

CONTENT BASED VIDEO CATEGORIZATION USING RELATIONAL CLUSTERING WITH LOCAL SCALE PARAMETER

Mohamed Maher Ben Ismail and OuiemBchir

Computer Science Department, College of Computer and Information Sciences,
King Saud University, Riyadh, KSA

ABSTRACT

This paper introduces a novel approach for efficient video categorization. It relies on two main components. The first one is a new relational clustering technique that identifies video key frames by learning cluster dependent Gaussian kernels. The proposed algorithm, called clustering and Local Scale Learning algorithm (LSL) learns the underlying cluster dependent dissimilarity measure while finding compact clusters in the given dataset. The learned measure is a Gaussian dissimilarity function defined with respect to each cluster. We minimize one objective function to optimize the optimal partition and the cluster dependent parameter. This optimization is done iteratively by dynamically updating the partition and the local measure. The kernel learning task exploits the unlabeled data and reciprocally, the categorization task takes advantages of the local learned kernel. The second component of the proposed video categorization system consists in discovering the video categories in an unsupervised manner using the proposed LSL. We illustrate the clustering performance of LSL on synthetic 2D datasets and on high dimensional real data. Also, we assess the proposed video categorization system using a real video collection and LSL algorithm.

KEYWORDS

Video categorization; unsupervised clustering; parameter learning; Gaussian function.

1. INTRODUCTION

The widespread use of smart devices with built-in cameras, the recent advances in high-performance networking, and the devices storage capacity reaching a level of hundreds of gigabytes yielded tremendous amount of non-textual data, such as digital images and videos. Meanwhile, video sharing communities through the internet are becoming more and more popular. If no automated tools for storing, searching, and retrieving videos are proposed, this proliferation of video databases may be counterproductive. In fact, finding videos of interest and navigating through these video collections is naturally challenging due to their large volumes and to the computer's inability to recognize the semantic of videos. This problem is known as the semantic gap. To overcome this challenge, video database categorization based on the video content has become an active research topic [46]. One of the most known solutions relies on unsupervised learning techniques, and aims to categorize the video collection into homogeneous classes based on their visual content. The obtained categories are then adopted to index the video database and reduce the search space during the retrieval process. Also, they make the user navigation through the database much easier. The cluster's representatives obtained using categorization algorithms may be used as page zero of a Content Based Video Retrieval (CBVR) system. In other words, instead of starting by displaying random videos, such CBVR system starts by showing representative videos of the classes obtained by categorizing the video

collection. Thus, users would have an overview on the video database content before submitting their queries.

During the past few years, various CBVR systems have been proposed [46], and considerable efforts have been deployed on related research topics such as system design [47], high dimensional indexing structures [49], feature extraction [48], and similarity measures [50]. Despite these efforts the performance of most CBVR systems has proved to be inherently constrained by the performance of the machine learning technique used to categorize video collections into visually homogeneous clusters. The problem is more acute when videos are represented using high dimensional low level descriptors yielding hardly separable video categories. Grouping videos into homogeneous categories may be posed as an unsupervised learning problem. More specifically, the challenge consists in partitioning the frame collection of given videos into subsets, so that frames in each subset share some visual properties. In other words, according to a defined distance measure, frames in the same cluster should be as similar as possible and frames in different clusters should be as dissimilar as possible. Clustering has been used in many applications related to understanding and exploring the structure of the data. In particular, fuzzy clustering techniques have been shown to be suitable to describe real world situations with overlapping boundaries [1]. Typically, the set of video frames to be clustered can be described in two ways: frame based representation, and relational based representation. While for frame representation, each frame is represented using one feature vector, for relational representation, information on how two frames are related is considered. Recently, relational clustering emerged as an active alternative to exploit the adjacency structure of the data and avoid dealing with a prefixed shape of clusters. Relational clustering can be formulated as a kernel based approach because kernel function can be perceived as pairwise dissimilarity function. The choice of such a kernel function yields the mapping of the input data into a new space in such a way that computing nonlinear partitioning in the input space can reduce to a simple partitioning in the feature space. One of the most common dissimilarity function is the Gaussian kernel. The performance of this function depends on the choice of the parameter σ which is usually chosen by trying several values. Moreover, since one global parameter is considered for the entire dataset, finding an optimal σ may be impossible when there are large variations between the distributions of the different clusters in the feature space.

In Figure 1, we use a simple example to illustrate the advantage of using cluster dependent kernel resolution σ_i instead of one global σ . We fit the two clusters of the dataset using two different Gaussian kernels. More specifically, we run kNERF [12] several times using all pairwise combinations of $\sigma_i=0.001,0.01,0.02,0.03\dots,0.2$. Figure 2 reports the accuracy obtained using various combinations of σ_1 and σ_2 . The reddish colors indicate high accuracy, while bluish ones represent low accuracy. As one can notice, the clustering accuracies corresponding to the specific cases where $\sigma_1 = \sigma_2$ (i.e. along the diagonal) do not exceed 80%. On the other hand, combinations of σ (such as $\sigma_i = 0.001$ and $\sigma_j = 0.03$) yield an accuracy of 100%. This illustrative example proves the need for cluster dependent parameter σ_i , and that one global parameter cannot handle effectively the dataset in Figure 1.

The naive solution to learn optimal parameters to fit a dataset is to try several combination (as illustrated in the above example), evaluate the obtained partitions using some validity measure [2], and adopt the parameter value corresponding to the best/optimal partition. However, this exhaustive search of parameters with respect to all clusters of the dataset is not practical. It is computationally expensive and increases significantly with the number of clusters.

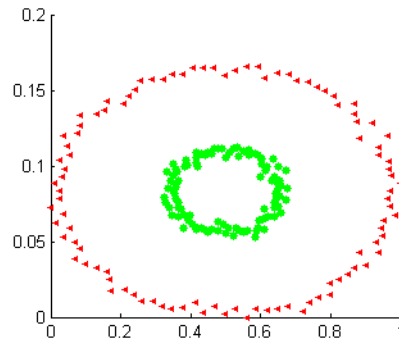


Figure 1 2-D dataset with 2 clusters with different densities

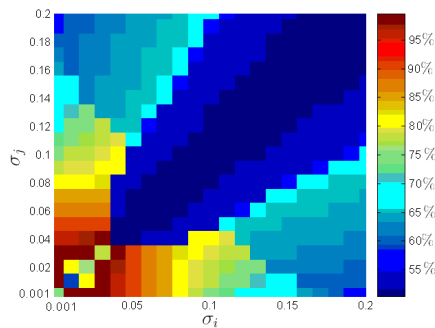


Figure 2 Accuracy results obtained on the dataset of Figure 1 using an extensive search of cluster dependent parameters

1.1. MAJOR CONTRIBUTIONS

The major contributions of this work consist of the design, implementation, and analysis of a video categorization system which relies on a novel clustering algorithm. The proposed unsupervised learning algorithm addresses the challenges raised above. It partitions the video frames, learns the parameters for each cluster, and assigns membership degrees to each frame with respect to all clusters. These memberships allow the algorithms to deal with overlapping clusters, and provide a richer description of the video collection by distinguishing between the frames at the core and at boundary of the cluster. The learned parameters yield an optimal recognition of clusters of different local geometric characteristics and, can be used in subsequent steps to provide better cluster assignment.

1.1.1. CLUSTERING AND LOCAL SCALE LEARNING ALGORITHM

In this work, we introduce a new relational clustering technique with Local Scale parameter Learning (LSL). This approach learns the underlying cluster dependent dissimilarity measure while finding compact clusters in the given data. The learned measure is a Gaussian dissimilarity function defined with respect to each cluster that improve the final partition. LSL minimizes one objective function to optimize the optimal partition and the cluster dependent parameter. This optimization is done iteratively by dynamically updating the partition and the local measure. This allows the kernel learning task exploit the unlabeled data and reciprocally, the categorization task takes advantages of the local learned kernel. Moreover, as we assume that the data is available in a relational form, the proposed approach is applicable even when only the degree to which pairs

of objects in the data are related is available. It is also more practical when similar objects cannot be represented efficiently by a single prototype.

1.1.2. VIDEO CATEGORIZATION SYSTEM

The proposed video categorization system adopts LSL algorithm in order to group similar video frames of the same video into clusters, and obtain key frames summarizing each video. Also, LSL is used to cluster these key frames in order to discover video categories existing in the collection.

1.2 PAPER OVERVIEW

The organization of the rest of the paper is as follows. In section 2, we outline some related work to the proposed approaches. We outline the new unsupervised relational clustering approach LSL in section 3. The proposed video categorization system is presented in section 4. We describe the experiments conducted to validate the proposed approaches in section 5. Finally, Section 6 contains the conclusions and future work.

2. RELATED WORKS

Traditional video categorization systems start with video clips parsing/segmentation into a collection of scenes. These scenes are further segmented into shots containing a reduced number of key frames. Then, a clustering technique is used to organize the shots for efficient indexing, browsing, retrieval, and viewing. In [52], the authors outlined a story extraction approach from long video programs using time-constrained unsupervised learning of video shots. This approach categorizes the video shots without key frames extraction. Also, they used scene transition graph to model the story flow across scenes. Another video categorization alternative consists in clustering key frames summarizing the video shots. Over the past few years, various image indexing, browsing, and clustering systems have been outlined in the literature [54, 53]. In [55], the authors proposed QBIC system to retrieve images of interest based their content from large on-line image collections. QBIC lets the users to submit sketches as query, layout or structural descriptions, texture, color, and sample images in order to retrieve relevant images. The authors in [56] proposed Photobook which allows interactive browsing and searching of image sequences. A general-purpose image retrieval system which selects in an unsupervised manner the appropriate the image features for retrieval is proposed in [57]. Since frames in the same shot show obvious redundancies, some frames which summarize the shot contents are appointed as key frames [58, 59] to represent the shot. The descriptors used for key frame extraction include colors, edges, shapes, and spatial distribution of motion activity, etc... These low-level features are then used to group visually similar frames into homogeneous categories. Finally, the closest frames to the obtained cluster centers are selected as video key frames. In [60], the authors proposed key frames extraction based onagglomerative hierarchical clustering. The researchers in [62] used fuzzy C-means algorithm in the color feature subspace to select key frames. Similarly, the authors in [61] adopted Gaussian mixture models (GMM) in the eigenspace of the frames to determine the video key frames. Theeffectiveness of these key frame extraction techniques rely on the quality of the clustering results. In other words, such solutions depend on the adopted unsupervised learning technique.

For unsupervised learning techniques, the set of frames to be clustered may be represented using either frame data or relational data. For frame data, each frame is described using a feature vector. For relational data, information that represents the degree to which pairs of frames in the video are related is exploited. Thus, Relational clustering is applicable when data instances to be clustered cannot be represented by numerical features. Moreover, it is more effective for applications which rely on expensive distance computation in terms of time complexity. In other words, relational clustering is an efficient alternative when the adopted distance measure does not

have a closed form solution, or when groups of similar objects cannot be represented efficiently using one single prototype/centroid.

Recently, clustering approaches which map data into a new feature space in order to linearize the partitioning problem have been proposed [5, 6, 7, 8]. These approaches rely mainly on kernel and spectral clustering algorithms. Most kernel based solutions are kernel versions of classical clustering algorithms such as kernel K-means [5, 9], kernel FCM [10, 11], kernel SOM [6] and kernel Neural Gas [7]. Among all work based on C-Means clustering algorithm [5, 9], only the Kernelization Non-Euclidean Relational Fuzzy c-Means Algorithm (kNERF) [12] has extended the kernelization to relational data clustering. kNERF [12] adopts the Gaussian kernel with one global parameter for the whole dataset. The parameter σ is set empirically. In [13], the authors suggested tuning σ automatically through multiple runs of their clustering algorithm for a number of values of σ . Then, the value yielding the least distorted clusters is adopted. However, this approach increases the computation time significantly. Additionally, the range of values to be tested is set manually. Moreover, when the input data includes clusters with different local statistics there may not be a single value of σ that works well for all the data. In fact, when the data contains multiple scales, one optimal σ fails to result in a good clustering result. Instead of selecting a single parameter σ , the authors in [14] estimate a local parameter σ_j for each data point x_j and define the similarity between a pair of points. In [14], σ_j is defined as the distance between x_j and its p^{th} neighbor x_p . Although, spectral clustering can accurately discover the structure of small datasets, it shows limitation when dealing with large-scale problems due to its computational complexity of $O(N^3)$ where N is the number of data points [15]. Gaussian similarity function is also considered as a type of radial basis functions. In the context of RBF neural networks, a supervised procedure which takes into consideration the labels of the training data is proposed to determine a parameter per cluster [17]. Although this approach computes a parameter with respect to each cluster it can be applied only in the case of supervised machine learning task. In fact, it uses the label information in order to determine the parameter. Thus, it cannot be applied in the context of unsupervised clustering. Moreover, this approach is prototype based and thus, inherits from the drawbacks of object-based approaches with respect to the relational ones.

3. UNSUPERVISED RELATIONAL CLUSTERING WITH LOCAL SCALE PARAMETER

Let $\{x_1, \dots, x_N\}$ be a set of N data points to be partitioned into C clusters. We also assume that $\mathbf{R} = [r_{jk}]$ is a relational matrix where r_{jk} represents the degree to which pairs of objects x_j and x_k are related. The matrix \mathbf{R} could be given or it could be constructed from the features of the objects. Each object x_j belongs to cluster i with a membership u_{ij} that satisfies [42]:

$$0 \leq u_{ij} \leq 1 \text{ and } \sum_{i=1}^C u_{ij} = 1, \text{ for } i \in \{1, \dots, C\}, j \in \{1, \dots, N\}. \quad (1)$$

The clustering and Local Scale Learning algorithm (LSL) minimizes the proposed objective function below:

$$J_m = \sum_{i=1}^C \sum_{j=1}^N \sum_{k=1}^N u_{ij}^m u_{ik}^m \left(1 - \exp\left(-\frac{r_{jk}}{\sigma_i}\right) \right) - \sum_{i=1}^C \frac{K_1}{\sigma_i^2}, \quad (2)$$

subject to the membership constraint in (1).

In (2), $m \in (1, \infty)$ is the fuzzyfier. The term $u_{ij}^m u_{ik}^m$ can be regarded as the likelihood that two points x_j and x_k belong to the same cluster i . We use β_{jk}^i to denote this term. That is, we let

$$\beta_{jk}^i = u_{ij}^m u_{ik}^m. \quad (3)$$

LSL algorithm is based on minimizing a joint objective function with two terms. The first term seeks compact clusters using a local relational distance, \mathbf{D}_{jk}^i , with respect to each cluster i . This distance is defined as

$$D_{jk}^i = 1 - \exp\left(-\frac{r_{jk}}{\sigma_i}\right), \quad (4)$$

In (4), \mathbf{D}_{jk}^i is based on the Gaussian kernel function and is intimately related to the heat flow [36]. In fact, locally, the heat kernel is approximately equal to the Gaussian, i.e.,

$$H_\sigma(x, y) \approx (4\pi\sigma)^{-\frac{p}{2}} \exp\left(-\frac{d(x, y)}{4\sigma}\right), \quad (5)$$

When the squared distance between x and y , $d(x, y)$, is sufficiently small [36].

In (4), the parameter σ_i controls the rate of decay of \mathbf{D}_{jk}^i as a function of the distance between \mathbf{x}_j and \mathbf{x}_k with respect to cluster i . Using a cluster dependent parameter σ_i allows LSL to deal with the large variations in the feature space between the distributions and the geometric characteristics of the different clusters. The second term in (2) is a regularization term aiming to avoid the trivial solution where all the parameters σ_i are infinitely large. In fact, without this term, the minimization of (2) with respect to σ_i yields the trivial solution of a very large σ_i that merges all points into a single cluster. Another trivial solution that minimizes the objective function in (2) is when one of σ_i is zero. In order to keep the derivation simple, we do not introduce another regulation term. We simply assume that σ_i is not null. Later in this section, we will discuss how to handle this case.

The goal of the proposed LSL algorithm is to learn the C clusters, the parameters σ_i of each cluster, and the membership values u_{ij} , of each sample \mathbf{x}_j with respect to each cluster i . This learning task is achieved by minimizing the objective function in (2) with respect to σ_i and u_{ij} . In order to optimize (2) with respect to u_{ij} , we use the relational dual of the fuzzy C -mean algorithm formulated by Hathaway et al. [37]. It has been proved in [37] that the Euclidean distance $d_{ik}^2 = \|\mathbf{x}_k - \mathbf{c}_i\|^2$, from feature \mathbf{x}_k to the center of the i^{th} cluster, \mathbf{c}_i , can be written in terms of the relational matrix \mathbf{D}^i as

$$d_{ik}^2 = \sum_{j=1}^N D_{jk}^i \frac{u_{ij}^m}{\sum_{v=1}^N u_{iv}^m} - \frac{1}{2} \sum_{j=1}^N \sum_{q=1}^N \frac{u_{ij}^m D_{jq}^i u_{iq}^m}{(\sum_{v=1}^N u_{iv}^m)^2}, \quad (6)$$

Using the implicit distance values, d_{ik}^2 , the objective function in (2) could be rewritten as

$$J = \sum_{i=1}^C \sum_{j=1}^N u_{ij}^m d_{ik}^2 - \sum_{i=1}^C \frac{K_1}{\sigma_i^2}, \quad (7)$$

To optimize J with respect to u_{ij} subject to (1), we use the Lagrange multiplier technique and obtain

$$J = \sum_{i=1}^C \sum_{j=1}^N u_{ij}^m d_{ik}^2 - \sum_{i=1}^C \frac{K_1}{\sigma_i^2} - \sum_{j=1}^N \lambda_j (\sum_{i=1}^C u_{ij} - 1), \quad (8)$$

By setting the gradient of J to zero, we obtain

$$\frac{\delta J}{\delta \lambda} = \sum_{i=1}^C u_{ij} - 1 = 0 \quad (9)$$

$$\frac{\delta J}{\delta u_{ij}} = m u_{ij}^{m-1} d_{ij}^2 - \lambda = 0 \quad (10)$$

Solving (10) for u_{ij} yields

$$u_{ij} = \left(\frac{\lambda}{md_{ij}^2} \right)^{\frac{1}{m-1}} \quad (11)$$

Substituting (11) back into (9), we obtain

$$\sum_{i=1}^C u_{ij} = \left(\frac{\lambda}{m} \right)^{\frac{1}{m-1}} \sum_{i=1}^C \left(\frac{1}{d_{ij}^2} \right)^{\frac{1}{m-1}}, \quad (12)$$

Thus,

$$\left(\frac{\lambda}{m} \right)^{\frac{1}{m-1}} = \frac{1}{\sum_{i=1}^C \left(\frac{1}{d_{ij}^2} \right)^{\frac{1}{m-1}}}, \quad (13)$$

Substituting this expression back in (11), we obtain

$$u_{ij} = \frac{\left(\frac{1}{d_{ij}^2} \right)^{\frac{1}{m-1}}}{\sum_{i=1}^C \left(\frac{1}{d_{ij}^2} \right)^{\frac{1}{m-1}}}, \quad (14)$$

Simplifying (14), we obtain the following update equation

$$u_{ij} = \frac{1}{\sum_{i=1}^C \left(\frac{d_{ij}^2}{d_{ij}^2} \right)^{\frac{1}{m-1}}}, \quad (15)$$

We can notice from equations (6) and (15) that the expression of u_{ij} does not depend on any notion of cluster prototype (e.g center). In fact, it depends only on the relational matrix \mathbf{R} , and the normalized membership vector \mathbf{v} .

In the objective function in (2), the resolution of the different clusters σ_i is independent of each other. Thus, in order to optimize (2) with respect to σ_i , we can reduce the optimization problem to C independent problems. That is, we convert the objective function in (2) to the following C simpler functions

$$J^i = \sum_{j=1}^N \sum_{k=1}^N u_{ij}^m u_{ik}^m \left(1 - \exp\left(-\frac{r_{jk}}{\sigma_i}\right) \right) - \frac{K_1}{\sigma_i^2}, \quad (16)$$

for $i = 1, \dots, C$. The optimal update equation of the parameters σ_i can be obtained using the Lagrange method by solving

$$\frac{\partial J^i}{\partial \sigma_i} = - \sum_{j=1}^N \sum_{k=1}^N u_{ij}^m u_{ik}^m \frac{r_{jk}}{\sigma_i^2} \exp\left(-\frac{r_{jk}}{\sigma_i}\right) + \frac{2K_1}{\sigma_i^3} = 0, \quad (17)$$

Using the duality between the Gaussian similarity and the heat flow function, it has been shown in [36] that

$$\sum_{j, r_{jk} < \epsilon} \exp\left(-\frac{r_{jk}}{\sigma_i}\right) = \frac{N}{(\pi\sigma_i)^{\frac{p}{2}}}, \quad (18)$$

When ϵ is sufficiently small. In (18), p is the dimension of the manifold. In the worst case, the point located in the neighborhood of a point j within the radius ϵ are distant from j with at most ϵ .

As ϵ is sufficiently small, we can assume that the distances \mathbf{r}_{jk} are almost constant in the neighborhood of j . Thus, (18) can be rewritten as

$$\exp\left(-\frac{r_{jk}}{\sigma_i}\right) = \frac{N}{|N|(\pi\sigma_i)^{\frac{p}{2}}} \quad (19)$$

Where $|N|$ is the cardinality of the neighborhood of j . Substituting (19) in (17) gives

$$\frac{\partial J^i}{\partial \sigma_i} = -\sum_{j=1}^N \sum_{k=1}^N u_{ij}^m u_{ik}^m \frac{Nr_{jk}}{2\pi^{\frac{2-p}{2}}|N|(\sigma_i)^{2-\frac{p}{2}}} + \frac{2K_1}{\sigma_i^3}, \quad (20)$$

Setting (20) to zero and solving for σ_i , we obtain

$$\sigma_i = \left(\frac{K}{\sum_{j=1}^N \sum_{k=1}^N u_{ij}^m u_{ik}^m r_{jk}}\right)^{\frac{2}{2+p}} \quad (21)$$

Where

$$K = K_1 \frac{2\pi^{\frac{2-p}{2}}|N|}{N}. \quad (22)$$

We should recall here that our assumption about a non null σ_i is reasonable. In fact, the update equation in (21) shows that σ_i is zero only when the distance \mathbf{r}_{jk} between two points, \mathbf{x}_j and \mathbf{x}_k , that belong to cluster i (non zero fuzzy memberships u_{ij} and u_{ik}) is infinitely large. This scenario is highly unlikely.

One can notice based on (21) and (4) that for each cluster i , σ_i controls the rate of decay of \mathbf{D}_{jk}^i with respect to the distance \mathbf{r}_{jk} . In fact, σ_i is inversely proportional to the intra-cluster distances with respect to each cluster i . Thus, when the intra-cluster dissimilarity is small, σ_i is large allowing the pairwise distances over the same cluster to be smaller and thus obtain a more compact cluster. On the other hand, when the intra-cluster dissimilarity is high, σ_i is small to prevent points which are not highly similar from being mapped to the same location. According to (21), σ_i can also be seen as the average time to move between points in cluster i . The proposed LSL approach is outlined in Algorithm 1.

Algorithm 1 The LSL algorithm

Set the number of clusters C , the parameter $K_1 \in]0 \infty)$, and the fuzzyfier $m \in [1 \infty)$;
 Initialize the partition matrix U ;
 Initialize the parameters σ_i ;
REPEAT Compute the dissimilarity D^i for all the clusters using (4);
 Compute the distances using (6);
 Update the membership using (15);
 Update the parameter σ_i using (21);
UNTIL (membership do not change or maximum number of iteration is reached)

Similarly to typical relational clustering algorithms such as Relational FCM [42] and Spectral clustering algorithms [5, 6, 7, 8], the worst case complexity of the proposed LSL is $O(N^2)$. Where N is the number of data samples.

4. PROPOSED VIDEO CATEGORIZATION APPROACH

In this section, we describe our video categorization approach that relies on LSL algorithm. Clustering is used to group frames of the same video into clusters and provides a set of key frames summarizing the video. These key frames are then used as basis for the categorization of the video collection. Since this unsupervised learning task involves high dimensional and hardly separable data, we use LSL to learn the video categories existing in the collection.

In Figure 3, we show the flowchart of the proposed video categorization system based on LSL algorithm. As it can be seen, three interactive phases form the proposed approach. Namely, the frame and low level features extraction, the automatic key frame selection, and the video categories learning. The first phase aims to extract frames from videos and represent them using various low-level descriptors such as color, texture and shape. Notice that the frame relational data is computed based on these features. The second phase (automatic key frame selection) inherits the feature space generated by phase 1, and runs LSL for each video frames in order to group features vectors into visually similar clusters. The representatives of each cluster are considered as video key frames. For the third phase, as shown in the right part of Figure 3, the obtained key frames are categorized using LSL algorithm into visually homogeneous categories which correspond to the video classes existing in the original video database.

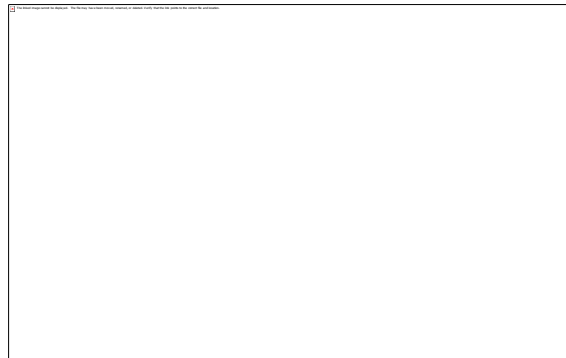


Figure 3. Overview of the proposed video categorization system

5. EXPERIMENTS

The objective of LSL algorithm is to partition the data and learn a kernel resolution for each cluster. In this section, we assess the performance of LSL and compare its performance to relevant clustering algorithms. First, we use 2D synthetic datasets. Then, in order to illustrate the ability of the proposed algorithm to learn local kernels and to cluster dissimilarity measure derived from real and high dimensional data, we use it to categorize real video collection where categories have different sizes, intra-group, and inter-group variations.

In our experiments, we assume that the ground truth is known and we compute the obtained partition accuracy. First, each cluster is assigned a label based on the majority of the true labels of its elements. Then, the correct classification rate of each cluster is computed. The overall accuracy of the partition is computed as the average of the individual clusters rates weighted by the clusters cardinality. Notice that our experiments showed that the value of the parameter $K1$ in (2) is related to inter-cluster similarity between the nearest clusters. Thus for the experiments aiming to cluster synthetic datasets we set $K1$ to 0.005. For the video database categorization, the parameter $K1$ is set to 0.8. Notice that in our experiments, we used an Intel core i7 computer with 16 G RAM and MATLAB 10 software.

In the following, we compare our unsupervised clustering algorithm, LSL, to those obtained using the FCM [42], DBSCAN [43], GK [44], kNERF[12], and self tuning spectral clustering [14]. For the relational approaches, the Euclidean distance is used to compute the pairwise distances.

5.1. SYNTHETIC DATASETS CLUSTERING

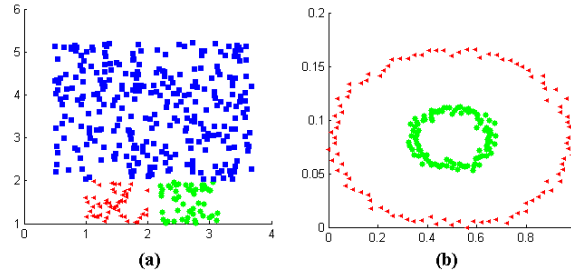


Figure 4. Datasets used to illustrate the performance of LSL. Each cluster is shown by a different color.

To illustrate the ability of LSL to learn appropriate local parameters and cluster the data simultaneously, we use it to partition synthetic 2D datasets. We should mention here that, for the purpose of visualizing the results, we use feature based and 2-dimensional data. Relational dissimilarity matrices are obtained by computing the Euclidean distance between these feature vectors. We use 2 datasets that include categories of different shapes with unbalanced sizes and densities. Figure 4 displays the 2 synthetic datasets. Each cluster is displayed with a different color. For all algorithms, we set the number of clusters C to the true one (see Figure 4), the fuzzyfiern to 1.1, and the maximum number of iterations to 100. As LSL requires the specification of one parameter K , we use $K=[0.001,0.01,0.05,0.1,0.5,1,1.5, 2,4,8,10]$ and select the best results. For DBSCAN and Self tuning spectral, we tune the neighborhood parameter from 1 to 20 by an increment of 1. For kNERF, we tune the parameter between 0.01 and 100 with a step of 0.1. The matrix of memberships is initialized randomly. Figure 5 displays the clustering results on dataset 1. As it can be seen, neither FCM, DBSCAN, GK, kNERF, or Spectral clustering were able to categorize this data correctly. On the other hand, LSL learned local exponential mapping of the data and was able to partition this data correctly.

The learned parameters of dataset 1 are reported in Table 1. For this example, cluster 3 has bigger size than the two other clusters, and the three clusters have comparable densities. As a result, LSL learns slightly small parameter for cluster 3 than for cluster 1 and 2. three parameters that are slightly different. However, some points along the cluster boundaries are not correctly categorized. In fact, as shown in figure 6, the returned memberships of cluster 2 and cluster 3 are too close (between 0.3 and 0.33). This is due to the characteristic of this dataset where the boundaries are not well defined. The learned parameters of dataset 1 are reported in Table 1. For this example, cluster 3 has bigger size than the two other clusters, and the three clusters have comparable densities. As a result, LSL learns slightly small parameter for cluster 3 than for cluster 1 and 2. three parameters that are slightly different.

However, some points along the cluster boundaries are not correctly categorized. In fact, as shown in figure 6, the returned memberships of cluster 2 and cluster 3 are too close (between 0.3 and 0.33). This is due to the characteristic of this dataset where the boundaries are not well defined.

Table 1. Partition and parameters learned by LSL for Dataset 2 displayed in Figure 4 (a) when $K = 0.05$

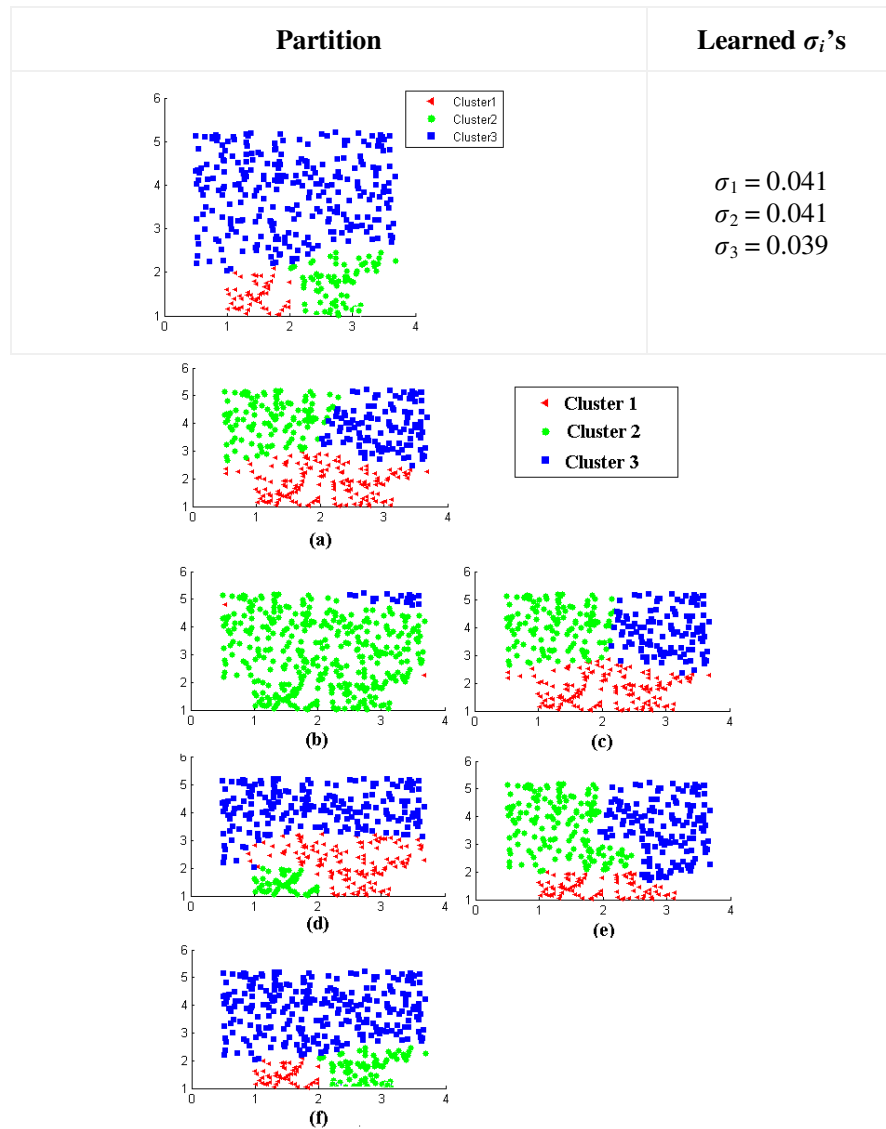


Figure 5. Results of clustering dataset 1 using (a) FCM, (b) DBSCAN, (c) GK, (d) kNERF, (e) Spectral, (f) LSL

The learned parameters of dataset 1 are reported in Table 1. For this example, cluster 3 has bigger size than the two other clusters, and the three clusters have comparable densities. As a result, LSL learns slightly small parameter for cluster 3 than for cluster 1 and 2. three parameters that are slightly different. However, some points along the cluster boundaries are not correctly categorized. In fact, as shown in figure 6, the returned memberships of cluster 2 and cluster 3 are too close (between 0.3 and 0.33). This is due to the characteristic of this dataset where the boundaries are not well defined.

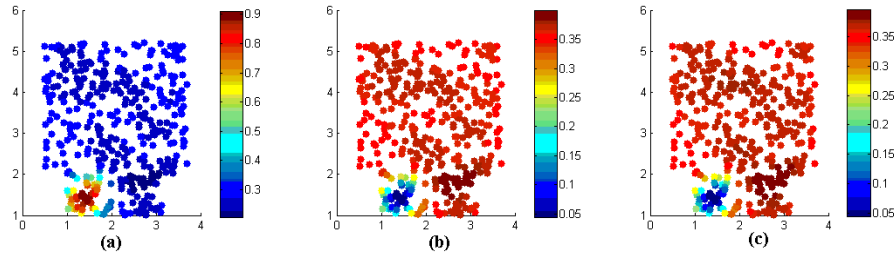


Figure 6. memberships learned by LSL on dataset 2 (Fig. 4 (b)) with respect to (a) cluster 1, (b) cluster 2, and (c) cluster 3.

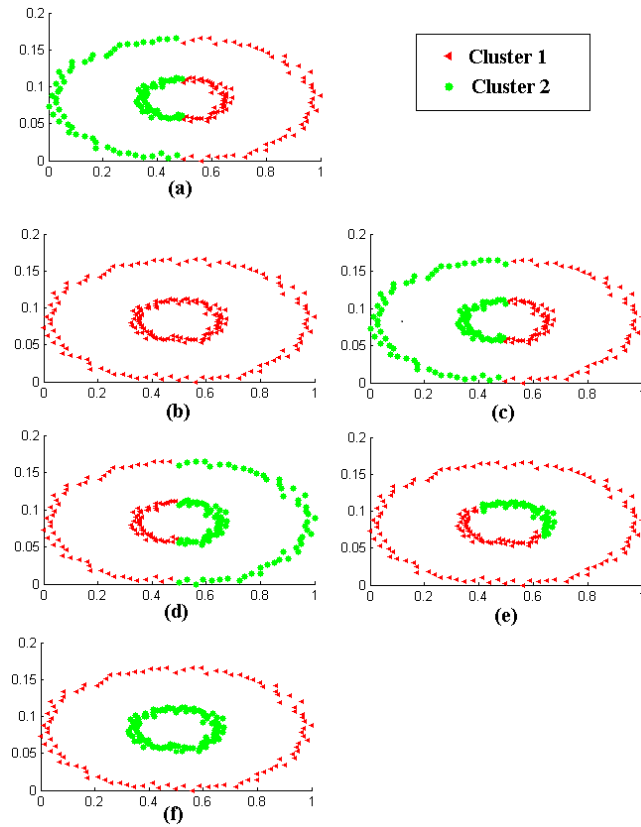


Figure 7. Results of clustering dataset 2 using (a) FCM, (b) DBSCAN, (c) GK, (d) kNERF, (e) Spectral, and (f) LSL.

Dataset 2 is constituted of two concentric ovals. It does not correspond to the classical way of perceiving intra-cluster and inter-cluster distances. As a result, only LSL was able to categorize this data correctly (Fig. 7 (f)). Although it is hard to define the notion of density on dataset 2, we can notice that the two learned parameters ($\sigma_1 = 0.0014$ and $\sigma_2 = 0.029$) are meaningful (Table 2). In fact, as cluster 2 is slightly denser than cluster 1, σ_2 is slightly higher than σ_1 . We should mention here that the learned parameters are equal to the ones found by extensive search in section 1. This shows the efficiency of LSL in learning the parameters. Figure 8 shows that some points have memberships around 0.5, indicating that they are close to both clusters in the mapped feature space. This is an inherit limitation of the LSL since it implicitly uses the Euclidean distance to map the data.

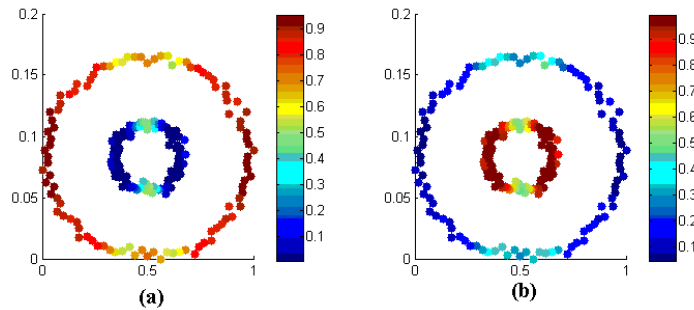
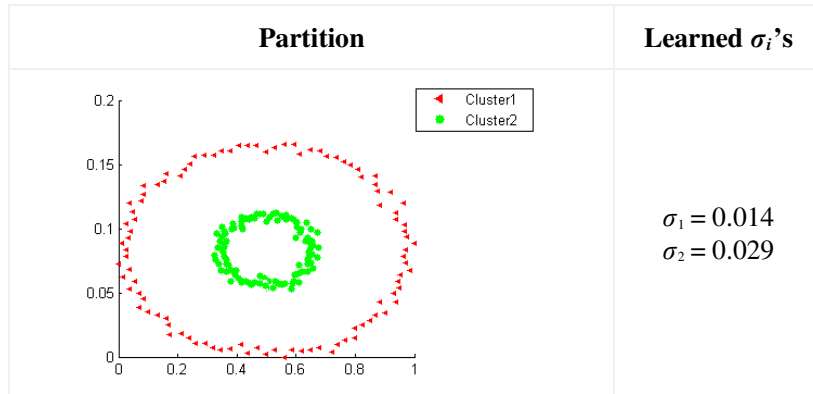
Table 2 Partition and parameters learned by LSL for Dataset 4 displayed in Figure 4 (b) when $K = 1.5$ 

Figure 8. memberships learned by LSL on dataset 4 (Fig. 4 (b)) with respect to (a) cluster 1, and (b) cluster 2.

5.2. VIDEO CATEGORIZATION

To illustrate the ability of LSL to model data containing hardly separable clusters, and cluster real video collection dataset. We use UCF Sports action dataset which consists of a set of actions collected from various sports [63]. It contains 150 video sequences of sport actions at a resolution of 720x480. Actions in this dataset include Diving (14 videos), Golf swinging (18 videos), Kicking (20 videos), Lifting (6 videos), Horseback riding (12 videos), Running (13 videos), Skating (12 videos), Swinging (33 videos) and Walking (22 videos). Figure 9 shows 12 sample videos from this dataset. The videos within most categories show high intra-class variations. For instances, the "Gulf" category includes videos of gulf swings taken from front, side and back. Similarly, the "Kicking" category contains front and side views of players kicking balls. Each video frame in the collection is represented using three low-level visual descriptors. These are some of the commonly used features in content based image retrieval. They may not be the optimal features for the selected video collection. However, our goal is to show that the clustering of these features using LSL provides better results than traditional clustering.

- RGB color histogram: Each image is mapped to the RGB color space and quantized into 32 bins. A 32-dimensional histogram is used to represent the distribution of the color of each image.

- HSV color moments: Each image is mapped to the HSV color space, and the mean, standard deviation, and skewness of the distribution of the H, S, and V components are computed. This feature subset is represented by a 9- dimensional vector.

- Edge histogram: We use the MPEG-7 edge histogram descriptor (EHD) [64] to represent the frequency and directionality of the edges. First simple edge detector operators are used to identify edges and group them into five categories: vertical, horizontal, 45% diagonal, 135% diagonal, and isotropic (non-edge). Thus, the global edge histogram has 5 bins. Similarly, for the semi global edge histograms, we define 13 different segments (i.e., 13 different subsets of the image-blocks) and for each segment we generate edge distributions for five different edge types from the 80 local histogram bins. Consequently, the EHD feature set is represented by a 150-dimensional vector (80 bins (local) + 5 bins (global) + 65 bins (13×5, semi-global)).

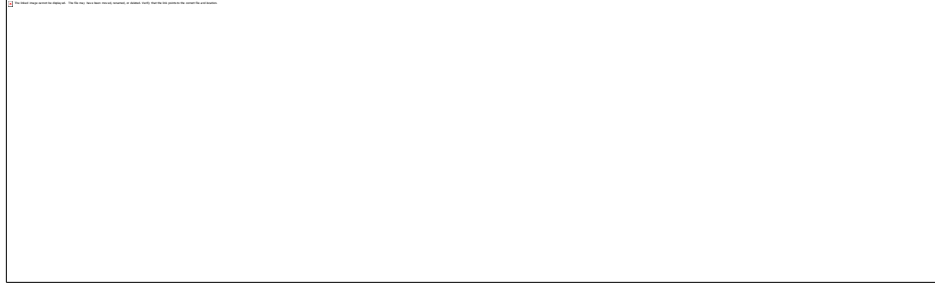


Figure 9. Sample videos from the the 9 categories from UCF sports action dataset [63].

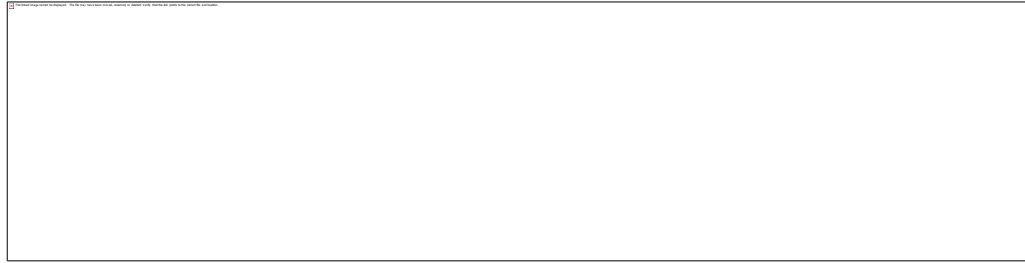


Figure 10. Sample videos key frames generated using LSL to represent videos from UCF sports action dataset [63].

After extracting these low level features from all frames, we concatenate them in order to represent each frame using one high-dimensional vector (191-dimensions). Then, we use the corresponding relational data as input to the clustering algorithm. More specifically, we LSL to group frames of the same video into visually homogeneous clusters and provide representative key frame. Notice that we empirically set the number of clusters to three. In Figure 10, we show sample key frames obtained using LSL to represent videos from UCF sports action dataset [63]. For each video, we consider the closest frame to the centroid of the largest obtained cluster as key frame. In fact, it has been proved in [37] that the Euclidean distance, $d_{ik}^2 = \|\mathbf{x}_k - \mathbf{c}_i\|^2$, from feature \mathbf{x}_k to the center of the i^{th} cluster, \mathbf{c}_i , can be written in terms of the relational matrix D^i as

$$d_{ik}^2 = (D^i v_i)_k - \frac{v_i^t D^i v_i}{2}. \quad (23)$$

In (23), D^i is the pairwise distance matrix, v_i is the membership vector of all N samples in cluster i defined by

$$v_i = \frac{(u_{i1}^m, \dots, u_{iN}^m)^t}{\sum_{j=1}^N u_{ij}^m}. \quad (24)$$

The obtained key frame collection is then used to categorize the video collection. In other words, we run LSL on the key frame vectors in order to cluster them into a set of visually homogeneous categories. The obtained key frame clusters correspond to the categories existing in the video collection. To assess the performance of the proposed video categorization approach, we assume that the ground truth is known, we consider the classification rate. More specifically, each cluster is assigned a label based on the majority of the true labels of its elements. Then, the correct classification rate of each cluster is computed. Since it is difficult to compare the individual clusters' rates for different partitions (requires cluster matching, and taking into account the cluster sizes), we simply compute the overall classification rate of the partition as the average of the individual clusters rates weighted by the clusters cardinality. The results were compared with those obtained using FCM [42] and DBSCAN [43] algorithms.

Since these algorithms require the specification of the number of clusters, first, we set the initial number of clusters to 9 and measure the performance of the different algorithms as shown in Table 3. All methods achieved reasonable overall performance with LSL based solution outperforming the method relying on FCM [42] and DBSCAN [43].


Table 3. Overall accuracy obtained using the proposed method, FCM [42] and DBSCAN [43].

	LSL	FCM	DBSCAN
Classification rate	94.12%	73.46%	81.86%
Folkes-Mallows	71.49%	46.60%	68.22%
Jaccard coefficient	49.97%	38.73%	42.04%

To illustrate the video categorization performance, we display in Table 4 sample videos from the 9 clusters obtained using the proposed approach and LSL. As it can be seen, our algorithm generates a partition that is highly similar to the ground truth partition. In other words, these clusters are more compatible with the users' notion of categories present in the video collection. This could be explained by the fact that the proposed algorithm learns better the structure of the data. Moreover, LSL makes frames clusters separable and yields better overall performance. However, as one can notice, few videos are wrongly assigned to different clusters. For instance, for cluster 6, the third video from the left is wrongly assigned to this cluster. In fact, the visual properties of this video (from the "Walking" category) are similar to those shared by most videos of the "Kicking" cluster. Thus, despite these videos are from different categories, the considered low-level features were unable to discriminate between them.

Table 5 displays two clusters obtained using FCM [42] and DBSCAN [43]. More specifically, we used these two algorithms, instead of LSL, to determine the key frames, and to cluster them into video categories. As it can be seen, the obtained video classes are less homogeneous. This is because FCM [42] and DBSCAN [43] do not have a provision for cluster dependent parameters. In other words, they learn one global sigma for the whole dataset. Thus, they are less capable of discriminating between frames sharing similar visual descriptors. For instance, for DBSCAN [43] "Diving" and "Lifting" images are categorized into the same cluster because their color descriptors are very similar.

Table 4. Sample videos from the 9 clusters generated by LSL

Cluster #	Sample videos
1	 The linked image cannot be displayed. The file may have been moved, renamed, or deleted. Verify that the link points to the correct file and location.
2	 The linked image cannot be displayed. The file may have been moved, renamed, or deleted. Verify that the link points to the correct file and location.
3	 The linked image cannot be displayed. The file may have been moved, renamed, or deleted. Verify that the link points to the correct file and location.
4	 The linked image cannot be displayed. The file may have been moved, renamed, or deleted. Verify that the link points to the correct file and location.
5	 The linked image cannot be displayed. The file may have been moved, renamed, or deleted. Verify that the link points to the correct file and location.
6	 The linked image cannot be displayed. The file may have been moved, renamed, or deleted. Verify that the link points to the correct file and location.
7	 The linked image cannot be displayed. The file may have been moved, renamed, or deleted. Verify that the link points to the correct file and location.
8	 The linked image cannot be displayed. The file may have been moved, renamed, or deleted. Verify that the link points to the correct file and location.
9	 The linked image cannot be displayed. The file may have been moved, renamed, or deleted. Verify that the link points to the correct file and location.

When we analyzed further the results we noticed that the FCM algorithm [42] uses more than one clusters for the “Gulf” and the “Kicking” categories. This is because these categories have large intra-cluster color variations. As a result, other categories (e.g. “Skating” and “Walking”) were not categorized correctly (since the number of clusters is fixed to 9). On the other hand, LSL suffers less from this problem because it handles better clusters of different shapes and sizes.

The obtained clusters are more compatible with the user’s notion of clusters. This improvement in performance is due mainly to the cluster dependent parameters learned by LSL that reflect the intra-cluster characteristics of the data.

Table 5 Sample “wrong” clusters obtained using FCM [42] and DBSCAN [43].

Sample videos					
FCM [42]	 The linked image...	 The linked image cannot be...	 The linked image c...	 The linked image...	 The linked image c...
DBSCAN [43]	 The linked image r	 The linked image r	 The linked image r	 The linked image r	 The linked image r

In Table 6, we show the per-cluster accuracies of the video categorization achieved by the different algorithms within the obtained clusters. We assign to each obtained cluster, the majority class of the videos assigned to it. For instance, if the majority of the videos assigned to a given cluster are from the class “Kicking”, then we consider that the correct class of this cluster is “Kicking”, and we calculate its accuracy based on this assumption. As it can be seen, video categorization using LSL outperforms the results obtained using FCM [42] and DBSCAN [43]. More specifically, LSL based categorization yields higher accuracies with respect to all clusters.

Table 6 Per-cluster accuracy obtained using the proposed method, FCM [42] and DBSCAN [43].

cluster #	LSL	FCM	DBSCAN
1	96.5%	73.46%	81.86%
2	97.32%	74.12%	83.05%
3	45.78%	18.23%	25.19%
4	57.91%	33.47%	57.91%
5	66.54%	21.98%	60.07%
6	90.11%	82.58%	90.11%
7	77.56%	47.71%	55.16%
8	95.66%	66.09%	88.20%
9	92.44%	58.63%	70.98%

6. CONCLUSIONS AND FUTURE WORK

Despite researcher efforts to propose efficient Content Based Video Retrieval (CBVR) systems, the proposed solutions have proved to be inherently constrained by the performance of the machine learning component which aims to categorize the video collections into visually

homogeneous clusters. In this paper, we have introduced a novel approach for efficient video categorization which relies on two main components. The first one is a new relational clustering techniques that identifies video key frames by learning cluster dependent Gaussian kernels. The proposed algorithm, called clustering and Local Scale Learning algorithm (LSL) learns the underlying cluster dependent dissimilarity measure while finding compact clusters in the given frame dataset. The learned measure is a Gaussian dissimilarity function defined with respect to each cluster. This is a major contribution to Gaussian based clustering approaches such as kernel and spectral clustering methods that suffer from their sensitivity to this parameter.

We have illustrated the clustering performance of LSL on synthetic 2D datasets and on high dimensional real data. Our experimental results have demonstrated the effectiveness of LSL. In addition, we have shown that the learned parameters and the memberships returned by LSL are meaningful and reflect the geometric characteristic of the data. Also, the proposed LSL algorithm has been adopted by our video categorization system in order to group similar video frames of the same video into clusters, and obtain key frames summarizing each video. LSL has been used to cluster these key frames in order to discover video categories existing in a real video collection.

Currently, we assume that the number of clusters is known priori. We plan to investigate relaxing this assumption by identifying the optimal number of clusters in an unsupervised manner. We would relax the constraint that the memberships of any data instance with respect to a given cluster must sum to 1, and use possibilistic memberships [45]. This way, we can over specify the number of clusters, then identify and merge similar clusters. In addition the optimal number of cluster identification, the possibilistic logic would improve LSL in terms of robustness to noise and outliers.

ACKNOWLEDGMENT

This work was supported by the Research Center of College of Computer and Information Sciences, King Saud University. The authors are grateful for this support.

REFERENCES

- [1] F. Klawonn, "Fuzzy Clustering: Insights and a New Approach", *Mathware& Soft Computing* 11, 2004.
- [2] C. Borgelt, R. Kruse, "Finding the number of fuzzy clusters by resampling", *IEEE Conf on Fuzzy Systems*, 2006.
- [3] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. "Constrained k-means clustering with background knowledge", *ICML*, pp. 577-584, 2001.
- [4] N. Grira, M. Crucianu, and N. Boujema. "Semi-supervised fuzzy clustering with pairwise-constrained competitive agglomeration", *IEEE Conf on Fuzzy Systems*, pp. 867-872, 2005.
- [5] M. Girolami. "Mercer kernel based clustering in feature space", *Neural Networks*, pp. 780-784, 2002.
- [6] R. Inokuchi, S. Miyamoto, "LVQ clustering and SOM using a kernel function", *Conf on Fuzzy Systems*, 2004.
- [7] A. K. Qinand, P. N. Suganthan, "Kernel neural gas algorithms with application to cluster analysis", *ICPR*, 2004.
- [8] Luxburg, "U. A tutorial on spectral clustering", *Statistics and Computing*, 2007.
- [9] B. Scholkopf, A.J. Smola, and K. R. Muller, "Nonlinear component analysis as a kernel eigenvalue problem", *Neural Computation*, 1998.
- [10] Z.-D. Wu, W.-X. Xie, and J.-P. Yu, "Fuzzy c-means clustering algorithm based on kernel method", *ICCIMA*, pp. 49-54, 2003.
- [11] D.-Q. Zhang and S.-C. Chen, "Fuzzy clustering using kernel method", *ICCA*, 2002.
- [12] R.J. Hathaway, J.M. Huband, and J.C. Bezdek, "A Kernelized Non-Euclidean Relational Fuzzy c-Means Algorithm", *FUZZ-IEEE*, 2005.
- [13] P. Perona and W. T. Freeman, "A Factorization Approach to Grouping", *ECCV*, 1998.
- [14] L. Zelnik-Manor, P. Perona, "Self-Tuning spectral clustering", *NIPS*, 2004.

- [15] I. Dhillon, Y. Guan, B. Kulis, "Kernel k-means: spectral clustering and normalized cuts", ACM SIGKDD, 2004.
- [16] B. Hammer, A. Hasenfuss, "Relational Neural Gas", KI, 2007.
- [17] M. J. Er, S. Wu, J. Lu and H. L. Toh, "Face Recognition With Radial Basis Function (RBF) Neural Networks", Neural Networks, 2002.
- [18] S. Basu, A. Banerjee, and R. J. Mooney, "Semisupervised clustering by seeding", ICML, 2002.
- [19] A. Demiriz, K. Bennett, and M. Embrechts, "Semi-supervised clustering using genetic algorithms", IESTANN, pp. 809-814, 1999.
- [20] K. Wagstaff and C. Cardie, "Clustering with instance-level constraints", ICML, pp. 1103-1110, 2000.
- [21] Stella X. Yu and Jianbo Shi, "Segmentation Given Partial Grouping Constraints", IEEE Trans. Pattern Anal. Mach. Intell., vol. 26, pp. 173-183.
- [22] E. Xing, A. Ng, M. Jordan, and S. Russell, "Distance metric learning with application to clustering with side-information," NIPS, 2003.
- [23] Shental, N., Bar-Hillel, A., Hertz, T., & Weinshall, D. (2003b). Enhancing image and video retrieval: Learning via equivalence constraints. CVPR. Madison, WI.
- [24] R. Yan, J. Zhang, J. Yang, & A. Hauptmann. "A discriminative learning framework with pairwise constraints for video object classification", IEEE TPAMI, pp. 578-593, 2006
- [25] S. Basu, et al. "Active semi-supervision for pairwise constrained clustering", ICML, 2003.
- [26] K. Q. Weinberger, J. Blitzer, and L. K. Saul. "Distance metric learning for large margin nearest neighbor classification", NIPS, 2005.
- [27] A. Globerson and S. T. Roweis. "Metric learning by collapsing classes", NIPS, 2005.
- [28] S. Shalev-Shwartz, Y. Singer, and A. Y. Ng. "Online and batch learning of pseudo-metrics", ICML, 2004.
- [29] J. V. Davis, et al. "Information-theoretic metric learning". ICML, pp. 209-216, 2007.
- [30] R. Chatpatanasiri, T. Korsrilabutr, P. Tangchanachaianan, and B. Kijirikul. "On kernelization of supervised Mahalanobis distance learners", ArXiv, 2008.
- [31] D. Kamvar, et. al., "Spectral Learning", IJCAI, 2003.
- [32] B. Kulis, S. Basu, I. Dhillon and R. Mooney. "Semi-supervised graph clustering: a kernel approach", Machine Learning, Vol. 74, pp. 1-22, 2009.
- [33] T. De Bie, J. Suykens, B. De Moor. "Learning from general label constraints", IAPR, pp. 671-679, 2004.
- [34] Kulis, et al. "Semi-supervised graph clustering: a kernel approach", ICML, 2005.
- [35] S. Basu, M. Bilenko, R. Mooney. "A probabilistic framework for semi-supervised clustering", KDD, 2004
- [36] M. Belkin, P. Niyogi, "Laplacian Eigenmaps and Spectral Techniques for Embedding", NIPS, 2001.
- [37] R. J. Hathaway, et al., "Relational duals of the c-means algorithms", Pattern Recognition, 1989.
- [38] S.C.H. Hoi, et al. "Semi Supervised Distance Metric Learning for Collaborative Image Retrieval", CVPR, 2008.
- [39] R. J. Hathaway and J. C. Bezdek, "NerF c-means: Non-Euclidean relational fuzzy clustering", Pattern Recognition, pp. 429-437, 1994.
- [40] F. Alimoglu, E. Alpaydin, "Methods of Combining Multiple Classifiers Based on Different Representations for Pen-based Handwriting Recognition," TAINN, June 1996, Istanbul, Turkey.
- [41] E. Levine and E. Domany, "Unsupervised estimation of cluster validity using resampling", Neural Computation, pp. 2573-2593, 2001.
- [42] J. Bezdek, "Pattern Recognition with fuzzy objective function algorithm", Plenum Press, New York, 1981.
- [43] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise", KDD-96, 1996.
- [44] E. Gustafson and W. Kessel, "Fuzzy clustering with a fuzzy covariance matrix", IEEE CDC, 1979.
- [45] Krishnapuram, R. and Keller, J., "A Possibilistic Approach to Clustering", IEEE Transactions on Fuzzy Systems, Vol. 1, No. 3, pp. 222-233, 1993.
- [46] Weiming H., Nianhua X., Li L., Xianglin Z., Stephen M., "A survey on Visual Content Based Video Indexing and Retrieval", IEEE Transactions on Systems, Man, and Cybernetics, Vol. 41, no. 6, pp. 797-819, 2011.
- [47] Lew S., N. Sebe, C. Djeraba, R. Jain, "Content-based multimedia information retrieval: State of the art and challenges," ACM Trans. Multimedia Comput. Commun. Appl., vol. 2, no. 1, pp. 1-19, 2006.
- [48] Rasheed Z., Y. Sheikh, and M. Shah, "On the use of computable features for film classification," IEEE Trans. Circuits Syst. Video Technol., vol. 15, no. 1, pp. 52-64, 2005.

- [49] Mittal A. and L. F. Cheong, "Addressing the problems of Bayesian network classification of video using high dimensional features," *IEEE Trans. Knowl. Data Eng.*, vol. 16, no. 2, pp. 230–244, 2004.
- [50] Browne P. and A. F. Smeaton, "Video retrieval using dialogue, keyframe similarity and video objects," in *Proc. IEEE Int. Conf. Image Process.*, vol. 3, pp. 1208–1211, 2005.
- [51] Frigui H., N. Boujemaa, and S. A. Lim, "Unsupervised clustering and feature discrimination with application to image database categorization", in *Proceedings of NAFIPS*, 2001.
- [52] Yeung M. and B.-L. Yeo, and B. Liu. Extracting story units from long programs for video browsing and navigation. In *Proc. International Conference on Multimedia Computing and Systems*, 1996.
- [53] Zhong D., et al. Clustering methods for video browsing and annotation. In *Proc. SPIE Conference on Storage and Retrieval for Image and Video Databases*, CA, 1995.
- [54] Ben Ismail M. M., Hichem Frigui: "Image Database Categorization using Robust Unsupervised Learning Of Finite Generalized Dirichlet Mixture Models". *ICIP*, pp. 2457 - 2460, 2011.
- [55] Faloutsos C., R. Barber, M. Flickner, J. Hafner, W. Niblack, D. Petkovic, and W. Equitz. Efficient and effective querying by image content. *Journal of Intelligent Information Systems*, 3:231–262, 1994.
- [56] Pentland A., R. W. Picard, and S. Sclaroff. Photobook: Content-based manipulation of image databases. *Proc. SPIE Vol 2185: Storage and Retrieval for Image and Video DB II*, 1994.
- [57] Minka T. P. and R. W. Picard. Interactive learning using a 'society of models'. Technical Report, MIT, 1995.
- [58] Sze K. W., K. M. Lam, and G. P. Qiu, "A new key frame representation for video segment retrieval," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 9, pp. 1148–1155, 2005.
- [59] Besiris D., et al, "Key frame extraction in video sequences: A vantage points approach," in *Proc. IEEE Multimedia Signal Process.*, Athens, Greece, pp. 434–437, 2007.
- [60] Girsensohn A. and J. Boreczky, "Time-constrained keyframe selection technique," *Multimedia Tools Appl.*, vol. 11, no. 3, pp. 347–358, 2000.
- [61] Gibson D., et al, "Visual abstraction of wildlife footage using Gaussian mixture models and the minimum description length criterion," in *Proc. IEEE Int. Conf. Pattern Recog.*, vol. 2, pp. 814–817, 2002.
- [62] Yu X. D. , L. Wang, Q. Tian, and P. Xue, "Multilevel video representation with application to keyframe extraction," in *Proc. Int. Multimedia Modelling Conf.*, pp. 117–123, 2004.
- [63] Rodriguez M. D., et al, "Action MACH: A Spatio-temporal Maximum Average Correlation Height Filter for Action Recognition", *IEEE ICPR*, 2008.
- [64] Manjunath, J. O., Vinod V. V., Akio Y., "Color and Texture Descriptors," *IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY*, vol. 11, no. 6, pp. 703-715, 2001.