# A Framework for Hypothesis Learning Over Sets of Vectors

Karim Abou–Moustafa
Dept. of Electrical and Computer Engineering
Centre of Intelligent Machines, McGill University
3480 University st., Montréal, QC, H3A 2A7, Canadas
karimt@cim.mcgill.ca

Frank Ferrie
Dept. of Electrical and Computer Engineering
Centre of Intelligent Machines, McGill University
3480 University st., Montréal, QC, H3A 2A7, Canada
ferrie@cim.mcgill.ca

## ABSTRACT

Sets of vectors, or bags of features, are a common data representation in domains such as computer vision and speech recognition. However, learning a hypothesis (classification, clustering, etc.) over sets of vectors is usually hindered by their particular structure, in which each object in a data set is represented by a different number of vectors of fixed dimensionality. This nonuniform format of the input data requires the learning algorithm to implicitly handle this non-regular type of input, either by unifying the format of the input, or by extracting the necessary information out of it.

In this paper we propose an unsupervised learning framework for unifying the representation of sets of vectors. The framework defines a metric space over probability distributions representing the sets of vectors, followed by a spectral embedding step for these distributions. The spectral embedding step offers an implicit clustering for the data, combined with a reduction – by orders of magnitude – in the data's space complexity, resulting in significantly faster hypothesis learning over the sets of vectors. Moreover, it allows the framework to easily generalize to out-of-sample examples using the Nyström formula. Although the framework is application independent, we test its validity in the context of human action recognition from video sequences. Besides the previously mentioned properties, the framework does indeed show better performance than other approaches in the literature.

## 1. INTRODUCTION

Sets of vectors are a common data representation in various domains such as computer vision in which an image is represented as a bag of features [31], motion analysis in video in which a short video segment is represented as set of spatio–temporal gradient vectors [29], and in speech recognition in which an utterance is represented as a set of MFCC vectors [19, 16], to mention a few. Despite their flexibility and richness as a representation, a major obstacle for directly learning a hypothesis (classification, clustering, etc.) over sets of vectors is their special structure, in which each object $D_i$ in a data set of objects $\mathscr{D}$ is represented by a different number of vectors of fixed dimensionality, forming that one *set of vectors* (SOV). This nonuniform format of the input data requires the learning algorithm, and consequently the algorithm designer, to implicitly handle this non-regular type of input, either by unifying the format of the input, or by extracting the necessary information out of it, such as the (dis)similarity between two SOVs.

In this paper we propose a principled, application independent framework that unifies the representation of SOVs in order to ease hypothesis learning over this type of data. In particular, as depicted in Figure (1), we propose an unsupervised learning approach that maps each SOV, or bag of features, to a single vector in a low dimensional Euclidean space. The advantages of our proposed framework are as follows. (i) The framework allows any learning algorithm to be transparently applied on SOVs through their images residing in the low dimensional subspace, and hence it frees the learning algorithm from the overhead of accommodating their special structure. (ii) The framework offers a reduction, by orders of magnitude, in the data's space complexity, which correlates directly with the computational complexity of the learning algorithm, resulting in significantly faster hypothesis learning. (iii) The framework is unsupervised, and hence it does not require labels nor side-information. However, if labels or side-information are available, they can be naturally integrated into the framework. (iv) The spectral embedding algorithm in the framework, together with the Bhattacharyya-Riemann metric [1] used to measure the similarity between two SOVs, reveal the natural clusters in $\mathscr{D}$; i.e. as a by-product, the framework performs implicit clustering for SOVs which is reflected on the images in the low dimensional subspace. (v) The framework has a well defined generalization to out-of-sample examples using the Nyström formula [5], and hence it does not require retraining the system whenever new data is available.

**Framework overview** We begin our discussion by formally defining SOVs. Let $\mathscr{D} = \{D_i\}_{i=1}^n$ be a set of $n$ objects $D_i$, where $D_i$ can be a speech utterance or a short video segment for instance[1]. Using a feature extraction function $\phi$, the data set $\mathscr{D} = \{D_i\}_{i=1}^n$ is transformed to a set $\mathscr{X} = \{\mathcal{X}_i\}_{i=1}^n$ where $\phi : D_i \longmapsto \mathcal{X}_i = \{\mathbf{x}_1^i, \ldots, \mathbf{x}_{t_i}^i\}$, $\mathbf{x}_j^i \in \mathbb{R}^p$, and $\mathcal{X}_i$ is one set of vectors. Note that $\mathscr{X}$ is now a

[1]**Notations:** Bold small letters $\mathbf{x}, \mathbf{y}$ are vectors. Bold capital letters $\mathbf{A}, \mathbf{B}$ are matrices. Calligraphic and double bold capital letters $\mathcal{X}$, $\mathcal{Y}$, $\mathbb{X}$, $\mathbb{Y}$ denote sets and/or spaces. tr is the matrix trace.


*MLG* '11 San Diego, CA, USA

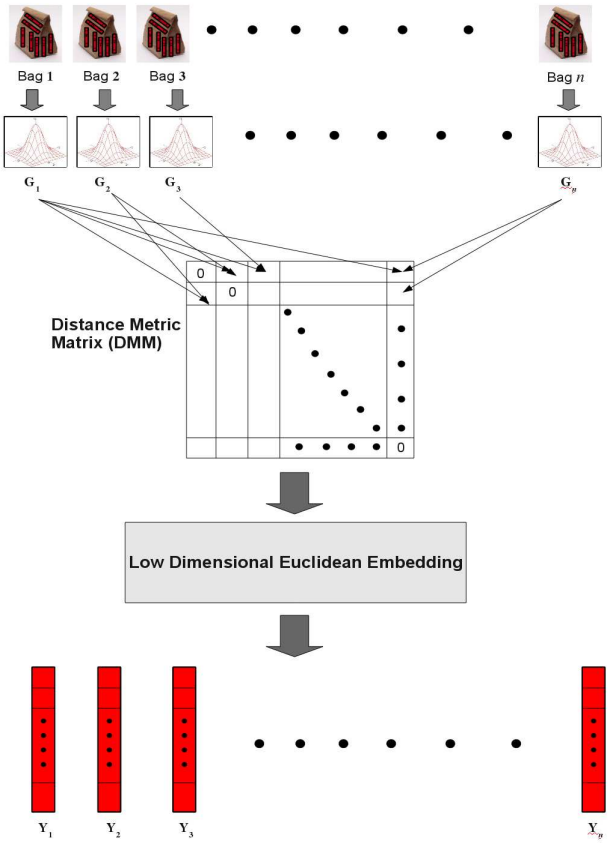


**Figure 1: Outline of the proposed framework for unifying the representation of sets of vectors. In the first step, each *bag of features*, or *set of vectors* (SOV) is modelled as a Gaussian distribution. In the second step, a dissimilarity measure for the difference between Gaussian densities is selected and used to fill a distance metric matrix (DMM) with the dissimilarity, or the distance between every pair of Gaussian distributions. Note that this matrix is symmetric with zero diagonal elements (or self distances). In the third step, Euclidean embedding, or classical MDS is used to collectively embed all SOVs in a low dimensional Euclidean space. The final result is that each bag $i$ is represented by a single vector $\mathbf{y}_i$.**

*set of sets* ; a.k.a. a family or a collection of sets. Note also that it is expected that each SOV $\mathcal{X}_i$ has a different number of vectors in it.

Our framework has slight overlap with some ideas proposed in [10, 12, 16, 31, 29], and before proceeding to our approach, we briefly review these ideas. Also, it is worth noting that the speech recognition community [19] has pioneered learning over time-series or sequential data, which are special cases of SOVs. In this work, we are concerned with SOVs in general including sequential and time-series data. Earlier approaches for hypothesis learning over SOVs focused on directly measuring the (dis)similarity between two SOVs using, for instance, dynamic time warping (DTW) [22], and the earth mover's distance [21]. Instead of measuring the similarity directly on the SOVs, a more popular approach in the computer vision community, is to construct a codebook of words (or visual words) from all the vectors of all SOVs, represent each SOV as a histogram of visual words, and then define kernels over the histograms [31] to be used for classification using support vector machines (SVMs).

A slightly different approach, which is adopted here, is to model each SOV $\mathcal{X}_i$ as a multivariate Gaussian distribution $\mathcal{G}_i$, where the mean vector $\boldsymbol{\mu}_i$ and the covariance matrix $\boldsymbol{\Sigma}_i$ are estimated using the sample mean and the sample covariance matrix for $\mathcal{X}_i$ respectively. Now that the set $\mathscr{X} = \{\mathcal{X}_1, \ldots, \mathcal{X}_n\}$ is replaced by the set $\mathscr{G} = \{\mathcal{G}_1, \ldots, \mathcal{G}_n\}$, a natural measure of (dis)similarity between two densities are divergence measures such as, the Bhattacharyya divergence $d_B$, the symmetric Kullback & Leibler (KL) divergence, a.k.a Jeffreys divergence $d_J$, and the Hellinger distance $d_H$ [13]. For instance, [12] uses SVMs with kernels based on $d_B$ to classify images represented as bags of pixels, while [16] uses SVMs with kernels based on $d_J$ to classify multimedia objects (video,audio) represented as bags of features.

In the context of supervised learning over time-series data, [10] models each class/category of SOVs using an HMM, followed by extracting the Fisher score for each SOV $\mathcal{X}_i$. The Fisher score is a fixed size high dimensional vector that is extracted from the HMMs' parameters and represents the time-series pattern $\mathcal{X}_i$, which in turn, unifies the representation of variable length time-series patterns. Following this representation, the authors define the Fisher kernel, and use SVMs to classify the Fisher scores. Note that this framework is completely different from the standard HMM based approach used in speech recognition [19]. The advantage of [10] is that it allows discriminative models such as SVMs, which can not handle variable length input, to be indirectly used for classifying such data.

In the same spirit of [10], but unlike the other approaches geared towards classification using SVMs and hence focused on the similarity between SOVs, we propose a principled, application independent framework that focuses on unifying the representation for SOVs, while discovering their latent natural clusters. That is, instead of relying on kernels to measure the similarity between two probability distributions, the Gaussian distributions $\{\mathcal{G}_1, \ldots, \mathcal{G}_n\}$ are collectively embedded in a low dimensional subspace $\mathbb{R}^{p_0}$, where in general $p_0 \ll p$, and $p_0 \ll n$. As shown in Figure (1), the first step in the framework is to model each bag of features, or SOV $\mathcal{X}_i$ as a Gaussian distribution $\mathcal{G}_i$. In the second step, a metric is used to fill a distance metric matrix (DMM) with the distance between every pair of Gaussian distributions. In the last step, using Euclidean embedding, each Gaussian $\mathcal{G}_i$ is finally embedded as a *single vector* $\mathbf{y}_i \in \mathbb{R}^{p_0}$.

The proposed framework is based on the theory of Euclidean embedding [30, 9] which is the core of the classical multidimensional scaling (MDS) algorithm [6]. The key ingredient for this embedding theorem is the distance measure that quantifies the dissimilarity between two Gaussian distributions. Since this dissimilarity measure has to satisfy all metric Axioms, we rely on a corrected divergence measure derived from the Bhattacharyya divergence $d_B$ [1] to define the distance between two Gaussian distributions. Although we show the validity of this framework in the context of human action recognition in video, our proposed framework is general enough to be applied to any type of data repre-

sented as SOVs, and it is not restricted to a particular class of learning algorithms.

## 1.1 Preliminaries

A metric space is an ordered pair $(\mathcal{M}, d)$ where $\mathcal{M}$ is a non-empty set, and $d$ is a distance function, or a metric defined as $d : \mathcal{M} \times \mathcal{M} \mapsto \mathbb{R}$, and $\forall\ a, b, c \in \mathcal{M}$, the following Axioms hold: (1) $d(a, b) \geq 0$, (2) $d(a, a) = 0$, (3) $d(a, b) = 0$ iff $a = b$, (4) symmetry $d(a, b) = d(b, a)$, and (5) the triangle inequality $d(a, c) \leq d(a, b) + d(b, c)$. Semi-metrics are relaxed versions of metrics, in which Axiom (3), and the triangle inequality are not required to hold. A result of this relaxation is that semi-metrics do not respect the geometry of metric spaces, and as a consequence, semi-metrics can mislead an algorithm that relies on distance metrics since $d(a, b)$ can be zero for any pair $a, b$ and $a \neq b$. Moreover, violating the triangle inequality results in violating the relative distances between the points. Note that the Euclidean distance $\|\mathbf{x} - \mathbf{y}\|_2$ is a metric, but $\|\mathbf{x} - \mathbf{y}\|_2^2$ is a semi-metric. Similarly, the generalized quadratic distance (GQD) $d(\mathbf{x}, \mathbf{y}; \mathbf{A}) = \sqrt{(\mathbf{x} - \mathbf{y})^\top \mathbf{A} (\mathbf{x} - \mathbf{y})}$, is a metric, but $d^2(\mathbf{x}, \mathbf{y}; \mathbf{A})$ is a semi-metric, where $\mathbf{A}$ is a symmetric positive definite (PD) matrix. If $\mathbf{A}$ is not strictly PD, then $d(\mathbf{x}, \mathbf{y}; \mathbf{A})$ is also a semi-metric.

The family of $p$–dimensional Gaussian distributions is denoted by $\mathbb{G}_p$, and for $\mathcal{G} \in \mathbb{G}_p$, it is defined as:

$$\mathcal{G}(\mathbf{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma}) = \frac{1}{(2\pi)^{\frac{p}{2}} |\boldsymbol{\Sigma}|^{\frac{1}{2}}} \exp\{-\tfrac{1}{2}(\mathbf{x} - \boldsymbol{\mu})^\top \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})\},$$

where $|\cdot|$ is the determinant, $\mathbf{x}, \boldsymbol{\mu} \in \mathbb{R}^p$, $\boldsymbol{\Sigma} \in \mathbb{S}_{++}^{p \times p}$, and $\mathbb{S}_{++}^{p \times p}$ is the manifold of symmetric PD matrices.

# 2. EUCLIDEAN EMBEDDING

Before proceeding to embedding SOVs, we begin with defining metric matrices for a set of points, their PD properties, and their low dimensional Euclidean embedding. For a set of $n$ unknown points, assume the matrix $[d_{ij}] = \mathbf{D} \in \mathbb{R}^{n \times n}$ is given with all the mutual distances (or dissimilarities) between the $n$ points, such that $d_{ij} = d_{ji}$, $d_{ii} = 0$, and $d_{ij} \geq 0$, $\forall\ i, j$. Note here that the points and the distance function are not specified. Gower and Legendre [9] define a metric matrix as follows:

**Definition D** is said to be a **distance metric matrix** (DMM) if the *metric (triangle) inequality* $d_{ij} + d_{ik} \geq d_{jk}$ holds for all triples $(i, j, k)$.

Note that the metric $d$ of any metric space $(\mathcal{M}, d)$ can define a DMM, while semi-metrics can not define DMMs since Axiom (3) and the triangle inequality of metrics are not required to hold. Euclidean distance matrices (EDMs), for example, share the same definition above since the Euclidean distance is a metric. However, an EDM has a more specific definition, which is Definition (2) in [9]:

**Definition D** is said to be an **Euclidean distance matrix** (EDM) if the $n$ points can be embedded in an Euclidean space as $\{\mathbf{p}_i\}_{i=1}^n$, such that the Euclidean distance between $\mathbf{p}_i$ and $\mathbf{p}_j$ is $d_{ij}$, $\forall\ i, j$.

The definition, alone, does not state how to formally validate whether $\mathbf{D}$ is an EDM or not. The necessary and sufficient condition for $\mathbf{D}$ to be an EDM is in Theorem (III) in [30], and Theorem (4) in [9] which is stated after the following definitions. Let $\mathbf{D}$ be defined as above, and let $[-\frac{1}{2} d_{ij}^2] = \mathbf{S} \in \mathbb{R}^{n \times n}$, $\forall\ i, j$. Define the centering matrix $\mathbf{H} \equiv \mathbf{H}_{n \times n} = \mathbf{I}_{n \times n} - \frac{1}{n} \mathbf{1}\mathbf{1}^\top$, where $\mathbf{I}$ is the identity matrix, and $\mathbf{1}$ is a vector of ones.

THEOREM 2.1. *$\mathbf{D}$ is an EDM if and only if the matrix $\mathbf{K} = \mathbf{HSH}$ is positive semi–definite (PSD).*

Young and Householder [30] further discuss the reverse direction of the theorem. That is, if $\mathbf{K}$ is symmetric and PSD, then there exist a set of $n$ real points in an Euclidean space with mutual distance $d_{ij} = d_{ji}$, and these points can be obtained as follows. Since $\mathbf{K}$ is symmetric and PSD, by eigen decomposition of $\mathbf{K}$ to $\mathbf{V}\mathbf{L}\mathbf{V}^\top$, where the columns of $\mathbf{V}$ are the eigenvectors of $\mathbf{K}$, $\mathbf{L} = \text{diag}\{\ell_1, \ldots, \ell_{p_0}, 0, \ldots, 0\}$ is its eigenvalue matrix, and $\ell_1 > \ell_2 > \cdots > \ell_{p_0}$, then the coordinates of these $n$ points are the rows of the matrix $\mathbf{Y} = \mathbf{V}\mathbf{L}^{\frac{1}{2}}$, where $\mathbf{Y} \in \mathbb{R}^{n \times p_0}$.

The key observation here is that from Theorem (2.1) and the previous definitions, it follows directly that if $\mathbf{K}$ is symmetric and PSD, then $\mathbf{D}$ is also a DMM. Hence, given only a DMM, and not necessarily an EDM, one can easily obtain its representing set of $n$ real points in an Euclidean space $\mathbb{R}^{p_0}$, with $p_0 \ll n$. Recalling the definition of a metric space $(\mathcal{M}, d)$ (see Preliminaries), a DMM can represent the mutual distances between all the elements of the non-empty set $\mathcal{M}$ since $d$ is a metric by definition. Therefore, for any metric space $(\mathcal{M}, d)$, where $\mathcal{M}$ can be any nonempty set of objects (images, video clips, speech utterances, etc.), it is possible to obtain an Euclidean embedding for this set as long as $d$ is a metric. Note that matrix $\mathbf{K}$ is in fact a centralized dot product matrix, or a centralized gram matrix, which describes the similarity between the original input points. If $d$ is a semi-metric, the similarity matrix $\mathbf{K}$ is not guaranteed to be PSD, and hence the resulting low dimensional subspace will be a semi-metric space where metric properties and relatives distances between points can be violated.

This is the rational for our proposed framework for SOVs. After modelling each SOV $\mathcal{X}_i$ as Gaussian distribution $\mathcal{G}_i$, we obtain the non-empty set $\mathscr{G} = \{\mathcal{G}_i\}_{i=1}^n$. If $d_{B\mathcal{R}}$ is the metric for the set $\mathscr{G} = \{\mathcal{G}_i\}_{i=1}^n$, where $d_{B\mathcal{R}}$ will be introduced shortly, then the ordered pair $(\mathscr{G}, d_{B\mathcal{R}})$ define a metric space, and a DMM can be defined with the mutual dissimilarities between all the elements of $\mathscr{G}$ using the metric $d_{B\mathcal{R}}$. It follows directly from Theorem (2.1) that the set $\mathscr{G} = \{\mathcal{G}_i\}_{i=1}^n$ can be embedded in a low dimensional Euclidean space $\mathbb{R}^{p_0}$, where in general $p_0 \ll p$, and $p_0 \ll n$.

## 2.1 Classical MDS and Graph Embedding

Before characterizing the Bhattacharyya–Riemann metric $d_{B\mathcal{R}}$, it is worth noting that Theorem (2.1) is the core of the classical MDS algorithm [6]. If $\mathbf{D}$ is obtained from the Euclidean distance between points, and the objective is to find a low dimensional embedding for the data such that it preserves all the distances between the points, then classical MDS minimizes the following objective function:

$$\mathcal{J}_1(\mathbf{Y}) = \|\mathbf{Y}\mathbf{Y}^\top - \mathbf{K}\|_F^2 \ , \tag{1}$$

for which it can be shown that $\mathbf{Y} = \mathbf{V}\mathbf{L}^{\frac{1}{2}}$ is its optimal solution [6]. Indeed, the matrix $\mathbf{D}$ can be replaced with any DMM without changing the final result. Further, using the property that for a matrix $\mathbf{A}$, $\|\mathbf{A}\|_F^2 = \text{tr}\{\mathbf{A}\mathbf{A}^\top\}$, then the

objective function in (1) turns to the following:

$$\mathcal{J}_2(\mathbf{Y}) = \mathrm{tr}\{\mathbf{Q}(\mathbf{Q}-2\mathbf{K})\} = \mathrm{tr}\{\mathbf{QB}\}\ , \qquad (2)$$

where the constant term $\mathbf{K}\mathbf{K}^\top$ is dropped, $\mathbf{Q} = \mathbf{Y}\mathbf{Y}^\top$, $\mathbf{B} = (\mathbf{Q}-2\mathbf{K})$ is symmetric and PSD, and tr is the matrix trace. If $\mathbf{B}$ in Equation (2) is replaced by the Laplacian function obtained from the neighbourhood graph of the original data points, then minimizing $\mathcal{J}_2$ yields the solution of Laplacian Eigenmaps [4]. Also, if $\mathbf{B} = (\mathbf{I}-\mathbf{W})^\top(\mathbf{I}-\mathbf{W})$, where $\mathbf{W}$ is the matrix of reconstruction weights obtained from the first step of LLE, or local linear embedding [23], then minimizing $\mathcal{J}_2$ yields the solution of LLE. Note that these reconstruction weights are also based on the neighbourhood graph of the data points. In a similar way, and despite the information they preserve through the embedding, minimizing $\mathcal{J}_2$ can be linked to isomap [25], spectral clustering algorithms [27, 17], tangent–corrected embedding [8], and other spectral embedding methods that are based on the neighbourhood graph of the data points.

Using this graph perspective, classical MDS is also a graph embedding algorithm, however instead of constructing a neighbourhood graph, it considers a fully connected undirected graph for all the $n$ points. In turn, the original data points are the vertices of this graph, and the weight on the edge between vertices $i$ and $j$, $i \neq j$, is the Euclidean distance $d_{ij}$, which induces the symmetric PSD similarity matrix $\mathbf{K}$. Using Theorem (2.1) and its consequences, each vertex $i$ will be finally embedded as a single vector $\mathbf{y}_i \in \mathbb{R}^{p_0}$.

This graph perspective for classical MDS gives another view for our proposed framework for SOVs. Since the set $\mathscr{X} = \{\mathcal{X}_1, \ldots, \mathcal{X}_n\}$ is replaced by the set $\mathscr{G} = \{\mathcal{G}_1, \ldots, \mathcal{G}_n\}$, we construct a fully connected undirected graph $G(\mathscr{G}, \mathscr{E})$, where $\mathscr{G}$ is the set of vertices, $\mathscr{E}$ is the set of edges, and the weight on each edge $e_{ij} \in \mathscr{E}$, $i \neq j$, is the dissimilarity between $\mathcal{G}_i$ and $\mathcal{G}_j$. However, in order to make use of Theorem (2.1), a distance metric is needed to describe the dissimilarity between two Gaussians. This metric is introduced in the next section.

## 3. CORRECTED DIVERGENCE MEASURES

The Bhattacharyya-Riemann metric used here for the dissimilarity between two Gaussian densities is based on the analysis of three closed form expressions for the divergence between two Gaussian densities, $\mathcal{G}_1$ and $\mathcal{G}_2$, with $\boldsymbol{\mu}_1 \neq \boldsymbol{\mu}_2$ and $\boldsymbol{\Sigma}_1 \neq \boldsymbol{\Sigma}_2$ [1]. The first expression is the symmetric Kullback–Leibler (KL) divergence, or Jeffreys divergence between $\mathcal{G}_1$ and $\mathcal{G}_2$ :

$$d_J(\mathcal{G}_1, \mathcal{G}_2) = \tfrac{1}{2}\mathbf{u}^\top\boldsymbol{\Psi}\mathbf{u} + \tfrac{1}{2}\mathrm{tr}\{\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_2 + \boldsymbol{\Sigma}_2^{-1}\boldsymbol{\Sigma}_1\} - p, \quad (3)$$

where $\boldsymbol{\Psi} = (\boldsymbol{\Sigma}_1^{-1} + \boldsymbol{\Sigma}_2^{-1})$, and $\mathbf{u} = (\boldsymbol{\mu}_1 - \boldsymbol{\mu}_2)$. The two other expressions are the Bhattacharyya divergence $d_B$ and the Hellinger distance $d_H$, which are based on the Bhattacharyya coefficient $\rho$ that measures the similarity between two probability distributions :

$$\rho(\mathcal{G}_1, \mathcal{G}_2) = |\boldsymbol{\Gamma}|^{-\frac{1}{2}}|\boldsymbol{\Sigma}_1|^{\frac{1}{4}}|\boldsymbol{\Sigma}_2|^{\frac{1}{4}}\exp\{-\tfrac{1}{8}\mathbf{u}^\top\boldsymbol{\Gamma}^{-1}\mathbf{u}\},$$

where $\boldsymbol{\Gamma} = (\frac{1}{2}\boldsymbol{\Sigma}_1 + \frac{1}{2}\boldsymbol{\Sigma}_2)$. The Bhattacharyya divergence is simply $-\log[\rho(\mathcal{G}_1, \mathcal{G}_2)]$ :

$$d_B(\mathcal{G}_1, \mathcal{G}_2) = \tfrac{1}{8}\mathbf{u}^\top\boldsymbol{\Gamma}^{-1}\mathbf{u} + \tfrac{1}{2}\ln\{\tfrac{1}{2}|\boldsymbol{\Sigma}_1|^{-\frac{1}{2}}|\boldsymbol{\Sigma}_2|^{-\frac{1}{2}}|\boldsymbol{\Gamma}|\}, \quad (4)$$

while the Hellinger distance is :

$$d_H(\mathcal{G}_1, \mathcal{G}_2) = \sqrt{2[1-\rho(\mathcal{G}_1, \mathcal{G}_2)]}. \qquad (5)$$

The measures $d_J$, $d_B$ and $d_H$ are symmetric, and by definition of a divergence [2], $d_J(\mathcal{G}_1, \mathcal{G}_2) \geq 0$, $d_B(\mathcal{G}_1, \mathcal{G}_2) \geq 0$, and $d_H(\mathcal{G}_1, \mathcal{G}_2) \geq 0$, where equality only holds when $\mathcal{G}_1 = \mathcal{G}_2$. This is equivalent to Axioms (1), (2), (3) & 4 of a metric. However, the triangle inequality Axiom is not satisfied for $d_B$ and $d_J$, while it is for $d_H$ [13, 11]. Therefore, $d_J$ and $d_B$ are semi-metrics, while $d_H$ is a metric.

The reason that $d_J$ and $d_B$ do not satisfy the triangle inequality for the case of Gaussian densities can be analyzed as follows. It is known from [13, pp. 6,7] that $d_J(\mathcal{G}_1, \mathcal{G}_2)$ is a sum of two components; one for the difference in means weighted by the covariance matrices (the first term), and the other for the difference in variances and covariances (the second term). Note that this explanation is also valid for $d_B(\mathcal{G}_1, \mathcal{G}_2)$. The first term in Equations (3) and (4) is equivalent to the GQD, up to a constant and a square root – i.e. semi-metrics. The second term in Equations (3) and (4) is a discrepancy measure between two covariance matrices that is independent from $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_2$. If $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 = \boldsymbol{\mu}$, the first term in Equations (3) and (4) disappears, and $d_J(\mathcal{G}_1, \mathcal{G}_2)$ and $d_B(\mathcal{G}_1, \mathcal{G}_2)$ yield the following:

$$d_J(\mathcal{G}_1, \mathcal{G}_2) = \mathrm{tr}\{\boldsymbol{\Sigma}_1^{-1}\boldsymbol{\Sigma}_2 + \boldsymbol{\Sigma}_2^{-1}\boldsymbol{\Sigma}_1\} - p, \text{ and} \qquad (6)$$

$$d_B(\mathcal{G}_1, \mathcal{G}_2) = \ln\left\{|\boldsymbol{\Gamma}||\boldsymbol{\Sigma}_1|^{-\frac{1}{2}}|\boldsymbol{\Sigma}_2|^{-\frac{1}{2}}\right\}, \qquad (7)$$

which are two semi-metrics for covariance matrices. Therefore, $d_J(\mathcal{G}_1, \mathcal{G}_2)$ and $d_B(\mathcal{G}_1, \mathcal{G}_2)$ are summations of two semi-metrics, where the second term in (3) and (4) does not define a proper metric for symmetric PD matrices on the manifold $\mathbb{S}_{++}^{d\times d}$ . Although $d_H$ is a metric by definition, and can be used for embedding, it is a product of two semi-metrics, which are the terms comprising the Bhattacharyya distance. This is unlike the corrected divergence measures introduced shortly, in which each term is a well defined metric.

A symmetric PD matrix is a geometric object, and the manifold $\mathbb{S}_{++}^{p\times p}$ has a specific structure with defined geometric properties, and equipped with a inner product that induces a natural distance metric, or a Riemannian metric, between all its elements. Förstner and Moonen [7], and independently X. Pennec [18], derived this distance metric for $\mathbb{S}_{++}^{d\times d}$. Due to space limitations, we do not derive the metric here, however a concise derivation can be found in [26]. The Riemannian metric, by default, respects the geometry of $\mathbb{S}_{++}^{p\times p}$, which is unlike the second term in Equations (3) and (4) that are derived from $d_J(\mathcal{G}_1, \mathcal{G}_2)$ and $d_B(\mathcal{G}_1, \mathcal{G}_2)$, and unaware of the geometry of $\mathbb{S}_{++}^{p\times p}$. The distance measure satisfies all the metric Axioms introduced earlier, it is invariant to inversion, and invariant to affine transformations of the coordinate system. For two matrices $\{\mathbf{A}, \mathbf{B} \in \mathbb{S}_{++}^{d\times d}\}$ the distance between them is:

$$d_{\mathcal{R}}(\mathbf{A}, \mathbf{B}) = \mathrm{tr}\{\ln^2\boldsymbol{\Lambda}(\mathbf{A}, \mathbf{B})\}^{\frac{1}{2}}, \qquad (8)$$

where $\boldsymbol{\Lambda}(\mathbf{A}, \mathbf{B}) = \mathrm{diag}(\lambda_1, \ldots, \lambda_d)$ is the solution of a generalized eigenvalue problem (GEP): $\mathbf{AV} = \boldsymbol{\Lambda}\mathbf{BV}$.

The metric $d_{\mathcal{R}}$ is the rational underlying the corrected divergence measures in [1], which for $\mathcal{G}_1$ and $\mathcal{G}_2$ are :

$$d_{J\mathcal{R}}(\mathcal{G}_1, \mathcal{G}_2) = (\mathbf{u}^\top\boldsymbol{\Psi}\mathbf{u})^{\frac{1}{2}} + \mathrm{tr}\{\log^2\boldsymbol{\Lambda}(\boldsymbol{\Sigma}_1, \boldsymbol{\Sigma}_2)\}^{\frac{1}{2}}\ , \qquad (9)$$

and

$$d_{B\mathcal{R}}(\mathcal{G}_1,\mathcal{G}_2) = (\mathbf{u}^\top \mathbf{\Gamma}^{-1}\mathbf{u})^{\frac{1}{2}} + \mathrm{tr}\{\log^2 \mathbf{\Lambda}(\mathbf{\Sigma}_1,\mathbf{\Sigma}_2)\}^{\frac{1}{2}} \ , \quad (10)$$

where $\mathbf{\Psi} \succ 0$, and $\mathbf{\Gamma}^{-1} \succ 0$. The first term in $d_{J\mathcal{R}}$ and $d_{B\mathcal{R}}$ is similar to the first term of $d_J(\mathcal{G}_1,\mathcal{G}_2)$ and $d_B(\mathcal{G}_1,\mathcal{G}_2)$ respectively, except for the square root, which together with the condition that $\mathbf{\Psi}$ and $\mathbf{\Gamma}^{-1}$ are strictly PD, ensure that the first terms are GQD (see Preliminaries). The second term in $d_{J\mathcal{R}}$ and $d_{B\mathcal{R}}$ is the Riemannian metric $d_{\mathcal{R}}$ which replaces the second term in Equations (3) and (4). Hence, $d_{J\mathcal{R}}$ and $d_{B\mathcal{R}}$ are both metrics by construction since each term is a properly defined metric. In addition, they are invariant to affine transformations of the coordinate system which is a property of quadratic distances, and a property of $d_{\mathcal{R}}$ as mentioned earlier. Finally, if $\boldsymbol{\mu}_1 = \boldsymbol{\mu}_2 = \boldsymbol{\mu}$, both measures will yield the metric $d_{\mathcal{R}}$, while if $\mathbf{\Sigma}_1 = \mathbf{\Sigma}_2 = \mathbf{\Sigma}$, then $d_{J\mathcal{R}}$ and $d_{B\mathcal{R}}$ will yield the Mahalanobis distance, and if $\mathbf{\Sigma} = \mathbf{I}$, both measures will be reduced to the Euclidean distance between the means.

It is worth noting that during our experiments, $d_{J\mathcal{R}}$ and $d_{B\mathcal{R}}$ yielded very similar results in various classification and clustering tasks. This shows that the main difference between Equations (3) and (4) is the dissimilarity measure of covariance matrices. Therefore, to reduce notations' cumbersomeness, in the following sections we continue our discussion using one corrected measure only, which is $d_{B\mathcal{R}}$.

# 4. EMBEDDING SETS OF VECTORS

It is possible now to introduce our framework for embedding SOVs. The set $\mathscr{G} = \{\mathcal{G}_i\}_{i=1}^n$, and the metric $d_{B\mathcal{R}}$ define a metric space $(\mathscr{G}, d_{B\mathcal{R}})$, and hence an Euclidean embedding for $\mathscr{G}$ can be obtained using the following procedure :

1. Define $\mathbf{D} \in \mathbb{R}^{n\times n}$ such that $[d_{ij}] = d_{B\mathcal{R}}(\mathcal{G}_i,\mathcal{G}_j)$, $\forall\ i,j$.

2. Define $\mathbf{K}_{B\mathcal{R}} = \mathbf{HSH}$, where $\mathbf{S} = [-\frac{1}{2}d_{ij}^2]$, and $\mathbf{H}$ is the centering matrix as defined earlier. Since $d_{B\mathcal{R}}$ is a metric, then according to Theorem (2.1) $\mathbf{K}_{B\mathcal{R}}$ is PSD.

3. Perform an eigen decomposition for $\mathbf{K}_{B\mathcal{R}}$ to $\mathbf{VLV}^\top$, and construct the matrix $\mathbf{Y}_{B\mathcal{R}} = \mathbf{VL}^{\frac{1}{2}}$, where $\mathbf{Y}_{B\mathcal{R}} \in \mathbb{R}^{n\times p_0}$.

Now, each row $i$ of $\mathbf{Y}_{B\mathcal{R}}$, which corresponds to SOV $\mathcal{X}_i$, is a vector $\mathbf{y}_i$ in a low dimensional Euclidean space. Hence any learning algorithm can be transparently applied on the set $\{\mathcal{X}_i\}_{i=1}^n$ through their corresponding images $\{\mathbf{y}_i\}_{i=1}^n$. Note that this procedure is totally unsupervised and does not require any labels nor side-information.

## 4.1 Generalization to New Sets of Vectors

The procedure above describes the training phase for embedding SOVs, where $\mathbf{V}$, $\mathbf{L}$, and $p_0$ are the parameters learned during that phase. Suppose we are given $m$ new SOVs $\{\mathcal{X}_1^*,\ldots,\mathcal{X}_m^*\}$ that were not included during the training phase, and it is desired to compute their low dimensional embeddings. This is the problem of generalizing Euclidean embedding to out-of-sample examples which was thoroughly studied in [5] for algorithms such as classical MDS, LLE, Isomap, Laplacian eigenmaps, and spectral clustering methods. Since all these algorithms share a spectral embedding step, it was shown that all these methods are learning eigenfunctions of similarity between input points, and for which the Nyström formula [3] provides a method for generalizing these algorithms to out-of-samples examples. Since our framework exchanges the Euclidean distance in classical MDS with the metric $d_{B\mathcal{R}}$, then the Nyström formula can be directly used to generalize our framework to the new SOVs $\{\mathcal{X}_1^*,\ldots,\mathcal{X}_m^*\}$ as follows :

1. Model each new $\mathcal{X}_j^*$ as a Gaussian $\mathcal{G}_j^*$, for $1 \le j \le m$.

2. Define $\mathbf{D}^* \in \mathbb{R}^{m\times n}$ such that $[d_{ji}^*] = d_{B\mathcal{R}}(\mathcal{G}_j^*,\mathcal{G}_i)$, for $1 \le j \le m$, and $1 \le i \le n$.

3. Define the similarity matrix $\mathbf{K}_{B\mathcal{R}}^*$:

$$\mathbf{K}_{B\mathcal{R}}^* = -\tfrac{1}{2}[\mathbf{D}^*\mathbf{H}_{n\times n} - \tfrac{1}{n}\mathbf{1}_m\mathbf{1}_n^\top\mathbf{DH}_{n\times n}] \ , \quad (11)$$

where $\mathbf{H}$ is the centering matrix defined earlier, and $\mathbf{D}$ is the DMM matrix for the training set defined in step (1) in the training phase.

4. Apply the Nyström formula on $\mathbf{K}_{B\mathcal{R}}^*$ to obtain the embedding for the out-of-sample examples $\{\mathcal{X}_1^*,\ldots,\mathcal{X}_m^*\}$:

$$\mathbf{Y}_{B\mathcal{R}}^* = \mathbf{K}_{B\mathcal{R}}^*\mathbf{VL}^{-\frac{1}{2}} \ , \quad (12)$$

where $\mathbf{Y}_{B\mathcal{R}}^* \in \mathbb{R}^{m\times p_0}$, $\mathbf{V}$ and $\mathbf{L}$ are the eigenvectors and eigenvalues matrices, respectively, obtained in step (3) in the training phase. Now row $i$ of $\mathbf{Y}_{B\mathcal{R}}^*$ represents the embedding of SOV $\mathcal{X}_i$.

From the generalization via the Nyström formula above, it is possible now to emphasize the advantage of adhering to metric properties via measures such as $d_{B\mathcal{R}}$ and $d_H$. The benefits of $d_{B\mathcal{R}}$ over $d_H$, however, will be demonstrated in the next section. Euclidean embedding via semi-metrics instead of metrics will result in the following consequences: *first*, a DMM can not be defined since Axiom (3) and the triangle inequality of a metric may not hold, and *second*, it follows that the resulting similarity matrix $\mathbf{K}$ will be indefinite.

A first option to overcome this situation is via metric MDS [6], which defines a transformation by minimizing a stress (or error) function. Unfortunately, this transformation does not provide an embedding nor it can be considered a mapping, and hence, generalization for out-of-sample examples can not be obtained [28]. This is unlike our approach that has a direct generalization via the Nyström formula. Another solution is to approximate the matrix $\mathbf{K}$ to a nearby PSD matrix by truncating the negative eigenvalues of $\mathbf{L}$, or using minimum shift embedding [20] which adds the smallest constant to $\mathbf{L}$ such that it transforms $\mathbf{K}$ to a PSD matrix. Although generalization via the Nyström formula can be obtained for the approximated matrix, our approach does not need to rely on such approximations. Therefore, our proposed framework using the metric $d_{B\mathcal{R}}$ provides a direct unsupervised low dimensional embedding for SOVs that does not require approximations, and has a direct generalization via the Nyström formula to out-of-sample examples.

# 5. EXPERIMENTS

We test the validity of our proposed representation in the context of human action recognition from video sequences. For this purpose, we use the KTH video data set for human action recognition shown in Figure (2) [24][2]. The data set consists of video clips for 6 types of human actions (boxing, hand clapping, hand waving, jogging, running, and walking) performed by 25 subjects in 4 different scenarios (outdoors,

[2] http://www.nada.kth.se/cvap/actions/

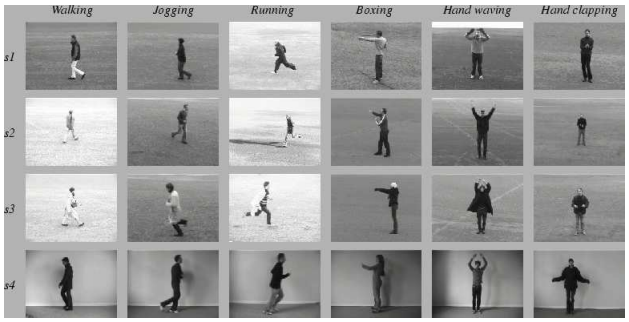


**Figure 2: Sample frames from the KTH video data set for human action recognition.**

outdoors with scale variation, outdoor with different clothes, and indoors), resulting in a total number of video clips $n = 6 \times 25 \times 4 = 600$. All sequences were taken over homogeneous backgrounds with a static camera with a frame rate of 25 fps. The spatial resolution of the videos is $160 \times 120$, and each clip has a length of 20 seconds on average.

## 5.1 Representing Motion as Sets of Vectors

To extract the motion information, a dense optical flow is computed for each video clip using the Lucas-Kanade algorithm [15][3], resulting in a large set of spatio-temporal gradients vectors describing the motion of pixels in each frame. The gradient vector is normal to the local spatio-temporal surface generated by the motion in the space–time volume. The gradient direction captures the local surface orientation which depends on the local behavioural properties of the moving object, while its magnitude depends mainly on the photometric properties of the moving object, and it is affected by its spatial appearance (color, texture, etc.) [14].

To capture the motion information encoded in the gradient direction, first we apply an adaptive threshold based on the norm of the gradient vectors to eliminate all vectors resulting from slight illumination changes and camera jitter. Second, each video frame is divided into $h \times w$ blocks – typically $3 \times 3$ and $4 \times 4$ – and the motion in each block is encoded by an $m$–bins histogram of gradient orientations. In all our experiments, $m$ is set to 4 and 8 bins. The histograms of all blocks for one frame are concatenated to form one vector of dimensionality $p = m \times h \times w$. Therefore, a video clip $D_i$ with $t_i$ frames is finally represented as a set $\mathcal{X}_i = \{\mathbf{x}_1^i, \ldots, \mathbf{x}_{t_i}^i\}$, where $\mathbf{x}_j^i$ is a $p$-dimensional vector of the concatenated histograms of frame $j$. Since histograms of orientations from optical flow vectors can not differentiate between two identical actions performed at different speeds, we excluded the `'walking'` and `'running'` classes from the data set. This resulted in $n = 400$ video clips, for 25 persons performing 4 actions in 4 different scenarios.

## 5.2 Experimental Setting

After extracting the motion information from each video clip $D_i$ and representing it as an SOV $\mathcal{X}_i$ as described above, each $\mathcal{X}_i$ is modelled as a Gaussian distribution $\mathcal{G}_i$ with mean vector $\hat{\boldsymbol{\mu}}_i = \frac{1}{t_i} \sum_{j=1}^{t_i} \mathbf{x}_j^i$ , and a covariance matrix $\hat{\boldsymbol{\Sigma}}_i = \frac{1}{t_i - 1} \sum_{j=1}^{t_i} (\mathbf{x}_j^i - \hat{\boldsymbol{\mu}}_i)(\mathbf{x}_j^i - \hat{\boldsymbol{\mu}}_i)^\top + \gamma \mathbf{I}$ , where $\gamma$ is a regularization parameter, and $\mathbf{I}$ is the identity matrix. The regularization here is necessary to avoid the expected rank deficiencies in $\boldsymbol{\Sigma}_i$'s, which can be due to the small number of samples in $\mathcal{X}_i$ with respect to the high dimensionality of the data, and hence, this helps avoid over-fitting and outlier reliance. In all our experiments $\gamma$ was set to 1.

Using the algorithm described in the previous section, all the Gaussians representing the motion of all video clips were embedded in four low dimensional subspaces $\mathbb{R}^{p_0}$ using four different dissimilarity measures; $d_J(\mathcal{G}_i, \mathcal{G}_j)$ used in [16] which is a semi-metric, $d_B(\mathcal{G}_i, \mathcal{G}_j)$ used in [12] which is also a semi-metric, $d_H(\mathcal{G}_i, \mathcal{G}_j)$ which is a metric, and the metric $d_{B\mathcal{R}}(\mathcal{G}_i, \mathcal{G}_j)$. This resulted in 4 similarity matrices, $\mathbf{K}_J$, $\mathbf{K}_B$, $\mathbf{K}_H$, and $\mathbf{K}_{B\mathcal{R}}$ respectively. Note that $p_0$, the dimensionality of the embedding space, is a free parameter that is either user defined, or selected by cross validation.

To classify the different actions embedded in the different low dimensional subspaces, we use a $k$–nearest neighbours ($k$–NN) classifier, with $k = \{1, 3, 5, 7\}$. The empirical error is measured using a 30 folds double cross validation procedure, in which the data set is randomly split into a training set (80%) and a test set (20%), and then search for $k$ that minimizes the training error of the current split. This optimal $k$ is used to obtain the test error of one trial. This process is repeated 30 times, and the final empirical error (with standard deviation) is the average test error over all the 30 trials. Since $p_0$ is a free parameter, the optimal $p_0$ for each embedding is selected based on the lowest empirical error, where $p_0 \in [2, 50]$.

Before proceeding to the results, it is worth noting that selecting optimal parameter values for $m$, $h$, $w$, and $\gamma$, and computing the optical flow vectors, is a fundamental question of model selection which is not addressed here. Nevertheless, even though we do not optimize all these parameters, the proposed framework using the metric $d_{B\mathcal{R}}$ appears indeed to be a valid framework for unifying the representation of SOVs with various desirable properties as it will be shown below.

## 5.3 Analysis of The Results

Our hypothesis before running the experiments is that the embeddings obtained via $d_J$ and $d_B$ will yield higher classification error than those embeddings obtained via $d_H$ and $d_{B\mathcal{R}}$ since $d_J$ and $d_B$ are semi-metrics. According to Theorem (2.1) and the definition of semi-metrics, the resulting similarity matrix $\mathbf{K}$ is not guaranteed to be PSD for semi-metrics, and hence the resulting embedding space will be a semi-metric space in which metric properties and the relative distances between points are violated. Table (1) shows the classification error (with standard deviation) and the dimensionality of the embedding space for each dissimilarity measure on the 4 feature sets extracted from the KTH data set. It can be clearly seen that despite the dimensionality $p_0$, $d_H$ resulted in lower classification error than $d_J$ and $d_B$ did, while the embedding based on the proposed metric $d_{B\mathcal{R}}$ yielded the lowest error among all other dissimilarity measures. Although $d_H$ is a metric, $d_{B\mathcal{R}}$ performed better since it was able to better capture the natural grouping in the data, and translate this in the low error rates in Table (1).

To see this natural grouping of the data, while being able to compare the difference between the 4 embeddings, we pick the $4 \times 4 \times 4$ feature set from the 4 sets of features shown in Table (1) since it yielded the lowest error rate

[3]Implemented in Piotr's Image and Video Toolbox for Matlab `http://vision.ucsd.edu/ pdollar/toolbox/doc/`

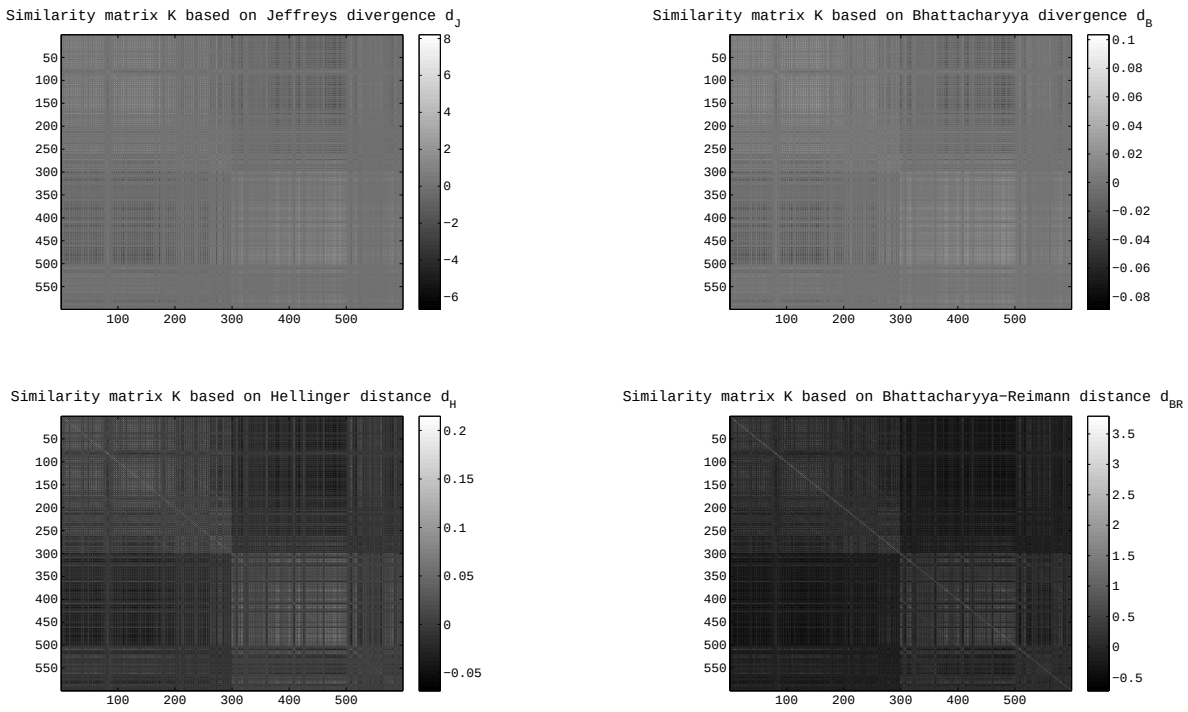


**Figure 3: The four similarity matrices $\mathbf{K}_J$ top left, $\mathbf{K}_B$ top right, $\mathbf{K}_H$ bottom left, and $\mathbf{K}_{B\mathcal{R}}$ bottom right. Note the clear block structure for $\mathbf{K}_{B\mathcal{R}}$ compared to other matrices.**

**Table 1: Empirical error (with standard deviation) and the dimensionality $p_0$ of the embedding space obtained by the four different dissimilarity measures on the four feature settings obtained from the KTH data set.**

| $m \times h \times w$ | $d_J$ | $d_B$ | $d_H$ | $d_{B\mathcal{R}}$ |
|---|---|---|---|---|
| $4 \times 3 \times 3$ | 21.2 (3.8), $p_0 = 11$ | 20.2 (3.4), $p_0 = 30$ | 19.7 (3.7), $p_0 = 45$ | **17.7 (4.7)**, $p_0 = 38$ |
| $4 \times 4 \times 4$ | 16.7 (3.6), $p_0 = 15$ | 17.0 (4.1), $p_0 = 19$ | 16.9 (3.7), $p_0 = 44$ | **15.9 (3.2)**, $p_0 = 47$ |
| $8 \times 3 \times 3$ | 24.3 (2.9), $p_0 = 43$ | 23.3 (4.9), $p_0 = 48$ | 22.1 (3.8), $p_0 = 44$ | **19.9 (3.8)**, $p_0 = 45$ |
| $8 \times 4 \times 4$ | 20.9 (4.6), $p_0 = 20$ | 20.4 (3.8), $p_0 = 22$ | 20.4 (3.7), $p_0 = 22$ | **18.8 (3.5)**, $p_0 = 47$ |

with all dissimilarity measures. Using this feature set, we obtain the 4 similarity matrices $\mathbf{K}_J$, $\mathbf{K}_B$, $\mathbf{K}_H$, and $\mathbf{K}_{B\mathcal{R}}$ and shown in Figure (3) – better seen on a display. It can be clearly seen that $\mathbf{K}_{B\mathcal{R}}$ has 3 clear block structures along the diagonal, indicating three main categories in the data, which has originally 4 classes. Further, the top-left block of $\mathbf{K}_{B\mathcal{R}}$ has further sub-blocks indicating finer categories within the data. This is less clear for $\mathbf{K}_H$, and obscured in the case of $\mathbf{K}_J$ and $\mathbf{K}_B$.

Further analysis can be made by comparing the eigen-spectrum of the four similarity matrices $\mathbf{K}_J$, $\mathbf{K}_B$, $\mathbf{K}_H$, and $\mathbf{K}_{B\mathcal{R}}$, and in particular, the tail of each eigen-spectrum which reflects the adherence of each dissimilarity measure to the metric properties. From Theorem (2.1), we know that only metrics will yield PSD similarity matrices $\mathbf{K}$. This is exactly depicted in Figure (4) where the smallest eigenvalues for $\mathbf{K}_H$ and $\mathbf{K}_{B\mathcal{R}}$, generated by $d_H$ and $d_{B\mathcal{R}}$ respectively, are greater than or equal to zero. This is unlike $d_J$ and $d_B$ which resulted in negative definite matrices $\mathbf{K}_H$ and $\mathbf{K}_B$ respectively, and hence the negative eigenvalues in Figure (4).

Finally, it is important to consider the reduction in space complexity achieved by the proposed framework. If the minimum representation of a single video frame, using the first feature set in Table (1) and a double precision format is $4 \times 3 \times 3$ $(m \times h \times w)$ $\times 4$ (Bytes) $= 144$ Bytes per frame, then for 400 clips, with 25 fps, and an average length for video clips of 20 seconds, the total space required for the data set is $400 \times 20 \times 25 \times 144 \approx 27$ MB. However, after using the proposed framework, the same data set will require 400 (clips) $\times$ $p_0 \times 4 = 73$ KB of memory for $p_0 = 47$ using $d_{B\mathcal{R}}$ (see Table (1)). This is a significant reduction in space complexity, and indeed learning a hypothesis over the embedded data set will be much faster than learning a similar hypothesis over the original representation.

## 6. CONCLUSION

We have proposed an unsupervised learning framework for unifying the representation of sets of vectors. The framework defines a metric space over Gaussian distributions representing the sets of vectors, followed by a spectral embedding step for these Gaussian distributions. The spectral embedding step offers an implicit clustering for the data, combined with a reduction in the data's space complexity, resulting in significantly faster hypothesis learning over the sets of vectors. Moreover, the metric space and the spectral

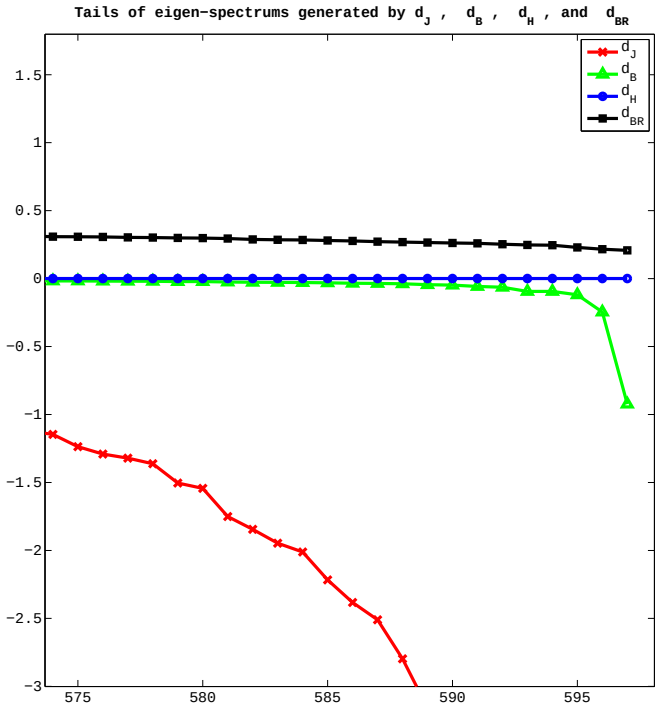


**Figure 4: Tails of eigen-spectrums for the four similarity matrices shown in Figure (3). Note how the semi-metrics $d_J$ and $d_B$ yield negative eigenvalues, indicating that $\mathbf{K}_J$ and $\mathbf{K}_B$ are negative definite matrices and not PSD as required by Theorem 2.1.**

embedding step allow the framework to easily generalize to out-of-sample examples using the Nyström formula.

Our experiments show (i) the validity of the proposed framework in terms of unifying the representation over sets of vectors while reducing their space complexity, (ii) the validity of the metric space defined over the Gaussian distributions, and (iii) the validity of the implicit clustering for the SOVs.

The proposed framework is not restricted to fully connected graphs as presented here, and can be easily extended to neighbourhood graphs defined over the SOVs. Moreover, while Euclidean embedding is a one way to achieve a low dimensional embedding in $\mathbb{R}^{p_0}$, the proposed framework can exchange the Euclidean embedding with embedding algorithms that rely on the graph Laplacian, achieving by that spectral clustering over sets of vectors. These different paths, however, remain to be explored in future venues.

## 7. REFERENCES


[1] K. T. Abou-Moustafa, F. De La Torre, and F. P. Ferrie. Designing a metric for the difference between two Gaussian densities. In *Advances in Intelligent and Soft Computing*, volume 83, pages 57 – 70. Springer, 2010.

[2] S. M. Ali and S. D. Silvey. A general class of coefficients of divergence of one distribution from another. *J. of the Royal Statistical Society.* Series B, 28(1):131–142, 1966.

[3] C. Baker, editor. *The Numerical Treatment of Integral Equations.* Clarendon Press, Oxford, 1977.

[4] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for data representation. *Neural Computation*, 15:1373–1396, 2003.

[5] Y. Bengio, O. Delalleau, N. L. Roux, J.-F. Paiement, P. Vincent, and M. Ouimet. Learning eigenfunctions links spectral embedding and kernel PCA. *Neural Computation*, 16:2197–2219, 2004.

[6] I. Borg and P. Groenen. *Modern Multidimensional Scaling: theory and applications.* Springer–Verlag, New York, 2005.

[7] W. Förstner and B. Moonen. A metric for covariance matrices. Technical report, Dept. of Geodesy and Geo–Informatics, Stuttgart University, 1999.

[8] A. Ghodsi, J. Huang, F. Southey, and D. Schuurmans. Tangent-corrected embedding. In *IEEE Proc. of CVPR*, pages 518–525, 2005.

[9] J. Gower and P. Legendre. Metric and Euclidean properties of dissimilarity coefficients. *Journal of Classification*, 3:5–48, 1986.

[10] T. Jaakkola and D. Haussler. Exploiting generative models in discriminative classifiers. In *NIPS 11*, pages 487–493. MIT Press, 1999.

[11] T. Kailath. The divergence and Bhattacharyya distance measures in signal selection. *IEEE Trans. on Communication Technology*, 15(1):52–60, 1967.

[12] R. Kondor and T. Jebara. A kernel between sets of vectors. In *ACM Proc. of ICML*, 2003.

[13] S. Kullback. *Information Theory and Statistics – Dover Edition.* Dover, New York, 1997.

[14] Z. Li and Y.-P. Tan. Event–based analysis of video. In *IEEE Proc. of CVPR*, pages 1063–6919, 2001.

[15] B. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *Proc. of IJCAI*, pages 674–679, 1981.

[16] P. Moreno, P. Ho, and N. Vasconcelos. A Kullback–Leibler divergence based kernel for svm classification in multimedia applications. In *NIPS 16*, 2003.

[17] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS 14*, pages 849–856. MIT Press, 2002.

[18] X. Pennec, P. Fillard, and N. Ayache. A Riemannian Framework for Tensor Computing. Technical Report RR-5255, INRIA, 7 2004.

[19] L. Rabiner and B. Juang. *Fundamentals of Speech Recognition.* Prentice–Hall, Inc., 1993.

[20] V. Roth, J. Laub, and J. Buhmann. Optimal cluster preserving embedding of nonmetric proximity data. *IEEE Trans. PAMI*, 25(12):1540–1551, 2003.

[21] Y. Rubner, C. Tomasi, and L. Guibas. The earth mover's distance as a metric for image retrieval. *Int. J. of Computer Vision*, 40(2):99–121, 2000.

[22] H. Sakeo and S. Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 26(1):43–49, 1978.

[23] L. Saul and S. Roweis. Think globally, fit locally: Unsupervised learning of low dimensional manifolds. *JMLR*, 4:119–155, 2003.

[24] C. SchâĹŽÂžldt, I. Laptev, and B. Caputo. Recognizing human actions: A local SVM approach.

In *In Proc. of ICPR*, pages 32–36, 2004.

[25] J. Tenenbaum, V. de Silva, and J. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, November 2000.

[26] O. Tuzel, F. Porikli, and P. Meer. Pedestrian detection via classification on Riemannian manifolds. *IEEE Trans. PAMI*, 30(10):1713–1727, 2008.

[27] Y. Weiss. Segmentation using eigenvectors: a unifying view. In *IEEE Proc. of ICCV*, pages 975–982, 1997.

[28] C. K. Williams. On a connection between kernel PCA and metric multidimensional scaling. *Machine Learning*, 46:11–19, 2002.

[29] Y. Yang, J. Liu, and M. Shah. Video scene understanding using multi–scale analysis. In *IEEE Proc. of ICCV*, 2009.

[30] G. Young and A. Householder. Discussion of a set of points in terms of their mutual distances. *Psychometrika*, 3(1):19–22, 1938.

[31] J. Zhang, M. Marszalek, S. Lazebnik, and C. Schmid. Local features and kernels for classification of texture and object categories: A comprehensive study. *Int. J. of Computer Vision*, 73:213–238, June 2007.