling data
$\widetilde{N}$: the number of sampling data in single ensemble
$\hat{X}$: the refining sampling data from the first generation stage

procedure AcceleratedKMeans(X, K)
  for $m = 1 \rightarrow T$ **do**
    $\widetilde{X} =$ **Bootstrap**$(X, \widetilde{N})$
    Initialize A, $\widetilde{X}'$, $\widetilde{C}'$ and R
    while the stop condition is satisfied **do**
      if the iteration is not first **then**
        $A_{new} =$ **GetBRP**()
        if A_{new} reduce the error than A **then**
          $A = A_{new}$, $\widetilde{X}' = A\widetilde{X}$, $\widetilde{C}' = A\widetilde{C}$
        end if
      end if
      $R =$ **MatchDataAndCluster**$(\widetilde{X}', \widetilde{C}')$
      $\widetilde{C}' =$ **UpdateCluster**$(\widetilde{X}', R)$
    end while
    for $j \rightarrow K$ **do**
      $\hat{x}_j^m = \frac{\sum r_{ij} \widetilde{x}_i}{\sum r_{ij}}$
      Add $\hat{x}_j^m$ to $\hat{X}$
    end for
  end for
  $\hat{C} =$**kmeans**$(\hat{X}$,K)
end procedure

Distance bound and triangle inequality Factors that can cause kmeans to be slow include processing large amounts of data, computing many point-center distances, and requiring many iterations to converge. A primary strategy of accelerating kmeans is applying geometric knowledge to avoid computing redundant distance. For example, Elkan kmeans [8] employs the triangle inequality to avoid many distance computations. This method efficiently updates the upper and lower bounds of point-center distances to avoid unnecessary distance calculations. The proposed method projects high dimensional data onto the lower dimensional subspace using the BRP matrix. It may be determined that each data of lower dimensional subspace cannot guarantee exact geometric information between data. However, our method is approximately preserved by the Johnson-Lindenstrauss lemma [25]: if points in a vector space are projected

on to a randomly selected subspace of a suitably high dimension, then the distances between the points are approximately preserved. Our algorithm thus can impose a distance bound characteristic to reduce the computational complexity.

2.2 Bootstrap sampling and ensemble learning

Our approach adopts an ensemble learning model because of statistical reasons and large volumes of data. The statistical reason is that combining the outputs of several models by averaging may reduce the risk of an unfortunate feature selection. Learning with such a vast amount of data is usually not practical. We therefore use a partitioning method that separates all dataset into several small subsets. Also, we learn each models with disjoint subdata. By adapting the ensemble approach to our work, we obtain diversity of models and decrease the correlation between ensemble models. The results of Eq. (5) thus are more stable and comparable to the results of Eq. (4).

To reduce the risk of an unfortunate feature selection, the diversity of each ensemble models should be guaranteed. The diversity of ensemble models can be generally achieved in two ways. The most popular way is to employ a different dataset in each model, and the other is to use different learning algorithms. We choose the first strategy, and the bootstrap is used for pre-processing of feature selection. We empirically show that our method produces sufficient diversity, even when the number of ensembles is limited.

As multiple candidate clusters are combined, our algorithm considers the compatibility with variants of kmeans methods and efficiency of the execution time. Our method simply combines multiple candidate clusters using the conventional kmeans algorithm to guarantee fast convergence. Finally, it affords K clusters by minimizing errors in Eq. (2) using the refined products of the generation stage, as mentioned above.

2.3 Time complexity

The time complexity for three accelerated algorithms is described in Table 1. We use lower case letters n, d, and k instead of N, D, and K for the readability. The total time is the summation of elapsed time in each kmeans iteration without the initialization step. The proposed total time in Table 1, the first part of the or statement represents executed total time without geometric knowledge to avoid computing the redundant distance, while the second part indicates total time with geometric knowledge. Our algorithm shows the highest simplicity, since the $\alpha\beta * T$ term is much smaller than 1.

The underline notation comes from Elkans kmeans, and $\underline{n}$ denotes the number of data, which need to be updated in every distance calculation. $\widetilde{n}$ indicates the number of reduced data using bootstrap, and d' denotes the number of reduced features. Let α denote $\widetilde{n}/n$, β denote d'/d, and γ denote Tk/n, which is a ratio of the number of data used in the generation and combination stage. As will be seen in Sec. 3, these values are much smaller than 1.

Table 1: The asymptotic total time for each examined algorithm.

	total time
kmeans	$O(ndk) * iter$
Elkan	$O(\underline{n}dk + dk^2) * iter$
proposed	$\alpha\beta * T * O(ndk) * iter$ or $T * O(\widetilde{\underline{n}}d'k + d'k^2) * iter$

3 Experiments

We extensively evaluate the performances on various datasets. Synthetic and real datasets are used for the explicit clustering evaluation in terms of accuracy and elapsed time. We also show offline and online training efficiency for building a vocabulary tree [1] and incremental vocabulary tree [5]. As mentioned earlier, the incremental vocabulary tree does not need heavy clustering in the offline training process due to the distributed online process. Thus, strict elapsed time is more important for online clustering.

Our algorithm has three parameters: α, β, and T. The default values of parameters are determined through several experiments. The values of α and β are set as [0.1, 0.3] and T is selected as [5, 7]. During the experiments, these values are preserved.

3.1 Data sets

Synthetic data We use synthetic datasets based on a standard cluster model using a multi-variated normal distribution. Synthetic data generation tool is available on the website[1]. This generator gives two datasets, Gaussian and ellipse cluster data. To evaluate the performance of the algorithm over various numbers of data (N), dimensions of data (D), and numbers of groups of data (K), we generated datasets having N = 100K, K $\in \{3, 5, 10, 100, 500\}$, and D $\in \{8, 32, 128\}$.

Tiny Images We use the CIFAR-10 dataset, which is composed of labelled subsets of 80 million tiny images [26]. CIFAR-10 consists of 10 categories and it contains 6000 images for each category. Each image is represented as GIST feature of dimension 384.

RGBD Images We collect about object images from the RGBD dataset [27]. RGBD images are randomly sampled with category information. We use a 384-dimensional GIST feature to represent each image.

Caltech101 It contains images of 101 categories of objects, gathered from the internet. This dataset is mainly used to benchmark classification methods. We extract dense multi-scale SIFT feature for each image, and randomly sample 1M features to form this dataset.

[1] http://personalpages.manchester.ac.uk/mbs/Julia.Handl/generators.html.

UKbench This dataset is from the Recognition Benchmark introduced in [1]. It consists of 10200 images split into four-image groups, with each of the same scene/object taken at different viewpoints. The features of the dataset and ground truth are publicly available.

Indoor/Outdoor One indoor and two outdoor datasets are used to demonstrate the efficiency of our approach. Indoor images are captured by a mobile robot that moves twice along a similar path in the building. This dataset has 5890 images. SURF features are used to represent each image. Outdoor datasets are captured by a moving vehicle. We refer to the datasets as small and large outdoor datasets for the sake of convenient reference. The vehicle moves twice along the same path in the small outdoor dataset. In the large outdoor dataset, the vehicle travels about 13km while making many loops. This large dataset consists of 23812 images, and we use sub-sampled images for test.

3.2 Evaluation metric

We use three metrics to evaluate the performance of various clustering algorithms, elapsed time, the within-cluster sum of squared distortions (WCSSD), and the normalized mutual information (NMI) [28]. NMI is widely used for clustering evaluation, and it is a measurement of how close clustering results are to the latent classes. NMI requires the ground truth of cluster assignments X for points in the dataset. Given clustering results Y, NMI is defined by NMI(X,Y) $= \frac{MI(X,Y)}{\sqrt{H(X)H(Y)}}$, where MI(X,Y) is the mutual information of X and Y and $H(\cdot)$ is the entropy.

To tackle a massive amount of data, distributed computing and efficient learning need to be integrated into vision algorithms for large scale image classification and image indexing. We apply our method to visual codebook generation for bag-of-models based applications. In our experiments, the precision/recall and similarity matrix are used for image indexing and the evaluation of classified images follows [29]. Our results show the quality and efficiency of the codebooks with all other parameters fixed, except the codebook.

3.3 Clustering performance

We compare our proposed clustering algorithm with three variations: Lloyd kmeans algorithm, Athur kmeans algorithm, and Elkan kmeans algorithm. All algorithms are run on a 3.0GHz, 8GB desktop PC using a single thread, and are mainly implemented in C language with some routines implemented in Matlab. We use the public releases of Athur kmeans and Elkan kmeans. The time costs for initialization and clustering are included in the comparison.

The results in Fig. 1 and Fig. 2 are shown for various dimensions of data and various numbers of clusters, respectively. The proposed algorithm is faster than Lloyds algorithm. Our algorithm consistently outperforms the other variations of kmeans in high dimensional large datasets. Also, our approach performs best regardless of K. However, from the results of this work, the accuracy of clustering

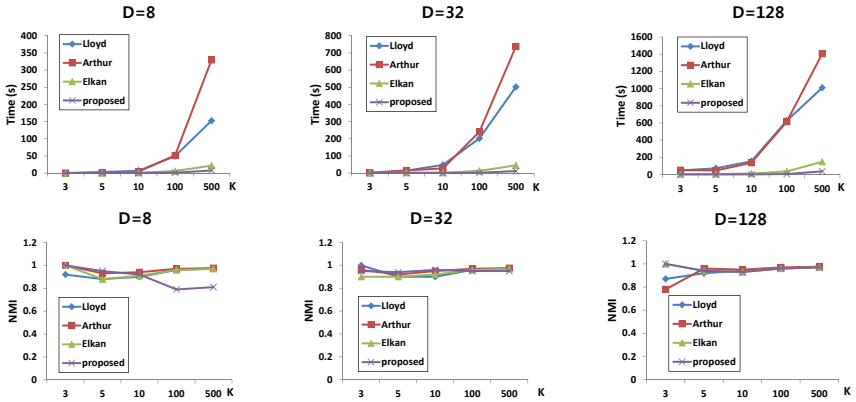


Fig. 1: Clustering performance in terms of elapsed time vs. the number of clusters and the clustering accuracy (N=100,000)

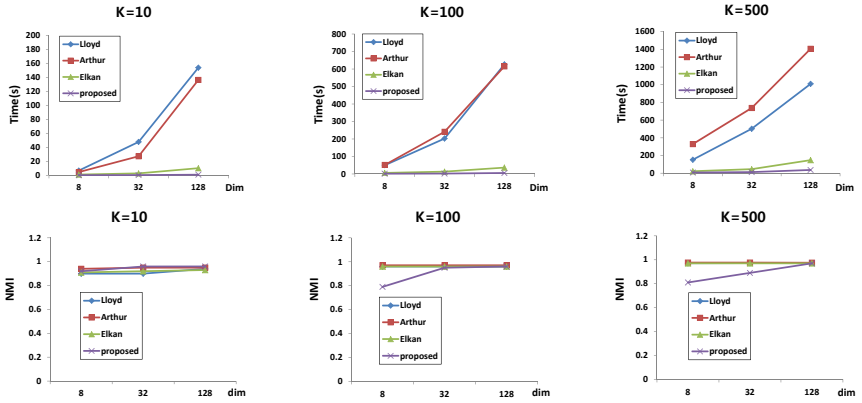
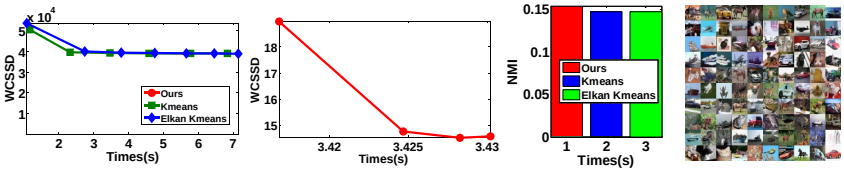


Fig. 2: Clustering performance in terms of elapsed time vs. the number of dimensions and the clustering accuracy (N=100,000)

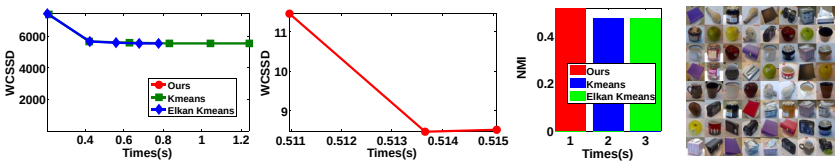
in low dimensional datasets is not maintained. Hecht-Nielsens theory [30] is not valid in low dimensional space, because a vector having random directions might not be close to orthogonal.

Our algorithm is also efficient for real datasets. We use CIFAR10 and RGBD image sub-datasets without depth. Fig. 3 and Fig. 4 show the clustering results in terms of WCSSD vs. time and NMI. As seen in these figures, the WCSSD of our algorithm is smaller than that of the earlier work and the NMI is similar. From this, we can see that our approach provides faster convergence with a small number of iterations.



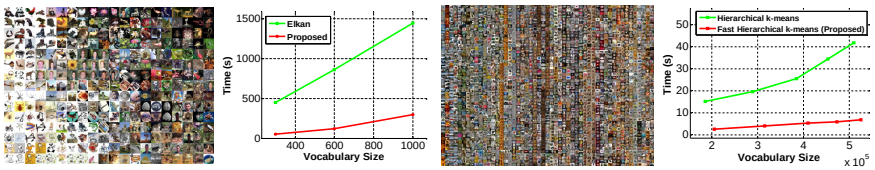
(a) WCSSD vs. time (b) final NMI

Fig. 3: Clustering performance of CIFAR-10



(a) WCSSD vs. time (b) final NMI

Fig. 4: Clustering performance of RGBD objects



(a) flat visual codebooks (b) hierachical visual codebooks

Fig. 5: The comparison of the clustering time with various vocabulary sizes. (a) is the result on the Caltech101. (b) is the result on the UKbench dataset.

4 Applications

4.1 Evaluation using object recognition

We compare the efficiency and quality of visual codebooks, which are respectively generated by flat and hierarchical clustering methods. A hierarchical clustering method such as HKM [1] is suitable for large data applications.

The classification and identification accuracy are similar and therefore we only present results in terms of elapsed time as increasing the size of visual words, namely vocabulary, in Fig. 5. We perform the experiments on the Caltech101 dataset, which contains 0.1M randomly sampled features. Following [29], we run the clustering algorithms used to build a visual codebook, and test only the codebook generation process in the image classification. Results of the Caltech101 dataset are obtained by 0.3K, 0.6K, and 1K codebooks, and a χ^2-SVM on top of 4×4 spatial histograms. From Fig. 5a, we see that for the same vocabulary size, our method is more efficient than the other approaches. However, the

accuracy of each algorithm is similar. For example, when we use 1K codebooks for clustering, the mAP of our approach is 0.641 and that of the other approach is 0.643.

In the experiment on the UKbench dataset, we use a subset of database images and 760K local features. We evaluate the clustering time and the performance of image retrieval with various vocabulary sizes from 200K to 500K. Fig. 5b shows that our method runs faster than the conventional approach with a similar mAP, about 0.75.

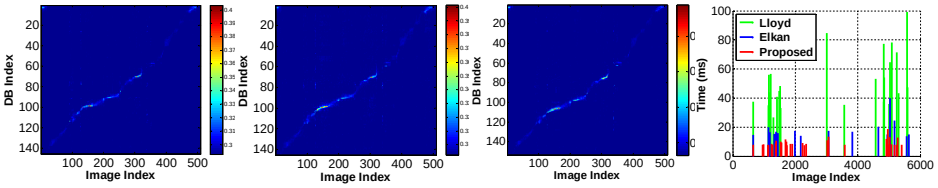
4.2 Evaluation using image indexing

The vocabulary tree is widely used in vision based localization in mobile devices and the robot society [4–6]. The problem with the bag-of-words model lies in that a codebook built by each single dataset is insufficient to represent unseen images. Recently, the incremental vocabulary tree was presented for adapting to dynamic environments and removing offline training [4, 5]. In this experiment, we use the incremental codebooks, as mentioned in AVT [5]. We demonstrate the accuracy of image indexing and the execution time for updating incremental vocabulary trees. Our visual search system follows [6], and we do qualitative evaluation by image to image matching. The clustering part of AVT is replaced by the state of the art kmeans and the proposed algorithms. We evaluate the online clustering process of modified AVT and show the performance of image matching for indoor and outdoor environments.

Three figures (from left to right) in Fig. 6 and Fig. 7 show the image similarity matrix that represents the similarity scores between training and test images. From this matrix, we can calculate the localization accuracy on each dataset, and diagonal elements show loop detection and the right-top part indicates loop closure. However, three similarity matrixes in each dataset have similar values and show similar patterns. These results mean that our clustering method runs well without losing accuracy. The last figure in Fig. 6 and Fig. 7 shows the execution time for the clustering process. This process runs when a test image is inserted. If a test image is an unseen one, features of the image are inserted into the incremental vocabulary tree, and all histogram of images are updated. When adaptation of the incremental vocabulary tree occurs, the graph of the last figure has a value. As we can see in figure (d), the elapsed time of our method is smaller and the number of executions is greater.

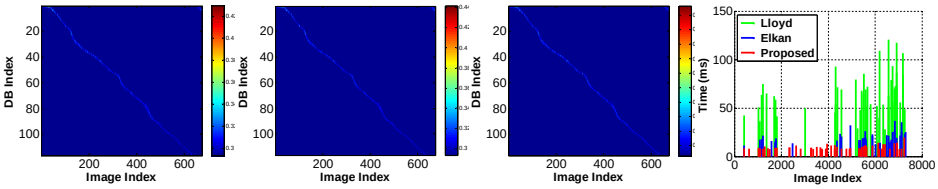
In Fig. 8, we use the precision-recall curve instead of a similarity matrix. The tendency of both results is similar to that seen above.

Two images (from left) of each row in Fig. 9 are connected with each dataset: images of the first row belong to the indoor dataset, the second belong to the small outdoor dataset, and the third belong to the large outdoor dataset. Images of the third column show total localization results. There are three circles: green, the robot position; yellow, added image position; and red, a matched scene position. In order to prevent confusion, we should mention that the trajectories of the real position (green line) are slightly rotated to view the results clearly and easily.



(a) kmeans (b) elkan kmeans (c) ours (d)

Fig. 6: The performance comparison on the indoor dataset. (a), (b), (c) are the image similarity matrix of the conventional approach and the proposed algorithm. (d) is the elapsed time of clustering.



(a) kmeans (b) elkan kmeans (c) ours (d)

Fig. 7: The performance comparison on the small outdoor dataset. (a), (b), (c) are the image similarity matrix of the conventional approach and the proposed algorithm. (d) is the elapsed time of clustering.

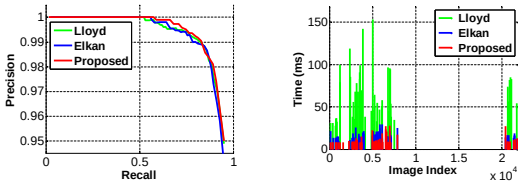


Fig. 8: The performance comparison on the large outdoor dataset. (a) is the precision-recall of the loop detection. (b) is the elapsed time of clustering.

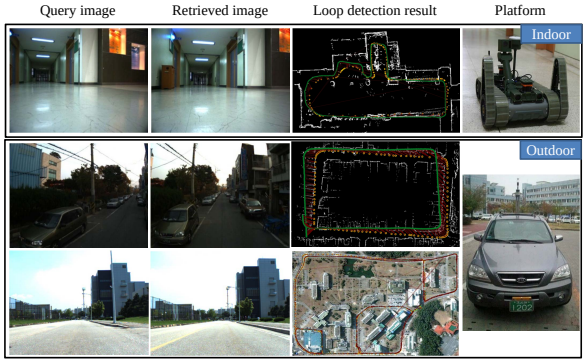


Fig. 9: Example images and the result of the loop detection in each dataset. Images of the first row belong to the indoor dataset, and images of the second and third rows belong to outdoor datasets.

5 Conclusions

In this paper, we have introduced an accelerated kmeans clustering algorithm that uses binary random projection. The clustering problem is formulated as a feature selection and solved by minimization of distance errors between original data and refined data. The proposed method enables efficient clustering of high dimensional large data. Our algorithm shows better performance on the simulated datasets and real datasets than conventional approaches. We demonstrate that our accelerated algorithm is applicable to an incremental vocabulary tree for object recognition and image indexing.

Acknowledgement We would like to thank Greg Hamerly and Yudeog Han for their support. This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(No. 2010-0028680).

References

1. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: International Conference on Computer Vision and Pattern Recognition. (2006) 2161–2168
2. Tsai, S.S., Chen, D., Takacs, G., Chandrasekhar, V., Singh, J.P., Girod, B.: Location coding for mobile image retrieval. In: Proceedings of the 5th International ICST Mobile Multimedia Communications Conference. (2009)
3. Straub, J., Hilsenbeck, S., Schroth, G., Huitl, R., Möller, A., Steinbach, E.: Fast relocalization for visual odometry using binary features. In: IEEE International Conference on Image Processing (ICIP), Melbourne, Australia (2013)
4. Nicosevici, T., Garcia, R.: Automatic visual bag-of-words for online robot navigation and mapping. Transactions on Robotics (2012) 1–13
5. Yeh, T., Lee, J.J., Darrell, T.: Adaptive vocabulary forests br dynamic indexing and category learning. In: Proceedings of the International Conference on Computer Vision. (2007) 1–8
6. Kim, J., Park, C., Kweon, I.S.: Vision-based navigation with efficient scene recognition. In: Journal of Intelligent Service Robotics. Volume 4. (2011) 191–202
7. Lloyd, S.P.: Least squares quantization in pcm. Transactions on Information Theory **28** (1982) 129–137
8. Elkan, C.: Using the triangle inequality to accelerate k-means. In: International Conference on Machine Learning. (2003) 147–153
9. Bradley, P.S., Fayyad, U.M.: Refining initial points for k-means clustering. In: International Conference on Machine Learning. (1998)
10. Arthur, D., Vassilvitskii, S.: k-means++: the advantages of careful seeding. In: ACM-SIAM symposium on Discrete algorithms. (2007)
11. Parsons, L., Haque, E., Liu, H.: Subspace clustering for high dimensional data: a review. In: ACM SIGKDD Explorarions Newsletter. Volume 6. (2004) 90–105
12. Khalilian M., Mustapha N., N.S.M., MD., A.M.: A novel k-means based clustering algorithm for high dimensional data sets. In: Internaional MultiConference of Engineers and Computer Scientists. (2010) 17–19
13. Moise, G., Sander: Finding non-redundant, statistically significant regions in high dimensional data: a novel approach to projected and subspace clustering. In: international conference on Knowledge discovery and data mining. (2008)

14. Achlioptas, Dimitris: Database-friendly random projections. In: ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems. (2001) 274–281
15. Ding, C., He, X., Zha, H., Simon, H.D.: Adaptive dimension reduction for clustering high dimensional data. In: International Conference on Data Mining. (2002) 147–154
16. Hinneburg, A., Keim, D.A.: Optimal grid-clustering: Towards breaking the curse of dimensionality in high-dimensional clustering. In: International Conference on Very Large Data Bases. (1999)
17. Bingham, E., Mannila, H.: Random projection in dimensionality reduction: applications to image and text data. In: international conference on Knowledge discovery and data mining. (2001)
18. Ana L. N. Fred, A.K.J.: Combining multiple clusterings using evidence accumulation. Transaction Pattern Analaysis Machine Intelligence **27(6)** (2005)
19. R.Polikar: Ensemble based systems in decision making. In: Circuits and systems magazine. Volume 6(3). (2006) 21–45
20. Fern, X.Z., Brodley, C.E.: Random projection for high dimensional data clustering: A cluster ensemble approach. In: International Conference on Machine Learning. (2003) 186–193
21. Kohavi, R., John, G.H.: Wrappers for feature subset selection. In: Artificial Intelligence. Volume 97. (1997) 273–324
22. Indyk, P., Motwani, R.: Approximate nearest neighbors: towards removing the curse of dimensionality. In: ACM symposium on Theory of computing. (1998) 604–613
23. Elhamifar, E., Vidal., R.: Sparse subspace clustering. In: International Conference on Computer Vision and Pattern Recognition. (2009)
24. Ehsan Elhamifar, R.V.: Sparse manifold clustering and embedding. In: Neural Information Processing Systems. (2011) 55–63
25. W.B.Johnson, J.Lindenstrauss: Extensions of lipschitz mapping into hilbert space. In: International conference in modern analysis and probability. Volume 26. (1984) 90–105
26. Krizhevsky, A.: Learning multiple layers of features from tiny images. Technical report (2009)
27. Kevin Lai, Liefeng Bo, X.R., Fox, D.: A large-scale hierarchical multi-view rgb-d object dataset. In: International Conference on Robotics and Automation. (2012) 1817–1824
28. Strehl, A., Ghosh, J.: Cluster ensembles – a knowledge reuse framework for combining multiple partitions. The Journal of Machine Learning Research **3** (2003) 583–617
29. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: International Conference on Computer Vision and Pattern Recognition. (2006) 2169–2178
30. R., H.N.: Context vectors: general purpose approximate meaning representations self-organized from raw data. Computational Intelligence: Imitating Life (1994) 43–56