Semi-supervised Activity Discovery via Nonlinear Dimension Reduction

Abstract. A key question in analyzing large datasets is identifying the number of clusters. We observe that most of the proposed approaches for determining the number of clusters do not provide clear values; partly this is due to unstable convergence around the actual number of clusters. Here, we propose a semi-supervised approach to estimate the number of classes in the context of unsupervised discovery of action categories. We estimated the number of classes by maximizing the mutual agreement between the manual input using cluster ensemble approach. We use agglomerative hierarchical clustering using a distance measure based on mutual acceptability of the HMM models. We find (to our surprise) that the proposed semi-supervised approach performs as well as the best supervised approaches on real life human activity recognition datasets.

1 Introduction

Time series data labelling or clustering has attracted considerable interests among variety of real world applications such as pattern recognition in finance and statistics, gene expression clustering, speech recognition or activity recognition from videos. We focused on clustering human activity videos in this work. Labelling large temporal data sets such as videos is more expensive than static data; and formulating a reliable oracle may require multiple opinions. Nonetheless, most activity classification approaches have been supervised [4, 10, 20, 30, 29, 24, 12, 1].

Unsupervised classification significantly reduces the requirement for expensive oracle consultations: once the N input instances have been reliably grouped into k clusters ($k << N$), the results can be tagged with category names based on k oracle consultations. A significant challenge for unsupervised approaches has been that of identifying the number of clusters k and achieving a high degree of reliability. The few supervised approaches for activity recognition often use prior knowledge of the number of categories (and also use manual intervention in the pre-processing [15, 2, 13]), which is generally unknown for the new datasets.

There have been many attempts to formulate a measure of optimal clustering in the literature such as co-association matrix based in consensus clustering [14], silhouette index [18] and gap statistic [25] and prediction based resampling [6]. They hold promise for identifying the number of clusters, in practice many of these are not able to predict k accurately for high dimensional time series data.

There are two approaches for temporal data classification: a) *sequentiality preserving* : Generative models like Hidden Markov Model (HMM)[32] Discriminative models like Conditional Random Fields (CRF) [21] and Dyanamic Time Warping (DTW), [26]. b) *sequentiality-agnostic*: bag of spatio-temporal interest points [15, 10] and set distance [30]. For recognizing activity sequences, most unsupervised approaches have been sequence-agnostic, and have not achieved high rates of performance. However, the complexity of markovian analysis makes it difficult to apply sequentiality-preserving techniques to high-dimensional datasets such as image sequences [16]. Another difficulty with sequence-preserving clustering of activities arises because each action sequence can be of different length. Conventional distance measures operate on a uniform-dimensional space and require each data point to be represented as a fixed length vector. These approaches thus cannot be applied directly to image sequences.

In this work, we propose two innovations to handle clustering of high-dimensional temporal data under the sequentiality-preserving paradigm. Specifically, the contributions of this work are:

1. *HMM distance measure on low-dimensional embedding*: Markovian processes are computationally expensive [16] and cannot work on sequences of large images. Hence, we use a locality preserving projections (LPP) which preserve the local structure of action sequence [8] and maps each action sequence to low dimension manifold. Next, we train N different HMMs on each action instance in the dataset. As a distance measure we use the mutual acceptability of the HMM models, and run hierarchical clustering with complete linkage.
2. *Semi-supervised ensemble Clustering:* We proposed a clustering algorithm which rely on small set of labelled data before reaching at final partitions. Using a fractional cross-validation approach, we apply hierarchical clustering repeatedly on the data to obtain a cluster ensemble. We note that the clusters detected are for slightly different subsets of the dataset, and hence it is an uneven ensemble. For identifying the number of clusters ($\hat{k}$) , we made our decision by the amount of shared information between the manual input and various clustering solution.

Finally, we validate our approach on two standard activity datasets, and show that our results (100% for Weizmann and 95.24% for Keck), is higher than most other approaches, even with supervision [1, 4, 30, 29, 24].

The flow diagram of our work is presented in figure 1. We next outline the two main contributions: the HMM based distance measure on the low-dimensional embedding; and the Semi-supervised Ensemble Clustering for estimating the number of clusters.

2 Dimension Reduction

The dimesionality of action sequence is very high, and we need to reduce the dimensionality of the input data before we can compute distance matrix which is used in clustering. We assume the set of vectors $\mathbf{x}$ in a high dimensional space $\mathcal{X} \subset \mathbf{R^D}$ have been sampled from a low-dimensional manifold embedded in that space, so that

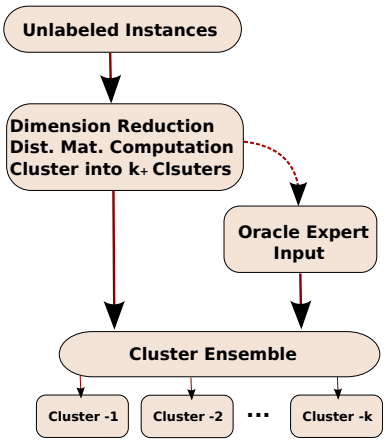


Figure 1. High level conceptual diagram of proposed semi-supervised clustering algorithm (Each steps is discussed in following sections)

any local neighbourhood on this manifold is homeomorphic to a disk in $\mathbf{R}^{\mathbf{d}}$ ($d \ll D$). Manifold discovery constitutes finding a mapping $f : \mathcal{X} \rightarrow \mathcal{Y}$, where $\mathcal{Y} \subset \mathbf{R}^{\mathbf{d}}$.

The mapping f in many situations (e.g when mapping from images to the human motion space) are very likely to be non-linear, and several techniques for discovering such mappings have been proposed within the last decade [23, 19, 3]. However, non-linearity imposes severe restrictions on using the mapping interpolation is not possible, and even a small change to the input set means that the entire mapping must be done afresh.

For this purpose, it has become important to seek Linear projections which however, preserve certain properties of the original image space. He et al. proposed Locality Preserving Projections (LPP)[8] that finds the optimal linear approximations to the eigenfunctions of the Laplace Beltrami operator on the data manifold.

The objective function of LPP is as follows:

$$\min \sum_{i,j} W_{ij}(\mathtt{y_i} - \mathtt{y_j})^2 \tag{1}$$

where $\mathtt{y_i}$ is the low-dimensional representation of $\mathtt{x_i}$ and the matrix W is a weight matrix. A possible way of defining W is as follows:

$$W_{i,j} = \begin{cases} \exp(-\frac{\|\mathtt{x_i}-\mathtt{x_j}\|^2}{t}) & \text{if } \mathtt{x_i} \text{ \& } \mathtt{x_j} \text{ are neighbors} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

The objective function with our choice of symmetric weights $W_{ij}(W_{ij} = W_{ji})$. W_{ij} incurs a heavy penalty if neighboring points $\mathtt{x_i}$ and $\mathtt{x_j}$ are mapped far apart, i.e., if $(\mathtt{y_i} - \mathtt{y_j})^2$ is large. Therefore, minimizing it is an attempt to ensure that, if $\mathtt{x_i}$ and $\mathtt{x_j}$ are "close," then $\mathtt{y_i}$ and $\mathtt{y_j}$ are close as well. The LPP embedding is as follows:

$$\mathtt{x_i} \rightarrow \mathtt{y_i} = \mathtt{A}^{\mathrm{T}}\mathtt{x_i} \tag{3}$$

A is transformation matrix which gives maps high dimensional points to low dimensional manifold.

Although LPP is linear, it differs from linear mappings like PCA which attempt to preserve the global structure. PCA identifies a subspace spanned by the largest eigenvectors of the global covariance matrix while LPP obtains a subspace spanned by the smallest eigenvectors of the local covariance matrix. The mapping results in a set of eigenvalues that characterize all points in the input space in the same manner, yet incorporating neighborhood information of the data set.

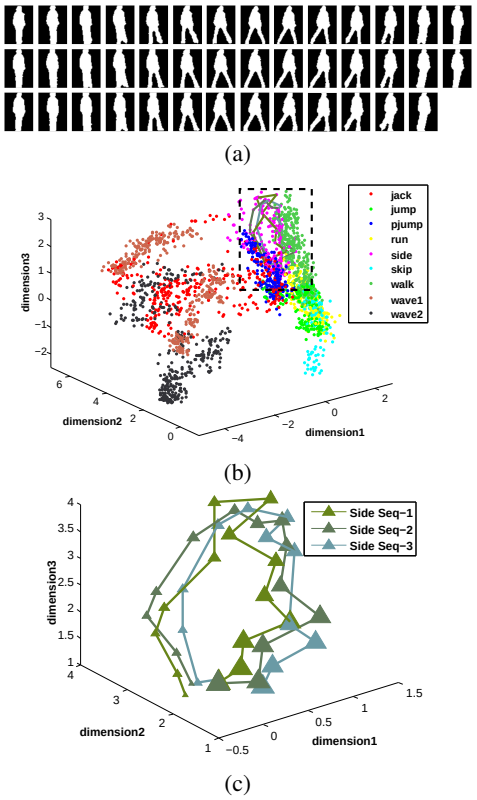


Figure 2. *LPP projection space*. b) Visualization of *side* action from the Weizmann dataset by three different subjects (each row in (a)). Note the analogous nature of the three 1-D sequences, shown enlarged in (c). The markers (triangles) in (c) denotes the poses and the size of markers is proportional to temporal order of pose in particular action sequence. For this dataset, the poses are also separable in each cluster, which is why set distance metrics like Hausdorff distance works quite well.

Further, because it is a linear mapping, LPP can generate an explicit map, which one may visualize and project the new test points onto Fig. 2. This linearity naturally leads to low computation complexity and is, thus, more efficient for practical applications. LPP can discover the nonlinear manifold structure to some extent. LPP explicitly models the manifold structure by an adjacency graph, which provides an efficient subspace learning algorithm to discover the intrinsic structure of the action space.

3 Distance Measure for Action Sequence

This section briefly describes some of the popular sequential approaches used for the activity data. Among the most popular are HMM and DTW. Both methods are sequential approaches which consider input video activities as sequences of features and try to learn particular sequence characteristic of activity. We briefly discuss these methods in the following paragraph:

3.1 Dynamic Time Warping

In DTW, activities are represented by a template activity. The similarity between new action sequence and template action sequence is

taken as measure between two sequence [28, 31]. Although the technique is very powerful for time series data but it is computationally too demanding for many real-time applications [31].

3.2 Hidden Markov Model

Human action recognition requires mapping temporal sequences that vary in speed and position. Hidden Markov Models are well known probabilistic transition automata that can model such dynamic processes. An HMM model $\lambda = \{A, B, \pi\}$ defines a number of hidden states, a static state transition probability distribution A, the observation symbol probability distribution B and the initial state distribution π. The distributions can be learned in hill-climbing mode using the Forward-Backward algorithm. The Forward or Viterbi algorithm is used to evaluate the likelihood between observation and HMM [17].

Several approaches have been proposed for computing the distances between two HMMs. Early approaches were based on the Euclidean distance of the discrete observation probability, others on entropy, or on co-emission probability of two models. In our work, we use mutual accepetedness of the models as a measure between two sequences, in which the distance between action sequences S_1 and S_2 is computed as follows:

$$dist(S_1, S_2) = \frac{|\log \mathcal{P}(S_1|\lambda_2)|}{T_2} + \frac{|\log \mathcal{P}(S_2|\lambda_1)|}{T_1} \quad (4)$$

where $S = \{s_1, s_2...., s_T\}$, $\mathcal{P}(\frac{S}{\lambda})$ is liklihood of a sequence of S of length T with respect to given HMM model λ.

Based on this distance, we may now use the standard proximity-based approach for clustering sequences using HMMs. Given an input set of N sequences $\langle S_1, S_2...S_N \rangle$ to be clustered:

1. Fit m-state HMMs λ_i, one for each sequence S_i, where $1 \leq i \leq N$. These HMMs can be initialized in a default manner.
2. For each fitted model λ_i evaluate the $\log \mathcal{P}(S_j|\lambda_i)$ log-likelihood of each of the N sequences given model λ_i where $1 \leq i, j \leq N$
3. Compute the distance matrix $Dist(i, j) = dist(S_i, S_j)$.

We applied pairwise distance-matrix-based method (e.g., an Hierarchical agglomerative method) for clustering. This clustering works well because it produces deterministic results and it is easy to vary the number of clusters (k), by cutting the tree at an appropriate point in the hierarchy. In a completely unsupervised learning environment, the number of classes labels, k, is not known, so the different levels of similarity/generalization provided by hierarchical clustering are appropriate. We used Complete linkage (furthest neighbor) for hierarchical clustering.

4 The Cluster Ensemble Approach

Ensemble Clustering aims to avoid the bias in individual clustering algorithms by considering a range of clustering solutions[22, 14]. Clusters may be obtained by using different parameters, different algorithms altogether, or with re-sampling somewhat differing between runs. It is important for clustering ensembles to ensure some degree of diversity among the solutions so that the better one may also have a chance of showing up. The basic idea behind the ensemble clustering is that make decision about clusters after multiple run on same database.

Definition 1 (Clustering Solution) *Let $D = \{S_1, S_2...., S_N\}$ be the set of data sequences. A clustering solution defined over D as $\pi = \{c_1, c_2...., c_k\}$ where k is the number of clusters such that $\forall S_i \in D \exists c_j \in \pi$ s.t. $\{(S_i \in c_j) \wedge (k \neq j) \Rightarrow S_i \notin c_k\}$*

Let $\pi = \{\pi_1, \pi_2, \cdots \pi_M\}$ be a cluster ensemble, with clustering solutions $\pi_g = \left\{C_1^g, C_2^g, \cdots, C_{k_g}^g\right\}$ where k_g is the number of clusters in the g^{th} clustering and M is size of cluster ensemble. These ensembles are now analyzed using *Consensus function*, Γ, which posits a new partition π^* of a data set X that summarizes the information from the cluster ensemble π.

A variety of consensus procedures have been proposed, including co-association-based methods [14], voting [7], information theoretic methods based measures such as shared mutual information [22]. Other approaches to improving clustering solutions include silhouette index [18] and gap statistic [25]; however, cluster ensembles have emerged as a popular choice owing to due to robustness of the clustering solution. Each of these approaches proposes an approach based on combinatorial optimization of a single objective (sometimes constructed as a combination of objective functions. In our analysis, we find that none of these methods predict a single k very reliably across differing types of datasets. Here we propose a The aim is to identify a k that overestimates the actual k, but not by much. If the unsupervised approach has a rich enough feature set, this will then ensure that the resulting clusters have high purity.

4.1 NMI based Clustering Evaluation

Information theoretic based measures such as Mutual Information [11, 27, 22] has been used for comparing clusterings besides well known pair counting based measure Adjusted Rand Index [9] because of their strong theoretical background. Let $\{S_1, S_2, \cdots, S_N\}$ be a set of N sequences in Dataset D. Let $Dist$ is the $N \times N$ dissimilarity matrix based on the distance functions discussed in previous section.

Given two clusterings solution of D, namely π_1 , p clusters $\{\pi_{11}, \pi_{12}, \cdots, \pi_{1p}\}$ with $n_1, n_2, \cdots, n_p$ members in respective clusters and similarly π_2 , q clusters $\{\pi_{21}, \pi_{22}, \cdots, \pi_{2q}\}$ with $m_1, m_2, \cdots, m_q$ members in respective clusters. The information on cluster overlap between π_1 and π_2 can be summarized in the form of a $p \times q$ contingency table $T = \left[n_{ij}\right]_{i=1,2,\cdots,p;j=1,2,\cdots,q}$ as illustrated in Table 1, where n_{ij} denotes the number of objects that are common to clusters π_{1i} and π_{2j} . Based on this contingency table, various cluster similarity indices can be built.

The Contingency Table, $n_{ij} = \pi_{1i} \cap \pi_{2j}$

π_1/π_2	π_{21}	π_{22}	$\cdots$	π_{2q}	Sum
π_{11}	n_{11}	n_{12}	$\cdots$	n_{1q}	n_1
π_{12}	n_{21}	n_{22}	$\cdots$	n_{2q}	n_2
$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
π_{1p}	n_{p1}	n_{p2}	$\cdots$	n_{pq}	n_p
Sum	m_1	m_2	$\cdots$	m_q	$\sum_{ij} n_{ij}$

MI measures how much knowing one of the clustering solutions reduces our uncertainty about the other [5]. This property is used to measure the information shared by two clusterings, and thus, assess their similarity by MI. Let us first define the entropy of a clustering π_1. Suppose that we pick an object at random from X, then the probability that the object falls into cluster π_{1i} is:

$$P(i) = \frac{|\pi_{1i}|}{N} = \frac{n_i}{N} \quad (5)$$

We define the entropy associated with the clustering π_1 as:

$$H(\pi_1) = -\sum_{i=1}^{K} P(i) \log P(i) \quad (6)$$

Mutual Information (MI) between two clusterings:

$$I(\pi_1, \pi_2) = -\sum_{i=1}^{K}\sum_{j=1}^{K} P(i,j) \log \frac{P(i,j)}{P(i)P(j)} \quad (7)$$

where $P(i,j)$ defined as follows:

$$P(i,j) = \frac{|\pi_{1i} \cap \pi_{1j}|}{N} = \frac{n_{ij}}{N} \quad (8)$$

MI is upper bounded by both the entropy $H(\pi_1)$ and $H(\pi_2)$, i.e. $I(\pi_1, \pi_2) \leq \max\{H(\pi_1), H(\pi_2)\}$.

A normalized version of MI is quite popular in literature because of easier interpretation and comparisons [11, 22]. The normalized mutual information (NMI) used is :

$$NMI(\pi_1, \pi_2) = \frac{I(\pi_1, \pi_2)}{\max\{H(\pi_1), H(\pi_2)\}} \quad (9)$$

The NMI will be one when the two clustering solutions π_1 and π_2 are identical in number of clusters as well as cluster members and zero when the two clustering don't share any information that means the clustering is random. Hence NMI captures the quality of clustering solution.

4.2 NMI based semi-supervised clustering

In this section we describe a semi-supervised approach for estimation of the number of natural groups in the activity data. One would expect that a clustering solution with large $k >> k_0$ have highly pure sub-clusters. Our objective is to identify a sufficient $\hat{K}$ which represents an close over-estimate of k_0 , based on the information available from cluster ensemble. The only condition is that the estimation of $\hat{K}$ should be independent of clustering algorithm and of any prior knowledge of the setup for the ensemble generation. Also the clusters obtained from running the clustering algorithm with different sub-samples of dataset expected to be similar. Based on these ideas, we develop an algorithm which estimate number of classes by finding correspondence between the predicted labels and actual label for various k over a large number of subsampled dataset.

We assumed that we are initially we are given unlabeled points. The task is to predict the labels of unlabeled points using partial domain knowledge. First, we choose a sufficiently high $k = k_+$ and get cluster label L_+ from oracle expert (manual input by choosing one sequence from each cluster and then assigning its predicted label to all members of the cluster). It is reasonable to postulate that partition into k_+ clusters (π_{K_+}) will have high purity. Now we treat L_+ as the ground truth for the dataset. Further we apply hierarchical clustering to get other clustering solutions: $\{\pi_{K_{min}}, \pi_{K_{min}+1}, \cdots, \pi_{K_{max}}\}$. Next step is to compare the various clustering solution with the assumed ground truth labels L_+. This NMI value should be large when the cluster are most similar with actual clusters. That means similarity between the expert input and clustering solution expected to be increasing as we reach towards the k_0 and after crossing k_0 it should decrease. Repeated run of this process mover different subset of same dataset (created by sub-sampling) will ensure the stability of the k estimate.

The proposed algorithm receives as input as Activity Dataset $(D = \{S_1, S_2 \cdots S_N\})$ and parameters f (fraction of dataset to be selected for clustering), K_{min}, K_{max} and M (maximum number of subsamples). The important steps of the algorithm are presented in next section:

Algorithm 1 Determination of number of classes $(\hat{K})$

Input: D, M, f, K_{min}, K_{max}
Output: Optimal number of Classes $(\hat{K})$
Require: $Hiecluster$ - Hierarchical clustering algorithm
$DistFunc$ - gives $N \times N$ pairwise distance Matrix

Select $K_+ = K_{max} + 1$
$Dist \leftarrow DistFunc(D)$
$\pi_+ = Hiecluster(Dist, K_+)$
$L_+ \leftarrow$ one label for each cluster by oracle expert
foreach i = *1 to* M **do**
  Select $\lfloor f * N \rfloor$ members randomly from the D and strore in (D')
  foreach $k = K_{min}$ *to* K_{max} **do**
    $\pi_{ik} = Hiecluster(D', Dist, k)$
    $L^{ik} \leftarrow estimateLabel(\pi_{ik}, \pi_+, L_+)$
    $NMI_{ik} = NMI(L_+, L_{ik})$
  end
end
foreach $k = K_{min}$ *to* K_{max} **do**
  $NMI_m(k)$ = median($\{NMI_{1k}, NMI_{2k} \cdots NMI_{Mk}\}$)
end
$\hat{K} = \arg\max_k NMI_m(k)$

Remark: The fraction f of the dataset chosen for each run should be high enough to capture members from all clusters, otherwise the clustering solution will not be stable. In our experiment the estimate didn't depend very much on specific f. For $k = 1$, all clustering solutions will be same, so $K_{min} > 1$ to avoid extra computation. We chose $K_{min} = 2$ for our simulation.

5 Experiments and Results

In this section we describe the two datasets used for experiments, human activity datasets (Weizmaan Activity dataset [4] and Keck Gesture Dataset [12]) and then we will discuss the clustering performance on these datasets.

5.1 Datasets

The Weizmann database [4] has sequences of bending, jumping jack, jumping-forward-on-two-legs, jumping-in-place-on-two-legs, running, galloping-sideways, skipping, walking, waving-one-hand, and waving-two-hands. We used the silhouette masks obtained in [4] directly for our experiments. A representative frame from each actions (or gestures) from the both datasets has been shown in figure 3. We selected 217 sequences from weizmann dataset and 126 sequences from keck gesture dataset for our experiment. We center and normalized all silhouette frames into the same 64×48 dimension, and vectorize them (raster-scan). Note that our training set is much smaller than other similar work on this model; this was necessary to rule out image sequences with the same subject performing the same task on a different cycle.

Keck database consist of 14 different gesture classes, used as military signals: turn left, turn right, attention left, attention right, flap, stop left, stop right, stop both, attention both, start, go back, close distance, speed up and come near. Each of the 14 gestures is performed by three people. In each sequence, the same gesture is repeated three

(a) (b)

Figure 3. Representative frames from each action of Keck Gesture (a) [12] and Weizmann Activity dataset (b)[4]

Table 1. Activity Dataset Details

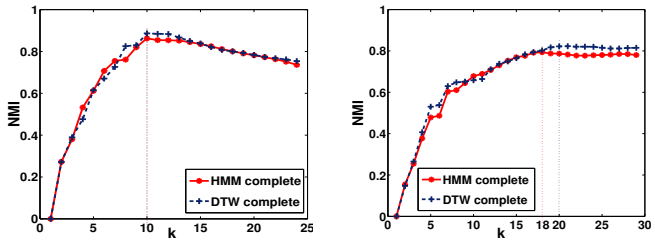
Dataset	Subjects	N	k_0
Weizmann Action [4]	10	217	9
Keck Gesture [12]	3	126	14

times by each person. Table 1 summarizes the details about the both dataset.

5.2 Discussion

We used the Matlab toolkits PMTK for HMM and simulated all other algorithms in Matlab.[1]. The estimated number of classes in the both datasets have been tabulated in Table 2. Also we used $M = 100$ and $f = 0.9$ for our simulation for both dataset.

Based on this approach, we note that the values of $\hat{K}$ as 18 for the Keck gesture dataset using HMM distance whereas DTW distance leads to slightly away from the $k_0 = 14$. For Weizmann dataset the shared mutual information is maximum at $\hat{K} = 10$ for both distance measures HMM and DTW. The results for these $\hat{K}$ are shown in figs. 4(a) and 4(b).



(a) (b)

Figure 4. [Determination of optimal number of classes using HMM (red) and DTW (blue) distance] NMI (relative to manual input) versus number of class plot for the] (a) Weizmann Dataset, both the distance measure give $\hat{k} = 10$ (b) Keck Dataset, Estimated number of classes in the dataset by HMM distance is 18 and by DTW distance is 20

We used Purity as a measure to evaluate the quality of clustering. Purity is the summation of number of elements of the dominant class in each cluster divided by size of the dataset (N). In general higher purity indicates better quality of clustering solutions.

$$Purity(\Omega, \mathbb{C}) = \frac{1}{N} \sum_{k} \max_{j} |c_k \cap \omega_j| \qquad (10)$$

[1] `http://code.google.com/p/pmtk3/` (Sept. 2011)

where $\Omega = \{\omega_1, \omega_2 .... \omega_j\}$ is the set of classes and $\mathbb{C} = \{c_1, c_2 .... c_k\}$ is the set of clusters. First we estimated the number of clusters using proposed semi-supervised approach using both distance measures. After getting an over-estimate of number of classes in the dataset we computed the purity of the clustering solution using ground truth. Table 2 summarizes the performance of the proposed algorithm for Wiezmann dataset and Table 3 for Keck Dataset. Clearly our approach outperforms existing unsupervised approaches for Keck Dataset [13] and Weizmann Dataset [15]. We got 100% accuaracy for weizmann for ($k \geq 12$) and 95.24% for Keck gesture Dataset for ($k = 18$) using complete linkage in hierarchical clustering. Surprisingly, we got better results as compare to some known supervised and semi-supervised approaches on Weizmann [10, 30, 2, 29, 24] and Keck dataset [1]. Only [4, 12] and [20] were able to perform better or comparable results on these datasets.

Table 2. Comparison of different recognition algorithms on Weizmann Action Database

Methods	Classifi.	Accu.
Blank et al.[4]	Supervised	99.61%
Niebles et al. [15]	Unsupervised	95%
Jhuang et al.[10]	Supervised	98.8%
Schindler et al.[20]	Supervised	100%
Wang at al.[30]	Supervised	92.22%
Ali et al.[2]	Semi-Supervised	92.60%
Wang et al. [29]	Supervised	97.80%
Thurau et al.[24]	Supervised	94.40%
Our approach ($k = 10$)	**Semi-supervised**	94.47%
Our approach ($k > 12$)	**Semi-supervised**	100%

Table 3. Comparison of different recognition algorithms on Keck Gesture Database

Methods	Classifi.	Accu.
Abdelkader et al. [1]	Supervised	82%
Lin et al.[12]	Supervised	95.24%
Lui et al.[13]	Unsupervised	82.7%
Our approach ($\hat{K} = 18$)	**Semi-supervised**	92.86%
Our approach ($\hat{K} = 20$)	**Semi-supervised**	95.24%

6 Conclusion

Semi-supervised algorithms have many advantages such as being able to work with partial knowledge of the domain. The main advantage of proposed algorithm is that there is no need for extracting

features by hand, etc. We adapt a pragmatic approach towards handling one of the difficulties of the approach, that of identifying the number of clusters.

The work is demonstrated on a problem of practical interest. Probabilistic sequence models naturally encode the sequence of most activity but are difficult to implement on high dimensional inputs. First, we show that action sequence is preserved on manifolds embedded in the image space. Next, we demonstrate that our proposed approach is able to find a good overestimate the number of classes for both Weizmann and Keck datasets. Our approach outperforms existing supervised algorithms which is a major contribution of this work. Also this applicarion of the approach is not only limited to acitivity clustering, it can be directly used in other domains which deal with high dimensional time series data.

REFERENCES

[1] M.F. Abdelkader, W. Abd-Almageed, A. Srivastava, and R. Chellappa, 'Silhouette-based gesture and action recognition via modeling trajectories on riemannian shape manifolds', *CVIU*, (2010).

[2] S. Ali, A. Basharat, and M. Shah, 'Chaotic invariants for human action recognition', in *ICCV*, pp. 1–8. IEEE, (2007).

[3] Mikhail Belkin and Partha Niyogi, 'Laplacian Eigenmaps for Dimensionality Reduction and Data Representation', *Neural Computation*, **15**, (2003).

[4] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, 'Actions as space-time shapes', in *ICCV*, volume 2, pp. 1395–1402. IEEE, (2005).

[5] T.M. Cover, J.A. Thomas, J. Wiley, et al., *Elements of information theory*, volume 6, Wiley Online Library, 1991.

[6] S. Dudoit and J. Fridlyand, 'A prediction-based resampling method for estimating the number of clusters in a dataset', *Genome biology*, **3**(7), research0036, (2002).

[7] A. Fred, 'Finding consistent clusters in data partitions', *Multiple Classifier Systems*, 309–318, (2001).

[8] Xiaofei He and Partha Niyogi, 'Locality preserving projections', in *In NIPS*, volume 16. MIT Press, (2003).

[9] L. Hubert and P. Arabie, 'Comparing partitions', *Journal of classification*, **2**(1), 193–218, (1985).

[10] H. Jhuang, T. Serre, L. Wolf, and T. Poggio, 'A biologically inspired system for action recognition', in ICCV, pp. 1–8. IEEE, (2007).

[11] A. Kraskov, H. Stögbauer, R.G. Andrzejak, and P. Grassberger, 'Hierarchical clustering using mutual information', *EPL (Europhysics Letters)*, **70**, 278, (2005).

[12] Z. Lin, Z. Jiang, and L.S. Davis, 'Recognizing actions by shape-motion prototype trees', in *ICCV*, pp. 444–451. IEEE, (2009).

[13] Y. Lui, 'Tangent bundles on special manifolds for action recognition', *CSVT*, (99), 1–1, (2011).

[14] S. Monti, P. Tamayo, J. Mesirov, and T. Golub, 'Consensus clustering: a resampling-based method for class discovery and visualization of gene expression microarray data', *Machine learning*, **52**(1), 91–118, (2003).

[15] J.C. Niebles, H. Wang, and L. Fei-Fei, 'Unsupervised learning of human action categories using spatial-temporal words', *IJCV*, **79**(3), 299–318, (2008).

[16] H. Othman and T. Aboulnasr, 'A separable low complexity 2d hmm with application to face recognition', *PAMI*, **25**(10), 1229–1238, (2003).

[17] L. Rabiner and B.H. Juang, 'Fundamentals of speech recognition', (1993).

[18] P.J. Rousseeuw, 'Silhouettes: a graphical aid to the interpretation and validation of cluster analysis', *IJCAM*, **20**, 53–65, (1987).

[19] Sam T. Roweis and Lawrence K. Saul, 'Nonlinear dimensionality reduction by locally linear embedding', *Science*, **290**, 2323–2326, (2000).

[20] K. Schindler and L. V. Gool, 'Action snippets: How many frames does human action recognition require?', in *CVPR*, pp. 1–8. IEEE, (2008).

[21] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas, 'Conditional models for contextual human motion recognition', in *ICCV*, volume 2, pp. 1808–1815. IEEE, (2005).

[22] A. Strehl and J. Ghosh, 'Cluster ensembles—a knowledge reuse framework for combining multiple partitions', *JMLR*, **3**, 583–617, (2003).

[23] Joshua B. Tenenbaum, Vin Silva, and John C. Langford, 'A Global Geometric Framework for Nonlinear Dimensionality Reduction', *Science*, **290**(5500), 2319–2323, (2000).

[24] C. Thurau and V. Hlavác, 'Pose primitive based human action recognition in videos or still images', in *CVPR*, pp. 1–8. IEEE, (2008).

[25] R. Tibshirani, G. Walther, and T. Hastie, 'Estimating the number of clusters in a data set via the gap statistic', *JRSS:B (Statistical Methodology)*, **63**(2), 411–423, (2001).

[26] A. Veeraraghavan, A.K. Roy-Chowdhury, and R. Chellappa, 'Matching shape sequences in video with applications in human movement analysis', *PAMI*, **27**(12), 1896–1909, (2005).

[27] N.X. Vinh, J. Epps, and J. Bailey, 'Information theoretic measures for clusterings comparison: is a correction for chance necessary?', in *ICML*, pp. 1073–1080. ACM, (2009).

[28] M. Vlachos, M. Hadjieleftheriou, D. Gunopulos, and E. Keogh, 'Indexing multi-dimensional time-series with support for multiple distance measures', in *SIGKDD*, pp. 216–225. ACM, (2003).

[29] L. Wang, X. Geng, C. Leckie, and R. Kotagiri, 'Moving shape dynamics: A signal processing perspective', in *CVPR*, pp. 1–8. IEEE, (2008).

[30] Liang Wang and D. Suter, 'Recognizing human activities from silhouettes: Motion subspace and factorial discriminative graphical model', in *CVPR*, pp. 1 –8, (june 2007).

[31] X. Xi, E. Keogh, C. Shelton, L. Wei, and C.A. Ratanamahatana, 'Fast time series classification using numerosity reduction', in *ICML*, pp. 1033–1040. ACM, (2006).

[32] L. Xie, S.F. Chang, A. Divakaran, and H. Sun, 'Unsupervised discovery of multilevel statistical video structures using hierarchical hidden markov models', in *ICME*, volume 3, pp. III–29. IEEE, (2003).