Dimensionality Reduction for Distance Based Video Clustering

Jayaraman J. Thiagarajan, Karthikeyan N. Ramamurthy, Andreas Spanias

Dimensionality Reduction for Distance Based Video Clustering

Jayaraman J. Thiagarajan, Karthikeyan N. Ramamurthy, and Andreas Spanias

SenSIP Center, School of ECEE, Arizona State University,
Tempe, AZ 85287-5706, USA.
{jjayaram,knatesan,spanias}@asu.edu

Abstract. Clustering of video sequences is essential in order to perform video summarization. Because of the high spatial and temporal dimensions of the video data, dimensionality reduction becomes imperative before performing Euclidean distance based clustering. In this paper, we present non-adaptive dimensionality reduction approaches using random projections on the video data. Assuming the data to be a realization from a mixture of Gaussian distributions allows for further reduction in dimensionality using random projections. The performance and computational complexity of the K-means and the K-hyperline clustering algorithms are evaluated with the reduced dimensional data. Results show that random projections with an assumption of Gaussian mixtures provides the smallest number of dimensions, which leads to very low computational complexity in clustering.

Key words: Clustering, Random projections, Gaussian mixtures, Video summarization.

1 Introduction

Classification of data is an important problem in machine learning, where data sets are separated into several disjoint classes based on predefined criteria. The predefined criteria, referred as the hypothesis, can be supplied by the user or learned by the machine itself from classes of labeled training samples. In supervised learning, hypotheses of multiple classes are learned from a set of labeled training data for each class [1]. Clustering is a more general problem in machine learning where the observed unlabeled data need to be grouped into different clusters. A cluster is a group of similar data where the similarity is quantified based on a well-defined measure. A useful similarity measure for clustering is the Euclidean distance measure and clustering based on Euclidean distance is an NP-hard problem [2].

Clustering of video frames is more than just a generalization of clustering of images. This is because the video frames that convey meaning as a group are both statistically and semantically related. One of the popular approaches to video clustering involves extracting the keyframes by shot boundary detection and clustering the keyframes together to derive a semantic interpretation [3].

However, it is important to understand that extraction of the keyframes by detecting the shot boundaries itself is a fundamental clustering problem which we address in this paper. Video frames of a single shot have similar background structure and they can be clustered together using color histograms or distance measures.

In this paper, we address the problem of clustering high dimensional long video sequences. In general, this kind of video clustering involves grouping the frames with similar background together for the purpose of extracting keyframes. Using the fact that the video frames that have similar backgrounds are close together in terms of the Euclidean distance measure (l_2 norm), we perform distance based clustering. It is important to clarify the notion of background in this problem. Background is the region in a frame that remains relatively motionless. Even if some objects in the foreground are relatively motionless they can be treated as background. The very high spatial and temporal dimensionality of the video data makes l_2 norm based clustering intractable in the absence of tremendous computational power. Therefore, it becomes essential to reduce the dimensionality of the video data in order to perform clustering with low complexity. This problem is highly significant in scenarios where fast summarization of video needs to be performed at a reduced computational cost. In this paper, we propose a framework for dimensionality reduction to cluster video frames having similar background structure. The framework is based on non-adaptive dimensionality reduction using the theory of random projections [4] and assumption of Gaussian mixture (GM) models for data.

2 K-Means and K-Hyperline clustering

The K-means clustering problem seeks to cluster the T data samples into K clusters by minimizing the sum of intracluster distances across all clusters. It converges to a locally optimal solution closest to the initial values. This is a 2-step alternating minimization problem where the samples are associated to the cluster centroids in the first step and the centroids are recalculated in the second step using the associated member samples. The member sample is associated to a cluster centroid that is closest in terms of the Euclidean distance measure. The centroid that minimizes the sum of distances to all its member samples is computed by solving,

$$\bar{\mathbf{x}}_j = \min_{\mathbf{r}} \sum_{i \in \Lambda_j} \|\mathbf{x}_i - \mathbf{r}\|^2, \tag{1}$$

where $\bar{\mathbf{x}}_j$ is the cluster centroid and Λ_j is the index set containing the memberships of the j^{th} cluster. The solution obtained for $\bar{\mathbf{x}}_j$ is the mean of member samples.

K-hyperline clustering seeks to compute a rank-1 subspace using Singular Value Decomposition (SVD) for each cluster that minimizes the sum of distance of the member data to the subspace [5]. K-hyperline clustering is more accurate and general than K-means clustering in that the minimization problem yields

lesser sum of distances than the K-means clustering and it can easily generalize to higher rank subspaces. The association rule for the member sample for the nearest rank-1 subspace is based on a maximum correlation measure. In K-hyperline clustering the rank-1 subspace is of unit norm, whereas no such constraint is imposed in K-means clustering on the cluster centroid.

3 Random Projections

Consider a high dimensional data matrix, $\mathbf{X}$, with dimensions $M \times T$, where each column represents a single data observation. In order to project this onto a random low dimensional space, we define a matrix $\mathbf{R}$ of dimensions $M \times N$ with $N < M$, whose entries are chosen independently from the standard normal distribution $\mathcal{N}(0, 1)$. The Random Projection (RP) of the data vectors is,

$$\mathbf{Y} = \frac{1}{\sqrt{N}}\mathbf{R}^T\mathbf{X}. \tag{2}$$

RP reduces the dimensionality of the data from M to N while approximately preserving pair-wise distances with high probability [4]. This is formalized by the *Johnson-Lindenstrauss* (JL) lemma, which states that for a large enough N ($N \geq C\frac{\ln T}{\epsilon^2}$) (3) holds with high probability.

$$(1-\epsilon)\left\|\mathbf{y}_i\right\|^2 \leq \frac{1}{N}\|\mathbf{R}^T\mathbf{x}_i\|^2 \leq (1+\epsilon)\left\|\mathbf{y}_i\right\|^2, \tag{3}$$

where $\mathbf{y}_i$ and $\mathbf{x}_i$ represent the columns of the matrices $\mathbf{Y}$ and $\mathbf{X}$ respectively and $0 < \epsilon < 1$. It is important to note that the JL lemma does not depend on the actual dimensionality of the data and depends only on the number of data vectors. The K-means clustering defines the cluster centroid as the mean of data vectors in a cluster. It is easy to observe from JL lemma that the distances between the cluster centroid and the data vectors will be approximately preserved even after random projections. Hence we can use the JL lemma to reduce the dimensionality of the data matrix for use in K-means clustering.

3.1 Computation of SVD using Random Projections

The computation of SVD can also be performed using the reduced dimensional matrix from random projections [4]. It can be shown that for the same low rank approximations of $\mathbf{Y}$ and $\mathbf{X}$, the Frobenius norms will be approximately preserved with high probability. This can be mathematically expressed as,

$$\sum_{i=1}^{s}\lambda_i^2 \geq (1-\epsilon)\sum_{i=1}^{s}\sigma_i^2, \tag{4}$$

where λ_i and σ_i are the singular values of $\mathbf{Y}$ and $\mathbf{X}$ respectively and s is the desired rank of the approximation [4]. In particular, considering a rank-1 approximation, it can be shown that, with high probability [4, 6]

$$(1-\epsilon)\sigma_1^2 \leq \lambda_1^2 \leq \sigma_1^2. \tag{5}$$

The existence of such upper and lower bounds motivates the use of K-hyperline clustering on $\mathbf{Y}$ instead of $\mathbf{X}$.

4 Gaussian Mixture Models for Clustering

Statistical clustering algorithms assume that the data is a realization from a mixture of probability distributions. In the case of mixture of K arbitrary distributions, the overall probability density is given by,

$$f = \sum_{i=1}^{K} w_i f_i \quad \text{s.t.} \quad w_i \geq 0 \quad \text{and} \quad \sum_{i=1}^{K} w_i = 1, \tag{6}$$

where f_i is the probability distribution function (pdf) and w_i is the non-negative weight of the i^{th} distribution.

The best SVD subspace for a spherical Gaussian distribution is any subspace through its mean [2]. More importantly, the best K-dimensional SVD subspace for a mixture of K Gaussians whose covariance is a scalar multiple of identity, contains the span of the means of the component distributions. This can also be extended to a mixture of arbitrary distributions. In general, we do not have the exact statistics of a GM and we have only the samples of realizations. The covariance matrix is also not a scalar multiple of identity. Even under these conditions, it has been proved that the SVD subspace of the sample matrix is not far from the subspace spanned by the actual component means [2].

4.1 Separation Between Spherical Gaussians

Two spherical Gaussians $\mathcal{N}(\mu_1, \sigma_1^2\mathbf{I})$ and $\mathcal{N}(\mu_2, \sigma_2^2\mathbf{I})$ are considered to be c-separated if $\|\mu_1 - \mu_2\|_2^2 \geq c^2 M \max(\sigma_1^2, \sigma_2^2)$, where M is the dimension of the Gaussian [6]. A 2-separated mixture corresponds to almost completely separated Gaussians, whereas a 1- or 1/2-separated mixture contains Gaussians which overlap significantly. By projecting the Gaussian mixtures on to a K-dimensional subspace spanned by the means of K Gaussians, we are equivalently projecting the Gaussian mixtures onto their best rank-K SVD subspace. This preserves the distance between the means (intercluster distance), whereas the intracluster distance reduces drastically [2]. Therefore, the separation between the Gaussians in the mixture increases and the clustering performance improves. Similar results can also be shown for mixtures of Gaussians with arbitrary covariances [2].

5 Proposed Clustering Framework

In this paper, we use both the K-means and the K-hyperline clustering algorithms for clustering the video data. We consider four different approaches: a) basic K-means/K-hyperline clustering on the high dimensional data, b) reducing the dimensions of the data using random projections prior to clustering, c)

reducing the dimensions of the data assuming it as a mixture of Gaussians prior to clustering and d) reducing the dimensions, first using RP and then under the mixture of Gaussians assumption, prior to clustering. Both centroid and left singular vector of a group of video frames retrieve the background information effectively. This motivates the use of both K-means/K-hyperline clustering in our approaches. We will assume that we have K clusters of the T video frames and we vectorize each video frame into a M dimensional vector thereby generating the $M \times T$ matrix $\mathbf{X}$. The K index sets of the clusters are given by $\{\Lambda_i\}_{i=1}^{K}$ and $T_i = |\Lambda_i|$. In the remaining part of this section, we describe the different approaches for clustering.

5.1 Random Projection based Clustering

The RP method can be used to reduce the dimensionality of the data matrix $\mathbf{X}$ according to (2), preserving the length of the data vectors, pairwise distances and angles with high probability. We have also seen in Section 3 that the centroid and SVD of a set of data vectors are approximately preserved with a high probability, given a sufficiently large number of measurements N.

Assuming that $K = 1$, the centroid and the first singular vector of $\mathbf{Y}$ are approximately equal to that of $\mathbf{X}$ for a sufficiently large N. If $K > 1$, the centroid and first singular vector of each cluster will still be approximately preserved because $T_i < T$. Therefore, the RP method will be useful regardless of the number of clusters, provided we choose N based on the assumption of single cluster. The linear increase in the number of data vectors T will not change N significantly because $N \propto \ln T$. Therefore, in order to perform RP based clustering, we use either the K-means or the K-hyperline clustering algorithm on the reduced dimensional data.

5.2 Gaussian Mixture based Clustering

We know from Section 4 that the rank-K SVD subspace of the sample matrix is not far from the space spanned by the K component means even when the Gaussians are not spherical. In this approach for clustering, we assume that $\mathbf{X}$ contains realizations from a mixture of K Gaussians, not necessarily spherical, where each Gaussian represents a cluster. We compute the best rank-K subspace of $\mathbf{X}$ using SVD and denote the basis vectors of the rank-K subspace (first K left singular vectors of $\mathbf{X}$) by $\mathbf{U}_K$. The projection to the rank-K subspace is given by,

$$\mathbf{W} = \mathbf{U}_K^T \mathbf{X}, \tag{7}$$

where $\mathbf{W}$ contains the K dimensional data vectors after projection. Because of the reasoning provided in Section 4.1, the clusters in the K dimensional space are more separated than the clusters in the M dimensional space. Therefore, we perform K-means or K-hyperline clustering on $\mathbf{W}$ and identify the index sets of the clusters.

Goal: To perform clustering of high-dimensional long video sequences using the approach given in Section 5.3

Variables

High-dimensional data matrix, $\mathbf{X}$ of size $M \times T$.
Intermediate data matrix after RP, $\mathbf{Y}$ of size $N \times T$.
Final data matrix used for clustering, $\mathbf{Z}$ of size $K \times T$.
Initial number of clusters, J.
Actual number of clusters, K.
Cluster centroid matrix (J Clusters), $\mathbf{B}$ of size $K \times J$.
Cluster centroid matrix (K Clusters), $\mathbf{A}$ of size $K \times K$.
Index set for the i^{th} cluster, $\mathbf{\Lambda}_i$.
Gaussian i.i.d. random matrix, $\mathbf{R}$ of size $M \times N$.

Algorithm

1. Compute the RP, $\mathbf{Y} = (1/\sqrt{N})\mathbf{R}^T\mathbf{X}$.
2. Compute rank-K SVD, $[\mathbf{U}_K, \mathbf{S}_K, \mathbf{V}_K] = \text{SVD}(\mathbf{Y}, K)$.
3. Project on to the K-dimensional SVD space, $\mathbf{Z} = \mathbf{U}_K^T\mathbf{Y}$.
4. Initialize $\mathbf{B}$ with J randomly chosen columns of $\mathbf{Z}$.
5. Perform K-means/K-hyperline clustering for J clusters.
6. Using greedy combinations of columns of $\mathbf{B}$, create $\mathbf{A}$.
7. Perform K-means/K-hyperline clustering for K clusters using $\mathbf{A}$ as initial centroids.
8. Using the index sets $\mathbf{\Lambda}_i$ obtained from clustering, identify the keyframes.

Table 1. Algorithm to cluster video data and identify keyframes.

5.3 Random Projection and Gaussian Mixture based Clustering

Similar to the previous case, in this approach we assume $\mathbf{X}$ to be a set of T realizations of K Gaussians. Furthermore, as RP approximately preserves the pairwise distances of the samples, realizations from a mixture of K Gaussians in M dimensions preserve their structure in N dimensions. This fact is used to further reduce the computational complexity of the framework.

In this approach, we first project $\mathbf{X}$ to obtain $\mathbf{Y}$ according to (2). The elements of $\mathbf{Y}$ are treated as realizations of a mixture of K Gaussians in N dimensions. From the arguments in Section 5.2, we project $\mathbf{Y}$ onto a K dimensional SVD subspace to obtain $\mathbf{Z}$. K-means or K-hyperline clustering can be performed on $\mathbf{Z}$ to identify the index sets of the clusters. Note that in this case, we first perform an initial level of dimensionality reduction using RP, which aids in a faster computation of the SVD subspace. In the second stage, we reduce the dimensionality further using the GM assumption. Hence, this approach combines the advantages of both the previous approaches in terms of a much reduced computational complexity due to RP and improved clustering performance along with further reduction in computational complexity due to the assumption of Gaussian mixtures. The outline of the algorithm to perform video clustering and identify the keyframes is shown in Table 1.

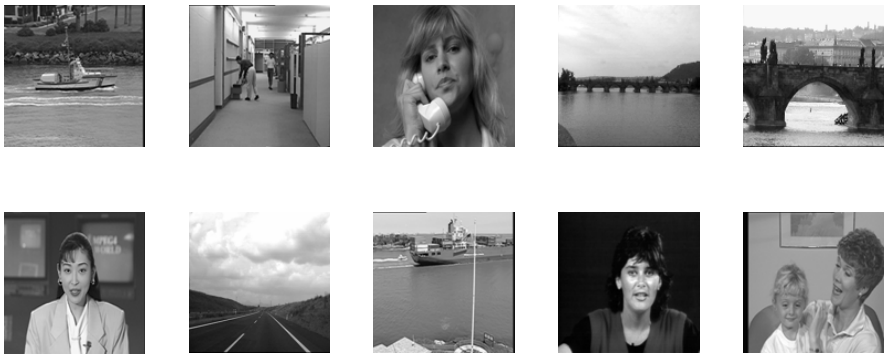

Fig. 1. Keyframes obtained by clustering the test data using the algorithm in Table 1.

To improve the clustering performance, we adopt a two stage approach to clustering. This reduces the possibility of the clustering algorithm being stuck in a local minima. Initially, we solve the clustering problem for a number of clusters J that is larger than the actual number K. Then, we greedily combine the columns of the cluster centroid matrix $\mathbf{B}$ in order to obtain the matrix $\mathbf{A}$ (step 6 of Table 1). In this greedy combination method we first choose the two most similar vectors of $\mathbf{B}$ and combine them. We repeat this procedure for two columns at a time until we are left with only K columns. In the K-means method two most similar vectors are the ones that have the minimum pairwise Euclidean distance and the combined vector is the mean of the two. In K-hyperline clustering two most similar vectors are the ones that have the maximum correlation and the combined vector is principal left singular vector of the matrix of the two vectors.

6 Experimental Results

The video sequences in QCIF format, used for evaluating the performance of the algorithms, were obtained from [7] and the spatial resolution was changed to 128×128. The first test data set was generated by stitching 10 different video sequences and it contains 1900 frames in total. The second test data set has a total of 550 frames obtained from 3 different video sequences. The initial number of clusters J is set to 3 times the actual number of clusters K. For the first data set $K = 10$ and for the second data set $K = 3$. The keyframes are identified using all the four approaches for both K-means and K-hyperline clustering. The keyframes obtained for the first data set are shown in Figure 1. The keyframes identified are similar with the all the four approaches and we also obtain 100% clustering performance.

Approach	Running time(s)		Number of
	K-means	Hyperline	Dimensions
Basic	696.51/37.69	774.23/102.22	16384/16384
RP	33.87/7.59	32.17/10.15	400/400
GM	-/10.18	-/10.30	10/3
RP and GM	29.41/7.36	26.84/7.45	10/3

Table 2. Comparison of running time for the different clustering approaches in MATLAB. Wherever applicable, the running times include dimensionality reduction phase also. The results for the first and the second data sets are separated by a slash (/). The third approach could not run for the first data set owing to memory issues because of high dimensionality.

The running times for the different approaches in MATLAB (version R2007b) to cluster the test data are listed in Table 2. It can be seen that the approach based on RP and GM is of least computational complexity. The running time for the K-means and the K-hyperline clustering algorithm are close to each other except for the case of the basic approach, which however is not the choice when clustering high dimensional data.

7 Conclusions

In this paper, we proposed different approaches for dimensionality reduction based on random projections in order to cluster video data. These approaches provide a practical solution to clustering video frames with similar background for fast video summarization. Incorporation of outlier rejection and compensating for global motion between the video frames are possible extensions to the proposed dimensionality reduction approaches for robust clustering.

References

1. Alpaydin, E.: Introduction to Machine Learning. Adaptive Computation and Machine Learning, The MIT Press (2004)
2. Kannan, R., Vempala, S.S.: Spectral Algorithms. Foundations and Trends in Theoretical Computer Science, Now Publishers Inc (2009).
3. Vailaya, A., Jain, A.K., Zhang, H.: Video Clustering. Technical report, Michigan State University (1996).
4. Vempala, S.S.: The Random Projection Method. Series in Discrete Mathematics and Theoretical Computer Science, American Mathematical Society (2004)
5. He, Z., Cichoki, A., Li, Y., Xie, S., S. Sanei: K-hyperline clustering learning for sparse component analysis. Signal Processing. 89, 1011–1022 (2009).
6. Dasgupta, S.: Learning Mixtures of Gaussians. In: Proceedings of the 40th Annual Symposium on Foundations of Computer Science, pp.634–644. Washington, DC, USA (1999).
7. YUV Video Sequences, `http://trace.eas.asu.edu/yuv/index.html`.