ISOMAP-KL: a parametric approach for unsupervised metric learning

Alaor Cervati Neto
Computing Department
Federal University of São Carlos
São Carlos, SP, Brazil
alaor_c_neto@yahoo.com.br

Alexandre L. M. Levada
Computing Department
Federal University of São Carlos
São Carlos, SP, Brazil
alexandre.levada@ufscar.br

***Abstract*—Unsupervised metric learning consists in building data-specific similarity measures without information of the class labels. Dimensionality reduction (DR) methods have shown to be a powerful mathematical tool for uncovering the underlying geometric structure of data. Manifold learning algorithms are capable of finding a more compact representation for data in the presence of non-linearities. However, one limitation is that most of them are pointwise methods, in the sense that they are not robust to the presence of outliers and noise in data. In this paper, we present ISOMAP-KL, a parametric patch-based algorithm that uses the KL-divergence between local Gaussian distributions learned from neighborhood systems along the KNN graph. We use this non-Euclidean measure to compute the weights and define the entropic KNN graph, whose shortest paths approximate the geodesic distances between patches of points in a parametric feature space. Results obtained in several datasets show that the proposed method is capable of improving the classification accuracy in comparison to other DR methods.**

I. Introduction

Dimensionality reduction (DR) methods have been successfully applied for unsupervised metric learning in high-dimensional data analysis. Besides, DR also avoids the curse of the dimensionality, a set of negative side-effects introduced by an arbitrary increase in the number of features in a small sample size problem [1].

The intuition behind these algorithms is that, often, the observed data samples lie along a low-dimensional structure embedded in a high-dimensional input space. The low-dimensional space encodes unknown underlying parameters (i.e., local coordinates) in the original feature space. Trying to recover this hidden structure is the main goal of DR algorithms. These methods are deeply connected to unsupervised metric learning, since besides learning a more compact and meaningful representation for a given dataset, they also learn a distance function that is geometrically better suited to represent a similarity measure between a pair of objects in the collection [2], [3]. In other words, by learning the hidden structure, in general, we earn a more powerful metric for granted.

Recently, deep learning neural networks have been pointed by many machine learning researchers as the state-of-the-art in feature extraction, especially from image data [4]. These models comprise a class of neural networks that uses multiple layers to progressively extract higher level features from the raw input [5]. A requirement for deep learning is to have a large sample size, that is, huge amounts of data are necessary to properly adjust millions of parameters in these mathematical models, which is not always possible in real world problems. DR algorithms on the other hand are able to learn good features from a few samples, producing reliable results even when the number of samples n is less than or equal the number of original features m [6]. Hence, it does not seem reasonable to assume that, eventually, deep learning will replace traditional DR methods and manifold learning. Furthermore, most deep learning models require some degree of supervision, since they are generalizations of multilayer perceptrons, meaning that we have to know the class labels, which is not always possible in pattern recognition tasks.

The study of manifold learning techniques for dimensionality reduction has begun in early 2000's, with the pioneering Isometric Feature Mapping (ISOMAP) algorithm [7]. Hence, this year we celebrate the 20th anniversary of this remarkable research field. A limitation of high-dimensional data analysis concerns the weak discrimination power of the Euclidean metric. It has been shown that, as the number of features increases, the degree of contrast provided by the usual Euclidean distance becomes poor [8]. In this paper, we propose a non-Euclidean parametric patch-based ISOMAP defined in terms of an information-theoretic measure: the relative entropy or KL-divergence [9]. The main goal is to replace the matrix of pairwise geodesic distances D, which is obtained from the KNN graph whose edges are weighted by Euclidean distances, by the entropic distance matrix E, which is obtained from the KNN graph whose edges are weighted by the symmetrized KL-divergence between multivariate Gaussians estimated from local neighborhood patches. Overall, the obtained results show that the proposed method is capable of improving two major aspects of traditional DR methods: 1) in general, the obtained clusters show a lower intra-class scattering, which is interesting for unsupervised classification; 2) ISOMAP-KL is a patch-based method, which makes it less sensitive to the presence of outliers and noise in data, in a way that the learned features show more discriminant power in supervised classification, making it a promising alternative for unsupervised metric learning.

The remaining of the paper is organized as: Section 2

describes the traditional ISOMAP algorithm for manifold learning. Section 3 presents the KL-divergence or relative entropy, and shows its calculation in the multivariate Gaussian case. Section 4 describes the proposed ISOMAP-KL method in details and Section 5 shows the experiments and obtained results. Finally, Section 6 brings the conclusions, final remarks and future directions in unsupervised metric learning.

II. Isometric Feature Mapping (ISOMAP)

ISOMAP was the first manifold learning algorithm for dimensionality reduction. This method combines the main algorithmic features of PCA and Multidimensional Scaling [10], [11] (MDS) - computational efficiency, global optimality, and asymptotic convergence guarantees - with the flexibility to learn a broad class of non-linear manifolds [7]. The basic idea is to build a graph by linking the k-nearest neighbors (KNN) in the input space, then compute the shortest paths between each pair of vertices in the graph and, knowing the approximate geodesic distances between the points, find a mapping to an Euclidean subspace of R^d that preserves those distances. The hypothesis assumed by the ISOMAP algorithm is that the shortest paths in the KNN graph are good approximations for the true geodesic distances in the manifold. In summary, ISOMAP can be divided in three main steps:

1) From the input data $\vec{x}_1, \vec{x}_2, ..., \vec{x}_n \in R^m$ build an undirected proximity graph using the KNN rule [12];
2) Compute the pairwise distance matrix D using n executions of the Dijkstra's algorithm or one execution of the Floyd-Warshall algorithm [13];
3) Estimate the new coordinates of the points in an Euclidean subspace of R^d by preserving the distances through the Multidimensional Scaling (MDS) method.

A. Multidimensional Scaling

Basically, the main goal of MDS is, given an $n \times n$ matrix of pairwise distances, recover the coordinates of the n points $\vec{x}_r \in R^d$ for $r = 1, 2, ..., n$ in an Euclidean subspace, where d, the target dimensionality, is a parameter of the algorithm [10], [11].

We begin by noting that the pairwise distance matrix is given by $D = \{d_{rs}^2\}$, for $r, s = 1, 2, ..., n$ where the distance between two arbitrary points $\vec{x}_r$ and $\vec{x}_s$ is:

$$d_{rs}^2 = \|\vec{x}_r - \vec{x}_s\|^2 = (\vec{x}_r - \vec{x}_s)^T(\vec{x}_r - \vec{x}_s) \tag{1}$$

Let B denote Gram matrix of inner products, that is $B = \{b_{rs}\}$, where $b_{rs} = \vec{x}_r^T \vec{x}_s$. To find the embedding, MDS needs the matrix B, not D. First, we need to assume a hypothesis that the data has zero mean, otherwise there would be infinitely many different solutions, since the application of any arbitrary translation in the set of points, would preserve the pairwise distances. From equation (1), applying the distributive law we have:

$$d_{rs}^2 = \vec{x}_r^T \vec{x}_r + \vec{x}_s^T \vec{x}_s - 2\vec{x}_r^T \vec{x}_s \tag{2}$$

From the matrix D, we can calculate the mean of an arbitrary column s by:

$$\frac{1}{n}\sum_{r=1}^{n} d_{rs}^2 = \frac{1}{n}\sum_{r=1}^{n} \vec{x}_r^T \vec{x}_r + \vec{x}_s^T \vec{x}_s \tag{3}$$

Similarly, we compute the mean of an arbitrary row r as:

$$\frac{1}{n}\sum_{s=1}^{n} d_{rs}^2 = \vec{x}_r^T \vec{x}_r + \frac{1}{n}\sum_{s=1}^{n} \vec{x}_s^T \vec{x}_s \tag{4}$$

Finally, we can compute the mean of all elements of D as:

$$\frac{1}{n^2}\sum_{r=1}^{n}\sum_{s=1}^{n} d_{rs}^2 = \frac{2}{n}\sum_{r=1}^{n} \vec{x}_r^T \vec{x}_r \tag{5}$$

Note that from equation (2), it is possible to define b_{rs} as:

$$b_{rs} = \vec{x}_r^T \vec{x}_s = -\frac{1}{2}(d_{rs}^2 - \vec{x}_r^T \vec{x}_r - \vec{x}_s^T \vec{x}_s) \tag{6}$$

Combining equations (3), (4) and (5) we have:

$$b_{rs} = -\frac{1}{2}\left(d_{rs}^2 - \frac{1}{n}\sum_{r=1}^{n} d_{rs}^2 - \frac{1}{n}\sum_{s=1}^{n} d_{rs}^2 + \frac{1}{n^2}\sum_{r=1}^{n}\sum_{s=1}^{n} d_{rs}^2\right) \tag{7}$$

Making $a_{rs} = -\frac{1}{2}d_{rs}$ we can write:

$$a_{r.} = \frac{1}{n}\sum_{s=1}^{n} a_{rs} \quad a_{.s} = \frac{1}{n}\sum_{r=1}^{n} a_{rs} \quad a_{..} = \frac{1}{n}\sum_{r=1}^{n}\sum_{s=1}^{n} a_{rs} \tag{8}$$

leading to:

$$b_{rs} = a_{rs} - a_{r.} - a_{.s} + a_{..} \tag{9}$$

which in matrix notation becomes $B = HAH$, where:

$$H = I - \frac{1}{n}\vec{1}\vec{1}^T \tag{10}$$

is the centring matrix. To find the embedding, that is, the coordinates of the points in R^d, we have to perform an eigendecomposition of the matrix B, that is:

$$B = V\Lambda V^T \tag{11}$$

where $\Lambda = diag(\lambda_1, \lambda_2, ..., \lambda_n)$ is the diagonal matrix with the eigenvalues of B and V is the matrix whose columns are the eigenvectors of B. Algorithm 1 summarizes the whole process in a sequence of logical and objective steps.

Algorithm 1 Isometric Feature Mapping

function ISOMAP(X)
  From the input data $X_{m\times n}$ build a KNN graph.
  Compute the pairwise distances matrix $D_{n\times n}$.
  Compute $A = -\frac{1}{2}D$.
  Compute $H = I - \frac{1}{n}U$, where U is a $n \times n$ matrix of 1's.
  Compute $B = HAH$.
  Find the eigenvalues and eigenvectors of the matrix B.
  Select the top $d < m$ eigenvalues and eigenvalues of B and define:

$$\tilde{V} = \begin{bmatrix} | & | & \dots & \dots & | \\ | & | & \dots & \dots & | \\ \vec{v}_1 & \vec{v}_2 & \dots & \dots & \vec{v}_d \\ | & | & \dots & \dots & | \\ | & | & \dots & \dots & | \end{bmatrix}_{n\times d} \tag{12}$$

$$\tilde{\Lambda} = diag(\lambda_1, \lambda_2, ..., \lambda_d) \tag{13}$$

  Compute $\tilde{X} = \tilde{\Lambda}^{1/2}\tilde{V}^T$
  return $\tilde{X}$
end function

III. Kullback-Leibler divergence

The problem of quantifying a suitable similarity measure between different objects in a dataset is a challenging task in many pattern recognition and machine learning applications [14]. Finding alternative similarity measures is crucial for modern data analysis, especially in situations where the standard Euclidean distance becomes an unreasonable choice [15]. Among feature selection methods it is usual to adopt stochastic divergences to build the set of features that maximize class separability [16]. Information-theoretic measures have been successfully applied in statistics to quantify the degree of similarity between random variables [17]. In this context, the concepts of entropy and relative entropy can be used as a solid mathematical background for metric learning. First, we define the entropy of a random variable x as the expected value of the self-information, that is, the average of the negative of the logarithm of the probabilities:

$$H(p) = -\int p(x)[log\ p(x)]dx = -E\left[log\ p(x)\right] \tag{14}$$

where $p(x)$ is the probability density function (pdf) of x. When $\vec{x} \in R^m$ is a random vector following a multivariate Gaussian distribution $N(\vec{\mu}, \Sigma)$, the pdf $p(\vec{x})$ is expressed by:

$$p(\vec{x}) = \frac{1}{(2\pi)^{m/2}|\Sigma|^{1/2}} exp\left\{-\frac{1}{2}(\vec{x}-\vec{\mu})^T\Sigma^{-1}(\vec{x}-\vec{\mu})\right\} \tag{15}$$

where $\vec{\mu}$ is the location parameter (mean) and Σ is the covariance matrix of the random vector $\vec{x}$. By taking the logarithm and computing the expected value we have:

$$H(p) = \frac{1}{2}log\ |\Sigma| + \frac{d}{2}(1 + log\ 2\pi) \tag{16}$$

In a similar way, we can define the cross-entropy between two probability density functions as:

$$H(p,q) = -\int p(x)[log\ q(x)]dx \tag{17}$$

The Kullback-Leibler divergence, or simply relative entropy, is the difference between the cross-entropy of $p(x)$ and $q(x)$ and the entropy of $p(x)$, that is:

$$\begin{aligned} D_{KL}(p,q) = H(p,q) - H(p) &= \int p(x) log\left(\frac{p(x)}{q(x)}\right)dx \\ &= E_p\left[log\left(\frac{p(x)}{q(x)}\right)\right] \end{aligned} \tag{18}$$

It should be mentioned that the relative entropy is always non-negative, that is, $D_{KL}(p,q) \geq 0$, being equal to zero if, and only if, $p(x) = q(x)$. Now suppose that we want to compute the KL-divergence between two multivariate Gaussian densities: $N(\vec{\mu}_1, \Sigma_1)$ and $N(\vec{\mu}_2, \Sigma_2)$. Let the parameter vector $\vec{\theta} = \{\vec{\mu}, \Sigma\}$. Then, we have:

$$\begin{aligned} D_{KL}(p,q) &= E_p\left[log\ p(\vec{x};\vec{\theta}) - log\ q(\vec{x};\vec{\theta})\right] \\ &= E_p\left[-\frac{1}{2}log|\Sigma_1| - \frac{1}{2}(\vec{x}-\vec{\mu}_1)^T\Sigma_1^{-1}(\vec{x}-\vec{\mu}_1) + \frac{1}{2}log|\Sigma_2| + \frac{1}{2}(\vec{x}-\vec{\mu}_2)^T\Sigma_2^{-1}(\vec{x}-\vec{\mu}_2)\right] \\ &= \frac{1}{2}log\left(\frac{|\Sigma_2|}{|\Sigma_1|}\right) - \frac{1}{2}E_p\left[(\vec{x}-\vec{\mu}_1)^T\Sigma_1^{-1}(\vec{x}-\vec{\mu}_1)\right] + \frac{1}{2}E_p\left[(\vec{x}-\vec{\mu}_2)^T\Sigma_2^{-1}(\vec{x}-\vec{\mu}_2)\right] \\ &= \frac{1}{2}log\left(\frac{|\Sigma_2|}{|\Sigma_1|}\right) - \frac{1}{2}E_p\left[Tr\left[\Sigma_1^{-1}(\vec{x}-\vec{\mu}_1)(\vec{x}-\vec{\mu}_1)^T\right]\right] + \frac{1}{2}E_p\left[Tr\left[\Sigma_2^{-1}(\vec{x}-\vec{\mu}_2)(\vec{x}-\vec{\mu}_2)^T\right]\right] \\ &= \frac{1}{2}log\left(\frac{|\Sigma_2|}{|\Sigma_1|}\right) - \frac{1}{2}Tr\left[\Sigma_1^{-1}\Sigma_1\right] + \frac{1}{2}E_p\left[Tr\left[\Sigma_2^{-1}(\vec{x}\vec{x}^T - 2\vec{x}\vec{\mu}_2^T + \vec{\mu}_2\vec{\mu}_2^T)\right]\right] \\ &= \frac{1}{2}log\left(\frac{|\Sigma_2|}{|\Sigma_1|}\right) - \frac{m}{2} + \frac{1}{2}Tr\left[\Sigma_2^{-1}\left(\Sigma_1 + \vec{\mu}_1\vec{\mu}_1^T - 2\vec{\mu}_1\vec{\mu}_2^T + \vec{\mu}_2\vec{\mu}_2^T\right)\right] \\ &= \frac{1}{2}log\left(\frac{|\Sigma_2|}{|\Sigma_1|}\right) - \frac{m}{2} + \frac{1}{2}Tr\left[\Sigma_2^{-1}\Sigma_1\right] + \frac{1}{2}\left(\vec{\mu}_1^T\Sigma_2^{-1}\vec{\mu}_1 - 2\vec{\mu}_1^T\Sigma_2^{-1}\vec{\mu}_2 + \vec{\mu}_2^T\Sigma_2^{-1}\vec{\mu}_2\right) \\ &= \frac{1}{2}\left[log\left(\frac{|\Sigma_2|}{|\Sigma_1|}\right) - m + Tr\left[\Sigma_2^{-1}\Sigma_1\right] + (\vec{\mu}_2-\vec{\mu}_1)^T\Sigma_2^{-1}(\vec{\mu}_2-\vec{\mu}_1)\right] \end{aligned} \tag{19}$$

Note that the KL-divergence is not symmetric. Similarly, it can be shown that the KL-divergence $D_{KL}(q,p)$ is given by:

$$D_{KL}(q,p) = \frac{1}{2}\left[log\left(\frac{|\Sigma_1|}{|\Sigma_2|}\right) - m + Tr\left[\Sigma_1^{-1}\Sigma_2\right] + (\vec{\mu}_1 - \vec{\mu}_2)^T\Sigma_1^{-1}(\vec{\mu}_1 - \vec{\mu}_2)\right] \quad (20)$$

The symmetrized KL-divergence can be computed by:

$$D_{KL}^{sym}(p,q) = \frac{1}{2}\left[D_{KL}(p,q) + D_{KL}(q,p)\right] \quad (21)$$

which has a closed-form expression:

$$D_{KL}^{sym}(p,q) = \frac{1}{2}\left[\frac{1}{2}Tr\left(\Sigma_1^{-1}\Sigma_2 + \Sigma_2^{-1}\Sigma_1\right) + \frac{1}{2}(\vec{\mu}_1 - \vec{\mu}_2)^T\Sigma_1^{-1}(\vec{\mu}_1 - \vec{\mu}_2) + \frac{1}{2}(\vec{\mu}_2 - \vec{\mu}_1)^T\Sigma_2^{-1}(\vec{\mu}_2 - \vec{\mu}_1) - m\right] \quad (22)$$

In this paper, we use equation (22) as a parametric similarity measure between patches, that is, the Euclidean distance between two points in the KNN graph will be replaced by the symmetrized KL-divergence between two local neighborhoods, under the hypothesis that, locally, these points are normally distributed.

IV. Proposed method

The main motivation of the proposed parametric method is the investigation of a surrogate for the usual KNN graph, by replacing the pointwise Euclidean distance in the feature space by KL-divergences between Gaussian models estimated in local patches. We denote by $X = \{\vec{x}_1, \vec{x}_2, ..., \vec{x}_n\}$, with $\vec{x}_i \in R^m$ the input data matrix. We can build a KNN graph $G = (V, E)$, with $|V| = n$, by connecting each sample $\vec{x}_i$ with its k nearest neighbors. Since the neighborhood can be well approximated by a linear patch, we use the Euclidean distance as similarity measure in this step. Let a patch P_i be the set $\{\vec{x}_i\} \cup \{\vec{x}_j \in N(i)\}$, where $N(i)$ is the neighborhood set of $\vec{x}_i$. Then, we can define the patch matrix P_i as:

$$P_i = [\vec{x}_i, \vec{x}_{i1}, \vec{x}_{i2}, ..., \vec{x}_{ik}] \quad (23)$$

to denote the $m \times (k+1)$ data matrix that compose the i-th patch. We assume P_i is a random sample of a multivariate Gaussian distribution of size k. Hence, we can compute the maximum-likelihood estimators of the model parameters as:

$$\vec{\mu}_i = \frac{1}{(k+1)}\sum_{j=1}^{k+1}\vec{x}_{ij} \quad (24)$$

$$\Sigma_i = \frac{1}{k}\sum_{j=1}^{k+1}(\vec{x}_{ij} - \vec{\mu}_i)(\vec{x}_{ij} - \vec{\mu}_i)^T \quad (25)$$

Figure 1 illustrates the mapping of local patches in the KNN graph to a parametric representation. In this parametric feature space, the relative entropy is a more meaningful measure of similarity than the usual Euclidean distance.

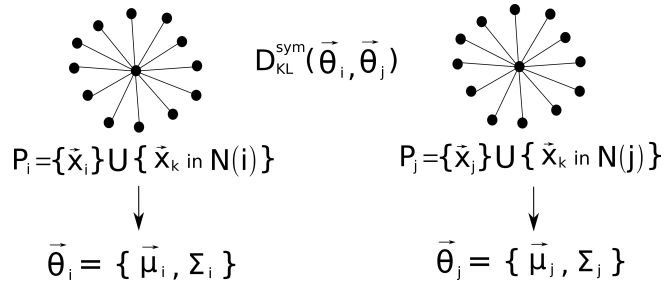

Fig. 1. Mapping from a patch P_i on the graph to a parametric feature vector.

The next step consists in building the entropic KNN graph (or KL-KNN graph), as a replacement for the traditional KNN graph used in ISOMAP. Basically, this is done by updating the edge weights in the KNN graph. Instead of using the Euclidean distance between the vectors $\vec{x}_i$ and $\vec{x}_j$, we compute the symmetrized KL-divergence D_{KL}^{sym} between the respective patches P_i and P_j using equation (22). Note that $D_{KL}^{sym}(P_i, P_j)$ is a patch-based similarity measure, which means that it is less sensitive to the presence of outliers and random noise in data than the pointwise Euclidean distance, employed by the traditional ISOMAP algorithm.

Given the entropic KNN graph, we proceed to the computation of the geodesic distances by finding the pairwise shortest paths. At the end of this procedure, ISOMAP-KL produces the so called entropic distance matrix E, which is a surrogate for the pairwise distance matrix of ISOMAP. The next steps are identical to those in ISOMAP, that is, from the entropic distance matrix E, we compute the Gram matrix of the inner products B and through spectral decomposition we find the leading eigenvectors.

V. Experiments and results

In order to test and evaluate the proposed method for unsupervised metric learning in classification tasks, we compared its performance against the usual PCA, Kernel PCA, ISOMAP, LLE and Laplacian Eigenmaps in several public datasets available at www.openml.org. It is worth mentioning that the selected datasets have significant variations in the number of samples and features, as well as different number of classes.

In the first set of experiments, we used an internal index to assess the quality of the clusters obtained after the unsupervised metric learning provided by different dimensionality reduction methods. Our choice was the Silhouette coefficient, which is a method of interpretation and validation of consistency within clusters of data [18]. Let C_i denote the i-th cluster, then for each data point $i \in C_i$ let $a(i)$ be the mean distance between i and all other points in the same cluster C_i:

$$a(i) = \frac{1}{|C_i| - 1}\sum_{j \in C_i, j \neq i} d(i,j) \quad (26)$$

TABLE I
SILHOUETTE COEFFICIENTS FOR CLUSTERS PRODUCED BY PCA, KERNEL PCA, ISOMAP, ISOMAP-KL, LLE AND LAPLACIAN EIGENMAPS FOR SEVERAL DATASETS FROM OPENML.ORG (2D CASE).

	PCA	KPCA	ISO	ISO-KL	LLE	LAP
iris	0.401	0.469	0.423	**0.576**	0.297	0.539
wine	0.526	0.610	0.533	0.656	0.140	**0.728**
mfeat-four.	0.000	0.011	0.016	**0.145**	-0.073	-0.006
texture	-0.058	-0.050	0.086	**0.348**	0.068	0.245
satimage	0.219	0.247	0.232	**0.349**	0.037	0.233
theorem	-0.168	-0.105	**-0.099**	-0.156	-0.113	-0.466
synthetic	0.346	0.459	0.361	**0.512**	0.146	0.501
car	-0.110	-0.129	-0.075	-0.034	**0.000**	-0.111
tae	-0.059	**-0.004**	-0.069	-0.118	-0.017	-0.019
transplant	0.485	0.436	0.483	**0.582**	0.407	0.439
hayes	-0.023	0.038	-0.020	**0.234**	0.085	-0.012
SPECTF	-0.018	0.093	-0.028	**0.106**	-0.083	0.046
servo	**0.121**	0.105	0.102	0.034	0.097	0.085
mu284	0.301	0.338	0.288	0.306	**0.346**	0.306
triazines	0.009	**0.064**	0.017	0.023	0.001	0.017
pageblock	0.419	0.218	**0.534**	0.450	0.402	0.299
male-lung	0.563	-0.182	0.676	**0.988**	0.629	0.019
retinol	-0.008	0.004	0.004	**0.038**	0.001	0.015
diggle	0.406	0.409	0.412	**0.430**	0.272	0.363
rmftsa	0.228	0.242	0.235	**0.258**	0.188	0.231
Average	0.179	0.164	0.206	**0.286**	0.142	0.173
Std. Dev.	0.236	0.229	0.243	0.292	**0.196**	0.271

where $d(i,j)$ is the distance between data points i and j in the cluster C_i. In other words, we can interpret $a(i)$ as a measure of how well the data point i is assigned to its cluster (the smaller the value, the better). Then, we define the mean dissimilarity of a data point i to a cluster C as the mean of the distances from i to all points in C. For each data point i, let $b(i)$ be the smallest mean distance of i to all points in any other cluster which i is not a member:

$$b(i) = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} d(i,j) \tag{27}$$

The cluster with the smallest mean dissimilarity is the neighboring cluster of i because it is the next best fit cluster for point i. Let:

$$s(i) = \frac{b(i) - a(i)}{max\{a(i), b(i)\}}, \text{ if } |C_i| > 1 \tag{28}$$

be the silhouette value of the data point i and

$$s(i) = 0. \text{ if } |C_i| = 1 \tag{29}$$

Combining both definitions we have:

$$s(i) = \begin{cases} 1 - \dfrac{a(i)}{b(i)} & \text{if } a(i) < b(i) \\ 0 & \text{if } a(i) = b(i) \\ \dfrac{b(i)}{a(i)} - 1 & \text{if } a(i) > b(i) \end{cases} \tag{30}$$

Note that $-1 \leq s(i) \leq 1$. An $s(i)$ close to one means that the data is appropriately clustered. If $s(i)$ tends to negative one, then i should be clustered in its neighboring cluster. An $s(i)$ near zero means that the data point i is on the border of two natural clusters. The mean $s(i)$ over all points of a cluster is a measure of how tightly grouped all the points in the cluster are. Therefore, the mean $s(i)$ over all data of the entire dataset, known as the Silhouette coefficient is a measure of how appropriately the data have been clustered.

Table I shows the obtained results for 20 different datasets, where column ISO-KL denotes the proposed parametric method under multivariate Gaussian hypothesis. It is worth mentioning that the definition of the parameter K (patch size) plays an important role in the proposed ISOMAP-KL. Our method is sensitive to variations on this parameter, which essentially controls the patch size. Different values of K can lead to significantly different classification results. In all experiments, we performed a simple heuristic: to evaluate the classification accuracy for the all values of K ranging from 10 to 200. using an increment of 10 units. In other words, we considered as candidates the values of K belonging to the interval $S = [10.20.30.40...., 100...., 200]$. The best result was then selected for each dataset. An intuition behind this choice is that a low K is usually preferred in small datasets, to preserve the locality of the patches. Ideally, one should keep in mind the tradeoff between locality preservation, which means choosing a small K, and having a large enough sample size for suitable parameter estimation. The value of K is global, but our goal in future works is to investigate adaptive ways to set the patch size based on local properties of the data.

The results suggest that, in average, the proposed ISOMAP-KL is more efficient in building a meaningful representation in terms of the consistency within clusters of data than the other methods for these datasets. Moreover, note that in 12 of 20 datasets, ISOMAP-KL obtained the highest Silhouette coefficient, that is, in 60% of the cases the proposed method produced better defined clusters. To test if the differences are significant, we performed a statistical test to compare the different groups. According to a non-parametric Friedman test, there is strong evidences against the null hypothesis that there are no significant differences between the groups (p-value = 7.49×10^{-5}) for a significance level $\alpha = 0.05$. According to a post-hoc Nemenyi test, for the same significance level, ISOMAP-KL produced significantly better clusters (in terms of Silhouette coefficient) than PCA (p-value = 4.15×10^{-5}), Kernel PCA (p-value = 0.0425), ISOMAP (p-value = 0.0201), LLE (p-value = 2.87×10^{-5}) and Laplacian Eigenmaps (p-value = 0.0046).

In the second set of experiments, we compared the performance of the proposed method against PCA, Kernel PCA, ISOMAP, LLE and Laplacian Eigenmaps in supervised classification. For this purpose, eight different parametric and non-parametric classifiers were selected: K-Nearest Neighbors (KNN) with $K = 7$, Support Vector Machine (SVM) (linear), Naive Bayes (NB), Decision Trees (DT), Quadratic Discriminant Analysis (QDA) under Gaussian hypothesis, Multilayer Perceptron (MPL), Gaussian Process Classifier (GPC) and Random Forest Classifier (RFC). In all experiments, we selected 50% of the samples for training and 50% of the

TABLE II
SUPERVISED CLASSIFICATION ACCURACY OBTAINED BY DIFFERENT CLASSIFIERS AFTER PCA, KERNEL PCA, ISOMAP, ISOMAP-KL, LLE AND LAPLACIAN EIGENMAPS FOR SEVERAL DATASETS FROM OPENML.ORG (2D CASE).

	PCA	**KPCA**	**ISO**	**ISO-KL**	**LLE**	**LAP**
iris dataset (k = 20)						
KNN	**0.960**	0.866	0.866	**0.960**	**0.960**	0.826
SVM	**0.946**	0.800	0.880	**0.946**	0.413	0.440
NB	0.906	0.826	0.826	**1.000**	0.906	0.866
DT	0.933	0.800	0.760	**0.960**	0.933	0.800
QDA	**0.946**	0.800	0.866	**0.946**	0.946	0.813
MPL	0.946	0.826	0.866	0.946	**0.960**	0.306
GPC	0.906	0.826	0.853	**0.946**	0.613	0.440
RFC	0.920	0.880	0.840	0.946	**0.960**	0.813
wine dataset (k = 40)						
KNN	0.966	0.977	**0.988**	0.966	0.752	**0.988**
SVM	0.955	0.977	0.943	0.966	0.382	0.382
NB	0.943	**0.955**	**0.955**	0.943	0.730	**0.955**
DT	0.943	0.932	0.943	**0.977**	0.629	0.966
QDA	0.966	0.966	0.966	**0.977**	0.808	0.955
MPL	0.955	**0.988**	0.966	0.977	0.382	0.382
GPC	0.966	0.966	0.966	**0.988**	0.404	0.382
RFC	0.966	0.955	0.932	**0.988**	0.685	0.977
mfeat-fourier dataset (k = 40)						
KNN	0.415	0.435	0.398	**0.626**	0.454	0.451
SVM	0.424	0.382	0.401	**0.450**	0.088	0.088
NB	0.415	0.431	0.428	**0.576**	0.469	0.409
DT	0.366	0.400	0.351	**0.542**	0.405	0.405
QDA	0.436	0.459	0.439	**0.595**	0.482	0.428
MPL	0.433	0.450	0.430	**0.637**	0.370	0.088
GPC	0.428	0.410	0.406	**0.547**	0.179	0.088
RFC	0.389	0.430	0.370	**0.580**	0.427	0.448
texture dataset (k = 40)						
KNN	0.583	0.543	0.712	**0.846**	0.460	0.622
SVM	0.579	0.469	0.730	**0.732**	0.083	0.083
NB	0.491	0.460	0.594	**0.800**	0.485	0.621
DT	0.485	0.470	0.646	**0.810**	0.408	0.545
QDA	0.541	0.461	0.661	**0.819**	0.705	0.760
MPL	0.568	0.419	0.714	**0.840**	0.474	0.087
GPC	0.578	0.463	0.732	**0.826**	0.304	0.083
RFC	0.538	0.522	0.704	**0.844**	0.408	0.559
satimage dataset (k = 200)						
KNN	0.826	0.800	0.837	**0.852**	0.621	0.835
SVM	0.835	0.792	0.852	**0.854**	0.230	0.230
NB	0.806	0.781	0.794	**0.835**	0.616	0.739
DT	0.779	0.753	0.780	**0.803**	0.534	0.798
QDA	0.827	0.787	**0.830**	0.827	0.622	0.822
MPL	0.828	0.788	0.840	**0.847**	0.604	0.230
GPC	0.837	0.778	0.845	**0.852**	0.372	0.230
RFC	0.818	0.799	0.829	**0.846**	0.605	0.831

TABLE III
SUPERVISED CLASSIFICATION ACCURACY OBTAINED BY DIFFERENT CLASSIFIERS AFTER PCA, KERNEL PCA, ISOMAP, ISOMAP-KL, LLE AND LAPLACIAN EIGENMAPS FOR SEVERAL DATASETS FROM OPENML.ORG (2D CASE).

	PCA	**KPCA**	**ISO**	**ISO-KL**	**LLE**	**LAP**
first-order-theorem dataset (k = 40)						
KNN	0.460	0.478	0.467	**0.511**	0.421	0.445
SVM	0.447	0.410	0.496	**0.521**	0.410	0.410
NB	0.410	0.410	0.413	**0.422**	0.418	0.138
DT	0.423	0.449	0.457	**0.498**	0.377	0.434
QDA	0.410	0.423	**0.430**	0.405	0.418	0.150
MPL	0.412	0.410	0.431	**0.457**	0.410	0.410
GPC	0.443	0.414	0.493	**0.512**	0.410	0.410
RFC	0.483	0.492	0.490	**0.516**	0.409	0.442
hayes-roth dataset (k = 15)						
KNN	0.424	0.651	0.545	0.742	**0.833**	0.590
SVM	0.606	0.606	0.530	**0.742**	0.606	0.606
NB	0.606	0.560	0.606	**0.803**	0.636	0.606
DT	0.621	0.803	0.636	**0.818**	0.757	0.636
QDA	0.606	0.575	0.606	**0.833**	0.681	0.606
MPL	0.606	0.606	0.606	**0.848**	0.606	0.606
GPC	0.500	0.606	0.515	**0.818**	0.606	0.606
RFC	0.666	0.696	0.636	**0.803**	**0.803**	0.621
SPECTF dataset (k = 80)						
KNN	0.771	0.754	0.685	**0.788**	0.725	0.731
SVM	0.742	0.742	0.714	**0.811**	0.742	0.742
NB	0.725	0.742	0.720	**0.754**	0.640	0.702
DT	0.782	0.754	**0.794**	**0.794**	0.714	0.760
QDA	0.742	0.742	0.720	**0.760**	0.605	0.742
MPL	0.742	**0.794**	0.771	0.771	0.742	0.742
GPC	0.754	**0.777**	0.691	0.765	0.742	0.742
RFC	0.794	0.731	0.771	**0.840**	0.742	0.777
servo dataset (k = 10)						
KNN	0.761	0.750	0.821	**0.916**	0.797	0.750
SVM	0.797	0.750	0.750	**0.916**	0.750	0.750
NB	0.928	0.738	0.821	**0.905**	0.809	0.821
DT	**0.904**	0.738	0.773	0.845	0.690	0.797
QDA	**0.916**	0.714	0.821	**0.916**	0.809	0.809
MPL	0.821	0.809	0.821	**0.905**	0.750	0.750
GPC	0.833	0.750	0.821	**0.905**	0.750	0.750
RFC	**0.916**	0.773	0.785	0.845	0.726	0.821
page-blocks dataset (k = 100)						
KNN	0.921	0.928	0.952	**0.957**	0.932	0.941
SVM	0.925	0.925	0.953	**0.956**	0.900	0.900
NB	0.879	0.922	0.897	**0.942**	0.903	0.942
DT	0.889	0.916	0.940	**0.955**	0.904	0.924
QDA	0.892	0.924	0.901	0.941	0.926	**0.942**
MPL	0.904	0.911	**0.949**	0.948	0.920	0.900
GPC	0.923	0.924	0.950	**0.962**	0.900	0.900
RFC	0.919	0.931	0.955	**0.963**	0.927	0.943

samples for testing. Tables II and III show the classification accuracies for several datasets after dimensionality reduction to 2D spaces. The results show that there is no method that is uniformly superior to all the other ones. However, looking at the average accuracy, the results are more conclusive. Table IV shows the average and standard deviation of all accuracies for each dimensionality reduction algorithm. The results indicate that for these datasets, in average, the proposed parametric ISOMAP-KL outperformed all the other methods. We also performed a hypothesis test to check whether the differences are statistically significant. According to a non-parametric Friedman test, there are strong evidences for rejecting the null hypothesis that all DR methods are equivalent (p-value = 1.12×10^{-15}) for a significant level $\alpha = 0.05$. According to a post-hoc Nemenyi test, ISOMAP-KL produced significantly better classification accuracies than PCA (p-value = 3.11×10^{-12}), Kernel PCA (p-value = 10^{-18}), LLE (p-value = 10^{-19}) and Laplacian Eigenmaps (p-value = 10^{-19}).

The obtained results emphasize that the proposed ISOMAP-KL is competitive with the existing dimensionality reduction

TABLE IV
AVERAGE CLASSIFICATION ACCURACIES OBTAINED BY DIFFERENT CLASSIFIERS FOR OPENML.ORG DATASETS IN TABLES II, AND III.

	PCA	KPCA	ISO	ISO-KL	LLE	LAP
Average	0.721	0.698	0.723	**0.807**	0.621	0.613
Std. Dev.	0.203	0.191	0.184	**0.160**	0.218	0.263

algorithms, since, overall, it is capable of producing features that are more discriminant than those generated by PCA, Kernel PCA and some manifold learning algorithms. In other words, we conclude that ISOMAP-KL is a viable option for unsupervised metric learning in pattern classification tasks. To illustrate how the proposed method is capable of producing better defined clusters, we present some scatter plots for the two dimensional case, comparing ISOMAP and ISOMAP-KL. Figures 2 and 3 show the clusters for the mfeat-fourier and texture datasets. Note that the clusters produced by ISOMAP-KL show less overlapping, that is, they tend to be easier to discriminate by pattern classifiers.

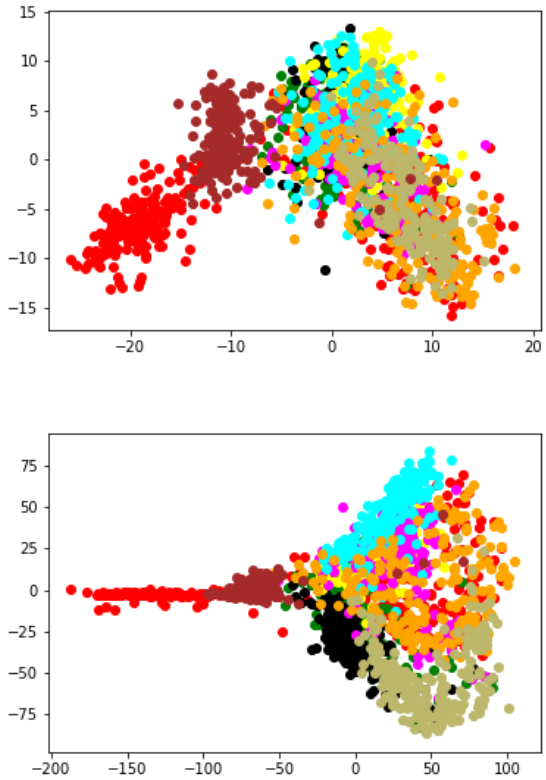

Fig. 2. Scatterplots of mfeat-fourier dataset for the 2D case: ISOMAP (above) versus ISOMAP-KL (below)

VI. CONCLUSION

Unsupervised metric learning is a fundamental step in many pattern recognition problems dealing with high-dimensional data. In this scenario, algorithms for dimensionality reduction

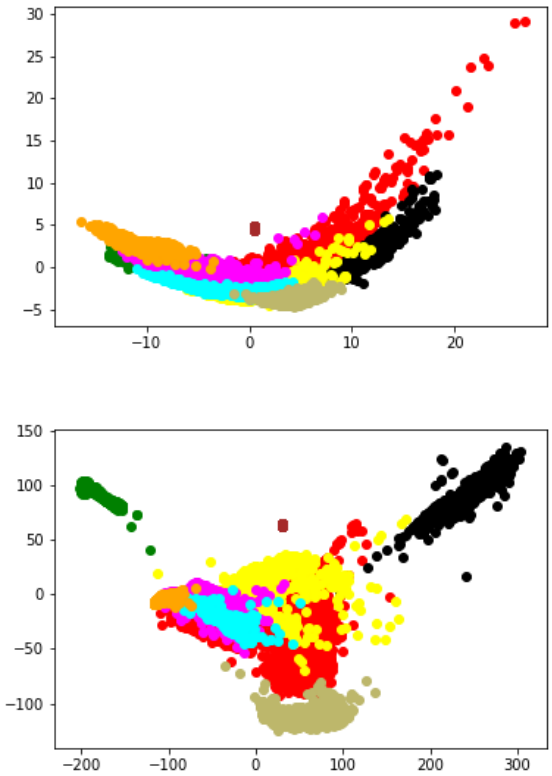

Fig. 3. Scatterplots of texture dataset for the 2D case: ISOMAP (above) versus ISOMAP-KL (below)

play an important role, since besides learning an adaptive distance function for each dataset, they also learn an optimal representation for the observed data in terms of compression. In this paper, we presented ISOMAP-KL, a parametric patch-based method based on the KL-divergence that maps neighborhoods of the KNN graph to a feature space in which a surrogate for the pairwise distance matrix is obtained by replacing the usual Euclidean distance by the symmetrized relative entropy between local statistical models. Results with several real datasets indicate that besides improving the quality of the clusters, which is a desirable feature in unsupervised classification, the proposed method can also improve the supervised classification accuracy, indicating that it can be better suited to unsupervised metric learning than regular PCA, Kernel PCA and some manifold learning algorithms.

Basically, the main positive points of ISOMAP-KL can be summarized as: 1) ISOMAP-KL is a patch-based approach so it is less sensitive to the presence of noise and outliers in data (the entropic distance matrix is computed between pairs of patches instead of pair of isolated points); 2) the method can be easily extended to different statistical models and divergences, such as Bhattacharyya and Hellinger distances. On the other hand, ISOMAP-KL has limitations, the major one being the sensitivity to the patch size K. Experiments have shown that variations on this parameter can produce significantly

different classification results. We still do not have a complete solution regarding the estimation of this parameter for each specific dataset, but we hope to study this problem as a future improvement.

Future works may include the incorporation of Fisher information based distances, which has been shown to be the metric tensor of the parametric space. Besides considering other information-theoretic measures, we aim to use different statistical models. For instance, it has been observed that several datasets have multi-modal features. Gaussian Mixture Models (GMM's) can be used to model this kind of behavior in an elegant way. Gaussian-Markov random fields (GMRF's) are particularly interesting mathematical structures, since it is possible to replace the usual statistical independence assumption by a more realistic conditional independence hypothesis. In other words, unlike most classical statistical models, we can incorporate the dependence between random variables in a formal way. Multivariate Generalized Gaussian distributions (MGGD's) can also be an alternative, since in many cases data shows signs of non-Gaussian behavior. Finally, another relevant problem is the adaptive definition of the appropriate patch size. We intend to perform local analysis of the Hessian matrix in order to bring insights about how to adjust the K parameter. Points with higher curvature should have a smaller neighborhood whereas points with lower curvature could have larger neighborhoods.

Acknowledgments

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

References

[1] E. Debie and K. Shafi, "Implications of the curse of dimensionality for supervised learning classifier systems: theoretical and empirical analyses," *Pattern Analysis and Applications*, vol. 22, pp. 519–536, 2019.

[2] D. Li and Y. Tian, "Survey and experimental study on metric learning methods," *Neural Networks*, vol. 105, pp. 447–462, 2018.

[3] F. Wang and J. Sun, "Survey on distance metric learning and dimensionality reduction in data mining," *Data Min. Knowl. Discov.*, vol. 29, no. 2, pp. 534–564, Mar. 2015.

[4] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[5] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning." *Nature*, vol. 521, pp. 436–444, 2015.

[6] P. Hurtik, V. Molek, and I. Perfilieva, "Novel dimensionality reduction approach for unsupervised learning on small datasets," *Pattern Recognition*, vol. 103, p. 107291, 2020.

[7] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, pp. 2319–2323, 2000.

[8] J. A. Lee and M. Verleysen, *Nonlinear Dimensionality Reduction*. Springer, 2007.

[9] S. Kullback and R. A. Leibler, "On information and sufficiency," *Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[10] T. F. Cox and M. A. A. Cox, *Multidimensional Scaling*, ser. Monographs on Statistics and Applied Probability. Chapman & Hall, 2001, vol. 88.

[11] I. Borg and P. Groenen, *Modern Multidimensional Scaling: theory and applications*, 2nd ed. Springer-Verlag, 2005.

[12] U. von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, pp. 395–416, 2007.

[13] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 3rd ed. MIT Press, 2009.

[14] A. S. Shirkhorshidi, S. Aghabozorgi, and T. Y. Wah, "A comparison study on similarity and dissimilarity measures in clustering continuous data," *PLOS ONE*, vol. 10, no. 12, pp. 1–20, 12 2015.

[15] C. C. Aggarwal, A. Hinneburg, and D. A. Keim, "On the surprising behavior of distance metrics in high dimensional space," in *Database Theory — ICDT 2001*, J. Van den Bussche and V. Vianu, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 420–434.

[16] Y. Zhang, S. Li, T. Wang, and Z. Zhang, "Divergence-based feature selection for separate classes," *Neurocomputing*, vol. 101, pp. 32 – 42, 2013.

[17] F. Pérez-Cruz, "Estimation of information theoretic measures for continuous random variables," in *Advances in Neural Information Processing Systems 21*, D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, Eds. Curran Associates, Inc., 2009, pp. 1257–1264.

[18] P. J. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Comp. and Appl. Math.*, vol. 20, pp. 53–65, 1987.