Seminar Report

Semi-Supervised Locally Linear Embedding: Application & Sensitivity Analysis of Critical Parameters

Department of Statistics
Ludwig-Maximilians-Universität München

By Lisa Wimmer
Under the supervision of Jann Goschenhofer, Ph.D.
Munich, April 2nd, 2021

Abstract

foo

Contents

1 Introduction **1**

2 Manifold Learning Problem **2**
2.1 Manifolds . . . 2
2.2 Formal Goal of Manifold Learning . . . 2

3 Local Graph-Based Manifold Learning (LGML) **3**
3.1 Overview . . . 3
3.2 Concept of LGML . . . 4

4 LGML Techniques **5**
4.1 Unsupervised Techniques . . . 5
4.1.1 Laplacian Eigenmaps (LEM) . . . 5
4.1.2 Locally Linear Embedding (LLE) . . . 6
4.1.3 Hessian Locally Linear Embedding (HLLE) . . . 6
4.2 Semi-Supervised Locally Linear Embedding (SSLLE) . . . 6
4.3 Particular Challenges . . . 6

5 Experimental Results **6**
5.1 Experimental Design . . . 6
5.1.1 Sensitivity Analysis . . . 6
5.1.2 Evaluation Framework . . . 6
5.1.3 Data . . . 6
5.2 Results . . . 6
5.2.1 Location of Prior Points . . . 6
5.2.2 Level of Label Noise . . . 6
5.3 Concluding Comparison . . . 6

6 Discussion **6**

7 Conclusion **6**

A Appendix **V**
A.1 Basic Concepts in Topology . . . V
A.2 Neighborhood Definitions . . . VII
A.3 Eigenanalysis . . . VIII
A.4 Generation of Synthetic Manifolds . . . IX

B Electronic Appendix **X**

List of Abbreviations

HLLE	Hessian locally linear embedding
kPCA	kernel principal component analysis
LEM	Laplacian eigenmaps
LGML	local graph-based manifold learning
LLE	locally linear embedding
PCA	principal component analysis
SSLLE	semi-supervised locally linear embedding

List of Symbols

$N \in \mathbb{N}$	number of observed data points
$D \in \mathbb{N}$	number of observed dimensions
$d \in \mathbb{N}$	number of intrinsic dimensions
$\mathcal{M} \subset \mathbb{R}^D$	d-manifold embedded in $\mathbb{R}^D$
$\mathcal{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N) \in (\mathbb{R}^D)^N$	observed coordinates
$\mathcal{Y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_N) \in (\mathbb{R}^d)^N$	embedding coordinates

List of Figures

1 S-curve manifold 2
2 Overview on selected methods of manifold learning 3
3 S-curve neighborhood graph 5
4 LGML concept 5

1 Introduction

Machine learning problems increasingly employ data of high dimensionality. While a large amount of samples is beneficial to learning, high-dimensional feature spaces, such as in speech recognition or gene processing, pose serious obstacles to the performance and convergence of most algorithms (Cayton, 2005). Three aspects strike as particularly problematic: computational complexity, interpretability of results, and geometric idiosyncrasies of high-dimensional spaces. Computational cost must be considered but is becoming less of an issue with technological evolution (Leist et al., 2009). By contrast, explainable results are increasingly in demand, but virtually inaccessible in more than a few dimensions (Doshi-Velez and Kim, 2017). The geometric aspect entails, among others, a sharp incline in the number of points required to sample spaces and a loss in meaningfulness of distances (Verleysen and Francois, 2005).

Manifold assumption. These challenges make the case for *dimensionality reduction*. Far from undue simplification, the endeavor is justified by the belief that the data-generating process is indeed of much lower dimension than is observed[1]. More formally, the data are assumed to lie on a d-dimensional *manifold*, i.e., the d-dimensional generalization of a curved surface, embedded in the D-dimensional observation space with $d \ll D$ (Cayton, 2005). A crucial property of d-manifolds is their local topological equivalence to $\mathbb{R}^d$ (Ma and Fu, 2011). It is precisely this locally Euclidean behavior that allows manifold coordinates to be mapped to $\mathbb{R}^d$ in a structure-preserving manner (Cayton, 2005). Finding this mapping constitutes an unsupervised task where models must learn the intrinsic manifold structure (Ma and Fu, 2011).

Local graph-based manifold learning (LGML). Various approaches have been proposed to retrieve points' intrinsic coordinates. A taxonomy may be found in van der Maaten et al. (2009). Many can be subsumed under the framework of *kernel principal component analysis (kPCA)*, characterizing the data by a specific matrix representation whose principal eigenvectors are used to span a d-dimensional embedding space (Ham et al., 2003). As manifolds may exhibit complicated surfaces, methods that find non-linear representations are often more successful (van der Maaten et al., 2009). LGML techniques achieve this by approximating the manifold with weighted neighborhood graphs. They pay particular heed to local environments and are thus able retrieve highly non-linear structures (Belkin and Niyogi, 2003). *Locally linear embedding (LLE)* is one of the earliest such techniques (Roweis and Saul, 2000). It is based on a rather heuristical notion of preserving local neighborhood relations. *Laplacian eigenmaps (LEM)* was proposed somewhat later on a more rigid theoretical foundation that is also extendable to LLE (Belkin and Niyogi, 2003). Both ideas are straddled by *Hessian LLE (HLLE)*, a conceptual variant of LEM algorithmically akin to LLE (Donoho and Grimes, 2003). The fully unsupervised functionality of these methods offers a drawback: they may fail to find an embedding that has an actual reflection in the real-life setting. Therefore, Yang et al. (2006) incorporate prior information in *semi-supervised LLE (SSLLE)* to produce more meaningful embeddings[2].

Outline. Indeed, their results indicate considerable success of SSLLE. It is the aim of this work to (1) reproduce these results, creating an open-source implementation, and (2) to assess its performance under different parameter settings. The remainder of the report is organized as follows: first, the problem of manifold learning is formalized. The subsequent chapters sketch the idea of LGML and lay out the above named unsupervised techniques and SSLLE in more detail. Afterwards, the results of the conducted experiments are presented, before the report concludes with a brief discussion.

[1] Consider, for example, image data of objects in different poses. Such data are typically stored in large pixel representations, yet it is reasonable to suppose the true sources of variability are few.

[2] Note that this is rather different from general semi-supervised learning: SSLLE supports an inherently unsupervised task by some labeled data points. Alternative proposals for a semi-supervised LLE have been made, e.g., by Zhang and Chau (2009), that build upon a fully supervised LLE (de Ridder and Duin, 2002).

2 Manifold Learning Problem

2.1 Manifolds

Before diving into the core concepts, some basic notation shall be fixed. A thorough introduction to manifold theory is beyond the scope, but section A.1 of the appendix provides some fundamental definitions for to make clear how these are understood in the remainder of this report.

Manifolds. A d-dimensional *manifold* $\mathcal{M} \subset \mathbb{R}^D$ is a topological space with some additional properties. $\mathcal{M}$ is most easily imagined as the d-dimensional generalization of a curved surface that behaves locally Euclidean, i.e., is locally homeomorphic to an open subset of $\mathbb{R}^d$ (Ma and Fu (2011); please refer to the appendix for a more rigorous derivation). Consider, for instance, the *S-curve* manifold (figure 1), embedded in $\mathbb{R}^3$, that will serve as a running example throughout the report. Clearly, the S-curve as a whole is far from linear, but it locally homeomorphic to $\mathbb{R}^2$ and thus intrinsically two-dimensional. In fact, it is generated from a planar patch of two-dimensional points by some trigonometric transformations.

Figure 1: 1,000 points sampled from the S-curve. *Source:* own representation.

Geodesic distance. Euclidean distance is not meaningful on general manifolds. Rather than measuring "shortcuts" between points across $\mathbb{R}^D$ (where, for instance, points in the red upper part of figure 1 would be considered deceptively close to the cyan mid area), it seems reasonable to constrain distances to the manifold surface. Put simply, *geodesic distance* between two points on $\mathcal{M}$ is the length of the shortest curve (*geodesic*) on $\mathcal{M}$. Intuitively, geodesic distance can be identified with Euclidean distance in Euclidean spaces where shortest curves are just straight lines (Ma and Fu, 2011).

2.2 Formal Goal of Manifold Learning

The manifold learning situation might be summarized as follows: data are observed in $\mathbb{R}^D$ but assumed to be really samples from a d-manifold $\mathcal{M}$ embedded in $\mathbb{R}^D$, meaning they can be interpreted in $\mathbb{R}^d$ if a faithful translation between $\mathcal{M}$ and $\mathbb{R}^d$ is found[3]. The challenge is thus to unravel the manifold in a maximally structure-preserving way (Saul et al., 2006). This goal may be formalized as follows, inspired by Cayton (2005) and Saul et al. (2006):

Given. Data $\mathcal{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N)$, with $\boldsymbol{x}_i \in \mathbb{R}^D \ \forall i \in \{1, 2, ..., N\}$ and $N, D \in \mathbb{N}$.
The true data-generating process is taken to have dimensionality $\mathbb{N} \ni d \ll D$, such that $\mathcal{X}$ is in fact a sample from a smooth, connected d-manifold with $\mathcal{X} \sim \mathcal{M} \subset \mathbb{R}^D$. $\mathcal{M}$ may be described by a single coordinate chart $\psi : \mathcal{M} \to \mathbb{R}^d$. For manifold learning methods to yield satisfying results, $\mathcal{M}$ is always assumed to be sampled well by $\mathcal{X}$.

Goal. Find the d-dimensional representation of the data, i.e., compute
$\mathcal{Y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_N)$, where $\boldsymbol{y}_i = \psi(\boldsymbol{x}_i) \in \mathbb{R}^d \ \forall i \in \{1, 2, ..., N\}$.
The map ψ itself is not always explicitly retrieved.

Note that, while D is given a priori, the intrinsic dimensionality d is often unknown in real-life applications. $\mathcal{Y}$ must therefore be expected to differ from the true coordinates and, in particular, to even have incorrect dimension (Saul et al., 2006). Notwithstanding this potential gap, solutions of the subsequently presented methods will be denoted by $\mathcal{Y} \in (\mathbb{R}^d)^N$ to avoid overloading notation.

[3]It is actually a simplification to assume all data to lie *on* $\mathcal{M}$, but the more general case of data lying *near* $\mathcal{M}$ is rarely considered explicitly.

3 Local Graph-Based Manifold Learning (LGML)

3.1 Overview

In the following, it shall be laid out how the manifold learning problem is approached by LLE as the conceptual parent of SSLLE (the incorporation of prior information is a rather different matter; aside from this, the functionalities of SSLLE and LLE are identical). Much of the theoretical foundation for LLE has been discussed only in later work. In order to provide a more integrated background, explanations will therefore be given in a broader context. LEM in particular provides much of the mathematical framework the original proposal of LLE lacks, and HLLE emerges as a combination of both ideas. All three may be viewed as instances of LGML.

Taxonomy. LGML arises from a variety of geometric intuitions and computational implementations. Nonetheless, methods share common structures that allow for interpretation in a more abstract framework[4] (Bengio et al. (2003), Bengio et al. (2004)). Figure 2 depicts a schematic overview on the models studied here. All of these belong to the realm of *spectral* models. The non-spectral group includes, among others, techniques based on neural networks and is not discussed here (van der Maaten et al., 2009).

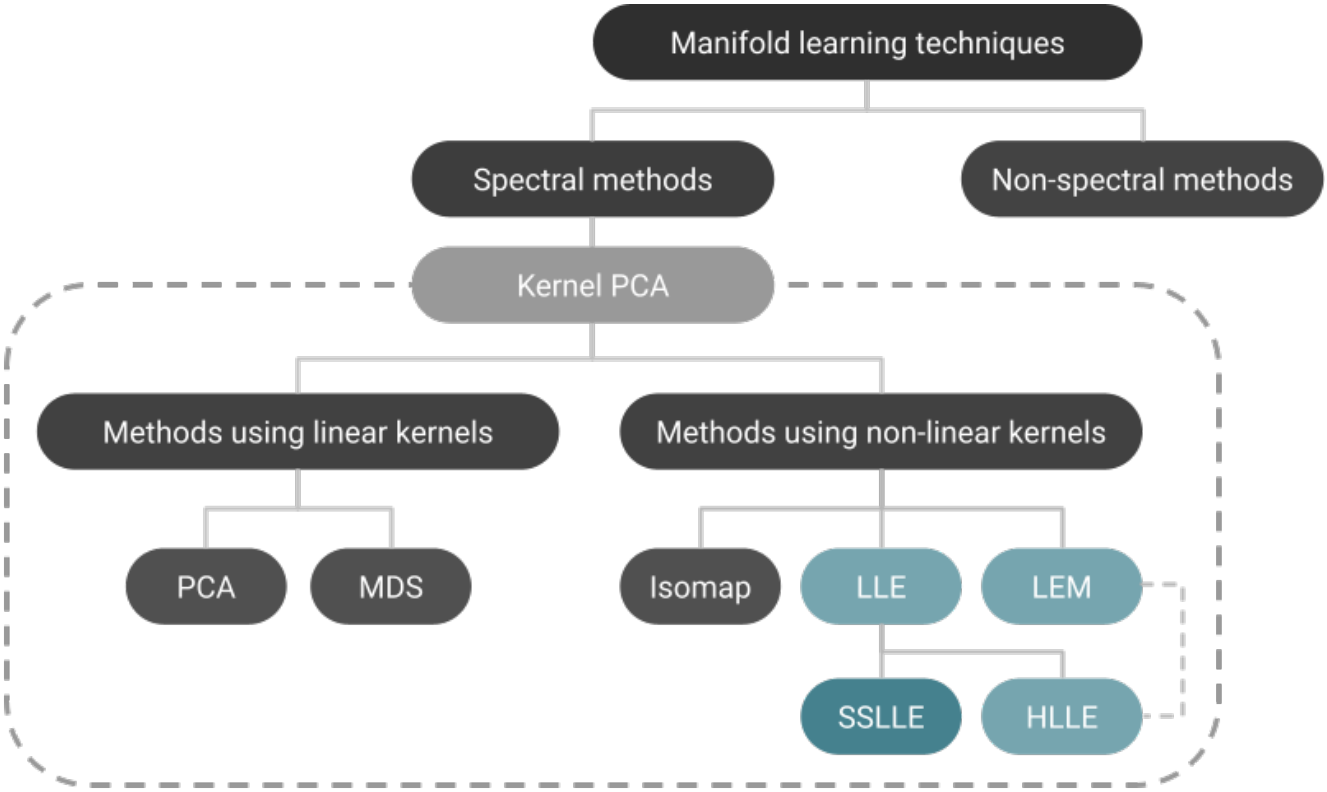

Figure 2: Overview on selected methods of manifold learning. *Source:* own representation, inspired by a similar example in van der Maaten et al. (2009) and re-interpreted with the findings in Bengio et al. (2004).

kPCA. As indicated in figure 2, LGML may be viewed in the light of *kernel principal component analysis (kPCA)*. kPCA was actually proposed earlier and later shown to link the other concepts by a unified idea (Ham et al. (2003). It provides a useful general intuition to manifold learning and subsumes the other methods in a way beneficial to the important task of out-of-sample extension (Bengio et al., 2004). kPCA builds upon two fundamental concepts in machine learning: it performs *principal component analysis (PCA)* on data transformed by the *kernel trick*. First, features of interest are extracted from the data by kernelization and coerced to a matrix representation.

[4]It should be noted that such a framework might be established from several angles; after all, the different approaches attempt to solve the same problem and can thus be translated into one another in various ways.

These are taken to capture the intrinsic data structure and may therefore be understood as an approximation to the latent manifold properties. Second, PCA finds the principal axes along which these intrinsic properties vary by eigenanalysis, yielding the desired reduction in dimensionality through preserving the most relevant latent dimensions (Schölkopf et al., 1998).

3.2 Concept of LGML

If kPCA sounds like a powerful concept, the crux of course lies in finding an appropriate kernel function. The nature of the feature map applied to the input data determines the kind of mapping that may be learned and serves to distinguish the various techniques. Methods using linear kernels, such as standard PCA, suffer from the confinement to finding linear embedding spaces (van der Maaten et al., 2009). As $\mathcal{X}$ must generally be assumed to lie on a non-linear manifold, kernelization is usually performed with non-linear feature maps (Schölkopf et al., 1998). Conceivably, there is no obvious way to arrive at such a mapping. *Graph-based* models therefore approach the problem from an alternative angle. In fact, they do not perform kernelization explicitly[5] but build on a different intuition.

Graph approximation. All LGML methods fundamentally rely on graph approximations of the manifold surface. These graphs are discretized models of the manifold and as such, in principle, able to capture arbitrarily structures. Distances may be then measured along the approximated manifold surface rather than in the ambient Euclidean space, effectively enabling non-linearity (Saul et al., 2006). Besides non-linearity, a second desideratum in manifold learning is the ability to treat possibly non-convex manifolds with sufficiently local focus. Non-convexity means $\mathcal{M}$ is not isometric to a convex subset of Euclidean space (Donoho and Grimes, 2003). Intuitively, such behavior requires careful tracing of the manifold surface. LGML methods therefore focus primarily local manifold properties[6] (Cayton, 2005).

Local neighborhoods. Graph approximations are constructed from neighborhood relations in the high-dimensional observation space, where neighborhoods are typically taken to be k-neighborhoods, i.e., based on a fixed number of neighbors. In principle, it is equally possible to restrict neighborhoods to a maximum distance. However, k-neighborhoods are often more easily specified due to the inherent scale invariance of k, and has attracted rather more attention in general research[7] (He et al., 2005). For a formal definition of both k- and ϵ-neighborhoods, see section A.2. Neighborhood size represents a crucial parameter in LGML. It reflects beliefs about the topological structure of $\mathcal{M}$ – smaller neighborhoods corresponding to a higher degree of non-linearity – and may affect performance rather strongly (Sudderth, 2002). Chapter 5 will discuss how this trade-off is addressed in practice.

[5] Explicit kernels may still be derived for all methods but as their illustrative ability is rather limited, this is not covered here. For the kernel perspective see for example Bengio et al. (2004) and Weinberger et al. (2004).

[6] As opposed to, for example, *Isomap* (Tenenbaum et al., 2000), one of the earliest and most prominent examples of global manifold learning. Isomap's central assumptions are global isometry and convexity of the parameter space (Tenenbaum et al., 2000). While it yields good results in many applications, Isomap does not sufficiently account for the curvature of strongly non-convex manifolds. In order to avoid this drawback, local methods limit isometry to only hold between neighboring points and relax the parameter space condition to open, connected subspaces (Donoho and Grimes, 2003).

[7] Yet, Tenenbaum et al. (2000) note that, when local dimensionality is not constant across the observed data, ϵ-neighborhoods might provide more reliable results.

Graph construction. $\mathcal{M}$ may be approximated by a *neighborhood graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, always assuming it is sampled well by $\mathcal{X}$. Inputs $\boldsymbol{x} \in \mathcal{X}$ form vertices $\mathcal{V}$ and edges $\mathcal{E}$ indicate neighborhood relations (Belkin and Niyogi, 2001). Each vertex is connected to its k nearest neighbors (or all points within ϵ-radius). It is easy to see that k-neighborhoods are an asymmetric notion and therefore lead to directed graphs. Conversely, the ϵ-distance boundary holds in both directions and produces undirected graphs (He et al., 2005). Figure 3 shows how a neighborhood graph may be used to approximate the S-curve manifold. It was built using k-neighborhoods with $k = 3$. Note that neighborhood construction solely relies on the observed data, not requiring any information about the intrinsic structure. For a densely sampled set of points, the graph representation should yield a fairly good approximation of the manifold surface.

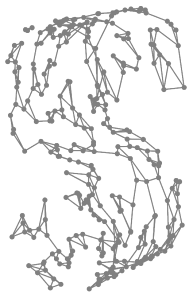

Figure 3: k-neighborhood graph for 300 points sampled from the S-curve with $k = 3$. *Source:* own representation.

Eigenanalysis. Eventually, spectral manifold learning boils down to eigenanalysis of a matrix derived from the graph approximation. This matrix representation is obtained by application of some graph functional. Precisely how the functional is defined reflects different beliefs about the intrinsic manifold structure and defines the core of each LGML method (Saul et al., 2006). The d principal (top or bottom) eigenvectors – as determined by the associated eigenvalues – span a subspace into which the data may be projected under minimal loss of information, preserving as much variability as possible along the axes of intrinsic structure (for a formal definition of eigenanalysis and generalized eigenvalue problems, see section A.3). The nature of different matrix representations across methods shall be made clear in the subsequent chapters. Figure 4 concludes the overview on LGML with a summary of the fundamental concept.

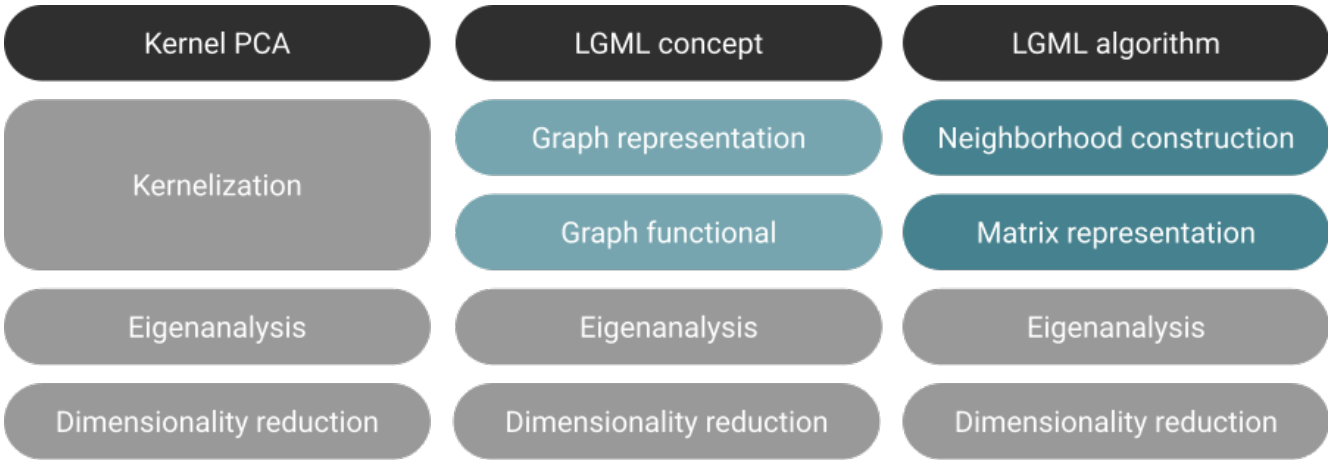

Figure 4: Concept of LGML in the kPCA framework. *Source:* own representation.

4 LGML Techniques

4.1 Unsupervised Techniques

4.1.1 Laplacian Eigenmaps (LEM)

The reason for LEM to appear in this report alongside the LLE family is its underlying theory both providing a foundation for LLE (Belkin and Niyogi, 2003), which was originally proposed lacking such, and closely relating to the theoretical concepts in HLLE (Donoho and Grimes, 2003). LEM are centered around the preservation of locality, i.e., mapping nearby inputs to nearby outputs. Locality is enforced via the *Laplace-Beltrami operator* defined on smooth, compact man-

ifolds, and operationalized by means of the *graph Laplacian* acting as a discrete approximator (Belkin and Niyogi, 2003). This idea is best understood recalling that the similarity of outputs for similar inputs is essentially a notion of smoothness and can thus be controlled by a size constraint on the gradient of the mapping function.

4.1.2 Locally Linear Embedding (LLE)

4.1.3 Hessian Locally Linear Embedding (HLLE)

4.2 Semi-Supervised Locally Linear Embedding (SSLLE)

4.3 Particular Challenges

5 Experimental Results

5.1 Experimental Design

5.1.1 Sensitivity Analysis

5.1.2 Evaluation Framework

5.1.3 Data

5.2 Results

5.2.1 Location of Prior Points

5.2.2 Level of Label Noise

5.3 Concluding Comparison

6 Discussion

Pros and Cons
Various extensions
See (van der Maaten et al., 2009) for extensive discussion of manifold learning
Theoretical convergence? (e.g., ISOMAP has this)
Determination of d: actually requires to know d, right? Must be automatically known if prior points are known
Potential shortcoming: what if manifold is not well-sampled? Not a problem with synthetic data, but IRL. But probably problematic with all manifold approaches
This is directly related to the COD – local methods require dense sampling (van der Maaten et al., 2009)
Also: generalization to new points (w/o recomputing everything) neighborhood-preserving propositions –> fundamental problem: except for prior points, it is deterministic (as opposed to generative approaches, such as autoencoders)

7 Conclusion

Lorem ipsum

A Appendix

A.1 Basic Concepts in Topology

This section contains definitions of the main geometric concepts considered above. Obviously, the list is by no means extensive; manifold theory is presented much more in detail (and mathematical rigor) in, for example, McCleary (2006) or Waldmann (2014).

Topological spaces. A *topological space* is constituted by a set T equipped with a *topology* $\mathcal{T}$. A topology is a general way of describing relations between elements in T. Consider a function $\mathcal{T} : T \to 2^T, t \mapsto \mathcal{T}(t)$, which assigns to $t \in T$ a set of subsets of T called *neighborhoods*. For $\mathcal{T}$ to be a topology[8] on T, the following properties must hold (Brown, 2006):

(T1) If $\mathcal{T}$ is a neighborhood of t, then $t \in \mathcal{T}$.
(T2) If $\mathcal{T}$ is a subset of T containing a neighborhood of t, then $\mathcal{T}$ is a neighborhood of t.
(T3) The intersection of two neighborhoods of t is again a neighborhood of t.
(T4) Any neighborhood $\mathcal{T}$ of t contains a neighborhood $\mathcal{T}'$ of t such that $\mathcal{T}$ is a neighborhood of each element in $\mathcal{T}'$.

Note that, in this general definition, neighborhoods are based on an abstract notion of "nearness". Learning the structure of a topological space effectively boils down to learning neighborhood relations. In Euclidean topological space these are directly based on distance: neighborhoods around t are constructed by ϵ-balls containing all elements within a Euclidean distance of ϵ from t. The resulting topology is also called the *metric topology* (McCleary, 2006).
Topological spaces in general are not accessible via distances (or angles, for that matter) known from Euclidean spaces. The ultimate goal in manifold learning therefore is the interpretation of the data in a space that is again Euclidean, albeit of lower dimensionality, where such concepts are meaningful.

Homeomorphisms. Consider two topological spaces $(S, \mathcal{T}_S)$, $(T, \mathcal{T}_T)$ (denoted by the respective shorthands S, T from here) and a mapping function $f : S \to T$. If f is bijective and continuous and $f^{-1} : T \to S$ is also continuous, f is called a *homeomorphism* (Brown, 2006). Topological spaces for which such a mapping exists are *homeomorphic* to each other. Any properties of S that T shares when it is homeomorphic to S are referred to as topological properties. Two homeomorphic spaces are thus topologically equivalent (McCleary, 2006).

If there exists a non-negative integer d such that for every s in a topological space S a local neighborhood $U \ni s$, $U \subset S$, is homeomorphic to an open subset of $\mathbb{R}^d$ (sometimes called *parameter space*), S is *locally Euclidean*[9] (Ma and Fu, 2011). In other words, there is a homeomorphism $f : U \to \mathbb{R}^d$ for every element in S. The neighborhoods U are also referred to as *coordinate patches* and the associated maps f are called *coordinate charts* (Cayton, 2005). In local neighborhoods S then behaves like $\mathbb{R}^d$ (Ma and Fu, 2011).

Manifolds. *Manifolds* are now precisely such locally Euclidean topological spaces, with some additional properties. For a topological space $\mathcal{M}$ to be a d-dimensional manifold[10] (also: d-manifold) it must meet the following conditions (Waldmann, 2014):

(M1) $\mathcal{M}$ is Hausdorff.
(M2) $\mathcal{M}$ is second-countable.
(M3) $\mathcal{M}$ is locally homeomorphic to $\mathbb{R}^d$.

[8] Alternative definitions employ open subsets of T, see for example Waldmann (2014).
[9] For locally Euclidean topological spaces it is thus meaningful to speak of elements as points.
[10] $\mathcal{M}$ is again a shorthand, omitting the explicit notation of the corresponding topology.

The Hausdorff condition is a separation property and ensures that for any two distinct points from $\mathcal{M}$ disjoint neighborhoods can be found (Brown, 2006). Second-countability restricts the manifold's size via the number of open sets it may possess (Waldmann, 2014).

Embeddings. Recall that the data are observed in $\mathbb{R}^D$ but taken to lie on $\mathcal{M}$, locally homeomorphic to $\mathbb{R}^d$. This implies the assumption $\mathcal{M} \subset \mathbb{R}^D$ and $\mathcal{M}$ is said to be *embedded* in the ambient D-dimensional Euclidean space (Cayton, 2005). The associated *embedding* is but a map $f : \mathcal{M} \to \mathbb{R}^D$ whose restriction to $\mathcal{M}$ is a homeomorphism (Brown, 2006), or, more specifically, the canonical inclusion map identifying points on the manifold as particular points of $\mathbb{R}^D$ (Waldmann, 2014). It can be shown that $K = 2d + 1$ is sufficient to create an embedding (Ma and Fu, 2011).

Geodesics. In order to enable the construction of a meaningful distance metric, manifolds must fulfill two additional properties: *smoothness*[11] and *connectedness*[12] (Ma and Fu, 2011). For smooth, connected manifolds, *geodesic distance* is the length of the shortest curve (*geodesic*) on $\mathcal{M}$ between two points on $\mathcal{M}$. A curve c in $\mathcal{M}$ is a smooth mapping from an open interval $\Lambda \subset \mathbb{R}$ into $\mathcal{M}$. c is parametrized by a point $\lambda \in \Lambda$, such that

$$c(\lambda) = (c_1(\lambda), ..., c_d(\lambda))^T \tag{1}$$

is a curve in $\mathbb{R}^d$ (all $c_j, j = 1, ..., d$ having a sufficient number of continuous derivatives). Componentwise differentiation with respect to λ yields *velocity* in λ:

$$c'(\lambda) = (c_1'(\lambda), ..., c_d'(\lambda))^T. \tag{2}$$

The *speed* of c is given by $\|c'(\lambda)\|_2^2$, where $\|\cdot\|^2$ denotes the square norm. Distance along this curve is measured by the arc-length

$$L(c) = \int_{\boldsymbol{p}}^{\boldsymbol{q}} \|c'(\lambda)\|^2 \, d\lambda.$$

Eventually, geodesic distance can be derived as the length of the shortest such curve, out of the set $\mathcal{C}(\boldsymbol{p}, \boldsymbol{q})$ of differentiable curves in $\mathcal{M}$ that connect $\boldsymbol{p}$ and $\boldsymbol{q}$ (Ma and Fu, 2011):

$$d^{\mathcal{M}}(\boldsymbol{p}, \boldsymbol{q}) = \inf_{c \in \mathcal{C}(\boldsymbol{p}, \boldsymbol{q})} L(c). \tag{3}$$

[11] The smoothness property is based on differentiability of coordinate charts and ensures that concepts of curvature, length and angle remain meaningful (Ma and Fu, 2011). A detailed derivation may be found, for example, in Mukherjee (2015).

[12] Connectedness means that no separation $\{U, V\}$ of a manifold $\mathcal{M}$ exists with open, non-empty and disjoint $U, V \subset \mathcal{M}$, $\mathcal{M} = U \cup V$. This may be loosely put as paths linking arbitrary pairs of manifold points (McCleary, 2006).

A.2 Neighborhood Definitions

A neighborhood of $\boldsymbol{x} \in \mathcal{X}$ is a subset of $\mathcal{X}$ containing another, open subset of $\mathcal{X}$ of which $\boldsymbol{x}$ is an element. Members of the neighborhood are called neighbors of $\boldsymbol{x}$. In metric spaces neighborhoods are defined via distances and therefore translate to open balls around each point (Waldmann, 2014). This distance-based construction locally applies to manifolds as a direct consequence of their local isometry to the Euclidean observation space (Ma and Fu, 2011). There are two principal ways to build a neighborhood around $\boldsymbol{x} \in \mathcal{X}$, both of which usually employ the squared Euclidean norm[13] $\|\cdot\|^2$. Let $\mathcal{N} : \mathcal{X} \to \mathcal{X}^\ell, \boldsymbol{x} \mapsto \mathcal{N}(\boldsymbol{x})$ be a constructor that assigns a set of neighbors to $\boldsymbol{x}$. The first possibility is to restrict the size of the neighborhood to the k points[14] with the smallest distance to $\boldsymbol{x}$, such that $\ell = k$ and

$$\mathcal{N}_k(\boldsymbol{x}) = \{\boldsymbol{x}_j \in \mathcal{X} : \|\boldsymbol{x} - \boldsymbol{x}_j\|^2 \leq d_{(k)}\}, \tag{4}$$

with $d_{(k)} \in \mathbb{R}$ being the k-th instance of ordered pairwise distances between $\boldsymbol{x}$ and all other points. Alternatively, the neighborhood may be constructed by collecting all points that have a maximum distance of $\epsilon \in \mathbb{R}$ to $\boldsymbol{x}$, yielding

$$\mathcal{N}_\epsilon(\boldsymbol{x}) = \{\boldsymbol{x}_j \in \mathcal{X} : \|\boldsymbol{x} - \boldsymbol{x}_j\|^2 \leq \epsilon\} \tag{5}$$

and $\ell = |\mathcal{N}_\epsilon(\boldsymbol{x})|$ (He et al., 2005).

[13] In principle, alternative metrics are applicable, for instance such that measure angles (Belkin and Niyogi, 2004).

[14] In presence of ties in pairwise distances k may vary across the data, but with zero probability in continuous feature spaces.

A.3 Eigenanalysis

Eigenanalysis yields the fundamental information stored in a matrix $\boldsymbol{A}$: its principal eigenvectors span a subspace into which the data may be projected under minimal loss of information (with respect to quadratic loss). Figure 5 depicts the idea of eigenanalysis in a schematic way.

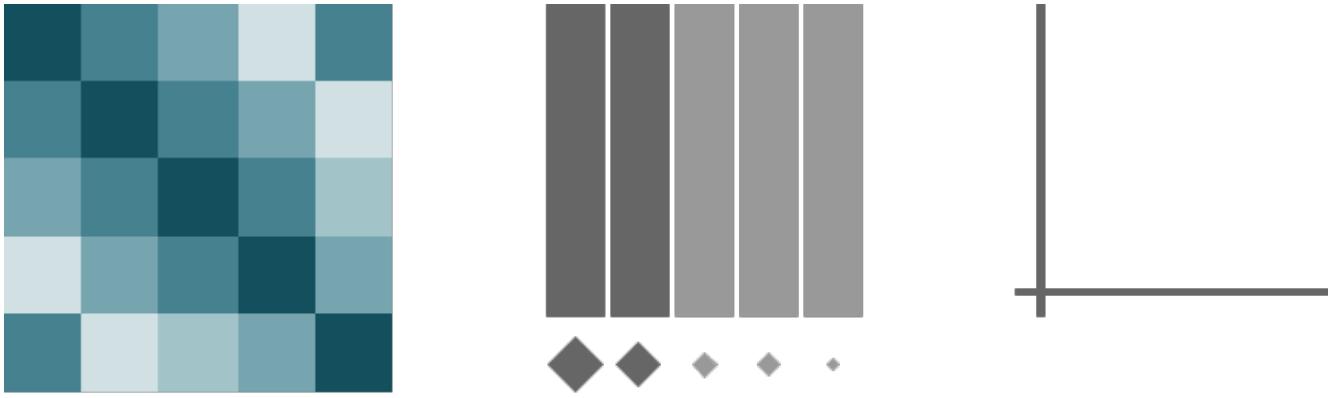

Figure 5: Conceptual idea of eigenanalysis. Eigenvectors of a matrix (*left*) point in the direction of greatest variability (*middle*), the degree of which is measured by the associated eigenvalues depicted as rhombi. Retaining the thus determined principal d eigenvectors allows to span a linear subspace of reduced dimensionality (*right*). *Source:* own representation.

Eigenvectors and eigenvalues. Formally, eigenanalysis is the decomposition of a square matrix into pairs of *eigenvectors* and *eigenvalues*. Let $\boldsymbol{A} \in \mathbb{R}^{N \times N}$ be a square matrix and $\lambda \in \mathbb{R}$ a scalar value. λ is said to be an eigenvalue to $\boldsymbol{A}$ if there exists $\boldsymbol{v} \in \mathbb{R}^N \setminus \{0\}$ such that $\boldsymbol{Av} = \lambda \boldsymbol{v}$. Then, $\boldsymbol{v}$ is the eigenvector corresponding to the eigenvalue λ, and their tuple is also called an *eigenpair*.

Null spaces. A closely related notion is that of the *null space*, consisting of the vectors that map $\boldsymbol{A}$ to 0 upon multiplication from the right: $\{\boldsymbol{v} \in \mathbb{R}^N : \boldsymbol{Av} = 0\}$. It can be easily seen that the null space consists of those eigenvectors of $\boldsymbol{A}$ that are associated with an eigenvalue of zero, and the zero vector itself. For a specific eigenvalue λ of $\boldsymbol{A}$, the null space of $\lambda \boldsymbol{I} - \boldsymbol{A}$ (with $\boldsymbol{I}$ the N-dimensional identity matrix) constitutes the *eigenspace* of $\boldsymbol{A}$ (Börm and Mehl, 2012).

Generalized eigenvalue problems. Eigendecomposition of a matrix $\boldsymbol{A}$ can be framed as the solution of a generalized eigenvalue problem. Generalized eigenvalue problems are posed subject to a constraint on a second, also symmetric matrix $\boldsymbol{B} \in \mathbb{R}^{N \times N}$. As the standard eigenvalue problem results immediately from $\boldsymbol{B} = \boldsymbol{I}$, the generalized form subsumes both cases. It is given by

$$\boldsymbol{AV} = \boldsymbol{BV\Lambda}, \tag{6}$$

where $\boldsymbol{V} = [\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_N] \in \mathbb{R}^{N \times N}$ is the matrix of eigenvectors of A and $\boldsymbol{\Lambda} = diag[\lambda_1, \lambda_2, ..., \lambda_N]^T \in \mathbb{R}^{N \times N}$ is the diagonal matrix of the associated eigenvalues (ordered from smallest to largest). The generalized eigenvalue problem may be stated equivalently as

$$\min_{\boldsymbol{V}} trace(\boldsymbol{V}^T \boldsymbol{AV}), \quad \text{s.t.} \quad \boldsymbol{V}^T \boldsymbol{BV} = \boldsymbol{I}, \tag{7}$$

and translated to the first form by means of a Lagrangian multiplier (Ghojogh et al., 2019).

A.4 Generation of Synthetic Manifolds

This section documents how the synthetic manifolds considered in the report may be generated.

S-curve.
Swiss roll.
Incomplete tire.
World data.

B Electronic Appendix

Data, code and figures are provided in electronic form.

References

Belkin, M. and Niyogi, P. (2001). Laplacian eigenmaps and spectral technique for embedding and clustering, *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, p. 585–591.

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

Belkin, M. and Niyogi, P. (2004). Semi-supervised learning on riemannian manifolds, *Machine Learning* **56**: 209–239.

Bengio, Y., Delalleau, O., Roux, N. L., Paiement, J.-F., Vincent, P. and Ouimet, M. (2004). Learning eigenfunction links spectral embedding and kernel pca, *Neural Computation* **16**: 2197–2219.

Bengio, Y., Paiement, J.-F., Vincent, P., Delalleau, O., Roux, N. L. and Ouimet, M. (2003). Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering, *Proceedings of the 16th International Conference on Neural Information Processing Systems*, MIT Press, p. 177–184.

Brown, R. (2006). *Topology and Groupoids. A Geometric Account of General Topology, Homotopy Types and the Fundamental Groupoid*, 2 edn, Createspace.

Börm, S. and Mehl, C. (2012). *Numerical Methods for Eigenvalue Problems*, De Gruyter.

Cayton, L. (2005). Algorithms for manifold learning, *Technical Report CS2008-0923*, University of California, San Diego (UCSD).

de Ridder, D. and Duin, R. P. (2002). Locally linear embedding for classification, *Technical Report PH-2002-01*, Delft University of Technology, Delft, The Netherlands.

Donoho, D. L. and Grimes, C. (2003). Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data, *Proceedings of the National Academy of Sciences of the United States of America* **100**(10): 5591–5596.

Doshi-Velez, F. and Kim, B. (2017). Towards a rigorous science of interpretable machine learning, *arXiv: Machine Learning* .

Ghojogh, B., Karray, F. and Crowley, M. (2019). Eigenvalue and generalized eigenvalue problems: Tutorial.

Ham, J., Lee, D. D., Mika, S. and Schölkopf, B. (2003). A kernel view of the dimensionality reduction of manifolds, *Technical Report TR-110*, Max-Planck-Institute for Biological Cybernetics.

He, X., Cai, D., Yan, S. and Zhang, H.-J. (2005). Neighborhood preserving embedding, *Proceedings of the Tenth IEEE International Conference on Computer Vision*.

Leist, A., Playne, D. P. and Hawick, K. A. (2009). Exploiting graphical processing units for data-parallel scientific applications, *Concurrency and Computation. Practice and Experience* **21**(18): 2400–2437.

Ma, Y. and Fu, Y. (2011). *Manifold Learning. Theory and Applications*, CRC Press.

McCleary, J. (2006). *A First Course in Topology. Continuity and Dimension*, American Mathematical Society.

Mukherjee, A. (2015). *Differential Topology*, 2 edn, Springer.

Roweis, S. T. and Saul, L. K. (2000). Nonlinear dimensionality reduction by locally linear embedding, *Science* **290**(5500): 2323–2326.

Saul, L. K., Weinberger, K. Q., Sha, F., Ham, J. and Lee, D. D. (2006). Spectral methods for dimensionality reduction, *in* O. Chapelle, B. Scholkopf and A. Zien (eds), *Semi-Supervised Learning*, MIT Press Scholarship Online, chapter 1.

Schölkopf, B., Smola, A. and Müller, K.-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem, *Neural Computation* **10**: 1299–1319.

Sudderth, E. B. (2002). Nonlinear manifold learning part ii 6.454 summary.

Tenenbaum, J. B., de Silva, V. and Langford, J. C. (2000). A global geometric framework for nonlinear dimensionality reduction, *Science* **290**(5500): 2319–2322.

van der Maaten, L., Postma, E. and van den Herik, J. (2009). Dimensionality reduction: A comparative review, *Technical Report TiCC TR 2009-005*, Tilburg University.

Verleysen, M. and Francois, D. (2005). The curse of dimensionality in data mining and time series prediction, *in* J. Cabestany, A. Prieto and F. Sandoval (eds), *Computational Intelligence and Bioinspired Systems*, Springer.

Waldmann, S. (2014). *Topology. An Introduction*, Springer.

Weinberger, K. Q., Sha, F. and Saul, L. K. (2004). Learning a kernel matrix for nonlinear dimensionality reduction, *Proceedings of the 21rd International Conference on Machine Learning*.

Yang, X., Fu, H., Zha, H. and Barlow, J. (2006). Semi-supervised nonlinear dimensionality reduction, *Proceedings of the 23rd International Conference on Machine Learning*.

Zhang, S. and Chau, K.-W. (2009). Dimension reduction using semi-supervised locally linear embedding for plant leaf classification, *Proceedings of the 2009 International Conference on Intelligent Computing*.

Declaration of Authorship

I hereby declare that the report submitted is my own unaided work. All direct or indirect sources used are acknowledged as references. I am aware that the Thesis in digital form can be examined for the use of unauthorized aid and in order to determine whether the report as a whole or parts incorporated in it may be deemed as plagiarism. For the comparison of my work with existing sources I agree that it shall be entered in a database where it shall also remain after examination, to enable comparison with future Theses submitted. Further rights of reproduction and usage, however, are not granted here. This paper was not previously presented to another examination board and has not been published.