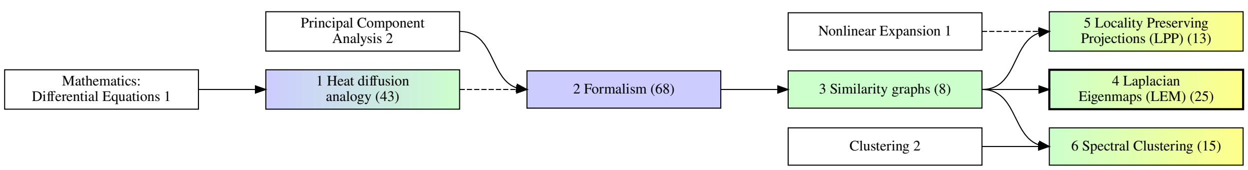
Laplacian Matrix for Dimensionality Reduction and Clustering

— Lecture Notes —

Laurenz Wiskott & Fabian Schönfeld
Institut für Neuroinformatik
Ruhr-Universität Bochum, Germany, EU

26 May 2022

— Summary —



4 Laplacian Eigenmaps (LEM) is an algorithm based on the Laplacian matrix for embedding non-vectorial data into a vector space for visualization or further processing. A toy example and an application to text analysis illustrate the algorithm.

Contents

If applicable, dark red marks core text and sometimes important formulas, a filled diamond ⧫ marks important formulas, and an empty diamond ◊ marks less important formulas or items that I would usually also present in a lecture to give context or dive into the details; core text and important formulas are worth remembering and learning for an exam; + marks sections that I would usually skip in a lecture.
You can also download the teaching material of this topic as zip files and then view them locally on your computer.

4 Laplacian Eigenmaps (LEM)

Learning objectives: The learning objective of this unit is that the student can

explain how the Laplacian eigenmaps (LEM) algorithm (Sec. 4) works.

Learning material:*

○ 5 min video 0 LEM in a Nutshell

□ 11 min video 4 Laplacian Eigenmaps

□ Text below

*Generic instruction: Consider the (possibly nested) list of resources like a horizontal tree with an invisible root on the very left, and decide from left to right what you want to select to work through. The invisible root node has to be selected. For any selected parent node all children nodes marked with ■ or ● are mandatory and have to be selected. Children nodes marked with □ or ○ are optional and may be selected in addition to get a better understanding of the material. If a parent node has no mandatory child, then at least one optional child has to be selected. Children marked with + provide additional voluntary material that can be safely ignored, typically going beyond the scope of the section. Children of non-selected parents may be ignored. ■ and □ indicate children that cover (almost) the whole material of the section. Missing content might then be indicated by struck through references to the corresponding learning objectives. Items tend to be ordered by precedence and/or recommended temporal order from top to bottom, assuming that you prefer to first watch a video before reading through lecture notes. If a detailed table of content for videos or lecture notes is given, references to learning objectives might be provided in green, 1:30 should be read as 1 min and 30 seconds, and 1'30 should be read as page 1 at about 30% of the page. Video times may be linked directly to the indicated position in the video, but be aware that the video might be downloaded anew each time you click on a time. Resources without author name are usually authored by Laurenz Wiskott and his team.

4.1 Motivation

Many algorithms work only on vectorial data and are limited in the dimensionality they can process efficiently. This causes problems if one has data that is either not vectorial, such as text, or too high dimensional, such as images, or both. If one can define a similarity function on the data, yielding a scalar similarity value for each pair of data samples, the Laplacian eigenmaps algorithm can provide a low-dimensional vectorial embedding of the data that tends to preserve similarity relationships and allows to apply other algorithms to the data that would not be applicable directly (Belkin and Niyogi, 2003). Laplacian eigenmaps are also very good for a 2- or 3-dimensional visualization of data.

For example, imagine a drone hovering through the air while equipped with a downward facing camera. Using the high dimensional pictures from its camera, we could, in theory, precisely compute the drone's current position and elevation. Unfortunately, the space of all possible high dimensional images is effectively intractable. Luckily though, we are merely interested in a small subset of this space, namely only those images the drone's camera can actually produce in a particular environment. And while each data point of this vastly smaller subset still is of the original, high dimensionality, it can be fully described by six dimensions alone: the position and orientation of the drone in 3D space. Laplacian eigenmaps can be used to find a low dimensional embedding of the images that still permits extracting positional and orientation information.

4.2 Objective

The objective of the Laplacian eigenmaps algorithm is to find an embedding of a set of I data samples (do not need to be vectors, but there must be a similarity function) **in a low-dimensional vector space $\{\boldsymbol{y}_1, ... \boldsymbol{y}_I\}$ such that samples with high similarity are close to each other in the embedding.** For dimensionality $M = 1$, i.e. an embedding in only a 1-dimensional space, this objective translates into minimizing

$$\frac{1}{2}\sum_{ij}(y_i - y_j)^2 W_{ij} \tag{4.1}$$

where the y_i are the values assigned to the samples and W_{ij} indicates the similarity between two samples. We have already seen above how this optimization problem is solved by the second eigenvector of the Laplacian matrix, (2.15) or (2.20) depending on the constraint. Each additional eigenvector adds one orthogonal (meaning the values are uncorrelated) dimension to the embedding provided by the other eigenvectors already. The quality of the embedding induced by each eigenvector is given by its associated eigenvalue, which directly relates to the actual value of sum (4.1). **The best M-dimensional embedding is** thus **given by the first M eigenvectors $\boldsymbol{u}_\alpha$ or $\boldsymbol{w}_\alpha$ of the Laplacian matrix with smallest eigenvalues** (**excluding the first one**).

Please notice that the dimension of the eigenvectors corresponds to the number I of data points, because the Laplacian matrix is $I \times I$ by construction. Thus, if you arrange the first M eigenvectors as rows in a matrix, this matrix will be $M \times I$ and the column vectors are the data points $\boldsymbol{y}_i$ in the M-dimensional embedding. For instance, three data samples embedded in a 2-dimensional space with LEM using (left) the ordinary eigenvalue problem and (right) the generalized eigenvalue problem could yield

$$\begin{pmatrix} & \boldsymbol{y}_1 & \boldsymbol{y}_2 & \boldsymbol{y}_3 \\ & \downarrow & \downarrow & \downarrow \\ \boldsymbol{u}_2 \rightarrow & -1/\sqrt{2} & 0 & +1/\sqrt{2} \\ \boldsymbol{u}_3 \rightarrow & -1/\sqrt{6} & +2/\sqrt{6} & -1/\sqrt{6} \end{pmatrix} \qquad \begin{pmatrix} & \boldsymbol{y}_1 & \boldsymbol{y}_2 & \boldsymbol{y}_3 \\ & \downarrow & \downarrow & \downarrow \\ \boldsymbol{w}_2 \rightarrow & w_{2,1} & w_{2,2} & w_{2,3} \\ \boldsymbol{w}_3 \rightarrow & w_{3,1} & w_{3,2} & w_{3,3} \end{pmatrix} \tag{4.2}$$

As usual, we have dropped $\boldsymbol{u}_1$ and $\boldsymbol{w}_1$, because they have equal components throughout, e.g. $\boldsymbol{u}_1 = (1,1,1)^T/\sqrt{3}$; $\boldsymbol{u}_2$ and $\boldsymbol{u}_3$ have zero mean, because they need to be orthogonal to $\boldsymbol{u}_1$; and $\boldsymbol{u}_2$ and $\boldsymbol{u}_3$ are orthogonal to each other as well. Analogous relations hold for $\boldsymbol{w}_\alpha$, but are numerically less intuitive.

We now have all the required components to formulate the Laplacian eigenmaps algorithm with constraints (2.21,2.22).

4.3 Algorithm

Laplacian eigenmaps algorithm (Belkin and Niyogi, 2003)

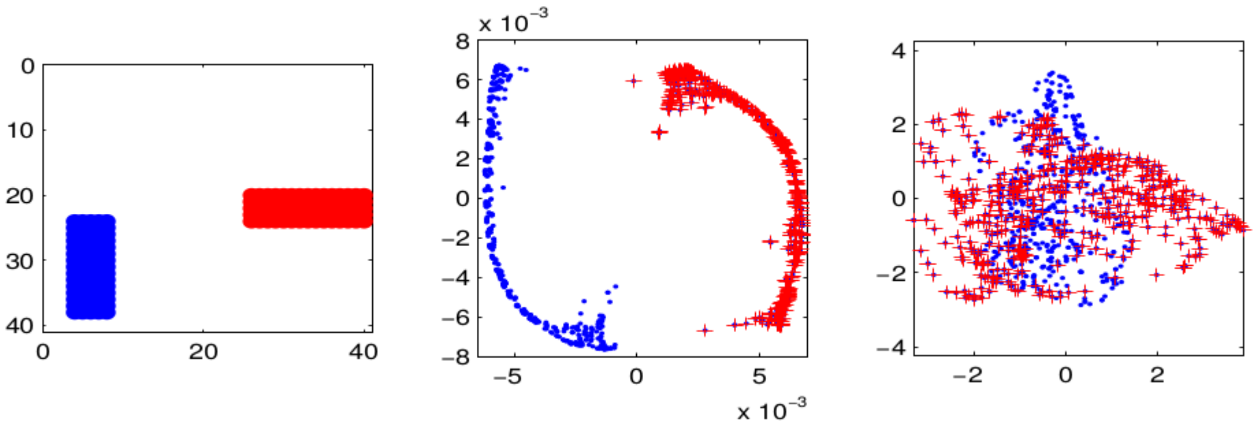
1. Given a set of I data samples, **construct a similarity graph G** according to one of the methods described in Section 3.
2. **Construct the** $I \times I$ weight matrix $\boldsymbol{W}$, degree matrix $\boldsymbol{D}$ (2.2), and **Laplacian matrix $\boldsymbol{L}$** (2.5) **for G** as described in Section 2.1.
3. **Compute the first $M+1$ eigenvectors $\boldsymbol{w}_\alpha$ of the generalized eigenvalue problem**

$$\Diamond \qquad \boldsymbol{L}\boldsymbol{w}_\alpha = \lambda_\alpha \boldsymbol{D}\boldsymbol{w}_\alpha \tag{4.3}$$

   ordered by increasing eigenvalues, see Section 2.3.
4. **An M-dimensional representation $\boldsymbol{y}_i$ of data sample i is now given by** $(w_{2,i}, ..., w_{M+1,i})^T$**,** see (4.2).

4.4 Sample Application

Figure 4.1 shows a toy example of dimensionality reduction of 1000 images of size 40×40 with either a vertical or a horizontal bar (Belkin and Niyogi, 2002). One can clearly see how the images with the horizontal bar are separate from the images with the vertical bar. It would be interesting to see a three dimensional Laplacian eigenmap, because presumably the red and blue points would each form a square manifold representing x- and y-position. A projection onto the first two principal components is shown for comparison.

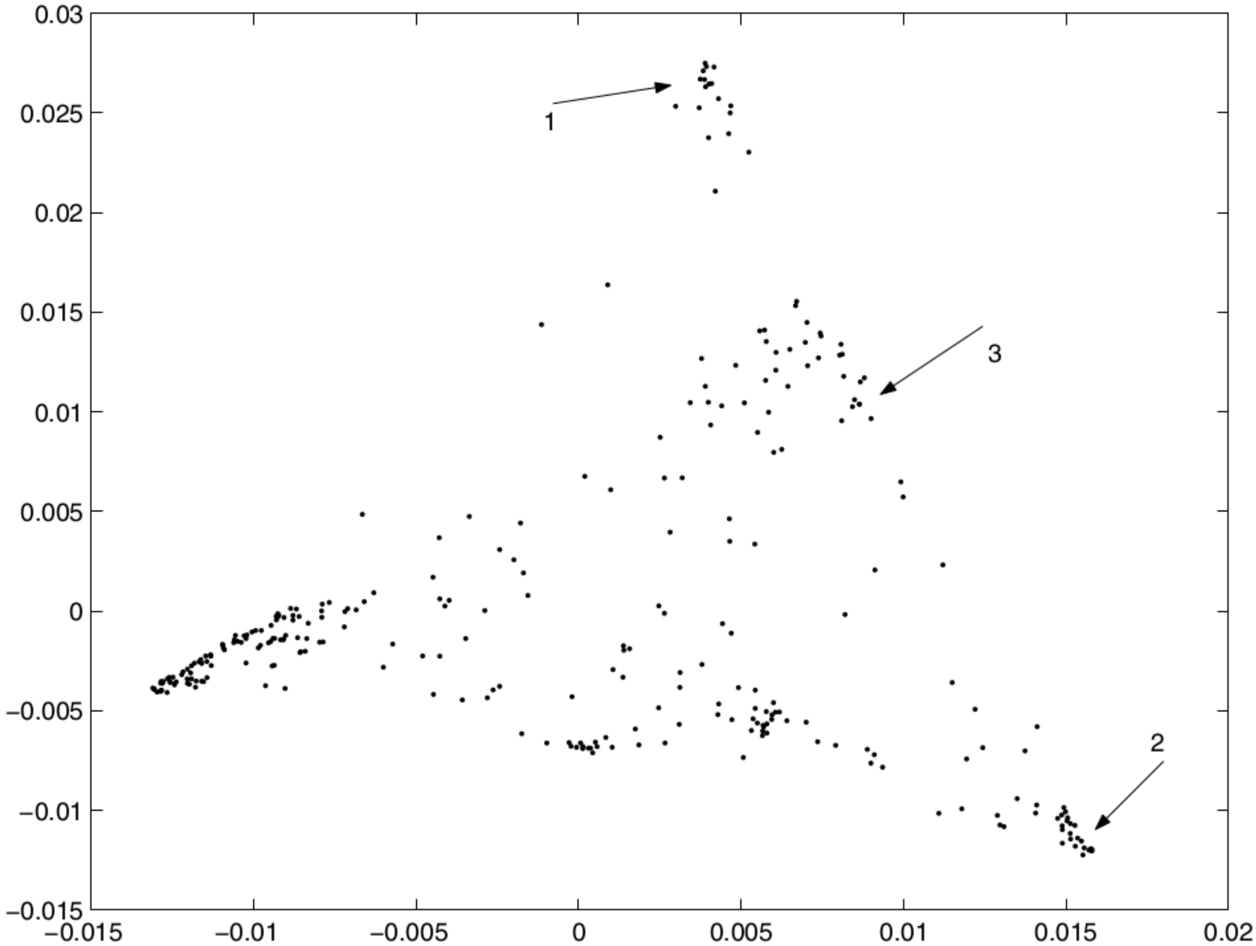


(Belkin and Niyogi, 2002, Fig. 1, URL)[4.1]

Figure 4.1: Dimensionality reduction of 1000 40×40 images with either a vertical or a horizontal bar (plotted together but distinguished by color for illustrative purposes, the original is in grayscale). Left: Two input images superimposed, one with a horizontal bar (red) one with a vertical bar (blue). Middle: Result of LEM. Right: Result of PCA for comparison.

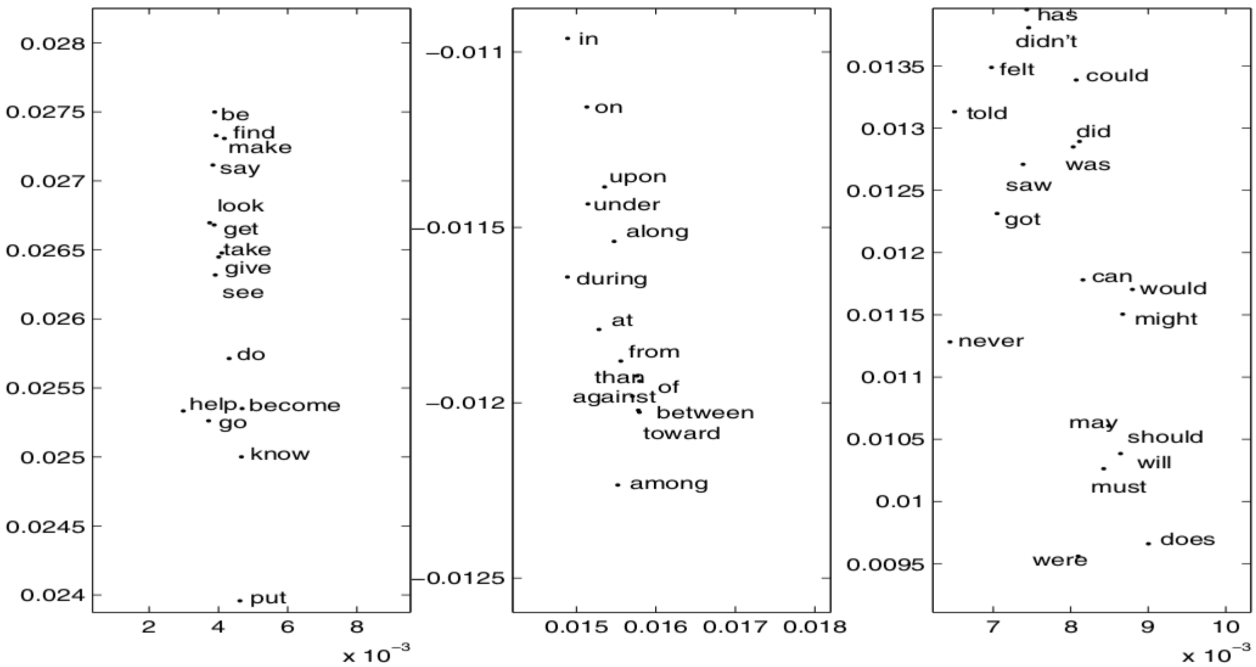
Figures 4.2 and 4.3 show an application of Laplacian eigenmaps to a set of 300 frequently used words (Belkin and Niyogi, 2003). Each word was represented by a 600-dimensional vector indicating how often any of the other words was found to the left or to the right of the considered word. Similarity was defined based on these 600-dimensional vectors. **Zooming into Figure 4.3 shows that grammatically closely related words are grouped together.**

Further reading: (Belkin and Niyogi, 2003).



(Belkin and Niyogi, 2003, Fig. 4, URL)[4.2]

Figure 4.2: Dimensionality reduction for 300 frequently used words from their word context data.



(Belkin and Niyogi, 2003, Fig. 5, URL)[4.3]

Figure 4.3: Zoom-in into the three subregions marked in Figure 4.2. Left infinitives, middle prepositions, and right mostly modal and auxiliary verbs.

Acknowledgments: We thank Jan Melchior and Merlin Schüler for valuable feedback on an earlier version of these lecture notes.

References

Belkin, M. and Niyogi, P. (2002). Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in neural information processing systems*, volume 14, pages 585–591.

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

Notes

[4.1] Belkin & Niyogi, 2002, NIPS, Fig. 1, http://papers.nips.cc/paper/1961-laplacian-eigenmaps-and-spectral-techniques-for-embedding-and-clustering.pdf

[4.2] Belkin & Niyogi, 2003, Neur. Comp., Fig. 4, https://pdfs.semanticscholar.org/989a/f45f8242b96cecb91d48b85620e7322e4aa7.pdf

[4.3] Belkin & Niyogi, 2003, Neur. Comp., Fig. 5, https://pdfs.semanticscholar.org/989a/f45f8242b96cecb91d48b85620e7322e4aa7.pdf