Geometry-based Adaptive Symbolic Approximation for Fast Sequence Matching on Manifolds: Applications to Activity Analysis

Rushil Anirudh, Student Member, IEEE, Pavan Turaga, Member, IEEE,

Abstract—In this paper, we consider the problem of fast and efficient indexing techniques for human activity sequences evolving in Euclidean and non-Euclidean spaces. This problem has several applications in the areas of human activity analysis, where there is a need to perform fast search and recognition in large databases. The problem is made more challenging when features such as landmarks, contours, and stick-figures etc. are naturally studied in a non-Euclidean setting where even simple operations are much more computationally intensive than their Euclidean counterparts. We propose a geometry and data adaptive symbolic framework that is shown to enable the deployment of fast and accurate algorithms for activity recognition and motif discovery. Toward this end, we present generalizations of key concepts of piece-wise aggregation and symbolic approximation for the case of non-Euclidean manifolds. We show that one can replace expensive geodesic computations with much faster symbolic computations with little loss of accuracy in activity recognition and discovery applications. The framework is general enough to work across both Euclidean and non-Euclidean spaces, depending on appropriate feature representations without compromising on the ultra-low bandwidth, high speed and high accuracy. The proposed methods are ideally suited for real-time systems and low complexity scenarios.

Index Terms—

1. INTRODUCTION

In this paper we consider the problem of indexing and analysis of multidimensional time-sequences, particularly in the context of human activity analysis where features are modeled as sequences evolving in either a Euclidean or non-Euclidean space. Such comparisons are computationally complex operations and are further complicated when the features used belong to non-Euclidean spaces such as shapes, optical flow, covariance matrices where underlying distance metrics are highly involved and even simple statistical operations are usually iterative. Applications that involve real-time analysis or remote monitoring require features that are simple for fast computations and/or low bandwidth transmissions while being robust enough to perform well for the desired application. Unfortunately most features in video analysis are either high dimensional, and hence not meant for low bandwidth transmissions, or have complex metrics defined in their feature spaces which makes real time implementations hard. For example, methods for video analysis have utilized non-linear metrics for shapes, textures etc. In this paper, we propose a geometry-based symbolic approximation framework, as a result of which low-bandwidth transmission and accurate real-time analysis become fairly straightforward. While this framework is applicable to general high-dimensional feature sequences, we demonstrate its utility on the problem of human activity analysis.

a) Non-Euclidean metrics for video analysis: Most traditional features in vision are studied in a Euclidean setting, however more recently there has been an interest in studying features that belong to non Euclidean spaces like manifolds. Examples of such features are shape contours, stick figures, histogram of oriented optical flow, etc. In this context, utilizing Riemannian geometric concepts has resulted in many advances in analysis of contours and landmarks. Other examples of such manifold-representations abound in computer vision literature. The space of $d \times d$ covariance matrices or tensors which appear both in medical imaging as well as texture analysis is a Riemannian manifold. The space of linear subspaces also called the Grassmann manifold, occurs in image set-modeling, video modeling by linear dynamic systems, and tensor decomposition. Long-term complex activities are often modeled as time-varying linear dynamical systems, which can be interpreted as a sequence of points on a Grassmann manifold, providing another motivating application for the problem of indexing of manifold sequences.

b) Sequence matching in non-Euclidean spaces: For manifolds such as the above, standard notions of distance, statistics, quantization etc. need modification to account for the non-linearity of the underlying space. As a result, basic computations such as geodesic distance, finding the sample mean etc are highly involved in terms of computational complexity, and often result in long iterative procedures further increasing the computational load. Not surprisingly, many standard approaches for sequence modeling and indexing which are designed for vector-spaces need significant generalization to enable application to these non-Euclidean spaces. Here, we study the problem of fast indexing of high dimensional sequences in both Euclidean and non-Euclidean spaces, which, to the best of our knowledge, is the first attempt in this direction. The problem of indexing of static data on manifolds is a related problem, and recent hashing based approaches have been proposed. This problem is quite distinct from indexing of sequences on manifolds, where we are not interested in indexing individual points, but sequences...
of points.

Signal approximation for manifolds using wavelets \[24\] is a related technique. It differs from our work in that it is non-adaptive to the data and requires observing the entire signal before it can be approximated, while the proposed framework allows for easy real-time implementation once the symbols are learned. Recent work also dealt with modeling human activity as a manifold valued random process \[25\] where the proposed techniques are theoretically and computationally involved due to the requirement of second-order properties such as parallel transports. Another related line of work in recent years has been advances in Riemannian metrics for sequences on manifolds \[12\]. These approaches consider a sequence as an equivalent vector-field on the manifold. A distance function is imposed on such vector-fields in a square-root elastic framework. This is applied to the special case of curves in 2D and nD Euclidean spaces \[12, 15\]. While such a distance function could be utilized for the purposes of indexing and approximation of sequences, it is offset by the computational load required in computing the distance function for long sequences. Further, the generalization of these metrics to sequences on manifolds is non-trivial and not yet fully explored.

c) Computationally efficient representations of images and video: In past decade, there has been a significant progress in speeding up retrieval and indexing images \[26, 27\] which are efficient in accurately retrieving similar images from very large datasets. There have also been extensions to video retrieval \[28, 29, 30\] from very large databases. These techniques have made it possible to search accurately through large image and video data bases, but there are two main differences to our method: 1) Most methods are for high dimensional vectors in the Euclidean space and 2) Most methods deal with static points and are designed specifically for retrieval purposes, while our representation approximates sequence metrics, which can then be used for any application. Perhaps \[29\] is the most related in philosophy to our work here, but it differs in two main aspects 1) They do not consider non-Euclidean features, since they use only SIFT descriptors while we have shown our method to work with three different kinds of feature manifolds and 2) They encode sequences using circulant temporal encoding by operating in the Fourier domain, while we operate on the feature manifold itself by comparing sequences while respecting the geometry of their space. In this work, we attempt to reduce the complexity and computational time in comparing such sequences and show that it can be applied to activity analysis among other applications. To the best of our knowledge such a problem has remained un-addressed until now. Benchmark activity and events datasets such as TRECVID \[31\] and VIRAT \[32\] are popularly used to evaluate large scale activity recognition algorithms. These datasets need some pre-processing like detection and tracking since it consist of multiple views and locations for each event, as a result the features extracted are often not smooth and may not have a well defined manifold interpretation. Since our method is a geometry based approach, we will focus on datasets that allow us to use features which lie on well defined smooth manifolds. Towards this end, we propose a novel approximation method for manifold sequences, which is consistent with the underlying geometry of the manifold, which also lends itself to fast algorithms for sequence indexing, motif discovery etc. among other applications. Solving the manifold valued problem provides us a framework that is general enough to deal with even vector features, wherein they fall under the special case where the manifold is \(\mathbb{R}^n\).

d) Generalizing sequence indexing to sequences on manifolds: In the case of scalar time-series, the pattern discovery indexing is defined as “finding the \(K\)-most frequent patterns occurring in a real-valued time series \(T\)”. This problem is computationally hard in general but motif discovery with strings can be achieved in linear time. A symbolic representation for time series data could potentially exploit these techniques. A recent approach to tackle this problem has been to discretize the sequence in a way such that the representative symbolic form contains most of the information as the original sequence, but enabling much faster computations in the symbolic space. This class of approaches are broadly termed as Symbolic Aggregate Approximation (SAX) \[33\]. Several problems of indexing and motif discovery from time series have been addressed using this framework \[33, 34, 35\]. Activity recognition using accelerometer data is another interesting new application of such symbolic approaches where it is posed as a time series classification problem. For example, Berlin et al. \[36\] use a symbol extraction technique that preserves the shape of a time series, the symbolic representation of data is used to build a suffix tree that allows for pattern discovery in linear time. Such techniques could also be useful for activity recognition, search and retrieval problems from video, however the extension from 1D to multidimensional and non-Euclidean spaces is not trivial. Multidimensional extensions to SAX have also been proposed such as \[37, 38, 39\], but these are trivial extensions which perform SAX on every dimension individually without considering the geometry of the ambient space.

Further, for manifolds such as the Grassmannian or the function-space of closed curves, there is no natural embedding into a vector space, thus motivating the need for a geometry-based intrinsic approach \[40, 12\]. We show that this class of approaches can be generalized to take into account geometry of the feature space resulting in several appealing characteristics for manifold-valued time-series, as they enable us to replace highly non-linear distance function computations with much faster and simpler symbolic distance computations. We show that, by considering the geometry of the high dimensional space in which these features lie, we can achieve significant improvement in compression and speed. However, to enable this generalization one needs to extend several key concepts to the manifold setting.

The ideal symbolic representation is expected to have the following properties: a) Be able to model the data with the least possible approximation error b) Should be robust to noise and outliers c) Finally, such symbols should enable the efficient use of data structures such as suffix trees, hash tables etc. (equiprobable symbols \[33, 41, 42\]). To generate symbols with these favorable properties, we extend ideas from
Desieno’s competitive learning mechanism [43] to make it adaptive to the geometry of the space and generate equiprobable symbols.

Contributions: The following are our primary contributions. 1) We present a geometry based data-adaptive strategy for indexing time series evolving on Euclidean and non-Euclidean spaces. 2) We propose effective algorithms for this task based on competitive learning and symbolic approximation. 3) We show that recognition and discovery of human activity patterns can be achieved with almost state of the art accuracy, while obtaining compression ratios of over 200× and up to 100× faster sequence to sequence comparisons using the proposed techniques.

2. Preliminaries for Manifolds in Activity Analysis

In this section, first we briefly describe the required notions and concepts to enable formalizing the sequence indexing problem on manifolds.

A topological space $\mathcal{M}$ is called a manifold [44, 45] if it is locally Euclidean i.e. for each point $p \in \mathcal{M}$, there exists an open neighborhood $U$ of $p$ and a mapping $\phi : U \rightarrow \mathbb{R}^n$ such that $\phi(U)$ is open in $\mathbb{R}^n$ and $\phi : U \rightarrow \phi(U)$ is called a diffeomorphism. The pair $(U, \phi)$ is called a coordinate chart for the points that fall in $U$.

To analyze sequences or curves on manifolds, one needs to take recourse to understanding tangent-space and exponential mappings. A tangent-space at a point of a manifold $\mathcal{M}$ is obtained by considering the velocities of differentiable curves passing through the given point. i.e. for a point $p \in \mathcal{M}$, a differentiable curve passing through it is represented as $\gamma : (-\delta, \delta) \rightarrow \mathcal{M}$ such that $\gamma(0) = p$. The velocity $\dot{\gamma}(0)$ refers to the velocity of the curve at $p$. This vector has the same dimension as the manifold and is a tangent vector to $\mathcal{M}$ at $p$. The set of all such tangent vectors is called the tangent space $T_p(\mathcal{M})$ is always a vector-space.

The distance between two points on a manifold is measured by means of the ‘length’ of the shortest curve connecting the points known as the geodesic. The notion of length is formalized by defining a Riemannian metric, which is a map $\langle \cdot, \cdot \rangle$ that associates to each point $p \in \mathcal{M}$ a symmetric, bilinear, positive definite form on the tangent space $T_p(\mathcal{M})$. The Riemannian metric allows one to compute the infinitesimal length of tangent-vectors along a curve. The length of the entire curve is then obtained by integrating the infinitesimal lengths of tangents along the curve. i.e. given $p, q \in \mathcal{M}$, the distance between them is the infimum of the lengths of all smooth paths on $\mathcal{M}$ which start at $p$ and end at $q$:

$$d(p, q) = \inf_{\gamma : [0,1] \rightarrow \mathcal{M}, \gamma(0) = p, \gamma(1) = q} L[\gamma], \text{ where,}$$

$$L[\gamma] = \int_0^1 \sqrt{\langle \dot{\gamma}(t), \dot{\gamma}(t) \rangle} dt$$

If $\mathcal{M}$ is a Riemannian manifold and $p \in \mathcal{M}$, the exponential map $\exp_p : T_p(\mathcal{M}) \rightarrow \mathcal{M}$, is defined by $\exp_p(\nu) = \gamma_\nu(1)$ where $\gamma_\nu$ is a specific geodesic in the direction of the tangent-vector $\nu$. The inverse mapping $\exp_p^{-1} : \mathcal{M} \rightarrow T_p$ called the inverse exponential map at a ‘pole’, takes a point on the manifold and returns a point on the tangent space of the pole. For the special case of $\mathbb{R}^n$, these concepts reduce to familiar notions of subtraction and addition. The exponential map for two points $a$ and $b$ is given by: $\exp_p(a+b) = a + \alpha b$, where $\alpha$ is some scalar that determines the extent to which one must travel from $a$ to reach $b$; the inverse exponential map is computed as: $\exp_p^{-1}(b) = a - b$.

In activity analysis, one is provided with several activity sequences per each activity class, denoted by $\gamma_{i,j}(t)$, where $i,j$ refer to the $i^{th}$ example of the $j^{th}$ class. Given a new sequence, the goal is to identify the class that best describes the sequence. Many approaches have been pursued including Dynamic Time Warping (DTW) [46], dynamical modeling etc. to solve this problem in the Euclidean settings. DTW based approaches can be extended to non-Euclidean manifolds [1], but involve significant computational load as the underlying distance metric can be highly non-linear and computationally intensive. Other approaches such as dynamic modeling [47] are defined for Euclidean spaces, and require knowledge of the entire feature sequence for estimation of the model parameters.

In the next section, we describe a geometry based symbolic approximation strategy that enables fast comparisons of sequences and allows us to exploit the favorable properties of approaches such as DTW for sequence comparison on manifolds. This approach also allows a coarse reconstruction of the signal thereby enabling dynamical modeling over approximated sequences, with little effect on recognition performance.

3. Symbolic Approach for Manifold Sequences

In this section, we describe the proposed representation for manifold sequences which allows efficient algorithms to be deployed for a variety of tasks such as motif discovery, low-complexity activity recognition. Toward this end, we focus on the piece-wise aggregate and Symbolic approximation (PAA, SAX) [48, 33] formulation, and present an intrinsic method to extend it to non Euclidean spaces like manifolds. Briefly, the PAA and SAX formulation consist of the following principal ideas - A given 1D scalar time-series is first divided into windows and the sequence in each window is represented by its mean value. This process is referred to as piece-wise aggregation. Then, a set of ‘break-points’ is chosen which correspond to dividing the range of the time-series into equiprobable bins. These break-points comprise the symbols using which we translate the time series into its symbolic form. For each window, the mean value is assigned to the closest symbol, this step is referred to as symbolic approximation. This representation has been shown to enable efficient solutions to scalar time-series indexing, retrieval, and analysis problems [33].

In the manifold case, to enable us to exploit the advantages offered by the symbolic representation of sequences, we need solutions to the following main problems - a) piece-wise
aggregation: which can be achieved by appropriate definitions of the mean of a windowed sequence on a manifold, and b) symbolic approximation: which requires choosing a set of points that are able to represent the data well. Here, we discuss how to generalize these concepts for the case of manifolds.

A. Piece-wise aggregation

Given a sequence $\gamma(t) \in \mathcal{M}$, we define its piece-wise approximation in terms of local-averages in small time-windows. To do this, we first need a notion of a mean of points on a manifold. Given a set of points on a manifold, a commonly used definition of their mean is the Karcher mean $[49]$, which is defined as the point $\mu$ that minimizes the sum of squared-distance to all other points: $\mu = \text{arg min}_{x \in \mathcal{M}} \sum_{i=1}^{N} d(x, x_i)^2$, where $d$ is the geodesic distance on the manifold.

Computing the mean is not usually possible in a closed form, and is unique only for points that are close together $[49]$. An iterative procedure is popularly used in estimation of means of points on manifolds $[50]$. Since in local time windows, points are not very far away from each other, the algorithm always converges. Thus, given a manifold-valued time series $\gamma(t)$, and a window of length $W$, we compute the mean of the points in the window and this gives rise to the piece-wise aggregate approximation for manifold sequences. When we consider vectors in $\mathbb{R}^n$, this reduces to finding the standard mean of $W$ $n$-dimensional vectors. The importance of this step is to reduce the number of points within the time series, a shorter sequence is computationally much faster to compute and store, but has a trade-off with increased error of approximation as shown in figure 3.

B. Symbolic approximation

As discussed above, one of the key-steps in performing symbolic approximation for manifold-valued time-series is to obtain a set of discrete symbols. One approach would be to try clustering approaches such as K-means by using the cluster centers as symbols. Further, the authors of SAX $[33]$ emphasize on using equi-probable symbols because they achieve optimal results for fast searching and retrieval using suffix trees, hashing and Markov models. However, standard clustering approaches do not necessarily result in equi-probable distributions of their centers $[51, 52, 53]$. It is also known that when symbols are not equi-probable, there is a possibility of inducing a probabilistic bias in the process $[54]$. Further, these symbols are also robust to noise and outliers, in recognizing activities from such data. Next, we describe how to generate such equi-probable symbols and K-means symbols and study their performance in activity recognition.

1) K-Means Symbols: While any clustering approach could be used for this step, we chose K-means because it is the most widely used clustering approach and its extension to non Euclidean spaces is well understood. For a set of points $D = (U_1, U_2, \ldots, U_n)$ we seek to estimate clusters $(C) = (C_1, C_2, \ldots, C_K)$ with centers $(\mu_1, \mu_2, \ldots, \mu_K)$ such that the sum of geodesic-distance squares, $\sum_{i=1}^{K} \sum_{U_j \in C_i} d^2(U_j, \mu_i)$ is minimized. Here $d^2(U_j, \mu_i) = |\exp_{\mu_i}^{-1}(U_j)|^2$, where $\exp^{-1}$ is the inverse exponential map as described in section 2. Alternatively, K-medoids could be used for non Euclidean data, however it is known to be more computationally complex, especially as the number of data points increases. More specifically, computational complexity of K-medoids has a quadratic dependence on the number of data points, whereas for the proposed algorithms (incl. K-means), this dependence is linear in the number of data points.

2) Equi-probable Symbols: Generation of equi-probable symbols is a hard problem in general. It has been observed that a ‘conscience’ based competitive learning approach does result in symbols that are much more equi-probable than those obtained from clustering approaches. However, the algorithm described in $[43]$ is devised only for vector-spaces. Here, we present a generalization of this approach to account for non-Euclidean geometries. The tools that we build upon include computation of geodesic distances, exponential maps and inverse-exponential maps. These are known for many standard manifolds commonly occurring in computer vision applications.

The conscience mechanism starts with a set of initial symbols/exemplars. When an input data-point is presented, a competition is held to determine the symbol closest in distance to the input point. Here, we use the geodesic distance on the manifold for this task. Let us denote the current set of $K$ symbols as $\{S_1, S_2, \ldots, S_K\}$, where each $S_i \in \mathcal{M}$. Let the input data point be denoted as $X \in \mathcal{M}$. The output $y_i$ associated with the $i^{th}$ symbol is described as

$$y_i = 1, \text{ if } d^2(S_i, X) \leq d^2(S_j, X), \forall j \neq i \quad (3)$$

$$y_i = 0, \text{ otherwise}$$

where, $d(\cdot)$ is the geodesic distance on the manifold. Since this version of competition does not keep track of the fraction of times each symbols wins, it is modified by means of a bias term to promote more equitable wins among the symbols. A bias $b_i$ is introduced for each symbol based on the number of times it has won in the past. Let $p_i$ denote the fraction of times symbol $i$ wins the competition. This is updated after each competition as

$$p_i^{\text{new}} = p_i^{\text{old}} + B(y_i - p_i^{\text{old}}) \quad (4)$$

where $0 < B << 1$. The bias $b_i$ for each symbol is computed as $b_i = C(\frac{1}{K} - p_i)$, where $C$ is a scaling factor chosen to make the bias update significant enough to change the competition (see below). The modified competition is given by

$$z_i = 1, \text{ if } d^2(S_i, X) - b_i \leq d^2(S_j, X) - b_j, \forall j \neq i \quad (5)$$

$$z_i = 0, \text{ otherwise}.$$

Finally, the winning symbol is adjusted by moving it partially towards the input data point. The key extension of this algorithm from vector space to non Euclidean spaces lies in this step. In the vector-space version this step is achieved by $S_i^{\text{new}} = S_i^{\text{old}} + \alpha((X) - S_i^{\text{old}})z_i$, but generalization of operations such as subtraction and multiplication to manifolds is not trivial. The partial movement of a symbol towards a
data-point can be achieved by means of the exponential and inverse-exponential map as

\[ S_i^{new} = \exp_{S_i^{old}}[\alpha \exp^{-1}_{S_i^{old}}(X)z_i]. \] (6)

The proposed algorithm for conscience based equi-probable symbol learning is summarized in algorithm 1.

**Algorithm 1** Equi-probable symbol generation on manifolds.

Input: Dataset \( \{X_1, \ldots, X_n\} \in \mathcal{M} \). Initial set of symbols \( \{S_1, \ldots, S_k\} \).

Parameters: Biases \( b_i = 0 \), learning rate \( \alpha \), win update factor \( B \), conscience factor \( C \).

**while** iter \( \leq \) maxiter **do**

**for** \( j = 1 \rightarrow n \) **do**

\( i \leftarrow \min_i d^2(X_j, S_i) - b_i \)

\( z_i = 1, z_i = 0, i \neq i \)

\( S_i \leftarrow \exp_{S_i}[\alpha \exp^{-1}_{S_i}(X_j)z_i] \)

\( p_i \leftarrow p_i + B(z_i - p_i) \)

\( b_i \leftarrow C(1/k - p_i) \)

**end for**

**end while**

Next, we illustrate the strength of this approach in obtaining equiprobable symbols on manifolds. For this experiment we chose the UMD human activity dataset [5] and pre-processed it such that we obtain the outer contour of the human. A detailed discussion of the dataset, processing, choice of shape metrics etc. appears in the experiments section. Here, we performed clustering of the shapes into 5 clusters and used the centroids as symbols. We show the histograms of the symbols as obtained in fig [1]. As can be seen, both K-means and affinity propagation result in symbols that are far from equi-probable. The proposed approach results in symbols which are much closer to a uniform distribution, the convergence plots for three different datasets is shown in fig [2] it is seen that the algorithm converges quickly in all cases. Once the symbols are obtained, transforming the feature sequence to its symbolic form is performed using algorithm 2. The clustering algorithm finds the best trade-off between equi-probability and approximation error. Therefore K-means clustering may result in lower approximation error but not necessarily better symbols due to noise and presence of outliers. (see figure 5(a) and 5(b))

![Figure 1: Probability Density Functions of the labels generated using (a) K-Means clustering, (b) Affinity Propagation and (c) Equi-Probable Clustering](image)

![Figure 2: Convergence for the equi-probable clustering algorithm on different feature manifolds - Grassmannian (UMD), Hypersphere (Weizmann) and \( \mathbb{R}^N \) (UCSD). Entropy is plotted as a measure of equi-probability.](image)
generalize to manifolds and thus the need for a training phase. It should also be noted that finding equi-probable clusters is slower than K-means, however since this is done only once during the training phase, it has no effect on the speed of comparisons during testing. Further, we run both algorithms until some convergence criterion has been met thereby ensuring that we don’t sacrifice quality of clusters for speed in training.

C. Limitations and special cases

Here, we discuss the limitations and some special cases of the proposed formulation. The overall approach assumes that a training set can be easily obtained from which we can extract the symbols for sequence approximation. In the 1D scalar case, this is not an issue, and one assumes that data distribution is a Gaussian, thus the choice of symbols can be obtained in closed-form without any training. If data is not Gaussian, a simple transformation/normalization of the data can be easily performed. In the manifold case, there is no simple generalization of this idea, and we are left with the option of finding symbols that are adapted for the given dataset.

For the special case of $\mathcal{M} = \mathbb{R}^n$, the approach boils down to familiar notions of piece-wise aggregation and symbolic approximation with the additional advantage of obtaining data-adaptive symbols, this ensures that the proposed approach is applicable even to the vast class of traditional features used in video analysis such as dynamic textures \cite{47}. Shape features \cite{13,14} etc. For the case of manifolds implicitly specified using samples, we suggest the following approach. One can obtain an embedding of the data into a Euclidean space and apply the special case of the algorithm for $\mathcal{M} = \mathbb{R}^n$. The requirement for the embedding here is to preserve geodesic distances between local pairs of points, since we are only interested in ensuring that data in small windows of time are mapped to points that are close together. Any standard dimensionality reduction approach \cite{55,56,57} can be used for this task. However, recent advances have resulted in algorithms for estimating exponential and inverse exponential maps numerically from sampled data points \cite{58}. This would make the proposed approach directly applicable for such cases, without significant modifications. Thus the proposed formalism is applicable to manifolds with known geometries as well as to those whose geometry needs to be estimated from data.

4. Speed up in sequence to sequence matching using symbols: applications in activity recognition and discovery

The applications considered in this paper are recognition and discovery of human activities. For recognition, a very commonly used approach involves storing labeled sequences for each activity, and performing recognition using a distance-based classifier, a nearest-neighbor classifier being the simplest one. When activity sequences involve manifold-valued time-series, distance computations are quite intensive depending on the choice of metrics. We explore here the utility of the symbolic approximation as an alternative way for approximate yet fast recognition of activities that can replace the expensive geodesic distance computations during testing. As we will show in the experiments, this is especially applicable in real-time deployments and in cases where recognition occurs remotely and there is a need to reduce the communication requirements between the sensor and the analysis engine. Before getting into the details of our experiments and distance metrics used, we define some of the terms used in this paper:

1. Activity - In this paper, we will consider an activity to be a high dimensional time series consisting of $N$ data points such that each data point is a feature extracted per frame of the original video. The features can be either Euclidean or belong to abstract spaces such as Riemannian manifolds. We consider cases where all activities may not be of equal lengths by using DTW as a distance metric.

2. Subsequence - A subsequence is defined as a contiguous subset of the larger time series, i.e. for a time series $T = (t_1, t_2, \ldots, t_n)$ a subsequence of length $n$ is $T_{i,n} = (t_i, t_{i+1}, \ldots, t_{i+n-1})$.

3. Motif Discovery - a pattern that repeats often within a larger time series is known as a motif. We say two patterns within the time series are similar if they are at a distance smaller than some threshold.

4. Trivial Match - Within a time series $T$, we say two subsequences $P$ at position $p$ and $Q$ at position $q$ are a trivial match if, $p \in (q - m + 1, \ldots, q, \ldots, q + m - 1)$ i.e $p$ and $q$ are different and within the neighborhood (as specified by $m$) of each other.

For an Activity of length $N$, we extract a symbolic representation in windows of size $W$ (where typically $W << N$). To replace geodesic distance computations for recognition, we

![Fig. 3: The trade-off between piece-wise aggregation and symbolic approximation is depicted here comparing the error in approximating the distance between two sequences from the Weizmann dataset. A symbol dictionary size of at least 40 and a approximation window size of up to 3 has negligible approximation error.](image)
will consider subsequences in their symbolic representations to calculate the distance between activities. Let \( p_{\text{sub}} \) (eg: ‘bccdea’) and \( q_{\text{sub}} \) (eg: ‘afffec’) be two such subsequences of length \( l \), then the distance metric \( d_{\text{symbol}} \), defined on symbols, is:

\[
d_{\text{symbol}}(p_{\text{sub}}, q_{\text{sub}}) = \sum_{i=1}^{l} d_{M}(D(p_{\text{sub}}(i)), D(q_{\text{sub}}(i)))
\]

where \( d_{M} \) is the metric defined on the manifold, \( D \) is the set of symbols or dictionary that is previously learned and \( D(a) \) is the point on the manifold corresponding to the symbol \( a \). Here we assume that the two sequences are of the same length, in other cases we use DTW as a metric or learn a dynamical model for each sequence and use the distance between them as a metric. Since the symbols are known apriori, the distance between them can be computed offline as part of training and stored as a look-up table of pairwise distances between symbols. This allows us to compute distances between sequences in near constant time, which is much faster than computing distances each time using DTW on actual features.

Before considering applications for the simplified distance measure, one must consider the trade-off between piecewise aggregation, number of symbols versus the error of approximation, this is shown in figure 3.

For activity discovery, we consider the problem as one of mining for motifs in time-series. In finding motifs, it is important to consider only non-trivial matches, for every such match we store its location and find the top \( k \) motifs. For each of the \( k \) motifs, we define a center for the motif as the sequence which is at minimum distance to all the sequences similar to it. These centers are the \( k \) most recurring patterns in the multidimensional time series. We use the brute-force algorithm given in [59] to extract our motifs.

5. **Experimental Evaluation**

In this section, we demonstrate the utility of the proposed algorithms for symbolic approximation and its application to activity recognition and discovery. We also study the complexity advantage in using these symbols as compared to original feature sequences. We first describe the datasets and choice of features.

a) **Datasets:** We performed experiments on three different datasets namely: the UMD Human Activity Dataset [5], the Weizmann Dataset for Human Actions [60] and the UCSD Traffic Dataset [61].

**The UMD database** consists of 10 different activities like bend, jog, push, squat etc., each activity was repeated 10 times, so there were a total of 100 sequences in the dataset.

**The Weizmann Dataset** consists of 93 videos of 10 different actions each performed by 9 different persons. The classes of actions include running, jumping, walking, side walking etc.

**The UCSD traffic database** consists of 254 video sequences of daytime highway traffic in Seattle in three patterns i.e. heavy, medium and light traffic. It was collected from a single stationary traffic camera over two days.

![Fig. 4: Sample images from the various data sets used in this paper. The Weizmann, UMD and the UCSD data sets are shown here from top to bottom in that order.](image)

**A. Choice of feature representation and metrics**

To show the generality and strength of the framework, we used a different feature on each data set such as landmarks on the silhouette of the subject for the UMD Dataset, Histogram of Oriented Optical Flow for the Weizmann Dataset and we stack the pixel values into a feature vector for the UCSD traffic dataset. For the features lying on manifolds, we present their representations and the choice of metrics used next.

**Landmarks on the Silhouette:** The background within the UMD Dataset is relatively static which allows us to perform background subtraction. From the extracted foreground, we perform morphological operations and extract the outer contour of the human. We sampled a fixed number of points on the outer contour of the silhouette to yield landmarks. The specific choice of shape space, and associated metrics are described next. To represent the points sampled on the outer contour of the human, we use an affine invariant representation of landmarks as follows. The set of landmark points is given by a \( m \times 2 \) matrix \( L = [(x_1, y_1); (x_2, y_2); \ldots; (x_m, y_m)] \), of the set of \( m \) landmarks of the centered shape. The affine shape space [62] is useful to remove the effects of small variations in camera location or small changes in the pose of the subject. Affine transforms of the base shape \( L_{\text{base}} \) can be expressed as \( L_{\text{affine}}(A) = L_{\text{base}} * A^T \), and this multiplication by a full-rank matrix on the right preserves the column-space of the matrix \( L_{\text{base}} \). Thus, the 2D subspace of \( \mathbb{R}^m \) spanned by the columns of the matrix \( L_{\text{base}} \) is an affine-invariant representation of the shape, i.e. \( \text{span}(L_{\text{base}}) \) is invariant to affine transforms of the shape. Subspaces such as these can be identified as points on a Grassmann manifold [21].

In this paper, we used extrinsic approaches for Grassmann manifold computations which are conceptually simpler and implemented more easily. A given \( d \)-dimensional subspace of \( \mathbb{R}^m \), \( Y \) can be associated with a idempotent rank-\( d \) projection...
matrix $P = YY^T$, where $Y$ is a $m \times d$ orthonormal matrix such as $\text{span}(Y) = \mathcal{Y}$. The space of $m \times m$ projectors of rank $d$, denoted by $\mathbb{P}_{m,d}$ can be embedded into the set of all $m \times m$ matrices $\mathbb{R}^{m \times m}$ which is a vector space. Using the embedding $\Pi : \mathbb{R}^{m \times m} \rightarrow \mathbb{P}_{m,d}$ we can define a distance function on the manifold using the metric inherited from $\mathbb{R}^{m \times m}$. 

$$d^2(P_1, P_2) = tr(P_1 - P_2)^T (P_1 - P_2)$$ (8)

The projection $\Pi : \mathbb{R}^{m \times m} \rightarrow \mathbb{P}_{m,d}$ is given by:

$$\Pi(M) = UU^T$$ (9)

where $M = USV^T$ is the $d$-rank SVD of $M$.

Given a set of sample points on the Grassmann manifold represented uniquely by projectors $\{P_1, P_2, \ldots, P_N\}$, we can compute the extrinsic mean $\mu_{\text{ext}}$ by first computing the mean of the $P_i$’s and then projecting it to the manifold as follows:

$$\mu_{\text{ext}} = \Pi(P_{\text{avg}}), \text{ where } P_{\text{avg}} = \frac{1}{N} \sum_{i=1}^{N} P_i$$ (10)

Histograms of Oriented Optical Flow (HOOF): As described in [10], optical flow is a natural feature for motion sequences. Directions of Optical Flow vectors are computed for every frame, then binned according to their primary angle with the horizontal axis and weighted according to their magnitudes. Using magnitudes alone is susceptible to noise and can be very sensitive to scale. Thus all optical flow vectors, $v = [x, y]^T$ with direction $\theta = \tan^{-1}(\frac{y}{x})$ in the range

$$-\frac{\pi}{2} + \frac{b - 1}{B} \leq \theta < -\frac{\pi}{2} + \frac{b}{B}$$ (11)

will contribute by $\sqrt{x^2 + y^2}$ to the sum in bin $b$, $1 \leq b \leq B$, out of a total of $B$ bins. Finally, the histogram is normalized to sum up to 1. Each frame is represented by one histogram and hence a sequence of histograms are used to describe an activity. The histograms $h_i = [h_{i,1}, \ldots, h_{i,B}]$ can be re-parameterized to the square root representation for histograms, $\sqrt{h_i} = [\sqrt{h_{i,1}}, \ldots, \sqrt{h_{i,B}}]$ such that $\sum_{i=1}^{B} (\sqrt{h_{i,j}})^2 = 1$. The Riemannian metric between two points $R_1$ and $R_2$ on the hypersphere is $d(R_1, R_2) = \cos^{-1}(R_1^T R_2)$. This projects every histogram onto the unit B-dimensional hypersphere or $\mathbb{S}^{B-1}$. From the differential geometry of the sphere, the exponential map is defined as

$$\exp_{\psi_i}(v) = \cos(||v||_{\psi_i})\psi_i + \sin(||v||_{\psi_i}) \frac{v}{||v||_{\psi_i}}$$ (12)

Where $v \in T_{\psi_i}(\Psi)$ is a tangent vector at $\psi_i$ and $||v||_{\psi_i} = \sqrt{(v, v)_{\psi_i}} = (\int_0^T v(s)v(s)ds)^{\frac{1}{2}}$. In order to ensure that the exponential map is a bijective function, we restrict $||v||_{\psi_i} \in [0, \pi]$. The truncation of the domain of the exponential map is made in accordance to the injectivity radius, which is the largest radius for which the exp map is a diffeomorphism. For the sphere, the injectivity radius is $\pi$. Points that lie beyond the injectivity radius have a shorter path connecting them to $\psi_i$, which determines their geodesic distance incorrectly. The logarithmic map from $\psi_i$ to $\psi_j$ is then given by

$$\log_{\psi_i}(\psi_j) = \frac{\int_0^T u(s) u(s)ds}{\int_0^T u^2(s)ds} \cos^{-1}(\psi_i, \psi_j),$$ (13)

with $u = \psi_i - (\psi_i, \psi_j) \psi_j$.

B. Activity discovery experiment

For this experiment, we randomly concatenated 10 repetitions of 5 different activities of the UMD dataset to create a sequence that was 50 activities long. Each activity consists of 80 frames which were sampled by a sliding window of size 20 frames with step size of 10 frames. After symbolic approximation, this resulted in 6 symbols per activity, chosen from an alphabet of 25 symbols. The motifs or repeating patterns, in five activities - Jogging, Squatting, Bending Knees, Waving and Throwing were discovered automatically using the proposed method. Each of the discovered motifs was validated manually to obtain a confusion matrix shown in table 1.

As can be seen, it shows a strong diagonal structure, which indicates that the algorithm works fairly well. Even though all executions of the same activity are not found, we do not find any false matches either.

| Activity Type       | Motif 1 | Motif 2 | Motif 3 | Motif 4 | Motif 5 |
|---------------------|---------|---------|---------|---------|---------|
| Jogging             | 7       | 0       | 0       | 0       | 0       |
| Squatting           | 0       | 7       | 0       | 0       | 0       |
| Bending Knees       | 0       | 0       | 8       | 0       | 0       |
| Waving              | 0       | 0       | 0       | 9       | 0       |
| Throwing            | 0       | 0       | 0       | 0       | 8       |

TABLE 1: Confusion matrix for the discovered motifs on the UMD database using a shape feature.

C. Non-equiprobable symbols

While the proposed formalism supports arbitrary symbols to be used, equiprobable symbols are a desirable feature in several applications [33]. However in our experiments we found that using equiprobable symbols, as generated by the conscience learning mechanism described in Sec 3-B, led to a recognition accuracy that was inferior to that obtained while using symbols generated by K-means. This is because reconstruction error (and therefore signal fidelity) is compromised to generate equiprobable symbols. However, even though K-means gives relatively better performance, its symbols are far from equiprobable. Therefore, as an intermediate solution, we considered a “hybrid”, which performs clustering in two stages. In the first stage we cluster the data using K-means into a small number of clusters. Each of these clusters is further split into smaller, equiprobable “sub-clusters” in the second stage. It should be noted that since this is done in the training stage, it does not affect the speed of translating features to symbols or the speed of comparing two activities. The number of clusters in the first stage is an empirical choice, we used values in the range of 5 to 10 for each data set. The number
of sub-clusters in the second stage varies according to the probability of their parent cluster. For example, if \( p_i \) was the probability of the smallest cluster and we decide to split it into \( r \) smaller sub-clusters, then the \( i^{th} \) cluster with probability \( p_i \) would be split into \( \lceil p_i \times r \rceil \) clusters. The parameter \( r \) indirectly controls the size of the final set of symbols, we used values of \( r \) in the range of 1 to 5. We chose these values to obtain similar number of clusters as K-means (~40 – 50) for a fair comparison. As we show, these symbols out-performed the K-means symbols in all our experiments.

D. Activity recognition using symbols

Symbolic approximation plays a significant role in reducing computational complexity since it allows us to work with symbols instead of working with high dimensional feature sets. In this experiment, we test the utility of the proposed symbolic approximation method for fast and approximate recognition of activities over three datasets. For each data set picking the number of symbols, \( K \) is an empirical choice, typically we picked \( K = K_{\min} \) where, for all \( K > K_{\min} \) the recognition performance shows no improvement. As shown in fig 6(c) using the DTW metric \( K_{\min} \) would lie between 60 to 80. We also picked a window size of \( W = 1 \) in our recognition experiments to achieve best performance. A detailed comparison between the window size, number of symbols and performance is given by the authors of SAX in [33]. In our experiments, we also chose to use the nearest neighbor classifier over others because it has been shown to be superior in performance as compared to many other classifiers [65], [66], and is also simple to execute. For the UMD dataset, we learned a dictionary of 60 symbols using algorithm [1]. Then, we performed a recognition experiment using a leave one-execution-out test in which we trained on 9 executions and tested on the remaining execution, the results are shown in Table 2. It can be seen that the recognition performance using symbols is very close to that obtained by using an oracle geodesic distance DTW based algorithm. We achieve this performance with matching times that are significantly faster, as will be described in section 5-E.

For the Weizmann dataset, we demonstrate the flexibility of the approximation strategy by learning linear dynamical models over the approximated sequences, which also serves as a fair comparison to the state of the art techniques. We performed the recognition experiment on all the 9 subjects performing 10 activities each with a total of 90 activities. The dictionary learned had 55 symbols which were used to map the activities to the approximated sequences. Next, we fit a linear dynamical model to the approximately reconstructed actions and perform recognition with a nearest neighbor classifier using the Martin metric on LDS parameters [47]. The results for the leave-one-execution-out recognition test are shown in Table 3 and it can be seen there is almost no loss in performance in comparison to state of the art techniques. Better results have been reported on this dataset by Gorelick et al. [60] etc., but there are no common grounds between their technique or feature and ours for it to be a fair comparison.

For the Traffic Database, we stacked every other pixel in the rows and columns of each frame to form our feature vector. We learned 45 symbols from the training set using these features. We performed the recognition experiment on 4 different test sets which contained 25% of the total videos. We used a 1-NN classifier with a DTW metric on the symbols. The results are shown in Table 4. We compare our results to Sankaranarayanan et al. [68] who used a similar feature. As it can be seen, recognition performance is clearly better when the feature is in its symbolic form as compared to when it was compressively sensed, given that both are significantly reduced versions of

| TABLE 3: Recognition Performance for the Weizmann dataset using the HOOF features. It is observed here that the approximated sequences perform almost as well as the original features sequences. |
|---|---|---|
| Expt 1 | Proposed - DTW | CS LDS | Oracle LDS |
| | 84.13 | 85.71 | 77.77 |
| | 82.81 | 73.43 | 82.81 |
| | 79.69 | 78.10 | 91.18 |
| | 79.37 | 76.10 | 80.95 |
| Average (%) | **81.50** | 78.33 | 83.25 |

| TABLE 4: Recognition performance2 for UCSD traffic data set. |
|---|
| Step | Complexity |
|---|
| Exponential map for \( \mathcal{M} \) (manifold specific) | \( O(\nu) \) |
| Inverse exponential map for \( \mathcal{M} \) (manifold specific) | \( O(\chi) \) |
| Intrinsic K-means clustering | \( O((NK+K\nu)\Gamma) \) |
| Equi-probable clustering | \( O((NK\chi+4N\nu)\Gamma) \) |
| Approximation of N-length activity to M symbols | \( O(M(w+\nu)\Gamma+MK\chi) \) |
| Symbolic DTW | \( O(M^2\delta) \) |
| Geodesic distance DTW | \( O(M^2\chi) \) |

| TABLE 5: Theoretical complexity analysis for the proposed algorithms. Notations used: N - number of data points, K - number of symbols, with \( O(\delta) \) the time required to read from memory, \( \Gamma \) maximum number of iterations, M and w are as defined in algorithm 2 and are usually much lesser than N. It can be seen that a huge complexity gain is achieved in using symbols over original features. |
|---|---|
| Step | Complexity |
|---|
| Exponential map for \( \mathcal{M} \) (manifold specific) | \( O(\nu) \) |
| Inverse exponential map for \( \mathcal{M} \) (manifold specific) | \( O(\chi) \) |
| Intrinsic K-means clustering | \( O((NK+K\nu)\Gamma) \) |
| Equi-probable clustering | \( O((NK\chi+4N\nu)\Gamma) \) |
| Approximation of N-length activity to M symbols | \( O(M(w+\nu)\Gamma+MK\chi) \) |
| Symbolic DTW | \( O(M^2\delta) \) |
| Geodesic distance DTW | \( O(M^2\chi) \) |
the original feature. We also perform nearly as well as the performance achieved using the original feature itself.

**E. Robustness to Outliers**

In this section we perform experiments to test the robustness of our competitive learning algorithm in learning symbols from data that is corrupted with outliers. We show that equi-probable symbols exhibit better recognition performance in addition to having other advantages like faster search and retrieval compared to other types of symbols. To test this, we performed an experiment on two types of data 1) synthetic data and 2) the UMD actions dataset. We tested the performance of both types of symbols learned from data corrupted by \( M \) outliers, where \( M \) varies between \( 0 \) − \( 15\% \) of the total data points. For the synthetic data, we generated 2D Gaussian noise with three distinct means and added \( M \) points of random Gaussian noise with means that were far away from the original centers as outliers. In figure 5(a) we demonstrate the effectiveness of modeling the original patterns using symbols learned from corrupted data. We calculate the centers of this corrupted data from both K-means and the competitive algorithm. We identify the corresponding pairs of original and estimated centers by the distance between them. Next, we average the distance between pairs of calculated mean and original mean for all the centers, and further average it over 20 iterations of randomly chosen initializations. It can be seen that the equi-probable centers are much closer to the original centers as compared to those generated by K-means.

For the UMD dataset, we generate random noise that is consistent with the dimensionality of the manifold, and enforce constraints such that these outliers also lie on the Grassmann manifold. To show the robustness of equi-probable symbols from corrupted data, we learned 3 and 5 symbols of both types and perform recognition as described earlier. Since our initialization is random, we repeat this experiment over 5 iterations and report the average recognition accuracy vs number of outliers in fig 5(b). It can be seen that at very low number of symbols, equi-probable symbols are able to recognize actions nearly 5−10% better than K-means symbols, demonstrating their robustness to outliers.

**F. Speed up and compression achieved using symbols**

A theoretical complexity analysis of the algorithm is shown in table 3. We also consider three metrics to study the time-complexity of the proposed framework. Namely 1) Time complexity of matching using symbols vs original feature sequences, 2) Time required to transform a given activity into a symbolic form, and 3) Number of bits required to store/transmit symbols as compared to feature sequences. Ideally, we require that the matching time be several orders of magnitude faster than using the original sequences, the transformation time to be small enough to enable real-time approximation, and very small bit-rate/storage requirement compared to original feature sequences. We show in the following that the proposed framework successfully satisfies all these criteria. We performed the experiments using MATLAB, on a PC with an i7 processor operating at 3.40Ghz with 16GB memory on Windows 7.

1) **Analysis of matching time:** In this experiment we show the gain in speed and compression achieved using symbols compared to using the original high-dimensional features with accompanying metrics. For the gain in speed, we measured the run-time of matching sequences using DTW on symbols vs oracle DTW. As shown in fig 6(a), the time taken to match two activity sequences using symbols is just 3.1ms which is two orders of magnitude faster than 100ms that it takes using the actual features.

2) **Analysis of approximation time:** Fig 6(b) shows the distribution of times taken over various activities to transform them into their respective symbolic forms. The average conversion time for an entire activity video is about 107ms. In other words, we can process the video at a speed of 445 frames per second (fps) which allows for easy real time implementation since most videos are recorded at 10-30fps.

3) **Bit-rate analysis:** Next, to demonstrate the gain in compression we compared our representation to a baseline using the original feature sequence. Assuming each dimension of the feature is coded as a 32-bit float number, we calculated the bits it would take to represent each feature and its symbolic representation. As shown in table 6 an activity video that is 4 seconds long is coded using 0.4Kb in its symbolic form. For a dictionary of size \( K \), the number of bits required to represent each symbol is \( \log_2(K) \). This provides enough flexibility for the user to choose the size of the dictionary and pick features of their choice without significantly affecting the bit-rate.

4) **Offline symbol learning:** Before coding the features, a dictionary needs to be learned offline. For the sake of clarity, we show the time and number of bits taken to learn this dictionary in table 7. Even though this is being done offline, it offsets the purpose of performing fast and low complexity analysis if it is extremely slow. However, a dictionary with 60 symbols even for a complex feature like shape silhouettes takes less than 30 minutes to learn, while smaller simpler features like the HOOF takes under 22 seconds. It is also seen that it takes lesser bits to store the dictionary than the actual feature.

| Activity    | Pick Object | Jog | Push | Squat | Wave | Kick | Bend | Throw | Turn | Talk | Average (%) |
|-------------|-------------|-----|------|-------|------|------|------|-------|------|------|-------------|
| Hybrid Symbols | 100         | 100 | 100  | 90    | 100  | 100  | 100  | 90    | 100  | 100  | 98          |
| K-means Symbols | 90          | 100 | 100  | 90    | 100  | 100  | 100  | 90    | 90   | 100  | 96          |
| Shape DTW | 100         | 100 | 100  | 100   | 100  | 100  | 100  | 100   | 100  | 100  | 100          |

**TABLE 2:** Recognition experiment for the UMD database with a shape silhouette feature. Here we see the performance achieved with symbolic approximation compared to an oracle geodesic distance based nearest neighbor classifier.
Fig. 5: Robustness to outliers - (a) shows that equi-probable symbols are able to model the underlying patterns significantly better than K-means symbols with a much lower distance between true and estimated centers. Fig 5(b) shows the improvement in recognition accuracy by using equi-probable symbols instead of K-means symbols for alphabets of size 3 and 5 symbols.

Fig. 6: Comparison of histograms for matching times when using symbolic v/s original feature sequences are shown in fig 6(a) for the UCSD traffic dataset. The times are shown in milliseconds on a log scale. As it can be seen, using symbols speeds up the process by nearly two orders of magnitude. Fig 6(b) shows a histogram of times taken to translate entire activities of 50 frames into symbols from the UCSD dataset. Fig 6(c) shows the trade off between accuracy and number of symbols for the Weizmann dataset.

| Data / Feature        | Actual Data                                                                 | Proposed Representation                              | Compression Factor |
|-----------------------|----------------------------------------------------------------------------|-------------------------------------------------------|--------------------|
| Weizmann data set     | Per activity video: 70 frames × 30 × 32 bits                               | Per activity video: 6 bits/symbol × 70 frames         | 160×               |
| HOOF feature ∈ R^{30} | Number of bits required: 65,625 Kb                                         | Number of bits required: 0.410 Kb                     |                    |
| UMD data set          | Per activity video: 80 frames × 200 × 32 bits                              | Per activity video: 6 bits/symbol × 80 frames         | 1068×              |
| Shape landmarks ∈ R^{200} | Number of bits required: 500 Kb                             | Number of bits required: 0.468 Kb                     |                    |
| UCSD traffic data set | Per activity video: 55 frames × 4800 × 32 bits                            | Per activity video: 6 bits/symbol × 55 frames         | 22,237×            |
| vector feature ∈ R^{4800} | Number of bits required: 8,250 Kb                                     | Number of bits required: 0.371 Kb                     |                    |

TABLE 6: Having learned the dictionary offline, most activity representations can be efficiently expressed on a minimal bit budget.
sequences, which can be further reduced by choosing fewer symbols depending on the application. This dictionary needs to be stored/transmitted only once during the learning phase.

| Data / Feature                  | Learning the Symbol Dictionary Offline |
|---------------------------------|----------------------------------------|
| UMD data set                    | Time taken to learn 60 clusters - 1667.7s |
| Shape landmarks ∈ ℝ200          | 60 clusters × 200 × 32 = 375 Kb         |
| Weizmann data set               | Time taken to learn 55 clusters - 21.4s |
| HOOF feature ∈ ℝ30              | 55 clusters × 30 × 32 = 51.56 Kb       |
| UCSD Traffic data set           | Time taken to learn 45 clusters - 190s  |
| Vector feature ∈ ℝ4800          | 45 clusters × 4800 × 32 = 6750 Kb      |

**TABLE 7:** Time required for offline learning of symbols. Storage requirement of the dictionary is also shown.

### 6. Discussion and Future Work

In this paper we presented a formalization of high dimensional time-series approximation for efficient and low-complexity activity discovery and activity recognition. We presented geometry and data adaptive strategies for symbolic approximation, which enables these techniques for new classes of features in activity analysis. The results show that it is possible to significantly reduce Riemannian computations during run-time by an intrinsic indexing and approximation algorithm which allows for easy and efficient real time implementation. This opens several avenues for future work like an integrated approach of temporal segmentation of human activities and symbolic approximation. A theoretical and empirical analysis of the advantages of the proposed formalism on resource-constrained systems such as robotic platforms would be another avenue of research.

Finally, the framework in this paper is general enough to deal with more abstract forms of information such as graphs [69] or bag-of-words [6]. In fact, any system that is sequential can be used within this framework, the key is to have a good understanding of metrics on these abstract models. Existing works have defined kernels for data on manifolds [70], for graphs [71] and a good starting point would be to use these to develop a kernel version of this framework that would allow us to learn symbols.

### References

1. A. Veeraraghavan, A. K. R. Chowdhury, and R. Chellappa, “Matching shape sequences in video with applications in human movement analysis,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 12, pp. 1896–1909, 2005.
2. P. K. Turaga and R. Chellappa, “Locally time-invariant models of human activities using trajectories on the Grassmannian,” in *CVPR*, 2009, pp. 2435–2441.
3. L. Zhao, X. Wang, G. Sukthankar, and R. Sukthankar, “Motif discovery and feature selection for CRF-based activity recognition,” in *ICPR*, 2010, pp. 3826–3829.
4. N. Raviv, N. D. Mysore, and M. L. Littman, “Activity recognition from accelerometer data,” in *IJAAL*, 2005, pp. 1541–1546.
5. A. Veeraraghavan and A. K. R. Chowdhury, “The function space of an activity,” in *CVPR* (1), 2006, pp. 959–968.
6. U. Gaur, Y. Zhu, B. Song, and A. K. R. Chowdhury, “A ‘string of feature graphs’ model for recognition of complex activities in natural videos,” in *ICCV*, 2011, pp. 2595–2602.
7. M. F. Abdelkader, W. Abd-Almageed, A. Srivastava, and R. Chellappa, “Silhouette-based gesture and action recognition via modeling trajectories on riemannian shape manifolds,” *Computer Vision and Image Understanding*, vol. 115, no. 3, pp. 438–455, 2011.
8. M. Müller, T. Röder, and M. Clausen, “Efficient content-based retrieval of motion capture data,” in *ACM SIGGRAPH 2005 Papers*, ser. SIGGRAPH ’05. New York, NY, USA: ACM, 2005, pp. 677–685. [Online]. Available: http://doi.acm.org/10.1145/1168622.1073247.
9. D. Kendall, “Shape manifolds, Procrustean metrics and complex projective spaces,” *Bulletin of London Mathematical society*, vol. 16, pp. 81–121, 1984.
10. R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal, “Histograms of oriented optical flow and Binet-Cauchy kernels on nonlinear dynamical systems for the recognition of human actions,” in *CVPR*, 2009, pp. 1932–1939.
11. O. Tuzel, F. M. Porikli, and P. Meer, “Region covariance: A fast descriptor for detection and classification,” *European Conference on Computer Vision*, pp. II: 589–600, 2006.
12. A. Srivastava, E. Klassen, S. H. Joshi, and I. H. Jermyn, “Shape analysis of elastic curves in Euclidean spaces,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, pp. 1415–1428, 2011.
13. M. Hu, “Visual pattern recognition by moment invariants,” *IRE Transactions on Information Theory*, vol. 8, pp. 179–187, 1962.
14. S. Belongie, J. Malik, and J. Puzicha, “Shape matching and object recognition using shape contexts,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 24, pp. 509–522, April 2002.
15. S. H. Joshi, E. Klassen, A. Srivastava, and I. Jermyn, “A novel representation for Riemannian analysis of elastic curves in R^n,” in *CVPR*, 2007.
16. W. Mio, A. Srivastava, and S. Joshi, “On shape of plane elastic curves,” *International Journal of Computer Vision*, vol. 73, no. 3, pp. 307–324, 2007.
17. I. Dryden and K. Mardia, *Statistical shape analysis*. John Wiley and Sons, 1998.
18. T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham, “Active shape models - their training and application,” *Computer Vision and Image Understanding*, vol. 61, no. 1, pp. 38–59, 1995.
19. X. Pennec, P. Fillard, and N. Ayache, “A Riemannian framework for tensor computing,” *International Journal of Computer Vision*, vol. 66, no. 1, pp. 41–66, 2006.
20. J. Ham and D. D. Lee, “Extended grassmann kernels for subspace-based learning,” *Neural Information Processing Systems*, pp. 601–608, 2008.
21. P. K. Turaga, A. Veeraraghavan, A. Srivastava, and R. Chellappa, “Statistical computations on Grassmannian and Stiefel manifolds for image and video-based recognition,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 33, no. 11, pp. 2273–2286, 2011.
22. Y. M. Liu, J. R. Beveridge, and M. Kirby, “Action classification on product manifolds,” in *CVPR*, 2010, pp. 833–839.
23. R. Chaudhry and Y. Ivanov, “Fast approximate nearest neighbor methods for non-Euclidean manifolds with applications to human activity analysis in videos,” in *European Conference on Computer Vision*, Crete, Greece, September 2010.
24. I. U. Rahman, J. Drioi, V. C. Stodden, D. L. Donoho, and P. Schrder, “Multiscale representations for manifold-valued data,” *SIAM J. MULTISCALE MODEL. SIMUL.*, vol. 4, no. 4, pp. 1201–1232, 2005.
25. S. Yi, H. Krim, and L. K. Norris, “Human activity as a manifold-valued random process,” *IEEE Transactions on Image Processing*, vol. 21, no. 8, pp. 3416–3428, 2012.
26. O. Chum, M. Perdoch, and J. Matas, “Geometric min-hashing: Finding a (thick) needle in a haystack,” in *CVPR*, 2009, pp. 17–24.
27. D. Nistér and H. Stewénius, “Scaleable recognition with a vocabulary tree,” in *CVPR* (2), 2006, pp. 2161–2168.
28. H. Jégou, M. Douze, and C. Schmid, “Product quantization for nearest neighbor search,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 117–128, 2011.
29. J. Revaud, M. Douze, C. Schmid, and H. Jégou, “Event retrieval in large video collections with circulant temporal encoding,” in *CVPR*, 2013, pp. 2459–2466.
30. M. Mua and D. G. Lowe, “Fast approximate nearest neighbors with automatic algorithm configuration,” in *VISAPP* (1), 2009, pp. 331–340.
31. P. Over, G. Awad, M. Michel, J. Fiscus, G. Sanders, W. Kraaij, A. F. Simpson, and G. Quenot, “Trecvid 2013 – an overview of the goals, tasks, data, evaluation mechanisms and metrics,” in *Proceedings of TRECVID 2013. NIST*, USA, 2013.
[32] S. Oh, A. Hoogs, A. G. A. Perera, N. P. Cuntoor, C.-C. Chen, J. T. Lee, S. Mukherjee, J. K. Aggarwal, H. Lee, L. S. Davis, E. Swears, X. Wang, Q. Ji, K. K. Reddy, M. Shah, C. Vondrcek, H. Pirsiavash, D. Ramanan, J. Yuen, A. Torralba, B. Song, A. Fong, A. K. R. Chowdhury, and M. Desai, “A large-scale benchmark dataset for event recognition in surveillance video,” in CVPR, 2011, pp. 3153–3160.

[33] J. Lin and H. Zha, “A symbolic representation of time series, with implications for streaming algorithms,” in DMKD, 2003, pp. 2–11.

[34] A. Mueen, E. J. Keogh, Q. Zhu, S. Cash, and M. B. Westover, “Exact discovery of time series motifs,” in SDM, 2009, pp. 473–484.

[35] B. Y. chi Chiu, E. J. Keogh, and S. Lonardi, “Probabilistic discovery of time series motifs,” in KDD, 2003, pp. 493–498.

[36] E. Berlin and K. Van Laerhoven, “Detecting leisure activities with dense motif discovery,” in Proceedings of the 2012 ACM Conference on Ubiquitous Computing. ACM, 2012, pp. 250–259.

[37] T. Kohonen, “Self-Organizing Maps,” Berlin: Springer - Verlag, 1995.

[38] D. Berndt and J. Clifford, “Using dynamic time warping to find patterns in time series,” in KDD workshop, vol. 10, no. 16. Seattle, WA, 1994, pp. 359–370.

[39] S. Soatto, G. Doretto, and Y. N. Wu, “Dynamic textures,” ICCV, vol. 2, pp. 439–446, 2001.

[40] K. Chakrabarti, E. J. Keogh, S. Mehrotra, and M. J. Pazzani, “Locally adaptive dimensionality reduction for indexing large time series databases,” ACM Trans. Database Syst., vol. 27, no. 2, pp. 188–228, 2002.

[41] H. Karcher, “Riemannian center of mass and mollifier smoothing,” in Data Mining, 2002. ICDM 2002. Proceedings. 2002 IEEE International Conference on, 2002, pp. 370–377.

[42] S. T. Roweis and L. K. Saul, “Nonlinear dimensionality reduction by locally linear embedding,” SCIENCE, vol. 290, no. 5500, pp. 2323–2326, 2000.

[43] T. Lin and H. Zha, “Riemannian manifold learning,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 30, pp. 796–809, 2008.