Supervised Laplacian Eigenmaps 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-2018-09

Abstract

With Laplacian eigenmaps the low-dimensional manifold of high-dimensional data points can be uncovered. This nonlinear dimensionality reduction technique is popular due to its well-understood theoretical foundation. This paper outlines a straightforward way to incorporate class label information into the standard (unsupervised) Laplacian eigenmaps formulation. With the example of hyperspectral data samples this supervised reformulation is shown to reinforce within-class clustering and increase between-class distances.

1 Introduction

Advances in hyperspectral image acquisition and domain-specific machine learning methods for clustering and classification of the acquired data are progressing in tandem. By its very nature hyperspectral images tend to be high-dimensional as sensors capture many narrow and contiguous spectral bands and algorithms for analysis often combine spatial as well as spectral information. Due to the high resolution of many hyperspectral image (HSI) data sets, it has been suggested to employ nonlinear dimensionality reduction (or *manifold learning*) as a method to embed high dimensional data points in a lower dimensional space,

while preserving the local geometry. Nonlinear dimensionality reduction has proven itself useful in many different domains, like face recognition [CK09], speech recognition [BO95] and image retrieval [LLC05]. In contrast to a principal component analysis, where data points are projected onto a linear subspace, nonlinear dimensionality reduction techniques can find the low-dimensional nonlinear manifold that is possibly embedded inside a higher-dimensional space. Manifold learning is therefore very suitable for data sets, where an intrinsic low-dimensional structure is suspected. A classical example of this is an image series of a person looking in various different directions. The images itself are rather high-dimensional, the intrinsic lower dimension however, can be characterized by a Euclidean space, where one axis represents looking right and left, and the other axis up and down [RYS04]. Manifold learning methods are generally able to find this structure. In this paper, we focus on Laplacian eigenmaps (LE), a classical manifold learning algorithm [BN03]. We show how the standard LE formulation can be adapted in order to take class labels into account and how this supervised reformulation is an improvement.

Different approaches to incorporate class label information into manifold learning applications have already been considered. In [RD12] for instance, one within-class and one between-class graph was constructed to achieve a supervised manifold learning formulation. In contrast to this, we show that one graph and the associated affinity matrix equipped with a certain kernel function is sufficient.

The remainder of this technical report is organized as follows: We first begin by revisiting the standard Laplacian eigenmaps formulation and motivate the importance of the Laplacian matrix and its relation to the Laplace operator. In the following section the supervised version of the Laplacian eigenmaps method is outlined. The algorithm is evaluated by using a hyperspectral dataset that was acquired by the AVIRIS sensor. We show quantitative and qualitative results.

2 Laplacian Eigenmaps

Given data samples $\mathcal{X} = \{\mathbf{x}_i\}_{i=0}^{n} \subseteq \mathbb{R}^m$ many classical manifold learning algorithms start with the construction of an undirected weighted graph $G = (V, W)$, where each node $v_j \in V$ represents one data point and W is the $n \times n$ affinity matrix. Affinity or similarity can intuitively be understood as an inverse distance

measure.[1] Note, that the dimensionality m of the data points does not appear in W, as it encodes all pairwise affinities. One common affinity measure is the so called Gaussian heat kernel

$$W_{ij} = k_{\text{Gauss}}(\mathbf{x}_i, \mathbf{x}_j; \beta) = e^{\frac{-||x_i - \mathbf{x}_j||_2^2}{\beta}},$$

with $\beta \in \mathbb{R}$. This Gaussian heat kernel is applied to all $\mathbf{x}_j, \mathbf{x}_k \in \mathcal{X}$, provided that the Euclidean distance between the samples is smaller than a certain $\varepsilon > 0$. In general, 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. Choosing an adequate affinity measure is of course one crucial aspect of LE. When dealing with signals or (hyper)spectral data, one might for instance consider the cosine similarity or dynamic time warping. For the time being, we focus on the Gaussian heat kernel. After the construction of the weighted graph G, the *eigenmaps*, which define the mapping to the low-dimensional space, must be computed. Therefore, the generalized eigenvector problem must be solved:

$$L\mathbf{y} = \lambda D\mathbf{y}, \tag{2.1}$$

where $L = D - W$ is the so called Laplacian matrix, and D the diagonal degree matrix.

Now, if $\mathbf{y}_0, \ldots, \mathbf{y}_{n-1}$ are the solutions to the above generalized eigenvalue equation (2.1), then order the equations according to the eigenvalues, such that λ_0 is the smallest:

$$\begin{aligned} L\mathbf{y}_0 &= \lambda_0 D\mathbf{y}_0 \\ L\mathbf{y}_1 &= \lambda_1 D\mathbf{y}_1 \\ &\vdots \\ L\mathbf{y}_{n-1} &= \lambda_{n-1} D\mathbf{y}_{n-1} \end{aligned}$$

[1] Note however, that affinity measures are not necessarily required to be an inverse metric. For instance, the inverse of the dynamic time warping distance could be used as an affinity measure (see e.g. [SNNS02]), although it is only a *semi*-metric, i.e. it does not satisfy the triangle inequality.

Finally, the mapping $\Phi : \mathbb{R}^m \to \mathbb{R}^p$ into a p-dimensional target space is defined by the p eigenvectors:

$$\Phi(\mathbf{x}_i) = [\,\mathbf{y}_1(i), \ldots, \mathbf{y}_p(i)\,]^T$$

The goal is to have a mapping, where $\mathbf{y}_i$ and $\mathbf{y}_j$ are "close" together, i.e. we want to minimize

$$\begin{aligned}\sum_{ij}(\mathbf{y}_i - \mathbf{y}_j)^2 W_{ij} &= \sum_{ij}(\mathbf{y}_i^2 + \mathbf{y}_j^2 - 2\mathbf{y}_i\mathbf{y}_j)W_{ij} \\ &= \sum_i \mathbf{y}_i^2 D_{ii} + \sum_j \mathbf{y}_j^2 D_{jj} - 2\sum_{ij}\mathbf{y}_i\mathbf{y}_j W_{ij} \\ &= 2\mathbf{y}^T L\mathbf{y}.\end{aligned}$$

The last step of the above derivation is true because by definition $D_{ii} = \sum_j W_{ji}$. Hence, the minimization problem reduces to

$$\operatorname*{argmin}_{\mathbf{y}^T D\mathbf{y}=1} \mathbf{y}^T L\mathbf{y}.$$

Finding a vector $\mathbf{y}$ that minimizes this objective function is equivalent in finding the eigenvectors of Equation (2.1).

3 Laplace Operator

The Laplacian matrix plays an important role in graph theory and can for instance be used to approximate the sparsest cut of a graph [AHK10] or to compute s-t flows [CKM$^+$11]. Another interesting property of the Laplacian matrix is that it can be understood as an discrete Laplace operator. In this subsection, we will motivate this aspect of the Laplacian matrix. To begin with, consider the definition of the Laplace operator, which is a second order differential operator of a function ϕ:

$$\Delta\phi = \sum_{i=1}^{n} \frac{\partial^2\phi}{\partial x_i^2},$$

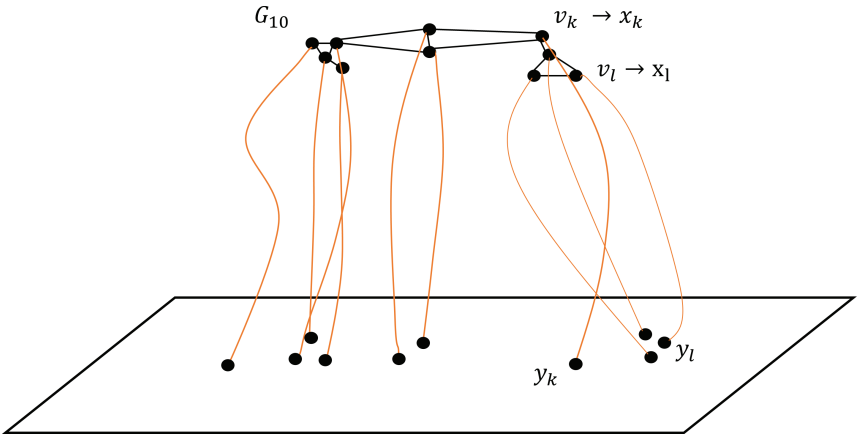


Figure 2.1: Conceptual projection of a data set with ten samples. Every of the vertices of G_{10} stands for one such data point. With $v_k \rightarrow x_k$ we denote that x_k is "represented" by v_k. The length of the edges is according to a certain affinity/kernel function, i.e. shorter edges mean a higher affinity. This must be reflected by the projection: A high affinity between nodes must lead to the points being "close" together in the lower-dimensional space.

where ϕ is a twice-differentiable function. $\Delta\phi$ is called the *Laplacian* of ϕ. Now, as we want to relate this continuous Laplace operator to the analogous discrete case, consider a grid, or rather any arbitrary undirected weighted graph G and let $\psi : V \rightarrow \mathbb{R}$ be a function that maps every node v_j of G to a real number $\nu_j = \psi(v_j)$. For the sake of vividness, assume that this number represents a temperature ν_j^t at a discrete time step t. The following derivation relates the temperature change $\Delta^{t:t+1}\nu_i = ||\nu_i^t - \nu_i^{t+1}||$ from time step t to $t+1$ to the difference of the neighboring nodes temperatures, where W_{ij} can be thought of as the heat conduction between nodes v_j and v_i.

$$\Delta^{t:t+1}\nu_i \propto -\sum_j W_{ij}(\nu_i - \nu_j)$$

$$
\begin{aligned}
&= -(\nu_i \sum_j W_{ij} - \sum_j W_{ij}\nu_j) \\
&= -(\nu_i \deg_W(v_i) - \sum_j W_{ij}\nu_j) \\
&= -\sum_j (\delta_{ij} \deg_W(v_i) - W_{ij})\nu_j \\
&= -\sum_j L_{ij}\nu_j
\end{aligned}
$$

$\deg_W(v_i)$ denotes i-th row (or column)[2] sum of W and δ_{ij} is the Kronecker delta.

We see that from the simple fact that the change in temperature is proportional to the difference of the neighboring temperatures (*Newton's law of cooling*), the Laplacian matrix L emerges. The above derivation of heat transfer on a graph resembles the heat equation

$$\frac{\partial u}{\partial t} = -\alpha \Delta u,$$

where the Laplacian matrix replaces the Laplace operator.

Consider the weighted graph from Figure 3.1 and its corresponding weight matrix W, diagonal degree matrix D and the resulting Laplacian matrix L. Multiplying the vector of node numbers with the Laplacian matrix, gives the negative change of those node temperatures for the next time-step.

4 Supervised Laplacian Eigenmaps

In order to incorporate class label information, we construct the affinity matrix as follows. Let $(\mathbf{x}_i, \ell) \in \mathbb{R}^m \times \{1, \ldots, L\}$ or $\mathbf{x}_i^\ell$ for short be a data sample and its associated label in a multi-class setting. The entries of $W \in (0, 1]^{n \times n}$ are then computed as:

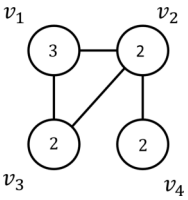
[2] Since W is symmetric, row and column sums are equal.

$$W_{i,j} := \begin{cases} \sqrt{k_{Gauss}(\mathbf{x}_i^\ell, \mathbf{x}_j^o; \beta)} & \text{if } \ell = o \\ \gamma\, k_{Gauss}(\mathbf{x}_i^\ell, \mathbf{x}_j^o; \beta) & \text{if } \ell \neq o, \end{cases} \tag{4.1}$$

where β is set to the average pairwise Euclidean distance of all $\{\mathbf{x}_i\}_{i=0}^n \subseteq \mathbb{R}^m$.

Speaking in terms of an heat distribution over the nodes, we want to inhibit the heat flow between two nodes if they do not have the same label. This is simply realized by the parameter $\gamma \in (0, 1)$ in Eq. (4.1). This *inhibition* parameter is a straightforward way to influence the affinity measure. Setting $\gamma = \varepsilon$ for $\varepsilon > 0$ will have the effect of increasing between-class distances in the target space, while $\gamma = 1 - \varepsilon$ will converge to the standard LE result.

However, similar data points should be close together on the projected space independent of their class, which is why the kernel function k is used in both cases. By this approach, both desired goals are achieved: First, samples sharing the same label are reinforced to fall into the same region. Second, data points with different labels are repelled from each other by a certain factor, but nevertheless their overall closeness is still defined by the inhibited but otherwise same kernel function.



$$\begin{bmatrix} 0.8 & 0 & 0 & 0 \\ 0 & 2.5 & 0 & 0 \\ 0 & 0 & 1.7 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} - \begin{bmatrix} 0 & 0.3 & 0.5 & 0 \\ 0.3 & 0 & 1.2 & 1 \\ 0.5 & 1.2 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} 0.8 & -0.3 & -0.5 & 0 \\ -0.3 & 2.5 & -1.2 & -1 \\ -0.5 & -1.2 & 0 & 0 \\ 0 & -1 & 0 & 1 \end{bmatrix}$$

Figure 3.1: A weighted undirected graph with $\boldsymbol{\nu} = [\psi(v_1), \psi(v_2), \psi(v_3), \psi(v_3)] = [3, 2, 2, 2]^T$, the corresponding diagonal degree matrix, weight matrix and Laplacian matrix. Multiplying $\boldsymbol{\nu}$ with the Laplacian matrix results in the negative change from one time step to another: $L\boldsymbol{\nu} = [0.8, -0.3, -0.5, 0.0]^T$.

4.1 Experiments & Results

We evaluate the performance of the proposed supervised Laplacian eigenmaps procedure using the k-nearest neighbor method. Independent of supervised or unsupervised, the expectation of different manifold learning algorithms is that similar data samples must be located close to each other in the projected low-dimensional space, given any suitable, i.e. domain-specific, distance function. Therefore, a reasonable approach is to inspect every data point and look for the label of its 1-nearest neighbor. For the evaluation of the proposed supervised Laplacian eigenmaps method, we use spectral information that was acquired by the AVIRIS sensor at the *Indian Pines* test site in Indiana–a standard dataset, that is commonly used in remote sensing research. There are all in all 16 different classes; two-thirds of the scene are composed of agriculture (corn, oats, soybean, wheat, etc.), one-thirds of forest.

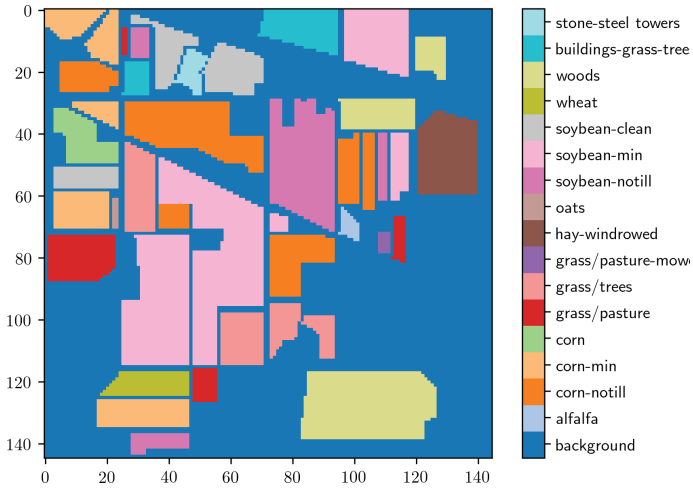


Figure 4.1: Indian pines acquired by the NASA AVIRIS sensor. The same color coding for the classes is also used for Figure 4.2.

The embedding in a low-dimensional target space depends very much of γ. Figure 4.2 depicts the embedding into the Euclidean space for two different values of $\gamma \in \{0.25, 0.75\}$ and also the standard embedding by LE. For $\gamma = 0.25$ some clusters are widely separated from each other, while for $\gamma = 0.75$ clusters are generally closer together.

All in all there are 200 spectral reflectance bands in the range of 400nm to 2500nm. As the data set consists of over twenty thousand hyperspectral pixels, it is computationally infeasible to solve the generalized eigenvalue decomposition as its complexity is $\mathcal{O}(n^3)$. Therefore, we subsample the data set, apply our method repeatedly to smaller chunks of the data and average the performance to get an overall score. We test this procedure for different values of $\gamma \in \{0.1, 0.2, \ldots, 0.9\}$.

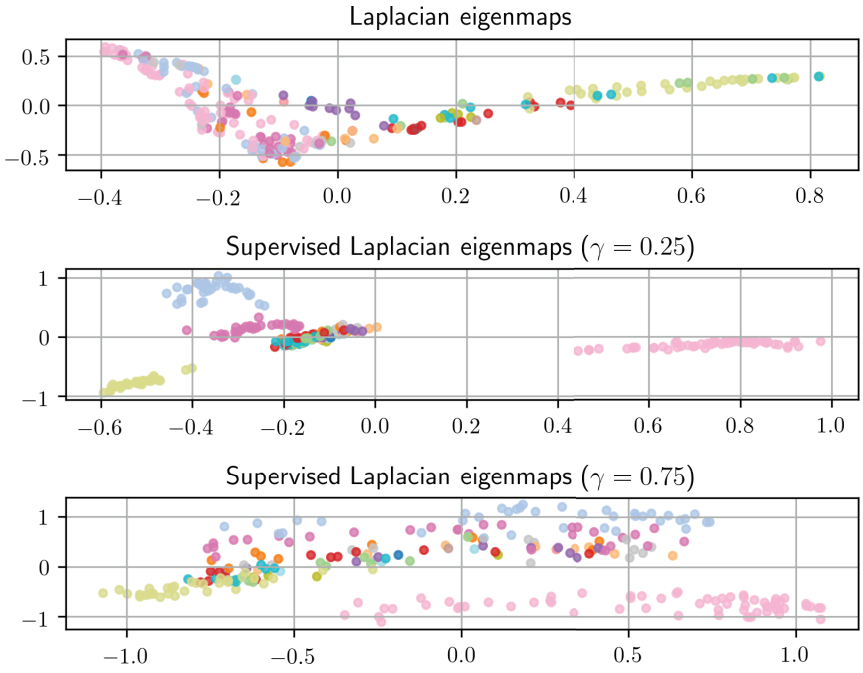


Figure 4.2: Qualitative results: 200-dimensional spectral data points are projected onto the Euclidean plane. It can be seen that in all cases spectra with the same label cluster together. However, when class label information is used in the construction of the weight matrix, same-label clusters are denser and different-label clusters are further apart. This effect is reinforced when using a small γ.

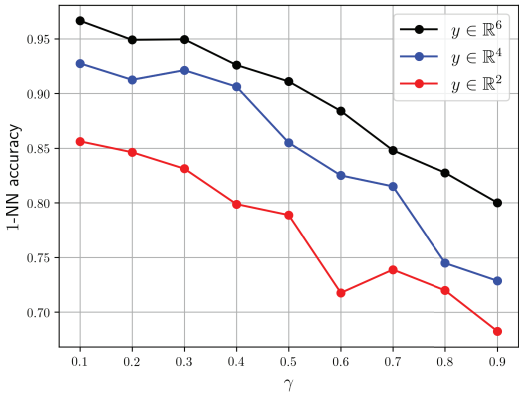


Figure 4.3: 1-nearest neighbor accuracy for different values of γ and target dimensions $\mathbb{R}^2$, $\mathbb{R}^4$ and $\mathbb{R}^6$.

Figure 4.3 shows the 1-nearest neighbor accuracy for different values of γ as well as for three different target dimensions. It is not surprising that a higher target dimension allows for a better performance. Note furthermore that γ has a large impact on the accuracy; decreasing the affinity between different-label data samples seems to ensure that the variance between the classes is larger than within.

5 Conclusion & Outlook

Nonlinear dimensionality reduction is a powerful tool for high-dimensional data analysis and visualization. We have shown an easy way to integrate class labels into the standard Laplacian eigenmaps formulation. By this it is possible to embed high-dimensional data into a low-dimensional space, while enhancing the within-class relations and extending the between-class distances.

Besides applying this method to other (hyperspectral) data sets, future work could include a way to parameterize γ and evaluate other procedures to build the affinity matrix. In this technical report, γ was used in order to decrease between-class affinity. Further research should investigate the impact of choosing γ according to the overall between-class dissimilarity of two different classes. This could

be computed using a the very same kernel function that is used to compute the affinity matrix. An important question in general is how to choose or design a kernel function k. In this technical report, we only considered the standard heat kernel. However, other affinity measures might be more suitable for spectral data. As future steps, we plan to evaluate the performance of various different kernel functions.

Bibliography

[AHK10] Sanjeev Arora, Elad Hazan, and Satyen Kale. $O(\sqrt{(logn)})$ approximation to sparsest cut in $O(n^2)$ time. *SIAM Journal on Computing*, 39(5):1748–1771, 2010.

[BN03] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

[BO95] Christoph Bregler and Stephen M Omohundro. Nonlinear manifold learning for visual speech recognition. In *iccv*, page 494. IEEE, 1995.

[CK09] Yeongjae Cheon and Daijin Kim. Natural facial expression recognition using differential-aam and manifold learning. *Pattern Recognition*, 42(7):1340–1350, 2009.

[CKM+11] Paul Christiano, Jonathan A Kelner, Aleksander Madry, Daniel A Spielman, and Shang-Hua Teng. Electrical flows, laplacian systems, and faster approximation of maximum flow in undirected graphs. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 273–282. ACM, 2011.

[LLC05] 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*, pages 249–258. ACM, 2005.

[RD12] Bogdan Raducanu and Fadi Dornaika. A supervised non-linear dimensionality reduction approach for manifold learning. *Pattern Recognition*, 45(6):2432–2444, 2012.

[RYS04] Bisser Raytchev, Ikushi Yoda, and Katsuhiko Sakaue. Head pose estimation by nonlinear manifold learning. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 4, pages 462–466. IEEE, 2004.

[SNNS02] Hiroshi Shimodaira, Ken-ichi Noma, Mitsuru Nakai, and Shigeki Sagayama. Dynamic time-alignment kernel in support vector machine. In *Advances in neural information processing systems*, pages 921–928, 2002.