Journal of Universal Computer Science, vol. 30, no. 5 (2024), 603-616
submitted: 27/5/2023, accepted: 29/12/2023, appeared: 28/5/2024 CC BY-ND 4.0

Probabilistic Nearest Neighbors Based Locality Preserving Projections for Unsupervised Metric Learning

Alaor Cervati Neto
(Computing Department, Federal University of São Carlos, São Carlos, SP, Brazil
https://orcid.org/0000-0001-6212-6205, alaor_c_neto@yahoo.com.br)

Alexandre L. M. Levada
(Computing Department, Federal University of São Carlos, São Carlos, SP, Brazil
https://orcid.org/0000-0001-8253-2729, alexandre.levada@ufscar.br)

Abstract: Dimensionality reduction based unsupervised metric learning consists in finding meaningful compact data representations previously to clustering and classification problems. One of the major aspects of these algorithms is the approximation of the underlying manifold by a weighted graph. A limitation with most manifold learning algorithms is that edge weights in the proximity graph rely heavily on the Euclidean distance, which is known to be quite sensitive to the presence of outliers. In this paper, we propose to improve the Locality Preserving Projections (LPP) algorithm by incorporating a recently proposed graph inference method called Probabilistic Nearest Neighbors (PNN), an extension of the Clustering with Adaptive Neighbors (CAN) approach, used with success in graph-based semi-supervised learning. The proposed PNN-LPP algorithm is able to achieve better classification results than regular LPP, showing competitive performance against state-of-the-art approaches for dimensionality reduction, such as the UMAP algorithm, especially in datasets with a limited number of samples.

Keywords: Manifold learning, Dimensionality reduction, Locality Preserving Projections, Probabilistic Nearest Neighbors, Unsupervised Metric Learning
Categories: I.5.0, I.5.1, I.5.4, I.5.5
DOI: 10.3897/jucs.107081

1 Introduction

Pattern analysis and classification are fundamental tasks in machine learning and data science [Sarker 2000]. A key concept in these computational tools is the notion of similarity measure, induced by a distance function [Kayabasi et al. 2021]. Usually, most data clustering and classification methods assume that the observed data lies in a linear space, which makes the Euclidean distance a suitable metric. However, several theoretical and empirical studies have shown that this is a weak assumption, that is, in fact, most datasets define a manifold, a curved geometric space with a intrinsic dimension much smaller than the dimension of the ambient space [Lin et al. 2015]. Hence, being able to learn a better and adaptive distance function while reducing data dimensionality is crucial before data analysis and classification [Xiao et al. 2010]. Despite finding a more compact low dimensional representation for data, we want to perform unsupervised metric learning [Dutta et al. 2020, Levada 2020, Levada 2021].

One of the most important manifold learning algorithms is Laplacian Eigenmaps. The basic idea behind this method is that if we approximate a manifold by a connected and undirected basic graph, then it is possible to find a map from the vertices of the graph

to an Euclidean subspace R^d, such that locality is preserved, or in other words, the map is smooth in the sense that neighboring points in the graph will remain close together after the mapping is performed. Such map is given by the eigenvectors of the graph Laplacian matrix [Bo et al. 2018]. The representation map generated by the algorithm may be viewed as a discrete approximation to a continuous map that naturally arises from the geometry of the manifold: the Laplace-Beltrami operator [Belkin and Niyogi 2003]. It has been shown the convergence of the eigenvectors of the graph Laplacian associated to a point cloud dataset to eigenfunctions of the Laplace-Beltrami operator when the data is sampled from a uniform probability distribution on an embedded manifold [Belkin and Niyogi 2007]. In machine learning, the Laplace Eigenmaps method is closely related to spectral clustering, an unsupervised learning approach for data clustering [Luxburg 2007].

However, manifold learning algorithms have a severe limitation, which is the out-of-sample problem, which means that although they perform quite well in the training set, it is not clear how to evaluate novel samples that do not belong to the training set [Taskin and Crawford 2019]. Often, it is necessary to include the new samples in the set and then to apply the method repeated times in these larger training sets, a procedure that is both time consuming and not scalable. To overcome this limitation, Locality Preserving Projections (LPP) was proposed to work as a linearized version of Laplacian Eigenmaps [He and Niyogi 2004]. The idea is to enforce a linear relationship between the high dimensional input data and the low dimensional output data, in a way that we can build a projection matrix. It has been verified that LPP also has some problems: first, as many manifold learning algorithms, the graph construction step of LPP is quite sensitive to noise and outliers [Hu et al. 2018]. LPP also suffers from the small sample size problem, that is, its performance is degraded when the number of samples is not sufficiently large [Ran et al. 2022]. Moreover, LPP's performance can become unstable when there are variations in the number of neighbors in the graph [Ran et al. 2022]. Recently, there has been great interested in improving the performance of LPP for supervised classification problems [Chen et al. 2020].

Graph-based semi-supervised learning (GSSL) is a relatively new research topic that has been drawing attention of the machine learning community, as its solid mathematical background often leads to closed-form optimal solutions, which means computational efficiency [Chong et al. 2020]. Particularly, in the last years, transductive GSSL has become a powerful framework for data classification through label propagation strategies [Ma et al. 2019]. Since GSSL consists of both graph construction and inference, our idea is to incorporate graph-based semi-supervised concepts in the discrete manifold approximation in dimensionality reduction methods to perform fully unsupervised metric learning.

Clustering with Adaptive Neighbors (CAN) is a probabilistic GSSL method that learns the data similarity matrix by solving a constrained least-square problem [Nie et al. 2014]. As CAN is more suitable for clustering tasks, later on, Probabilistic Nearest Neighbors (PNN) was proposed to be more optimized for supervised classification problems [Ma et al. 2020]. The main advantage of PNN is the incorporation of a min-max normalization process in the solution of the optimization problem, which makes method optimized for data discrimination [Ma et al. 2020].

In this paper, we propose PNN-LPP, a dimensionality reduction based unsupervised metric learning algorithm that replaces the extrinsic Euclidean distance used in the Gaussian kernel of the edge weighting function by a probabilistic distance. The main contributions of the proposed method can be summarized as: 1) PNN-LPP can be less sensitive to the presence of noise and outliers than regular LPP; 2) Overall, PNN-LPP has

superior performance than regular LPP in reduced sample size problems; 3) PNN-LPP is able to extract more discriminant features than regular LPP and other state-of-the-art dimensionality reduction methods, such as UMAP, showing that it is a valid alternative for unsupervised metric learning.

The remaining of the paper is organized as follows: Section 2 describes LPP in details, showing how it avoids the out-of-sample problem, introduce the graph-based learning algorithms CAN and PNN, used to approximate the manifold by a discrete structure avoiding the Euclidean distance and presents the proposed PNN-LPP method using an algorithmic approach. Section 3 shows the computational experiments and the obtained results. Finally, Section 4 presents our conclusions and final remarks.

2 PNN-LPP for Unsupervised Metric Learning

In this Section, we describe in details all the mathematical and computational tools employed by the proposed Probabilistic Nearest Neighbor based Locality Preserving Projections method for unsupervised metric learning.

2.1 Locality Preserving Projections

The out-of-sample problem, in which the map is defined only on the training data points and it is unclear how to evaluate the map for fresh test points, is one issue with Laplacian eigenmaps. The basic goal of the Locality Preserving Projection (LPP) methodology is to create a method that can be used to locate any new test data point in the reduced representation space [He and Niyogi 2004]. LPP is a linear approximation of the non-linear Laplacian Eigenmaps approach. We seek a smooth map that preserves locality, similar to the Laplacian Eigenmaps approach. That is, proximity in the graph must imply proximity in the line. In previous sections, we demonstrated that if the following criterion is minimized, the map $\vec{y} = [y1, y2, ..., yn]$ is optimal in that sense:

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

where L is the Laplacian matrix of the KNN graph induced by the $m \times n$ data matrix $X = [\vec{x}_1, \vec{x}_2, ..., \vec{x}_n]$. Note the major role of the edge weights w_{ij} in the optimization problem. The intuition behind this equations is that proximity in the graph must imply in proximity in the resulting embedding. In this paper, the objective is to find a better measure than the regular Euclidean distance, by employing the Probabilistic Nearest Neighbors method, knowing that the Euclidean distance is sensitive to outliers in data.

In LPP, it is assumed that the relationship between $\vec{x}_i \in R^m$ and $y_i \in R$ is linear, that is, $y_i = \vec{a}^T \vec{x}_i$, where $\vec{a} \in R^m$ is a column vector. Hence, the objective function is:

$$\begin{aligned}\vec{y}^T L\vec{y} &= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}(\vec{a}^T\vec{x}_i - \vec{a}^T\vec{x}_j)^2 \\ &= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}\left[\vec{a}^T\vec{x}_i\vec{x}_i^T\vec{a} - 2\vec{a}^T\vec{x}_i\vec{x}_j^T\vec{a} + \vec{a}^T\vec{x}_j\vec{x}_j^T\vec{a}\right] \\ &= \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} 2w_{ij}\vec{a}^T\vec{x}_i\vec{x}_i^T\vec{a} - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} 2w_{ij}\vec{a}^T\vec{x}_i\vec{x}_j^T\vec{a} \\ &= \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}\vec{a}^T\vec{x}_i\vec{x}_i^T\vec{a} - \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}\vec{a}^T\vec{x}_i\vec{x}_j^T\vec{a}\end{aligned} \tag{2}$$

Since $d_i = \sum_{j=1}^{n} w_{ij}$, we have:

$$\vec{y}^T L\vec{y} = \sum_{i=1}^{n}\vec{a}^T\vec{x}_i d_i\vec{x}_i^T\vec{a} - \sum_{i=1}^{n}\sum_{j=1}^{n}\vec{a}^T\vec{x}_i w_{ij}\vec{x}_j^T\vec{a} \tag{3}$$

Note that The previous equation is equivalent to:

$$\vec{y}^T L\vec{y} = \vec{a}^T XDX^T\vec{a} - \vec{a}^T XWX^T\vec{a} \tag{4}$$

where X is the $m \times n$ data matrix, D is the $n \times n$ diagonal matrix of the degrees and W is the $n \times n$ weight matrix. Knowing that $L = D - W$, we finally reach:

$$\vec{y}^T L\vec{y} = \vec{a}^T X(D-W)X^T\vec{a} = \vec{a}^T XLX^T\vec{a} \tag{5}$$

Thus, we have to solve the following constrained minimization problem:

$$\arg\min_{\vec{a}} \ \vec{a}^T XLX^T\vec{a} \qquad \text{subject to} \qquad \vec{a}^T XDX^T\vec{a} = 1 \tag{6}$$

where the constraint is a general form to express that the norm of the vector $\vec{a}$ is a constant (the important thing here is the direction of the vector, not its magnitude). The Lagrangian function is given by:

$$L(\vec{a},\lambda) = \vec{a}^T XLX^T\vec{a} - \lambda(\vec{a}^T XDX^T\vec{a} - 1) \tag{7}$$

Taking the derivative with respect to the vector $\vec{a}$ and setting the result to zero:

$$\frac{\partial}{\partial\vec{a}}L(\vec{a},\lambda) = XLX^T\vec{a} - \lambda XDX^T\vec{a} = 0 \tag{8}$$

Therefore, we have a generalized eigenvector problem:

$$XLX^T\vec{a} = \lambda XDX^T\vec{a} \tag{9}$$

$$(XDX^T)^{-1}XLX^T\vec{a} = \lambda\vec{a} \tag{10}$$

showing that in order to minimize the objective function, we shoud select the vector a as

the smallest eigenvector of the matrix $(XDX^T)^{-1}XLX^T$. The multivariate version of the problem considers a $m \times d$ matrix A where each column $\vec{a}_j$ represents a direction in which data will be projected:

$$(XDX^T)^{-1}(XLX^T)A = \lambda A \tag{11}$$

In this situation, we should choose the d eigenvalues associated with the d smallest eigenvalues of $(XDXT)^{-1}XLXT$ to build the columns of the matrix A. Note that the transformation matrix A has m rows and d columns, while the output matrix Y has d rows and n columns, implying that each column $\vec{y}_j$ for $j = 1, 2, ..., n$ stores the coordinates of the j-th sample in the output space (after dimensionality reduction).

2.2 Graph-Based Learning

Graph-based learning methods are usually organized in two categories: graph construction and label inference. In this work, our focus is on the graph construction process. To build a graph that is a discrete approximation of a manifold, there are two basic steps: first, we need to define, from the set of all possible edges, which of them will be created, and second, we need to assign a weight to every edge created in the previous step. In the following, we will discuss two strategies for graph construction: Clustering with Adaptive Neighbors (CAN) [Nie et al. 2014] and Probabilistic Nearest Neighbors (PNN) [Ma et al. 2020].

2.2.1 Clustering with Adaptive Neighbors

As the name suggests, this graph construction method is more suitable for data clustering than classification. Let s_{ij} denote the probability that the sample $\vec{x}_j$ becomes a neighbor of $\vec{x}_i$. Then, the vector $\vec{s}_i \in R^n$ is composed by the probabilities of each sample in the dataset become a neighbor of $\vec{x}_i$. Intuitively, the smaller the distance $d(\vec{x}_i, \vec{x}_j)$, the greater the probability s_{ij} should be. The optimal probabilities for a single sample $\vec{x}_i$ is given by the solution to the following optimization problem:

$$min \;\; J(\vec{s}_i) = \sum_{j=1}^{n} \left(\|\vec{x}_i - \vec{x}_j\|^2 s_{ij} + \gamma_i s_{ij}^2 \right) \tag{12}$$

subject to $\vec{s}_i^T \vec{1} = 1$, where $0 \leq s_{ij} \leq 1$, n denotes the number of samples, $\gamma_i > 0$ is a regularization parameter and the constraint is necessary to enforce that the sum of the probabilities is equal to one. The first term of the objective function says that the edge weights are penalized by the Euclidean distance between the vertices, while the second term plays the role of a smooth constraint about the solution, trying to assess that all samples can be neighbors of $\vec{x}_i$ with roughly the same probability. Clearly, none of the limiting cases are especially interesting, but the goal is to find a good trade-off between data fidelity and prior knowledge.

Let $\vec{d}_i$ be the vector of squared Euclidean distances between the i-th sample $\vec{x}_i$ and all the other samples in the dataset. Then, we can express the optimization problem in terms of a vector norm:

$$min \;\; J(\vec{s}_i) = \left\| \vec{s}_i + \frac{1}{2\gamma_i} \vec{d}_i \right\|^2 \tag{13}$$

subject to $\vec{s}_i^T \vec{1} = 1$, where $0 \leq s_{ij} \leq 1$.

The Lagrangian function incorporates all the constraints into the objective function, leading to:

$$L(\vec{s}_i, \eta, \vec{\beta}_i) = \frac{1}{2} \left\| \vec{s}_i + \frac{1}{2\gamma_i} \vec{d}_i \right\|^2 - \eta(\vec{s}_i^T \vec{1} - 1) + \vec{\beta}_i^T \vec{s}_i \tag{14}$$

where η and $\vec{\beta}_i$ are the Lagrange multipliers. Differentiating with respect to s_{ij} and setting the result to zero leads to:

$$s_{ij} = -\frac{d_{ij}}{2\gamma_i} + \eta \tag{15}$$

where $d_{ij} = \|\vec{x}_i - \vec{x}_j\|^2$. To preserve the local geometric properties of the manifold, it is recommended that we link each sample only to a fixed number $k < n$ of neighbors. Without loss of generality, suppose the distances in $\vec{d}_i$ are sorted in ascending order. Considering that only the first k components in the optimal $\vec{s}_i$ are non-zero, then:

$$-\frac{d_{ik}}{2\gamma_i} + \eta > 0 \qquad\qquad -\frac{d_{ik+1}}{2\gamma_i} + \eta = 0 \tag{16}$$

By the constraint that the sum of all the elements of $\vec{s}_i$ must equal (since they denote probabilities) one, we have:

$$\sum_{j=1}^{k} \left(-\frac{d_{ij}}{2\gamma_i} + \eta \right) = 1 \tag{17}$$

From here, it can be shown that the solution to this minimization problem has a closed-form solution, given by [Nie et al. 2014]:

$$s_{ij} = \frac{d_{ik+1} - d_{ij}}{k d_{ik+1} - \sum_{j=1}^{k} d_{ij}} \tag{18}$$

where the vector $\vec{d}_i$ stores all the distances from $\vec{x}_i$ to other samples in ascending order.

2.2.2 Probabilistic Nearest Neighbors

The Probabilistic Nearest Neighbors (PNN) approach is more suitable for classification problems in label propagation algorithms [Ma et al. 2020]. Intuitively, the reason is that, the larger the s_{ij}, more influence the sample $\vec{x}_i$ has on the labeling of a neighboring sample $\vec{x}_j$. In classification problems, ideally, the variance of the propagation probabilities should be large enough to reflect the decision boundaries defined by samples belonging to different classes. To achieve this goal, a min-max normalization scheme is employed, which is a a usual to normalize data. After this transformation, all variables have the exact same scale, converting the data into the $[0, 1]$ interval. According to this, we have:

$$x_{ij}^* = \frac{x_{ij} - min(\vec{x}_i)}{max(\vec{x}_i) - min(\vec{x}_i)} \tag{19}$$

The main difference between CAN and PNN is the computation of γ_i, which is:

$$\gamma_i = \frac{d_{ik+1} - d_{i1}}{d_{ij} - d_{i1}} \left(\frac{k}{2} d_{ik+1} - \frac{1}{2} \sum_{j=1}^{k} d_{ij} \right) \tag{20}$$

The optimal probabilities in the PNN method are given by [Ma et al. 2020]:

$$s_{ij} = \frac{d_{ik+1} - d_{ij}}{d_{ik+1} - d_{i1}} \tag{21}$$

where the vector $\vec{d_i}$ stores all the distances from $\vec{x}_i$ to other samples in ascending order. It is worth to mention that the computational complexity of the KNN graph building method is $O(mn^2)$ (quadratic), where m is the dimension of the input space and n is the number of samples. According to the authors, CAN and PNN methods, on the other hand, have closed-form solutions, leading to a $O(n)$ (linear) computational complexity [Ma et al. 2020].

Algorithm 1 Probalistic Nearest Neighbors based Locality Preserving Projections

function PNN-LPP($X, n, k, d = 2, t = 1, \epsilon = 10^{-5}$)
  $A \leftarrow kNNGraph(X, k)$ ▷ Weighted adjacency matrix of the kNN graph
  $W \leftarrow exp\left\{-\frac{A^2}{t}\right\}$ ▷ A^2 denotes the pointwise power of the weights
  $P \leftarrow zeros(n, n)$ ▷ Matrix to store the final PNN weights
  for $i \leftarrow 1; i \leq n; i++$ **do** ▷ Loop for PNN weights computation
    $D \leftarrow W[i, :]$ ▷ D is the i-th row of W
    $order \leftarrow D.argsort()$ ▷ Indices of the weights in ascending order
    $D.sort()$ ▷ Sort the weights in ascending order
    for $j \leftarrow 1; j \leq n; j++$ **do**
      if $D[k+1] - D[1] \neq 0$ **then** ▷ To avoid division by zero
        $P[i, order[j]] \leftarrow \frac{D[k+1] - D[j]}{D[k+1] - D[1]}$ ▷ Compute the PNN weights
      else
        $P[i, j] \leftarrow 0$
      end if
    end for
  end for
  $\Delta \leftarrow diag(sum(P))$ ▷ Diagonal matrix: v_i degree is the sum of the i-th row
  $L \leftarrow \Delta - P$ ▷ Laplacian of the probabilistic kNN graph
  $M_1 \leftarrow X^T \Delta X$ ▷ Matrix to be inverted
  $M_1 \leftarrow M_1 + \epsilon I$ ▷ Regularization: add small value to the main diagonal
  $M_2 \leftarrow X^T L X$
  $M \leftarrow M_1^{-1} M_2$ ▷ Compute the LPP matrix
  $V, U \leftarrow eigen(M)$ ▷ Perform eigendecomposition of the matrix M
  $K \leftarrow U[:, 1:d]$ ▷ First d columns of U span the output space
  $Y \leftarrow K^T X$ ▷ Project the data in the output space
  return Y ▷ Y is a $d \times n$ data matrix
end function

2.3 The PNN-LPP Algorithm

The main goal of the proposed PNN-LPP method is to combine the Probabilistic Nearest Neighbors graph construction method to approximate the underlying data manifold in Locality Preserving Projections to perform dimensionality reduction based unsupervised metric learning. Our goal with PNN-LPP is to overcome two known limitations of regular LPP: 1) to make it less sensitive to noise and outliers in data, through the replacement of the Euclidean distance by a probabilistic measure; 2) to improve the performance of regular LPP in small sample size problems. Algorithm 1 shows the proposed PNN-LPP for dimensionality reduction based unsupervised metric learning. Basically, the method has four parameters: the data matrix X, the number of samples n, the number of neighbors k, the dimension of the output space $d = 2$, the variance of the Gaussian kernel $t = 1$ and the regularization parameter $\epsilon = 10^{-5}$, to avoid numerical problems in the inversion of a matrix. Another aspect that should be mentioned is that the k-NN graph used in the proposed PNN-LPP algorithm is the regular version, not the mutual k-NN graph. For the interested reader, the Python source code used to generate all the results presented in this paper can be found at https://github.com/alexandrelevada/PNN_LPP.

3 Experiments and Results

A set of computational experiments was conducted to compare the average classification accuracies obtained by eight supervised classifiers (KNN, Naive Bayes, SVM, Decision Trees, Bayesian classifier under Gaussian hypothesis, Multilayer Perceptron, Gaussian Process Classifier, and Random Forest classifiers) after dimensionality reduction to 2-D spaces in order to test and evaluate the proposed PNN-LPP method. An objective comparison of the proposed PNN-LPP against six unsupervised metric learning techniques based on dimensionality reduction: PCA, ISOMAP, LLE, Laplacian Eigenmaps, the regular LPP [He and Niyogi 2004], and UMAP [McInnes et al. 2018]. We chose 30 publicly accessible datasets from www.openml.org. It's worth noting that Table 1 has specific information about each of them, including the number of samples, features and classes. In all computational experiments, we chose 50% of the samples for training and 50% of the samples for testing.

It is widely known that state-of-the-art dimensionality reduction based unsupervised metric learning algorithms, such as t-SNE and UMAP have excellent performance in large datasets, in which the density of points in the underlying data manifold is high. However, when the number of samples is somehow limited, lowering the density of points in the input space, the performance of these algorithms tends to significantly drop. As deep neural networks, these methods require numerical optimization algorithms for error/distance minimization, such as stochastic gradient descent. Table 2 contains all of the acquired findings. The approach with the bold value is the best for that dataset. We can see that, for these datasets, PNN-LPP outperformed not only regular LPP, but also UMAP, a state-of-the-art algorithm, implying that the proposed method is a viable alternative for dimensionality reduction based unsupervised metric learning.

In order to verify whether the PNN-LPP performance is significantly superior than the performances of the other methods in these datasets, we performed a Friedman test [Friedman 1937], a statistical test that is considered to be a non-parametric version of ANOVA (Analysis of Variance). For a significant level $\alpha = 0.01$, we conclude that there are strong evidences against the null hypothesis (all methods have the same performance) ($p = 1.21 \times 10^{-16}$). Moreover, to check which methods are statistically different, we

#	Dataset	#samples	#fetaures	#classes
1	SPECTF	349	44	2
2	veteran	137	7	2
3	sleuth_ex1605	62	5	2
4	AIDS	50	4	2
5	cloud	108	7	2
6	FL2000	67	15	5
7	analcatdata_creditscore	100	6	2
8	corral	160	6	2
9	cars1	392	7	3
10	LED-display-domain-7digit	500	7	10
11	hayes-roth	160	4	3
12	Diabetes130US (1%)	1017	49	3
13	blogger	100	5	2
14	user-knowledge	403	5	5
15	rabe_131	50	5	2
16	haberman	306	3	2
17	prnn_synth	250	2	2
18	visualizing_environmental	111	3	2
19	vineyard	52	2	2
20	monks-problems-1	566	6	2
21	acute-inflammations	120	6	2
22	planning-relax	182	12	2
23	sensory	576	11	2
24	auto_price	159	15	2
25	wisconsin	194	32	2
26	fri_c4_250_100	250	100	2
27	thoracic_surgery	470	16	2
28	conference_attendance	246	6	2
29	analcatdata_boxing1	120	3	2
30	fri_c2_100_10	100	10	2
31	lupus	87	3	2
32	fruitfly	125	4	2

Table 1: Number of samples, features and classes of the selected openML datasets.

performed a Nemenyi post-hoc test [Nemenyi 1963, Demsar 2006] to perform pairwise comparisons. According to the test, there are strong evidences that PNN-LPP produced significantly higher average accuracies than PCA ($p < 10^{-3}$), ISOMAP ($p < 10^{-3}$), LLE ($p < 10^{-3}$), Laplacian Eigenmaps ($p < 10^{-3}$), regular LPP ($p < 10^{-3}$), and UMAP ($p < 10^{-3}$). A visual comparison of the clusters obtained after dimensionality reduction based metric learning is performed in the user-knowledge dataset is shown in Figure 1. The discrimination between the classes is more evident in the proposed method, as there is less overlap between the clusters.

Dataset	**PCA**	**ISO**	**LLE**	**LAP**	**LPP**	**UMAP**	**PNNLPP**
SPECTF	0.754	0.720	0.715	0.759	0.724	0.764	0.794
veteran	0.666	0.666	0.666	0.682	0.677	0.708	0.735
sleuth_ex1605	0.556	0.608	0.641	0.540	0.588	0.604	0.734
AIDS	0.374	0.355	0.370	0.415	0.360	0.410	0.630
cloud	0.657	0.643	0.664	0.625	0.634	0.655	0.680
FL2000	0.632	0.643	0.562	0.562	0.654	0.61	0.691
analcatdata_creditscore	0.795	0.795	0.730	0.750	0.737	0.757	0.825
corral	0.829	0.814	0.826	0.751	0.853	0.810	0.911
cars1	0.684	0.700	0.658	0.648	0.690	0.689	0.705
LED-display-domain-7digit	0.574	0.553	0.337	0.361	0.546	0.585	0.599
hayes-roth	0.615	0.476	0.479	0.403	0.465	0.440	0.633
Diabetes130US (1%)	0.523	0.525	0.527	0.529	0.534	0.526	0.565
blogger	0.679	0.667	0.707	0.600	0.697	0.622	0.795
user-knowledge	0.505	0.581	0.448	0.450	0.518	0.628	0.785
rabe_131	0.740	0.910	0.865	0.815	0.899	0.904	0.940
haberman	0.750	0.748	0.725	0.717	0.742	0.732	0.752
prnn_synth	0.857	0.858	0.761	0.708	0.857	0.846	0.868
visualizing_enviromental	0.671	0.649	0.587	0.560	0.687	0.642	0.714
vineyard	0.793	0.778	0.812	0.759	0.793	0.807	0.817
monks-problems-1	0.583	0.580	0.549	0.566	0.529	0.581	0.605
acute-inflammations	0.893	0.929	0.760	0.762	0.906	0.968	1.000
planning-relax	0.673	0.666	0.666	0.693	0.657	0.681	0.710
sensory	0.563	0.554	0.567	0.560	0.559	0.579	0.600
auto_price	0.924	0.920	0.762	0.840	0.914	0.929	0.958
wisconsin	0.610	0.595	0.552	0.567	0.592	0.614	0.639
fri_c4_250_100	0.549	0.559	0.571	0.588	0.487	0.536	0.595
thoracic_surgery	0.813	0.814	0.813	0.805	0.820	0.815	0.823
conference_attendance	0.853	0.852	0.845	0.847	0.856	0.851	0.861
analcatdata_boxing1	0.691	0.639	0.679	0.622	0.668	0.641	0.729
fri_c2_100_10	0.700	0.617	0.537	0.585	0.702	0.635	0.722
lupus	0.784	0.803	0.707	0.673	0.69	0.792	0.818
fruitfly	0.521	0.500	0.567	0.521	0.539	0.507	0.623
Average	0.682	0.679	0.645	0.633	0.674	0.683	0.746
Median	0.676	0.658	0.665	0.624	0.682	0.649	0.732
Minimum	0.374	0.355	0.337	0.361	0.360	0.41	0.565
Maximum	0.924	0.929	0.865	0.847	0.914	0.968	1.000

Table 2: Average classification accuracies produced after dimensionality reduction based unsupervised metric learning with PCA, ISOMAP, LLE, Laplacian Eigenmaps (LAP), LPP, UMAP and PNN-LPP for 32 openML datasets (2-D case).

Despite the fact that the results are intriguing, the proposed strategy has certain drawbacks. The most important is that PNN-LPP requires is quite sensible to the parameter K, which governs the size of the neighborhood system in Probabilistic Nearest Neighbors. The way this parameter is defined has a direct impact on the results: during the experiments, we have observed that the classification accuracies are quite sensitive to changes in the value of K. We used the a simple strategy in this study: to perform a line search in the integers belonging to the interval $[2, max\{n/2, 50\}]$ for each dataset. The best model is defined as the one that optimizes classification accuracy over all K values. It is worth mention that, although we use the class labels to perform model selection, the dimensionality reduction based metric learning is completely unsupervised. At the present moment, we still do not have an automated strategy for the estimation of the

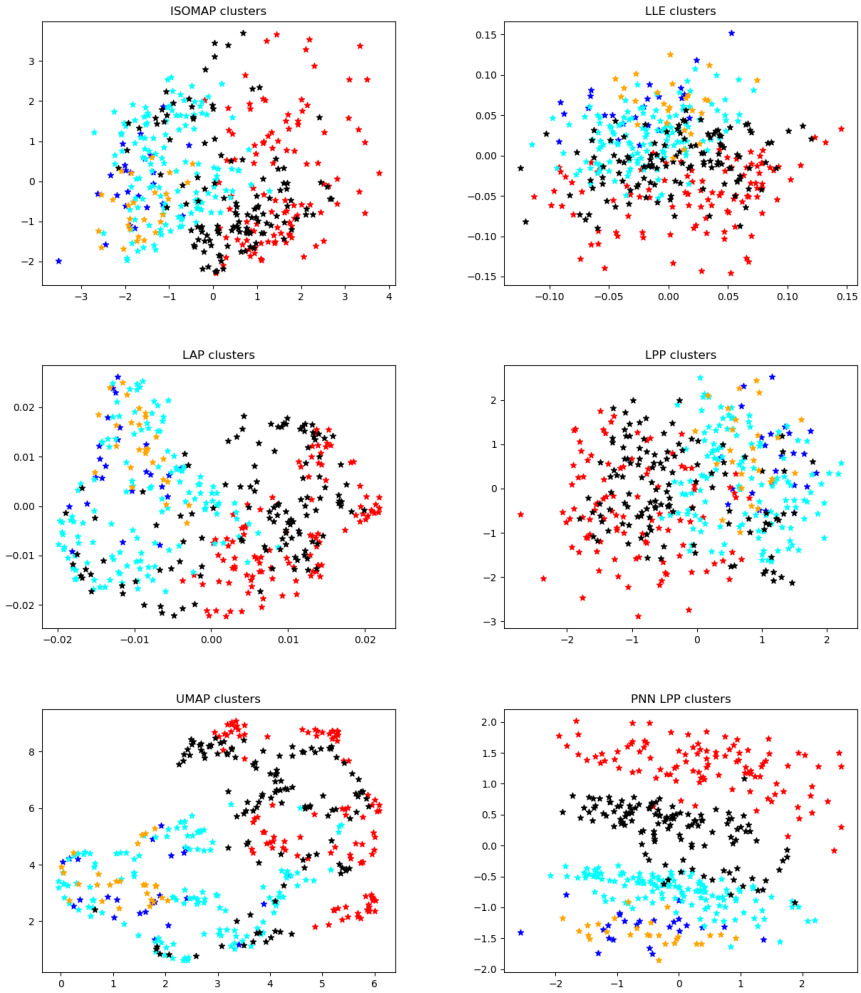

Figure 1: Scatterplots of the user-knowledge dataset for the 2D case. From left to right, top to bottom: ISOMAP, LLE, Laplacian Eigenmaps, LPP, UMAP and the proposed PNN-LPP for $K = 9$.

optimal parameter K. For all experiments described in this paper, we fixed the parameter $t = 1$ and $d = 2$. We expect that even better results can be obtained by optimizing the values of these parameters.

On the other hand, one advantage of the proposed technique is that it has been shown in various computational experiments that PNN-LPP usually performs better than its competitors when the number of samples is limited. In other words, the proposed strategy appears to be promising for dealing with difficulties involving small sample sizes. The methods t-SNE and UMAP, for example, are state-of-the-art algorithms for dimensionality reduction based metric learning that require a large number of samples

for providing good results, since as the optimization problems do not have closed-form solutions, they require numerical algorithms (stochastic gradient descend) that require more data for convergence.

4 Conclusions and Final Remarks

Unsupervised metric learning and manifold learning are tightly related. Many methods for determining the underlying geometric structure from data have been developed, with Laplacian Eigenmaps being one of the most important. Nevertheless, it does suffer from the out-of-sample problem. Locality Preserving Projections was created to address this specific limitation in Laplacian Eigenmaps. However, most versions use the Euclidean metric to compare samples in the KNN graph (discrete approximation for the manifold), which is problematic because we know that most real-world datasets do not have a linear geometric structure.

In this paper, we proposed Probabilistic Nearest Neighbor Locality Preserving Projections as a non-parametric approach for dimensionality reduction based unsupervised metric learning. Our goal was to replace the pointwise Euclidean distance by a patch-based probabilistic distance to make LPP more resilient against the presence of variations in data, such as noise and outliers. The proposed PNN-LPP features can be more discriminative in supervised classification than features produced from conventional manifold learning techniques, according to our computational experiments. Moreover, one of the main problems with state-of-the-art approaches such as t-SNE and UMAP is the unreasonable performance in small sample size problems due to the necessity of numerical optimization algorithms. The results indicate that PNN-LPP improves the performance of regular LPP in situations where the number of samples is limited, showing that it can be a viable option in unsupervised metric learning.

Future works may include the estimation of the intrinsic dimensionality d for each dataset, the incorporation of information-theoretic distances, such as KL, Bhattacharyya, Hellinger, and Cauchy-Schwarz divergences, as well as geodesic distances based on the Fisher information matrix of a parametric statistical model. Another alternative is to estimate non-parametric local densities using kernel density estimation techniques (KDE). By establishing the adjacency relations that define the discrete approximation for the manifold, the ϵ-neighborhood approach may also be utilized to build non-regular graphs. Additionally, a supervised PNN-LPP can be created by combining the Euclidean distance and the probabilistic measures in the following way: the edges of the KNN graph for which the endpoints belong to the same class are weighted with the minimum of the two distances, while the edges for which the endpoints belong to different classes are weighted with the sum of the distances, ensuring that intra-class variations are smaller than inter-class variations. Other graph construction techniques can also be employed instead of the regular k-NN graph, such as the mutual k-NN graph and ϵ-ball graphs (instead of making the number of neighbors fixed, we fix the radius around a sample). Finally, we plan to investigate a method for automatically estimating an ideal value for the patch size control parameter K.

Acknowledgements

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

References

[Belkin and Niyogi 2003] Belkin M, Niyogi P. Laplacian eigenmaps for dimensionality reduction and data representation, Neural Computation, 2003, 15(6), pp. 1373–1396.

[Belkin and Niyogi 2007] Belkin M, Niyogi P. Convergence of laplacian eigenmaps. In: Schölkopf B, Platt J C, Hoffman T. (Eds.) Advances in Neural Information Processing Systems, MIT Press, 2007, 19, pp. 129–136.

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

[Chen et al. 2020] Chen S, Wu X, Xu J. Locality preserving projection least squares twin support vector machine for pattern classification, Pattern Analysis and Applications, 2020, 23, pp. 1–13.

[Chong et al. 2020] Chong Y, Ding Y, Yan Q, Pan S. Graph-based semi-supervised learning: a review, Neurocomputing, 2020, 408, pp. 216–230.

[Demsar 2006] Demsar J. Statistical comparisons of classifiers over multiple data sets, Journal of Machine Learning Research, 2006, 7, pp. 1–30.

[Dutta et al. 2020] Dutta U K, Harandi M, Sekhar C C. Unsupervised metric learning with synthetic examples, In: Proceedings of the AAAI Conference on Artificial Intelligence, 2020, 34(4), pp. 3834–3841.

[Friedman 1937] Friedman M, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, Journal of the American Statistical Association, 1937, 32(200), pp. 675–701.

[He and Niyogi 2004] He X, Niyogi P. Locality preserving projections, In: Thrun S, Saul L K, Schölkopf B. (Eds.), Advances in Neural Information Processing Systems, 2004, 16, pp. 153–160.

[Hu et al. 2018] Hu X, Sun Y, Gao J, Hu Y, Yin B. Locality preserving projection based on f-norm, In: Proceedings of the AAAI Conference on Artificial Intelligence, 2018, 32(1), pp. 1330–1337.

[Kayabasi et al. 2021] Kayabasi A, Karaman K, Akkaya, I B. Comparison of distance metric learning methods against label noise for fine-grained recognition, in: Hammoud R I, Overman T L, Mahalanobis A (Eds.), Automatic Target Recognition XXXI, Vol. 11729, International Society for Optics and Photonics, SPIE, 2021, pp. 107–118.

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

[Levada 2021] Levada A L M. PCA-KL: a parametric dimensionality reduction approach for unsupervised metric learning, Advances in Data Analysis and Classification, 2021, 15, pp. 829–868.

[Lin et al. 2015] Lin B, He X, Ye J. A geometric viewpoint of manifold learning, Applied Informatics, 2015, 2, pp. 3.

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

[Ma et al. 2019] Ma J, Wang N, Xiao B. Semi-supervised classification with graph structure similarity and extended label propagation, IEEE Access, 2019, 7, pp. 58010–58022.

[Ma et al. 2020] Ma J, Xiao B, Deng C. Graph based semi-supervised classification with probabilistic nearest neighbors, Pattern Recognition Letters, 2020, 133, pp. 94–101.

[McInnes et al. 2018] McInnes L., Healy J., Melville J. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction, http://arxiv.org/abs/1802.03426, 2018.

[Nie et al. 2014] Nie F, Wang X, Huang H. Clustering and projected clustering with adaptive neighbors. In: Proceedings of the 20th ACM International Conference on Knowledge Discovery and Data Mining (SIGKDD), 2014, p. 977–986.

[Nemenyi 1963] Nemenyi P. Distribution-free multiple comparisons, Ph.D. thesis, Princeton University, 1963.

[Ran et al. 2022] Ran R, Qin H, Zhang S, Fang B. Simple and robust locality preserving projections based on maximum difference criterion, Neural Processing Letters, 2022, 54, pp. 1783–1804.

[Sarker 2000] Sarker I. Machine learning: Algorithms, real-world applications and research directions, SN Computer Science, 2021, 2, pp. 60.

[Taskin and Crawford 2019] Taskin G, Crawford M M. An out-of-sample extension to manifold learning via meta-modeling, IEEE Transaction on Image Processing, 2019, 28(10), pp. 5227–5237.

[Xiao et al. 2010] Xiao R, Zhao Q, Zhang D, Shi P. Data classification on multiple manifolds, in: Proceedings of the 20th International Conference on Pattern Recognition (ICPR), 2010, pp. 3898–3901.