Entropic Laplacian eigenmaps for unsupervised metric learning

Alexandre Luís Magalhães Levada [*], Michel Ferreira Cardia Haddad [†‡]
[*] Computing Department, Federal University of São Carlos, São Carlos, SP, Brazil
alexandre.levada@ufscar.br
[†] Department of Land Economy, University of Cambridge, Cambridge, United Kingdom
[‡] School of Business and Management, Queen Mary University of London, London, United Kingdom
m.haddad@qmul.ac.uk

***Abstract*—Unsupervised metric learning is concerned with building adaptive distance functions prior to pattern classification. Laplacian eigenmaps consists of a manifold learning algorithm which uses dimensionality reduction to find more compact and meaningful representations of datasets through the Laplacian matrix of graphs. In the present paper, we propose the entropic Laplacian eigenmaps (ELAP) algorithm, a parametric approach that employs the Kullback–Leibler (KL) divergence between patches of the KNN graph instead of the pointwise Euclidean metric as the cost function for the graph weights. Our objective with such a modification is increasing the robustness of Laplacian eigenmaps against noise and outliers. Our results using various real-world datasets indicate that the proposed method is capable of generating more reasonable clusters while reporting greater classification accuracies compared to existing widely adopted methods for dimensionality reduction-based metric learning.**

I. Introduction

Dimensionality reduction-based metric learning aims to recover the underlying geometric structure of data while capturing a non-Euclidean distance function that is better suited to represent similarity between samples. Manifold learning algorithms are capable of finding more compact and meaningful representations of the observed data by preserving the intrinsic non-Euclidean geometry of the data. The Laplacian eigenmaps is a manifold learning algorithm for non-linear dimensionality reduction based on the Laplacian matrix of graphs [1]. One of the main drawbacks of Laplacian eigenmaps (LAP) is that such a method may be remarkably influenced by the presence of noise and outliers. The performance of the LAP is severely affected in datasets which do not lead to smooth manifolds.

The rationale of the Laplacian eigenmaps method is that if we approximate a manifold by a connected and undirected basic graph, then it would be feasible to find a map from the vertices of the graph onto a d-dimensional Euclidean subspace R^d, such that the locality is preserved (i.e., the map is smooth in the sense that neighboring points in the graph remains spatially close after the mapping being performed). Such a map is produced by the eigenvectors of the graph Laplacian matrix [2]. The representation map generated by the respective algorithm resembles to a discrete approximation to a continuous map that naturally arises from the geometry of the manifold (i.e., the Laplace-Beltrami operator) [3]. The eigenvectors of the graph Laplacian associated to a point cloud converges to the eigenfunctions of the Laplace-Beltrami operator in the case data follows a uniform probability distribution on an embedded manifold [4]. In the machine learning (ML) literature, the Laplacian eigenmaps method is closely related to spectral clustering (i.e., unsupervised classification approach for data clustering) [5].

In the present paper, we propose the entropic Laplacian eigenmaps (ELAP), a parametric method that incorporates the relative entropy in the estimation of the Laplacian matrix of the KNN graph. There are two main contributions of this paper. Firstly, we replace the pointwise Euclidean distance by a patch-based information-theoretic distance (Kullback–Leibler or KL-divergence), resulting in a less sensitive method to noise and outliers. Secondly, our results exploring more than 20 real-world datasets suggest that the proposed method produces more reasonable clusters as well as larger classification accuracies compared to existing popular supervised classifiers and manifold learning algorithms, as ISOMAP [6] and LLE [7].

The remainder of this paper paper is organized as follows. Section 2 summarizes related methods. Section 3 details the proposed ELAP algorithm. Section 4 reports computational experiments and results. Section 5 concludes and suggests future directions in the literature of metric learning-based dimensionality reduction.

II. Related work

In this section, we discuss the relative entropy or KL-divergence, Laplacian matrix, Laplacian eigenmaps algorithm and its relation with graph-cuts in graphs.

A. Relative entropy

In ML applications, the problem of quantifying similarity between objects or clusters is a challenging task, especially in cases in which the standard Euclidean distance is not a reasonable alternative. Many studies on feature selection adopt statistical divergences to select the set of features that maximize some measure of separation between classes. Part of their success is due the fact that most dissimilarity measures are related to distance metrics. In such a context, information theory provides a solid mathematical background for metric learning in pattern classification. The entropy of a continuous random vector x is given as follows:

$$H(p) = -\int p(x)[log\ p(x)]dx = -E\left[log\ p(x)\right] \quad (1)$$

where $p(x)$ is the probability density function (PDF). In a similar fashion, we may define the cross-entropy between the PDFs $p(x)$ and $q(x)$ as follows:

$$H(p,q) = -\int p(x)[log\ q(x)]dx \quad (2)$$

The relative entropy consists of the difference between the cross-entropy and entropy [8], as formulated below:

$$\begin{aligned} D_{KL}(p,q) &= H(p,q) - H(p) \quad (3) \\ &= -\int p(x)[log\ q(x)]dx + \int p(x)[log\ p(x)]dx \\ &= \int p(x) log\left(\frac{p(x)}{q(x)}\right)dx = E_p\left[log\left(\frac{p(x)}{q(x)}\right)\right] \end{aligned}$$

A relevant property is that the relative entropy is non-negative, in the sense that $D_{KL}(p,q) \geq 0$. First, it is worth noting that $log\ x \leq x - 1$, which leads to:

$$\begin{aligned} -D_{KL}(p,q) &= \int p(x) log\left(\frac{p(x)}{q(x)}\right)dx \quad (4) \\ &\leq \int p(x)\left[\frac{q(x)}{p(x)} - 1\right]dx \\ &= \int q(x) - \int p(x) = 0 \end{aligned}$$

B. Laplacian matrix of a graph

Let $G = (V, E)$ be an undirected graph with vertex set $V = \{v_1, v_2, ..., v_n\}$. We consider that the graph is weighted, where each edge between v_i and v_j has a non-negative weight $w_{ij} \geq 0$. Typically, the weights w_{ij} represent a similarity measure or a pairwise distance between vectors $\vec{x}_i \in R^m$ and $\vec{x}_j \in R^m$.

Definition 1. *The weighted adjacency matrix of an undirected graph $G = (V, E)$ with $|V| = n$ is the symmetric matrix $W = \{w_{ij}\}$ for $i, j = 1, 2, ..., n$. If $w_{ij} = 0$ the vertices v_i and v_j are not connected through an edge.*

Definition 2. *The degree of a vertex $v_i \in V$ is defined by the sum of the elements of the i-th row of W:*

$$d_i = \sum_{j=1}^{n} w_{ij} \quad (5)$$

The degree matrix D is defined as the diagonal matrix with degrees $d_1, d_2, ..., d_n$.

Laplacian matrices and their properties have been vastly explored in spectral graph theory, a rather mature research field which focus is on studying graphs regarding to the characteristic polynomial, eigenvalues, and eigenvectors of all types of matrices associated with a graph [9]–[13].

Definition 3. *The unnormalized graph Laplacian matrix is defined as follows:*

$$L = D - W \quad (6)$$

where D consists of the degree matrix and W is the adjacency matrix.

In the following, we present some elementary, although especially important, mathematical properties of the graph Laplacian [5]. More details about the Laplacian spectrum and advanced properties may be found in the study conducted by Mohar [14].

Theorem 1. *The Laplacian matrix L satisfies the following properties:*

1) *For every column vector $\vec{f} \in R^n$ we have:*

$$\vec{f}^T L \vec{f} = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}(f_i - f_j)^2 \quad (7)$$

2) *L is symmetric and positive semi-definite and f_i indicates the i-th coordinate of f.*
3) *The smallest eigenvalue of L is zero and the corresponding eigenvector is the constant $\vec{1}$ vector.*
4) *L has n non-negative, real eigenvalues $0 = \lambda_1 \leq \lambda_2 \leq ... \leq \lambda_m$.*

To prove the first statement mentioned above, it is worth noting that by the definition of L and D we have the following:

$$\begin{aligned} &\vec{f}^T L \vec{f} = \vec{f}^T D \vec{f} - \vec{f}^T W \vec{f} \\ &= \sum_{i=1}^{n} d_i f_i^2 - \sum_{i=1}^{n}\sum_{j=1}^{n} f_i w_{ij} f_j \\ &= \frac{1}{2}\left(2\sum_{i=1}^{n} d_i f_i^2 - 2\sum_{i=1}^{n}\sum_{j=1}^{n} f_i w_{ij} f_j\right) \\ &= \frac{1}{2}\left(\sum_{i=1}^{n} d_i f_i^2 - 2\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} f_i f_j + \sum_{j=1}^{n} d_j f_j^2\right) \\ &= \frac{1}{2}\left(\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} f_i^2 - 2\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} f_i f_j + \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij} f_j^2\right) \\ &= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}(f_i - f_j)^2 \quad (8) \end{aligned}$$

The second statement is divided into two parts. The first part is about symmetry and it follows directly from the symmetry of the matrices D and W. In the second part concerned positive semi-definiteness, it is clear that $(f_i - f_j)^2 \geq 0, \forall f_i, f_j \in R$, and because $w_{ij} \geq 0$ for $i, j = 1, 2, ..., n$, then $\vec{f}^T L \vec{f} \geq 0$. To prove the third statement, noteworthy the following equation:

$$L\vec{1} = (D - W)\vec{1} = D\vec{1} - W\vec{1} = \sum_{i=1}^{n} d_i - \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}$$
$$= \sum_{i=1}^{n} d_i - \sum_{i=1}^{n} d_i = 0 \tag{9}$$

demonstrating that the constant eigenvector $\vec{1}$ contains zero eigenvalue. Lastly, the fourth statement is a direct outcome of statements 2 and 3.

C. Laplacian embedding on the line

The embedding that the LAP generates is optimal in terms of preserving local information. Thus, subsequently to the embedding process, neighboring points in the graph are close while distant points are far apart. Suppose we have a connected weighted graph $G = (V, E)$ which nodes are the data points in $X = [\vec{x}_1, \vec{x}_2, ..., \vec{x}_n]$. The problem may be formulated through the following question: How mapping the nodes of G onto a line so that connected points stay as close as possible? Finding an appropriate answer for this question is the objective of the LAP.

Let $\vec{y} = [y_1, y_2, ..., y_n]^T \in R^n$ be a map of vertices $v_1, v_2, ..., v_n$ onto the real line. An adequate objective function should heavily penalize neighboring points that are mapped far apart. A suitable choice for a given adjacency matrix W is the following function:

$$J(\vec{y}) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}(y_i - y_j)^2 = \vec{y}^T L\vec{y} \tag{10}$$

where L is the Laplacian matrix. It is worth mentioning that $J(\vec{y})$ consists of a measure of dispersiveness of the points in the real line. Thus, minimizing such a measure aims to guarantee that if $\vec{x}_i$ and $\vec{x}_j$ are close in the input space, then the coordinates y_i and y_j should also be close in the line. Therefore, we may formulate the constrained optimization problem as follows:

$$\arg\min_{\vec{y}} \; y^T L y \qquad \text{subject to} \quad y^T D y = 1 \tag{11}$$

where the constraint $y^T D y = 1$ removes an arbitrary scaling factor in the embedding [3]. Specifically, we are interested in the direction of the vector $\vec{y}$. If there is no constraint, we could then further minimize the objective function by simply dividing the components of $\vec{y}$ by a constant. We may then express the Lagrangian function as follows:

$$L(\vec{y}, \lambda) = y^T L y - \lambda(y^T D y - 1) \tag{12}$$

Differentiating with respect to $\vec{y}$ and setting the result to zero results in the following:

$$\frac{\partial}{\partial \vec{y}} L(\vec{y}, \lambda) = 2L\vec{y} - 2\lambda D\vec{y} = 0 \tag{13}$$

which leads to:

$$L\vec{y} = \lambda D\vec{y} \tag{14}$$
$$(D^{-1}L)\vec{y} = \lambda\vec{y} \tag{15}$$

elucidating that we have a generalized eigenvector problem. Since it consists of a minimization problem, it is then required to select the eigenvector of $D^{-1}L$ associated to the smallest eigenvalue. As the constant eigenvector $\vec{1}$ contains zero eigenvalue, this must then be discarded. It is reasonable mapping all points onto the same coordinate in order to minimize their dispersion level. However, such a trivial solution is of no practical use. Therefore, $\vec{y}$ should be the eigenvector associated to the smallest non-zero eigenvalue, also known as the Fiedler vector [9], [15].

D. Laplacian embedding on R^d

Consider the generalized problem of embedding the graph $G = (V, E)$ into an d-dimensional Euclidean space. Each node $v_i \in V$ needs to be mapped onto a point in R^d, requiring the estimation of d coordinates for each node. We denote the final embedding by an $n \times d$ matrix $Y = [\vec{y}_1, \vec{y}_2, ..., \vec{y}_d]$, where the i-th row, $\vec{y}^{(i)}$, provides the coordinates of v_i in the manifold. The objective function is generalized to the following:

$$J(Y) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} W_{ij}\left\|\vec{y}^{(i)} - \vec{y}^{(j)}\right\|^2 \tag{16}$$

where $\vec{y}^{(i)} = [\vec{y}_1(i), \vec{y}_2(i), ..., \vec{y}_d(i)]$ is the d-dimensional representation of v_i. It is worth noting that, considering Y as an $n \times d$ matrix in which each row represents a $\vec{y}^{(i)}$, for $i = 1, 2, ..., n$, we then reformulate the objective function as follows:

$$J(Y) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} W_{ij}(\vec{y}^{(i)} - \vec{y}^{(j)})(\vec{y}^{(i)} - \vec{y}^{(j)})^T \tag{17}$$

Expanding the expression for $J(Y)$, we may simplify to the following:

$$J(Y) = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\Big[W_{ij}\vec{y}^{(i)}\vec{y}^{(i)^T} - W_{ij}\vec{y}^{(i)}\vec{y}^{(j)^T}$$
$$-W_{ij}\vec{y}^{(j)}\vec{y}^{(i)^T} + W_{ij}\vec{y}^{(j)}\vec{y}^{(j)^T}\Big]$$
$$= \frac{1}{2}\Bigg[\sum_{i=1}^{n} d_i\vec{y}^{(i)}\vec{y}^{(i)^T} - 2\sum_{i=1}^{n}\sum_{j=1}^{n} W_{ij}\vec{y}^{(i)}\vec{y}^{(j)^T}$$
$$+\sum_{j=1}^{n} d_j\vec{y}^{(j)}\vec{y}^{(j)^T}\Bigg]$$
$$= \frac{1}{2}\Bigg[2\sum_{i=1}^{n} d_i\vec{y}^{(i)}\vec{y}^{(i)^T} - 2\sum_{i=1}^{n}\sum_{j=1}^{n} W_{ij}\vec{y}^{(i)}\vec{y}^{(j)^T}\Bigg]$$
$$= \sum_{i=1}^{n} d_i\vec{y}^{(i)}\vec{y}^{(i)^T} - \sum_{i=1}^{n}\sum_{j=1}^{n} W_{ij}\vec{y}^{(i)}\vec{y}^{(j)^T} \tag{18}$$

Considering the matrix $Y_{n\times d}$ of the coordinates for the n points, the diagonal matrix $D_{n\times n}$ of degrees d_i, and the adjacency matrix $W_{n\times n}$, we may rewrite the equation adopting a matrix-vector notation as follows:

$$J(Y) = Tr(DYY^T) - Tr(WYY^T) \tag{19}$$

As the trace is an operator that is invariant under cyclic permutations, then we have the following:

$$\begin{aligned} J(Y) &= Tr(Y^T DY) - Tr(Y^T WY) = Tr(Y^T(DY - WY)) \\ &= Tr(Y^T(D - W)Y) = Tr(Y^T LY) \end{aligned} \tag{20}$$

Thus, we have the following constrained optimization problem:

$$\underset{Y}{\arg\min}\ Tr(Y^T LY) \qquad \text{subject to} \quad Y^T DY = I \tag{21}$$

whose Lagrangian function is given by:

$$L(Y, \lambda) = Tr(Y^T LY) - \lambda(Y^T DY - I) \tag{22}$$

Taking the derivative and setting the result to zero leads to the following:

$$\frac{\partial}{\partial Y} L(Y, \lambda) = 2LY - 2\lambda DY = 0 \tag{23}$$

resulting in the following eigenvector problem:

$$LY = \lambda DY \tag{24}$$

This result demonstrates that we should compose the columns of the matrix Y with d eigenvectors associated to the d smallest non-zero eigenvalues of the normalized Laplacian $D^{-1}L$. Some variants of the algorithm include the eigendecomposition of different versions of the graph Laplacian. The most common choices refer to another form of normalized Laplacian, given by $L_{sym} = D^{-1/2}LD^{-1/2}$, and the pure unnormalized Laplace $L = D - W$. While applying LAP to some real-world data, several limitations have been found, such as uneven data sampling, out-of-sample problem, small sample size, discriminant feature extraction and selection, among others. To overcome such problems, extensions of the LAP have been proposed [2]. Algorithm 1 summarizes the LAP method.

Algorithm 1 Laplacian eigenmaps (LAP)

```
function LaplaceEigen($X, K, d$)
    From input data $X_{m\times n}$ build an KNN graph.
    Select the weights to define the adjacency matrix $W$.
        $$W_{ij} = exp\left\{-\frac{\|\vec{x}_i - \vec{x}_j\|^2}{t}\right\} \quad \text{if} \quad v_j \in N(v_i) \tag{25}$$
    Compute the diagonal matrix $D$, with degrees $d_i$ for $i = 1, 2, ..., n$.
        $$d_i = \sum_{j=1}^{n} W_{ij} \tag{26}$$
    Compute the Laplacian matrix $L = D - W$.
    Select the bottom $d$ eigenvectors with non-zero eigenvalues of $D^{-1}L$ and define matrix $Y$, where each column is an eigenvector.
    return $Y$
end function
```

E. Graph cuts and Laplacian eigenmaps (LAP)

There exists a deep relation between the problem of finding the minimum cut in a weighted graph and spectral clustering, which is the application of the k-means algorithm subsequently to the LAP. In the following, we briefly discuss about such an intrinsic connection based on the seminal paper of Luxburg [5]. Firstly, recalling that the normalized cut ($RCut$) is formulated as follows:

$$RCut(A_1, A_2, ..., A_k) = \frac{1}{2}\sum_{i=1}^{k}\frac{w(A_i, \overline{A_i})}{|A_i|} \tag{27}$$

where $|A_i|$ denotes the number of elements in the partition A_i, the complement of A_i is represented by $\overline{A_i}$, and $w(A_i, \overline{A_i})$ is formulated as follows:

$$w(A_i, \overline{A_i}) = \sum_{i\in A_i; j\in \overline{A_i}} w_{ij} \tag{28}$$

refers to the summation of weights of the edges with one vertex in A_i and another vertex in $\overline{A_i}$. Typically, the problem of finding the cut that minimizes $RCut$ is NP-hard. For a binary problem $k = 2$, we have to minimize $RCut(A, \overline{A})$, where:

$$RCut(A, \overline{A}) = \frac{1}{2}\left[\frac{w(A, \overline{A})}{A} + \frac{w(\overline{A}, A)}{\overline{A}}\right] \tag{29}$$

It is possible to associate the value of $RCut$ with the Laplacian matrix of the graph. Let $\vec{f} \in R^n$ be defined as follows:

$$f_i = \begin{cases} \sqrt{\dfrac{|\overline{A}|}{|A|}} & v_i \in A \\ -\sqrt{\dfrac{|A|}{|\overline{A}|}} & v_i \in \overline{A} \end{cases} \tag{30}$$

We know that:

$$\vec{f}^T L \vec{f} = \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}(f_i - f_j)^2 \qquad (31)$$

The previous equation may also be computed as follows [5]:

$$\vec{f}^T L \vec{f} = |V| RCut(A, \overline{A}) \qquad (32)$$

Therefore, the minimization of $RCut$ is mathematically equivalent to the following:

$$\arg\min \ \vec{f}^T L \vec{f} \quad \text{s.t.} \quad \vec{f}^T \vec{1} = 0 \quad \text{and} \quad \|f\| = \sqrt{n} \qquad (33)$$

It is worth noting that the problem is NP-hard due to the fact that as the dimensionality of the solution vector $\vec{f}$ is n, and each component f_i may assume one of two possible values, we then have a total of 2^n candidate solutions. Exhaustive search becomes unfeasible for large values of n. By relaxing such a problem by allowing that $f_i \in R$, the solution to the relaxed problem is known to be $\vec{f} = \vec{v}_1$, where $\vec{v}_1$ is the eigenvector associated to the smallest non-zero eigenvalue of the Laplacian matrix - recalling that the smallest eigenvalue is zero. Subsequently, we quantize the components of the vector $\vec{f}$, considering $f_i = 0$ if $f_i < 0$ and $f_i = 1$ if $f_i \geq 0$, which may be performed by a clustering algorithm such as k-means. In fact, the regular Laplacian leads to an approximation to the minimization of the $RCut$, while the normalized Laplacian induces to an approximation for $NCut$, defined as follows:

$$NCut(A_1, A_2, ..., A_k) = \frac{1}{2}\sum_{i=1}^{k} \frac{w(A_i, \overline{A_i})}{vol(A_i)} \qquad (34)$$

where $vol(A_i)$ is the summation of the degrees of the nodes in A_i. Figure 1 depicts how relaxing the problem results in different sub-optimal solutions depending on the type of the Laplacian matrix (i.e., normalized versus unnormalized).

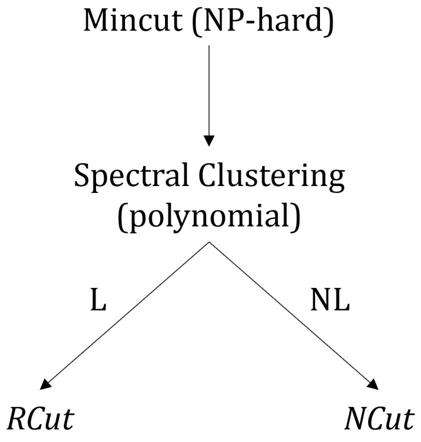

Fig. 1. Normalized versus unnormalized Laplacian leading to different approximations to the minimum cut problem.

III. Entropic Laplacian eigenmaps (ELAP)

The main motivation of the proposed ELAP method is to replace the pointwise Euclidean distance between $\vec{x}_i$ and $\vec{x}_j$ in the Gaussian kernel used to compute the edge weights by the relative entropy between patches P_i and P_j. We use an information-theoretic distance function, namely the KL-divergence. Our inspirations are the parametric principal component analysis (PCA) [16] and ISOMAP-KL [17], which consist of two recent dimensionality reduction-based unsupervised metric learning algorithms that are variations of the PCA and ISOMAP, respectively, with information-theoretic divergences.

Let $X = \{\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n\}$, with $\vec{x}_i \in R^m$, be our data matrix. The first step in the proposed method consists of building the KNN graph from X. At this early stage, we employ the extrinsic Euclidean distance to compute the nearest neighbors of each sample $\vec{x}_i$. Denoting by η_i the neighborhood system of $\vec{x}_i$, a patch P_i is defined as the set $\{\vec{x}_i \cup \eta_i\}$. It is worth noting that the number of elements of P_i is $K + 1$, for $i = 1, 2, ..., n$. In other words, a patch P_i is given by an $m \times (k + 1)$ matrix as follows:

$$P_i = [\vec{x}_i, \vec{x}_{i1}, \vec{x}_{i2}, ..., \vec{x}_{ik}] = \begin{bmatrix} x_i(1) & x_{i1}(1) & \ldots & x_{ik}(1) \\ x_i(2) & x_{i1}(2) & \ldots & x_{ik}(2) \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ldots & \vdots \\ x_i(m) & x_{i1}(m) & \ldots & x_{ik}(m) \end{bmatrix} \qquad (35)$$

The idea behind the proposed method is to consider each column of the matrix P_i as a sample of a multivariate Gaussian random variable of size $k+1$. Then, we compute the maximum likelihood estimators of the model parameters $\vec{\mu}_i$ (mean) and Σ_i (covariance matrix) as follows:

$$\vec{\mu}_i = \frac{1}{k+1}\sum_{j=1}^{k+1} \vec{x}_{ij} \qquad (36)$$

$$\Sigma_i = \frac{1}{k}\sum_{j=1}^{k+1} (\vec{x}_{ij} - \vec{\mu}_i)(\vec{x}_{ij} - \vec{\mu}_i)^T \qquad (37)$$

Let $p(x)$ and $q(x)$ be multivariate Gaussian densities, $N(\vec{\mu}_1, \Sigma_1)$ and $N(\vec{\mu}_2, \Sigma_2)$. Then, the relative entropy $D_{KL}(p, q)$ becomes:

$$D_{KL}(p, q) = \frac{1}{2}\left[log\left(\frac{|\Sigma_2|}{|\Sigma_1|}\right) + Tr\left[\Sigma_2^{-1}\Sigma_1\right] \right. \qquad (38)$$
$$\left. + (\vec{\mu}_2 - \vec{\mu}_1)^T \Sigma_2^{-1} (\vec{\mu}_2 - \vec{\mu}_1) - m \right]$$

Similarly, the relative entropy $D_{KL}(q, p)$ is formulated as follows:

$$D_{KL}(q, p) = \frac{1}{2}\left[log\left(\frac{|\Sigma_1|}{|\Sigma_2|}\right) + Tr\left[\Sigma_1^{-1}\Sigma_2\right] \right. \qquad (39)$$
$$\left. + (\vec{\mu}_1 - \vec{\mu}_2)^T \Sigma_1^{-1} (\vec{\mu}_1 - \vec{\mu}_2) - m \right]$$

As the relative entropy is not symmetric, it is then possible to compute its symmetrized counterpart as follows:

$$D_{KL}^{sym}(p,q) = \frac{1}{2}\left[D_{KL}(p,q) + D_{KL}(q,p)\right] \tag{40}$$

which contains the following closed-form expression:

$$\begin{aligned} D_{KL}^{sym}(p,q) = \frac{1}{2}\Bigg[& \frac{1}{2}Tr\left(\Sigma_1^{-1}\Sigma_2 + \Sigma_2^{-1}\Sigma_1\right) \\ & + \frac{1}{2}(\vec{\mu}_1 - \vec{\mu}_2)^T\Sigma_1^{-1}(\vec{\mu}_1 - \vec{\mu}_2) \\ & + \frac{1}{2}(\vec{\mu}_2 - \vec{\mu}_1)^T\Sigma_2^{-1}(\vec{\mu}_2 - \vec{\mu}_1) - m\Bigg] \end{aligned} \tag{41}$$

Figure 2 illustrates the mapping of local patches onto the KNN graph to a parametric representation. In this parametric feature space, the relative entropy is a more meaningful measure of similarity compared to the traditional Euclidean distance.

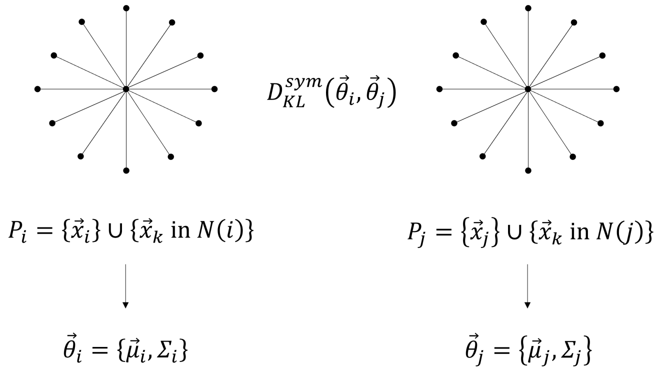
$D_{KL}^{sym}(\vec{\theta}_i, \vec{\theta}_j)$
$P_i = \{\vec{x}_i\} \cup \{\vec{x}_k \text{ in } N(i)\}$
$P_j = \{\vec{x}_j\} \cup \{\vec{x}_k \text{ in } N(j)\}$
$\vec{\theta}_i = \{\vec{\mu}_i, \Sigma_i\}$
$\vec{\theta}_j = \{\vec{\mu}_j, \Sigma_j\}$

Fig. 2. Mapping from a patch P_i onto the graph to a parametric feature vector.

Algorithm 2 summarizes the proposed method.

Algorithm 2 Entropic Laplacian eigenmaps (ELAP)

```
function EntLaplaceEigen($X, K, d$)
    From input data $X_{m \times n}$ build an KNN graph.
    Select the weights to define the adjacency matrix $W$.
        $W_{ij} = exp\left\{-\frac{D_{KL}^{sym}(P_i, P_j)^2}{t}\right\}$  if  $v_j \in N(v_i)$  (42)
    Compute the diagonal matrix $D$ with degrees $d_i$ for $i = 1, 2, ..., n$.
        $d_i = \sum_{j=1}^{n} W_{ij}$  (43)
    Compute the Laplacian matrix $L = D - W$.
    Select the bottom $d$ eigenvectors with non-zero eigenvalues of $L$ and define the matrix $Y$, where each column consists of an eigenvector.
    return $Y$
end function
```

There are two main differences between the LAP and its entropic counterpart. First, the distance function in the Gaussian kernel. Second, in the proposed ELAP method we do not normalize the Laplacian matrix L using the inverse of the degree matrix D.

IV. Experiments and results

In order to test and evaluate the proposed method, we performed two empirical experiments. In the first experiment, a quantitative comparison of clusters obtained after dimensionality reduction to 2-D spaces with the silhouette coefficient or SC (i.e., measure of the fit to a low-dimensional representation) [18]. In the second experiment, after dimensionality reduction to 2D spaces, we compare the average classification accuracies for four supervised classifiers, namely the KNN, decision trees (DT), Bayesian classifier under Gaussian hypothesis, and random forest (RF). We then compare the proposed ELAP method against the PCA, kernel PCA, ISOMAP, LLE, Hessian eigenmaps (HLAP), and regular LAP.

All datasets used in the experiments, along with detailed information regarding the number of instances, features and classes for each one of them, are publicly available at openML.org. The results of the first empirical experiment are reported in Table I. Bold values denote the best method for that particular dataset. Based on the averages and medians, we may realize that the proposed ELAP method shows a superior performance in comparison with the regular LAP. Moreover, the proposed ELAP method is the best method in terms of generating well formed clusters.

To check whether the results provided by the proposed method are statistically superior to the competing methods, we then performed a Friedman test (i.e., non-parametric test for paired data in case of more than two groups) [19]. For a significant level $\alpha = 0.01$, we conclude that there is strong evidence against the null hypothesis that all groups are identical ($p = 5.71 \times 10^{-13}$). In order to analyze which groups are significantly different, we perform the Nemenyi post-hoc test [20]. According to this test, there is strong evidence that the proposed ELAP method produces significantly superior SCs compared to the PCA ($p = 0.00166$), kernel PCA ($p < 10^{-3}$), ISOMAP ($p < 10^{-3}$), LLE ($p < 10^{-3}$), HLAP ($p < 10^{-3}$), and standard LAP ($p < 10^{-3}$).

In the second empirical experiment, subsequently to performing the dimensionality reduction-based metric learning, for each dataset we use 50% of the samples to train four distinct supervised classifiers, as follows: KNN ($K = 7$), DT, quadratic Bayesian classifier (QDA), and RF. Each of these methods is used to classify the 50% remaining samples from the test data and the average accuracy among them is selected to evaluate the behavior of the dimensionality reduction in supervised classification tasks. The results are reported in Table II.

The non-parametric Friedman test to verify if the classification accuracies obtained by the proposed ELAP method are statistically superior compared to existing methods shows that, for a significance level $\alpha = 0.01$, there is strong evidence

TABLE I

Silhouette coefficients for clusters produced by PCA, kernel PCA, ISOMAP, LLE, Hessian eigenmaps (HLAP), Laplacian eigenmaps (LAP), and entropic Laplacian eigenmaps (ELAP) for 25 datasets (2D case).

Dataset	**PCA**	**KPCA**	**ISO**	**LLE**	**HLAP**	**LAP**	**ELAP**
mammog	0.349	0.032	0.307	0.070	-0.747	-0.251	**0.703**
marketing	0.082	-0.006	-0.001	0.078	0.125	-0.273	**0.293**
Biodeg	0.094	0.126	0.031	-0.061	-0.321	-0.055	**0.336**
Tictac	-0.023	-0.019	-0.020	0.007	-0.043	-0.011	**0.354**
pc3	0.201	0.074	-0.017	-0.760	-0.768	-0.341	**0.537**
Blood	0.086	0.026	0.082	0.000	-0.327	0.004	**0.322**
kc1	0.371	0.210	0.187	0.202	-0.401	-0.480	**0.519**
parity5	-0.062	-0.047	-0.048	-0.051	-0.043	-0.036	**0.540**
thoracic	0.006	-0.002	-0.006	0.082	-0.018	-0.021	**0.319**
attendence	-0.034	0.002	-0.077	0.000	-0.053	-0.121	**0.233**
fl2000	0.180	0.043	0.119	0.073	0.253	0.025	**0.320**
creditscore	0.111	0.081	0.131	0.071	0.119	0.049	**0.249**
haberman	0.060	-0.024	0.062	-0.004	0.040	-0.032	**0.373**
newton	0.087	0.113	0.082	0.077	0.092	0.090	**0.142**
wildcat	0.151	0.081	0.125	0.149	-0.020	0.028	**0.348**
datatrieve	0.239	0.010	0.096	0.066	0.120	0.080	**0.248**
Grub	0.042	0.050	0.066	0.094	0.190	0.132	**0.285**
fem-blad	0.122	0.008	0.170	0.143	0.185	0.030	**0.336**
mw1	0.349	0.122	0.286	0.175	-0.841	0.180	**0.565**
ar1	0.265	0.028	0.216	-0.004	-0.835	-0.002	**0.350**
segment	-0.161	-0.028	-0.164	-0.227	-0.623	-0.240	**0.615**
kc3	0.386	0.103	0.233	0.062	-0.803	-0.129	**0.495**
boxing1	0.019	0.055	-0.030	0.016	-0.013	0.028	**0.188**
collins	-0.049	-0.056	-0.053	0.056	-0.674	-0.012	**0.3**
blogger	0.036	-0.011	0.052	0.029	-0.002	0.003	**0.333**
Mean	0.116	0.039	0.073	0.014	-0.216	-0.054	**0.372**
Median	0.087	0.028	0.066	0.062	-0.043	-0.011	**0.336**
Minimum	-0.161	-0.056	-0.164	-0.760	-0.841	-0.480	**0.142**
Maximum	0.386	0.210	0.307	0.202	0.253	0.180	**0.703**

TABLE II

Average classification accuracies generated subsequently to the PCA, kernel PCA, ISOMAP, LLE, Hessian eigenmaps (HLAP), Laplacian eigenmaps (LAP), and entropic Laplacian eigenmaps (ELAP) for 25 datasets (2D case).

Dataset	**PCA**	**KPCA**	**ISO**	**LLE**	**HLAP**	**LAP**	**ELAP**
Monks	0.593	0.578	0.584	0.635	0.599	0.610	**0.787**
Tictac	0.619	0.644	0.607	0.591	0.651	0.743	**0.760**
KNugget	0.739	0.731	0.760	0.755	0.567	0.752	**0.786**
cloud	0.657	0.601	0.615	0.606	0.587	0.601	**0.676**
kc1	0.834	0.827	0.822	0.818	0.698	0.697	**0.844**
parity5	0.453	0.390	0.421	0.359	0.343	0.406	**1.000**
attendence	0.835	0.829	0.825	0.833	0.847	0.825	**0.860**
AIDS	0.380	0.300	0.360	0.290	0.340	0.280	**0.810**
fl2000	0.654	0.610	0.610	0.654	0.603	0.566	**0.662**
creditscore	0.770	0.705	0.794	0.750	0.685	0.764	**0.800**
Hayes	0.594	0.670	0.666	0.632	0.723	0.625	**0.807**
crabs	0.605	0.592	0.607	0.650	0.635	0.607	**0.680**
haberman	0.733	0.668	0.732	0.704	0.722	0.683	**0.760**
newton	0.682	0.675	0.642	0.625	0.692	0.675	**0.714**
wildcat	0.789	0.768	0.756	0.740	0.719	0.746	**0.814**
veteran	0.655	0.659	0.619	0.605	0.630	0.594	**0.696**
datatrieve	0.907	0.915	0.934	0.919	0.930	0.892	**0.942**
ar1	0.959	0.938	0.942	0.950	0.741	0.938	**0.963**
segment	0.824	0.847	0.860	0.795	0.756	0.854	**0.887**
kc3	0.885	0.882	0.895	0.891	0.724	0.814	**0.895**
boxing1	0.691	0.658	0.670	0.629	0.620	0.587	**0.717**
collins	0.79	0.825	0.796	0.798	0.668	0.813	**0.847**
vineyard	0.769	0.721	0.769	0.74	0.74	0.759	**0.798**
kidney	0.605	0.671	0.684	0.644	0.69	0.611	**0.697**
mux6	0.609	0.683	0.656	0.726	0.605	0.511	**0.734**
Mean	0.705	0.695	0.705	0.694	0.661	0.678	**0.797**
Median	0.691	0.675	0.684	0.704	0.685	0.683	**0.798**
Minimum	0.380	0. 300	0.360	0.290	0.340	0.280	**0.662**
Maximum	0.959	0.938	0.942	0.950	0.930	0.938	**1.000**

against the null hypothesis that all groups are identical ($p = 1.91 \times 10^{-12}$). In addition, according to the Nemenyi post-hoc test to verify which groups are significantly different, there is strong evidence that the proposed ELAP method produces significantly higher classification accuracies in comparison with the PCA ($p < 10^{-3}$), kernel PCA ($p < 10^{-3}$), ISOMAP ($p < 10^{-3}$), LLE ($p < 10^{-3}$), HLAP ($p < 10^{-3}$), and standard LAP ($p < 10^{-3}$).

Despite of such promising results, the proposed ELAP method has some limitations. A negative aspect of manifold learning algorithms in general - including the proposed method - is the out-of-sample problem. Most unsupervised metric learning algorithms are not capable of dealing with new samples that are not part of the training data in a straightforward manner. A natural choice would be adding such new samples to the data and then perform a further full training round, which may be time consuming. Another caveat of the proposed method concerns the definition of the parameter K (number of neighbors) that controls the patch size. Our experiments reveal that the SC and classification accuracies are rather sensitive to changes in such a parameter. In the present study, we employ a strategy in which KKN graphs are built for each dataset considering all values of K in the interval $[2, 40]$. We select the best model as the one that maximizes the classification accuracy among all values of K. It is worth mentioning that we are using the class labels to perform model selection, however, the dimensionality reduction-based metric learning is fully unsupervised. A visual comparison of the clusters obtained through the standard LAP and the proposed ELAP method for the parity5 dataset is depicted in Figure 3. It is worth noting that the discrimination between classes is comparably more evident in the proposed ELAP method as a consequence of less overlaps between clusters. The Python source code for ELAP can be found at https://github.com/alexandrelevada/Entropic-Laplacian-Eigenmaps.

V. Conclusion

Unsupervised metric and manifold learning are intrinsically related. Many algorithms have been devised to learn underlying geometric structures from data, being the LAP among the most relevant ones. Many extensions have been proposed to avoid some limitations of the original method, such as the HLAP. However, one persistent problem is that most methodological variations adopt the Euclidean metric to measure similarity between samples in the KNN graph.

In the present study, we propose the ELAP algorithm as a parametric method that incorporates the relative entropy between local Gaussian distributions into the Laplacian matrix. The rationale is that replacing the pointwsie Euclidean distance by a patch-based information-theoretic distance would result in a more robust method against noise and outliers. Our claim is that the proposed ELAP algorithm is a promising alternative to existing manifold learning algorithms. Such a claim is based upon computational experiments, which report two main points. First, the quality of the clusters generated by the proposed method may be superior to those produced by

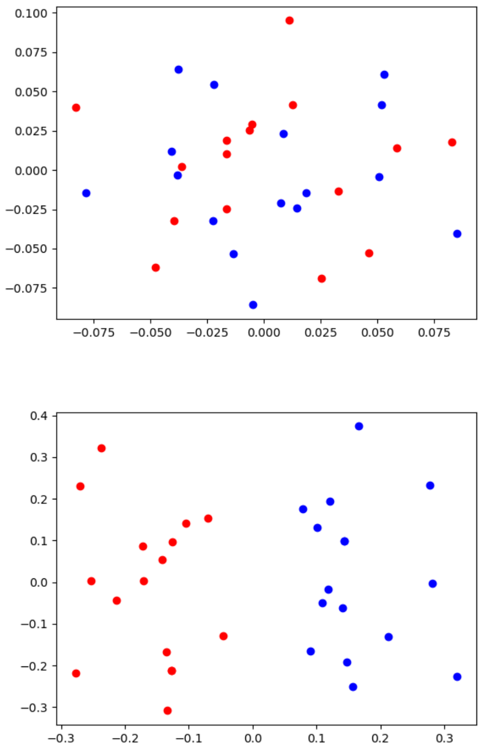

Fig. 3. Comparison between clusters generated after LAP and ELAP (number of neighbors K = 2) for the parity5 dataset.

competing manifold learning algorithms. Second, non-linear features of the proposed method may be more discriminative in supervised classification compared to features obtained through competing manifold learning algorithms.

Future research might include further information-theoretic distances, such as the Bhattacharyya, Hellinger and Cauchy-Schwarz divergences, as well as geodesic distances based in the Fisher information matrix. Another possibility is the non-parametric estimation of local densities using kernel density estimation techniques (KDE). In this case, non-parametric versions of the information-theoretic distances might be employed to compute a distance function between the patches of the KNN graph. The ϵ-neighborhood rule might also be used for building the adjacency relations that define the discrete approximation for the manifold, leading to non-regular graphs. Furthermore, a supervised ELAP algorithm might be devised by combining both Euclidean and information-theoretic divergences. In such a proposition, the edges of the KNN graph in which the endpoints belong to the same class are weighted with the minimum of the two distances, while the edges in which the endpoints belong to different classes are weighted with the sum of the distances to enforce smaller intra-class compared to inter-class variations.

Acknowledgments

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

References

[1] W. N. A. Jr. and T. D. Morley, "Eigenvalues of the laplacian of a graph," *Linear and Multilinear Algebra*, vol. 18, no. 2, pp. 141–145, 1985.

[2] L. Bo, L. Yan-Rui, and Z. Xiao-Long, "A survey on laplacian eigenmaps based manifold learning methods," *Neurocomputing*, vol. 335, pp. 336–351, 2018.

[3] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Computation*, vol. 15, no. 6, pp. 1373–1396, Jun. 2003.

[4] ——, "Convergence of laplacian eigenmaps," in *Advances in Neural Information Processing Systems 19*, B. Schölkopf, J. C. Platt, and T. Hoffman, Eds. MIT Press, 2007, pp. 129–136.

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

[6] 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.

[7] S. Roweis and L. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 2323–2326, 2000.

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

[9] F. R. K. Chung, Ed., *Spectral Graph Theory*. American Mathematical Society, 1997.

[10] A. E. Brouwer and W. H. Haemers, Eds., *Spectra of Graphs*. Springer, 2011.

[11] D. A. Spielman, "Spectral graph theory and its applications," in *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, Oct 2007, pp. 29–38.

[12] B. Nica, *A Brief Introduction to Spectral Graph Theory*. American Mathematical Society, 2018.

[13] P. van Mieghem, *Graph Spectra for Complex Networks*. Cambridge University Press, 2010.

[14] B. Mohar, "The laplacian spectrum of graphs," in *Graph Theory, Combinatorics, and Applications*. Wiley, 1991, pp. 871–898.

[15] M. Fiedler, "Laplacian of graphs and algebraic connectivity," *Banach Center Publications*, vol. 25, no. 1, pp. 57–70, 1989.

[16] A. L. M. Levada, "Parametric PCA for unsupervised metric learning," *Pattern Recognition Letters*, vol. 135, pp. 425–430, 2020.

[17] A. C. Neto and A. Levada, "Isomap-kl: a parametric approach for unsupervised metric learning," in *Anais do XXXIII Conference on Graphics, Patterns and Images*. Porto Alegre, RS, Brasil: SBC, 2020, pp. 57–64. [Online]. Available: https://sol.sbc.org.br/index.php/sibgrapi/article/view/14108

[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.

[19] M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *Journal of the American Statistical Association*, vol. 32, no. 200, pp. 675–701, 1937.

[20] M. Hollander, D. A. Wolfe, and E. Chicken, *Nonparametric Statistical Methods*, 3rd ed. Wiley, 2015.