Semi-Supervised Manifold Learning for Hyperspectral Data

Florian Becker

Vision and Fusion Laboratory
Institute for Anthropomatics
Karlsruhe Institute of Technology (KIT), Germany
florian.becker@kit.edu

Technical Report IES-2019-11

Abstract

There are real world data sets where a linear approximation like the principal components might not capture the intrinsic characteristics of the data. Nonlinear dimensionality reduction or *manifold* learning uses a graph-based approach to model the local structure of the data. Manifold learning algorithms assume that the data resides on a low-dimensional manifold that is embedded in a higher-dimensional space. For real world data sets this assumption might not be evident. However, using manifold learning for a classification task can reveal a better performance than using a corresponding procedure that uses the principal components of the data. We show that this is the case for our hyperspectral data set using the two manifold learning algorithms Laplacian eigenmaps and locally linear embedding.

1 Introduction

Nonlinear dimensionality reduction or *manifold learning* is a useful tool for high-dimensional data analysis. In contrast to linear dimensionality reduction as it is performed by a standard principal component analysis (PCA), with manifold learning the possibly low-dimensional manifold that is embedded in a high-dimensional space can be uncovered. This so-called *manifold assumption* is central to the theory of manifold learning and states that the data resides on a low-dimensional manifold in high-dimensional space. Manifold learning has been applied to many computer vision problem domains including face recognition [3], image retrieval [4] and medical image analysis [1]. Due to the high spectral resolution of many hyperspectral image data sets and the high correlation between adjacent and overtone bands, manifold learning has received some attention in the research community [5].

In this technical report, we will first review the basics of manifold learning, why it is a useful framework and how it can be utilized for classification in a semi-supervised manner. Finally, we will apply this semi-supervised procedure to a hyperspectral data set consisting of four different kinds of wood (chips): eucalyptus, poplar, beech and spruce. The results indicate that manifold learning outperforms a linear approach using PCA.

2 Classification with manifold learning

Discovering the low-dimensional manifold embedded in a higher-dimensional space can be utilized for classification. We aim to show two aspects of manifold learning: First, it can be employed for classification, second, manifold learning outperforms a corresponding linear procedure using principal component analysis. In general, dimensionality reduction is often used as a step prior to classification. This is due to the fact that for many datasets, the dimensions of individual data points might be correlated due to the physical nature of the process that has generated the data. For instance, in (near) infrared spectroscopy overtone bands can be observed that are a manifestation of the vibrational modes. As the resonant frequencies can be approximated by an harmonic oscillator, characteristic peaks in the spectrum might arise from the vibrational modes of the same chemical substance. For a classification task correlation means that specific dimensions might not carry valuable information, in the sense that the additional information does not lead to a better separability of the data and therefore also does not contribute to the classification performance. Removing correlated dimensions can therefore lead to a simpler classifier with less parameters. When applying manifold learning prior to classification, the objective is to exploit the manifold assumption. Manifold learning is a good fit to the data when there are non-linear dependencies between different dimensions. In practice, it is not evident that non-linear dependencies exist in high-dimensional data. However, if manifold learning leads to better classification results than a linear method, this might indicate the presence of an intrinsic low-dimensional manifold.

3 Laplacian eigenmaps

We briefly review the basics of one popular manifold learning algorithm called Laplacian Eigenmaps (LE). Given data samples $\mathcal{X} = \{\mathbf{x}_i\}_{i=0}^{N} \subseteq \mathbb{R}^n$, LE computes a Laplacian matrix according to a kernel function. The final mapping is then defined by the eigenvectors of the graph Laplacian matrix. A detailed description is given by Algorithm 3.1 below. Central to the algorithm is the choice of the kernel function. We call a symmetric function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ a kernel, if the induced *Gram matrix* defined by $K_{ij} = k(\mathbf{x_i}, \mathbf{x_j})$ is positive semi-definite, i.e.

$$\boldsymbol{x}^T \boldsymbol{K} \boldsymbol{x} = \sum_{i=1}^{N} \sum_{j=1}^{N} x_i \, K_{ij} \, x_j \geq 0, \tag{3.1}$$

for all $\boldsymbol{x} \in \mathbb{R}^n$. This is the discrete analog to Mercer's condition [6] which states that the function $K : [a, b] \times [a, b] \to \mathbb{R}$ fulfills the inequality

$$\int \int f(x) K(x, y) f(y) dx dy \geq 0 \tag{3.2}$$

for every function $f \in L^2(\mathbb{R})$. A symmetric kernel function satisfying Mercer's condition leads to nonnegative real eigenvalues and orthogonal eigenvectors for the corresponding kernel matrix.

Algorithm 3.1 Laplacian Eigenmaps [2]

```
procedure LAPLACIAN EIGENMAPS
    Input: data $\mathcal{X} = \{\mathrm{x}_i\}_{i=0}^{N} \subseteq \mathbb{R}^n$
    Output: embedding $\mathcal{Y} = \{\mathrm{y}_i\}_{i=0}^{N} \subseteq \mathbb{R}^m$
    1.) Build an adjacency graph $G = (V, E)$
        nodes $v_i \in V$ and $v_j \in V$ are connected if $||\boldsymbol{x_i} - \boldsymbol{x_j}||_2^2 < \varepsilon$
    2.) Pick weights
        Choose a kernel function $k(\boldsymbol{x}_i, \boldsymbol{x}_j)$ and set
        $\boldsymbol{W}_{ij} = \begin{cases} k(\boldsymbol{x}_i, \boldsymbol{x}_j) & (i,j) \in E \\ 0 & \text{else} \end{cases}$
    3.) Compute Eigenmap
        $\boldsymbol{Ly} = \lambda \boldsymbol{Dy}, \quad$ with $\boldsymbol{D}_{ii} = \sum_j \boldsymbol{W}_{ji}$ and $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W}$
        $\boldsymbol{x}_i \rightarrow (\boldsymbol{y}_1(i), \ldots, \boldsymbol{y}_m(i))$
end procedure
```

The embedding is found by computing the generalized eigenvalue problem involving the graph Laplacian and the corresponding degree matrix. The nonlinear nature of Algorithm 3.1 is due to the choice of the kernel function.

4 Semi-supervised manifold learning

Semi-supervised machine learning methods make use of unlabeled data points for training. Transductive learning is one variant of a semi-supervised learning setting where the correct labels of some given unlabeled data points must be inferred. This is in contrast to inductive learning where a function is learned that maps a data point to its label. Manifold learning algorithms are label-agnostic: In order to build the adjacency graph no information about class labels is necessary. The main idea behind a semi-supervised manifold learning approach is that the kernel matrix is built using labeled and unlabeled data points. The resulting matrix quantifies the similarity between all pairwise data points. As a subset of these data points is labeled, the kernel matrix relates each unlabeled data point to every labeled data point. The computation of the eigenmap and the projection of the high-dimensional data leads to an embedded space with partially labeled data points. Unlabeled points can be classified with a simple nearest-neighbor search. In this way, the intrinsic manifold structure–given that it exists–is put to use for a classification task.

Algorithm 4.1 Semi-Supervised Manifold Learning

```
procedure Semi-Supervised Manifold Learning
    Input:  labeled data $\{(\boldsymbol{x}_1, c_1), \ldots, (\boldsymbol{x}_p, c_p)\}$,
            unlabeled data $\{\boldsymbol{x}_1^u, \ldots, \boldsymbol{x}_q^u\}$
    Output: labels for $\{\boldsymbol{x}_1^u, \ldots, \boldsymbol{x}_q^u\}$
    1.) Compute embedding by manifold learning algorithm
        e.g. by $\boldsymbol{L}\boldsymbol{y} = \lambda\boldsymbol{D}\boldsymbol{y}$
    2.) Embed all data points
        $\boldsymbol{x}_i \longrightarrow \boldsymbol{y}_1(i), \ldots, \boldsymbol{y}_m(i)$
    3.) Classify unlabeled data points
    for all unlabeled data points $\boldsymbol{x}^u$ do
        get the labels of the $k$ nearest labeled points in the embedded space
        assign data point $\boldsymbol{x}^u$ the most common label
    end for
end procedure
```

The procedure described above can be used together with any manifold learning algorithm. In order to compare LE, we also apply a further manifold learning algorithm to the data set called locally linear embedding (LLE). For a given data set $\mathcal{X} = \{\mathbf{x}_i\}_{i=0}^N \subseteq \mathbb{R}^n$, LLE tries to reconstruct every data point from a linear combination of its k-nearest neighbors. LLE minimizes the following cost function:

$$
\begin{aligned}
E(W) = \sum_{i=1}^{N} ||\boldsymbol{x}_i - \sum_{\boldsymbol{x}_j \in \mathcal{N}_k(\boldsymbol{x}_i), j \neq i} w_{ij}\boldsymbol{x}_j||_2^2 \\
\text{s.t.} \sum_{i=1} W_{ij} = 1 \qquad \forall j \in \{1, \ldots, N\}
\end{aligned}
\tag{4.1}
$$

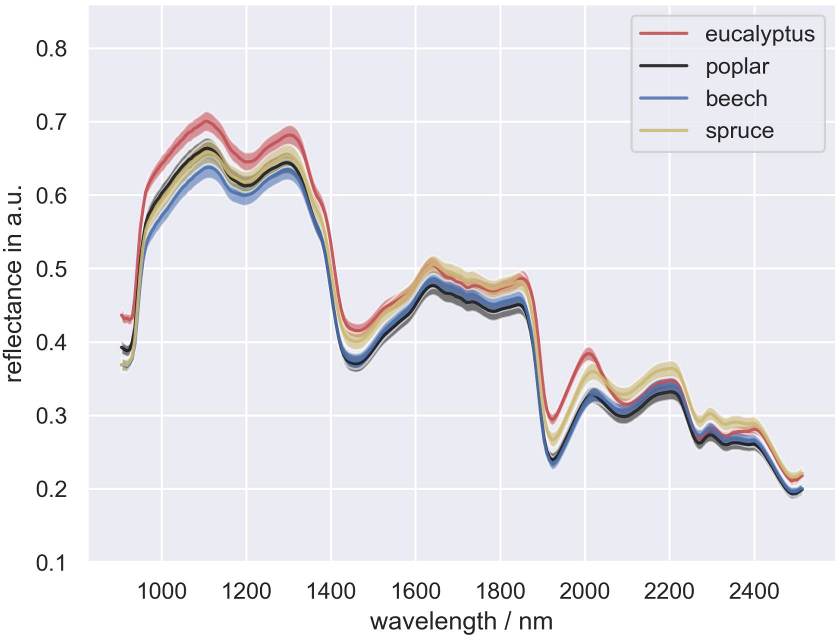

Figure 4.1: Spectra of the four different woods: eucalyptus, poplar, beech and spruce. This plot also shows the standard deviation (0.1σ) around the mean.

$\mathcal{N}_k(\boldsymbol{x})$ denotes the set of k-nearest neighbors of $\boldsymbol{x}$. In order to achieve a neighborhood preserving map, the resulting weight matrix from the optimization problem 4.1 above is used to find an embedding:

$$E(Y) = \sum_{i=1}^{N} ||y_i - \sum_{y_j \in \mathcal{N}(y_i), j \neq i} w_{ij} y_j||_2^2. \tag{4.2}$$

In the following, we describe the methodology that was used to apply and validate Algorithm 4.1 for hyperspectral data. The hyperspectral images were acquired using a Specim SWIR camera with spectral range from 950 nm–2500 nm and a spectral resolution of 10 nm. Figure 4.1 shows the entirety of the

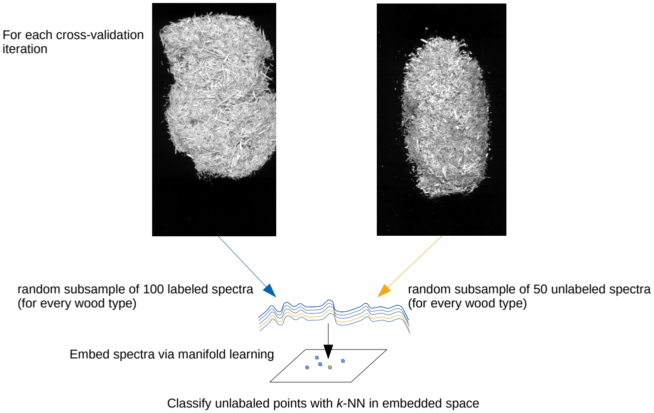
For each cross-validation iteration
random subsample of 100 labeled spectra (for every wood type)
random subsample of 50 unlabeled spectra (for every wood type)
Embed spectra via manifold learning
Classify unlabaled points with *k*-NN in embedded space

Figure 4.2: The proposed methodology. We acquired separate data sets for training and testing. For each cross-validation iteration, we sampled 100 labeled and 50 unlabeled spectra from every wood type. No further preprocessing of the spectra is applied. Based on this data, the Laplacian (and the locally linear embedding optimization problem) is computed. The images above of the fine wood chips are averages over all hyperspectral bands.

spectra for the four classes in terms of a mean spectrum with 0.1σ. Separate image sets were acquired for training and testing. To evaluate Algorithm 4.1, a target dimension of 2 was chosen for all dimensionality reduction procedures.

5 Results

The above methodology leads to the results given in Table 5.1. The results indicate that the used manifold learning algorithms outperform linear dimensionality reduction in terms of a 1-nearest neighbor classification in the embedded space. Furthermore, we used two different kernel functions k_{rbf} and $k_{\cos}$. The overall accuracy for $k_{\cos}$ leads to better results. As the spectra were not preprocessed, this result is not too surprising as the cos-similarity is invariant to linear shifts of the spectrum–which is in contrast to the rbf-kernel. We furthermore observe that LE outperforms LLE for our data set.

Table 5.1: Classification results for PCA and the different manifold learning algorithms using the the semi-supervised manifold learning procedure outlined in Algorithm 4.1. The overall accuracy (OA) is given in the last column.

Method	**Eucalyptus**	**Poplar**	**Beech**	**Spruce**	**OA** ($\mu + \sigma$)
PCA	0.61	0.74	0.76	0.58	0.67 + 0.036
LLE$_{k=30}$	0.68	0.81	0.76	0.60	0.71 + 0.019
LLE$_{k=40}$	0.72	0.83	0.76	0.63	0.74 + 0.018
LE$_{\mathrm{rbf}}$	0.75	0.86	0.76	0.61	0.74 + 0.020
LE$_{\cos}$	0.76	0.95	0.75	0.66	0.78 + 0.016

Especially LE$_{\cos}$ significantly outperforms the PCA-based approach. In addition, as indicated by the standard deviation, LE$_{\cos}$ is the most robust method, while throughout the cross-validation the variance of the PCA-based procedure is the highest.

6 Conclusion & Outlook

In essence, Laplacian eigenmaps and locally linear embedding build a discrete approximation of the underlying data manifold. By computing a weight matrix that captures the local structure of the data, the intrinsic characteristics are utilized for dimensionality reduction. The induced neighborhood preserving map is a suitable tool for high-dimensional data analysis. We have applied manifold learning for a semi-supervised classification task and showed that it outperforms classification in the space that is defined by the principal components. Our results indicate that choosing a kernel function is a critical step for LE. Manifold learning has the potential to uncover the low-dimensional manifold of the data. Future work should continue to examine this potential.

References

[1] Paul Aljabar et al. “Combining morphological information in a manifold learning framework: application to neonatal MRI”. In: *International Conference on Medical Image Computing and Computer-Assisted Intervention*. Springer. 2010, pp. 1–8.

[2] Mikhail Belkin and Partha Niyogi. “Laplacian eigenmaps for dimensionality reduction and data representation”. In: *Neural computation* 15.6 (2003), pp. 1373–1396.

[3] Xiaofei He et al. “Face recognition using laplacianfaces”. In: *IEEE Transactions on Pattern Analysis & Machine Intelligence* 3 (2005), pp. 328–340.

[4] Yen-Yu Lin, Tyng-Luh Liu, and Hwann-Tzong Chen. “Semantic manifold learning for image retrieval”. In: *Proceedings of the 13th annual ACM international conference on Multimedia*. ACM. 2005, pp. 249–258.

[5] Dalton Lunga et al. “Manifold-learning-based feature extraction for classification of hyperspectral data: A review of advances in manifold learning”. In: *IEEE Signal Processing Magazine* 31.1 (2013), pp. 55–66.

[6] Alex J Smola and Bernhard Schölkopf. “From regularization operators to support vector kernels”. In: *Advances in Neural information processing systems*. 1998, pp. 343–349.