Seminar Report

Semi-Supervised Locally Linear Embedding: Application and 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

List of Symbols

$D \in \mathbb{N}$	Number of observed dimensions
$d \in \mathbb{N}$	Number of dimensions of embedded manifold
$m \in \mathbb{N}$	Number of dimensions of low-dimensional representation
$N \in \mathbb{N}$	Number of observed data points
$\mathcal{M} \subset \mathbb{R}^D$	d-manifold embedded in $\mathbb{R}^D$
$\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N) \in (\mathbb{R}^D)^N$	Observed coordinates of data sampled from $\mathcal{M}$
$\boldsymbol{Y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_N) \in (\mathbb{R}^m)^N$	Learned coordinates of low-dimensional representation of data

List of Figures

List of Tables

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 PCA*, 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

3.1.1 Taxonomy

After the goal of manifold learning has been formalized, it shall now be laid out how the problem is approached by LLE as the conceptual parent of SSLLE (the incorporation of prior information is a rather different matter that will be addressed in chapter 4.2; apart from this, the basic functionalities of SSLLE and LLE are identical and will therefore be presented as one). 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 of local graph-based manifold learning (LGML), which also comprises LEM and HLLE. The particular relationship of the three methods shall be made clear along the way.
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 (Bengio et al. (2003), Bengio et al. (2004)). 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.
Figure 2 depicts a schematic overview on the models studied here, representing the specific perspective taken within this report. 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).

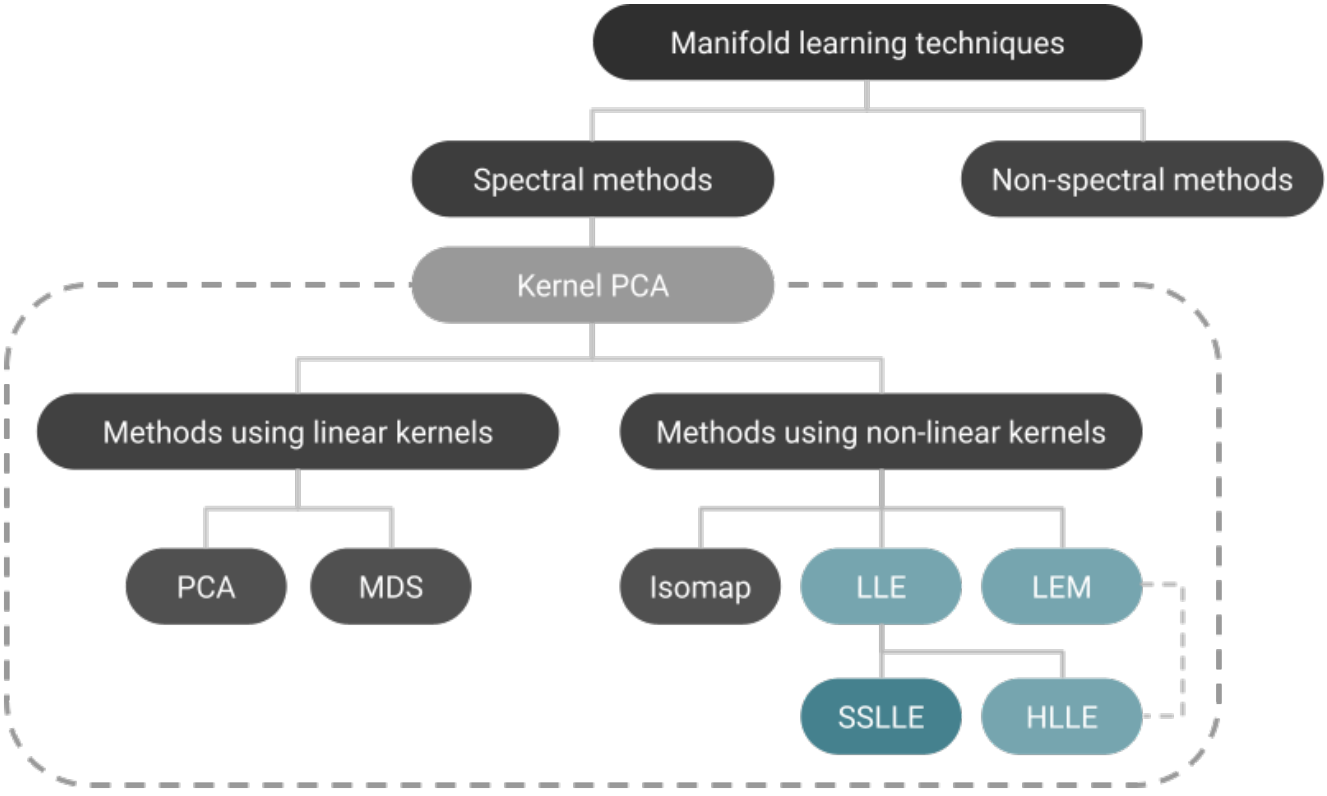

Figure 2: A schematic overview on selected methods of manifold learning (the list is by no means extensive and could arguably be ordered in several alternative ways). *Source:* own representation, inspired by a similar example given in van der Maaten et al. (2009) and re-interpreted with the findings in Bengio et al. (2004).

3.1.2 Intuition

As indicated in figure 2, this report will sketch the idea behind LGML in the light of *kernel principal component analysis (kernel PCA)*. Kernel PCA was actually proposed first and later shown to link the other concepts by a unified idea (Ham et al. (2003). It makes for an appealing

framework that 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).

Kernel PCA. Kernel PCA builds upon two fundamental concepts in machine learning: it performs *principal component analysis (PCA)* on data transformed by the *kernel trick*. It undertakes two subsequent steps. First, features of interest are extracted from the data by kernelization. These are taken to capture the intrinsic data structure and may therefore be understood as an approximation to the latent manifold properties. In the end, they constitute a matrix representation. Second, PCA finds the principal axes along which these intrinsic properties vary, yielding the desired reduction in dimensionality by preserving the most relevant latent dimensions (Schölkopf et al., 1998).

> **Kernelization.** By kernelization, mapping the data to a space $\mathcal{F}$ of arbitrarily high dimension, features may be obtained that relate to the input in a non-trivial way[4]. Crucially, the feature map $\phi : \mathbb{R}^D \to \mathcal{F}$ need not be computed explicitly, which might prove prohibitively expensive. Kernelization instead solely relies on inner products $\langle \phi(\boldsymbol{x}_i), \phi(\boldsymbol{x}_j) \rangle$ of the transformed inputs. Employing Mercer's theorem of functional analysis, these inner products may be interpreted as performed by a continuous kernel $\kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$ in some space with Hilbert property. Appropriate choice of κ then allows for the data to be represented by a matrix $\boldsymbol{K} \in \mathbb{R}^{N \times N}, \boldsymbol{K}_{ij} = \kappa(\boldsymbol{x}_i, \boldsymbol{x}_j)$. This matrix is the numerical data representation derived with respect to their latent properties. (Schölkopf et al., 1998). Precisely how it is computed depends on the choice of the kernel function κ and gives rise to different techniques (Ham et al., 2003).
>
> **PCA.** PCA is a quite powerful technique by itself. It finds the directions of maximum variance through eigenanalysis of the empirical covariance matrix, yielding the most important axes of inter-feature relations that coincide with the principal eigenvectors of the covariance matrix. The data are projected into the linear subspace spanned by these d eigenvectors, thereby mapping the observations to a coordinate system given by the linear feature combinations that represent the strongest (co)variability. Note that for this transformation to actually rotate the coordinate system about the origin, the data must be mean-centered beforehand. PCA thus performs an orthogonal input transformation that allows for dimensionality reduction at minimal information loss (Cayton, 2005). In kernel PCA, this eigenanalysis is implicitly performed in the feature space $\mathcal{F}$. Algorithmically, it boils down to diagonalizing the kernel matrix $\boldsymbol{K}$ (Schölkopf et al., 1998).

Figure 3 visualizes the idea of kernel PCA. The original data (*left*) are observed in two dimensions but clearly intrinsically one-dimensional, where the non-linear manifold feature is expressed by coloring. The kernel trick creates a feature map, visualized here as a projection of the intrinsic feature to a third coordinate axis (*middle*). Coercing the data to this dimension as the sole axis of variation yields the desired one-dimensional representation (*right*).

[4]Support vector machines use the kernel trick to achieve linear separability. An intuitive example may be given by data observed in two classes that form concentric circles in $\mathbb{R}^2$. While such data are not linearly separable in two dimensions, they are in three: mapping the classes to different heights enables separation by a horizontal hyperplane. This example also hints at the idea of (spectral) clustering to which kernel PCA is indeed intimately related (Bengio et al., 2004).

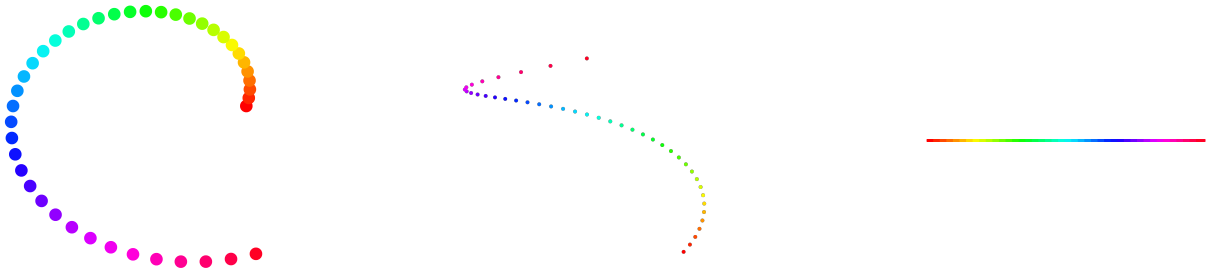

Figure 3: Schematic idea of kernel PCA: from data observed in two dimensions, but clearly of intrinsic dimensionality one (*left*), create a mapping to a higher-dimensional feature space (*middle*), reduction of which to its principal axes yields the desired one-dimensional representation (*right*). *Source:* own representation, using a subset of `mlbench`'s noise-free `spirals` data. Note that this is but a schematic depiction where the mid and right representation have not been created by an actual implementation of kernel PCA.

3.2 Conceptual Framework of LGML

3.2.1 Motivation

If kernel PCA 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. As foreshadowed in figure 2, spectral methods decompose into groups using *linear* and *non-linear* kernels, respectively. This distinction directly translates to the feature map ϕ. Linear methods suffer from the confinement to finding linear subspaces (van der Maaten et al., 2009). PCA in its standard form can be interpreted as kernel PCA by identifying the kernel function with the covariance function. It thus returns the subspace of greatest variability in the original input features (Ham et al., 2003). The closely related *multi-dimensional scaling (MDS)* yields the same result[5], albeit from a different intuition (Saul et al., 2006).
As extensively discussed above, $\mathcal{X}$ must often be assumed to lie on a non-linear manifold $\mathcal{M} \subset \mathbb{R}^D$, which is precisely why kernelization is usually performed such that the resulting feature space is related to the input space in a non-linear way (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 even perform kernelization explicitly: they transform the data in a way that can be shown to correspond to applying a (data-dependent) kernel function[6], but the fundamental intuition is a different one.

Non-linearity. The key idea in graph-based learning is to approximate the manifold by a discretized graph representation. Such a graph may be intuitively imagined as a skeletal model of $\mathcal{M}$ (an example is given in 5). This way, distances may be measured along the approximated manifold surface rather than in the ambient Euclidean space, effectively enabling non-linearity. Functionals that vary across methods are used to describe properties of the graph – essentially a proxy of the latent manifold properties (Saul et al., 2006).

Locality. A second desideratum in general manifold learning is the ability to treat highly non-linear manifolds with sufficiently local focus. Non-convexity means $\mathcal{M}$ is isometric to a non-convex subset of Euclidean space (Donoho and Grimes, 2003). Intuitively, such behavior requires careful

[5] At least, in its metric form; there are alternative formulations of MDS that are not equivalent to PCA (see, for example, Williams (2002)).

[6] The report does not discuss the actual kernel function as their illustrative ability is considered rather limited. For an explicit formulation of kernels in LLE, LEM and HLLE see for example Bengio et al. (2004) and Weinberger et al. (2004).

tracing of the manifold surface. LGML methods therefore focus on solely local manifold properties (Cayton, 2005). They are frequently contrasted to *Isomap*, one of the earliest and most prominent examples of global manifold learning. Isomap retains pairwise distances between points on the manifold surface as measured along graph edges via geodesic curves[7] (Tenenbaum et al., 2000). Its 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).

Algorithmic procedure. Summing up the above, LGML methods use functionals defined on graph approximations of the manifold to capture the intrinsic data structure. This information is stored in a matrix representation $\boldsymbol{M}$ that is directly linked to the kernel matrix $\boldsymbol{K}$. Eigenanalysis of $\boldsymbol{M}$ then leads to the sought-for low-dimensional subspace coordinates.
The resulting algorithmic pattern may be stated as follows (Bengio et al., 2003):

1. Construct a neighborhood graph $\mathcal{G}$ from the observed data.
2. Analyze the graph properties with an appropriate functional and derive a matrix representation $\boldsymbol{M}$ thereof.
3. Find the eigenvalues and associated eigenvectors of $\boldsymbol{M}$.
4. From the principal (top or bottom) eigenvectors, as determined by the ordered eigenvalues, retrieve the low-dimensional coordinates.

Figure 4 provides a final view on the LGML concept and its algorithmic operationalization in the kernel PCA context.

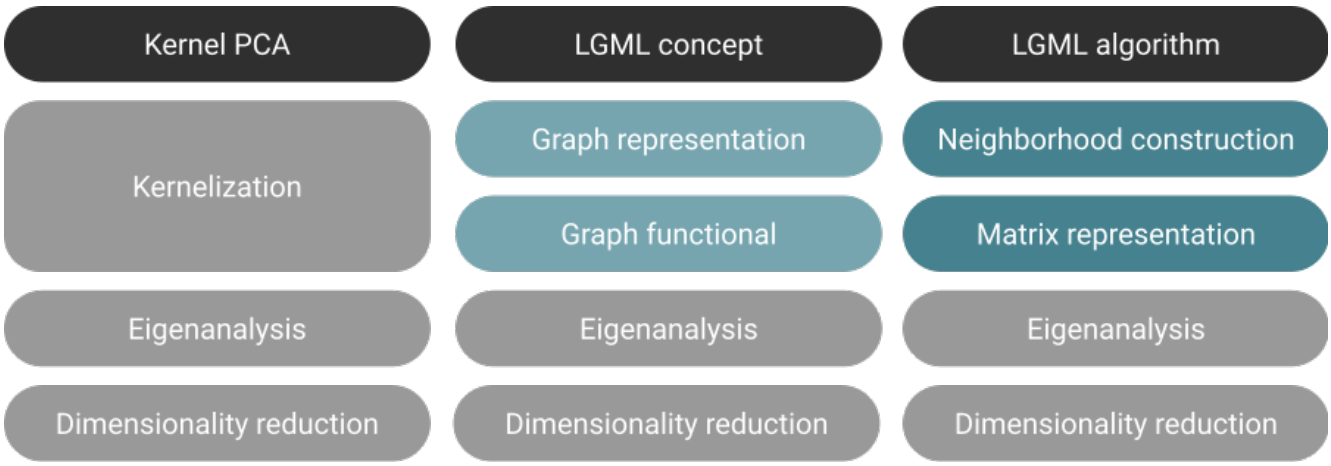

Figure 4: Conceptual view of LGML, interpreted as a form of kernel PCA. *Source:* own representation.

3.2.2 Graph Approximation

All LGML methods fundamentally build on graph approximations of the manifold surface. Note that these graphs are constructed from neighborhood relations in the high-dimensional observation space and do not require any prior information about the latent manifold coordinates.

Neighborhoods. 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

[7]It is thus a non-linear variant of MDS, which uses standard Euclidean distances (Tenenbaum et al., 2000).

are two principal ways to build a neighborhood around $\boldsymbol{x} \in \mathcal{X}$, both of which usually employ the squared Euclidean norm[8] $\|\cdot\|^2$. 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[9] with the smallest distance to $\boldsymbol{x}$, such that $\ell = k$ and

$$\mathcal{N}_k(\boldsymbol{x}) = \{\boldsymbol{x}_j \in \mathcal{X} : \|\boldsymbol{x} - \boldsymbol{x}_j\|^2 \leq d_{(k)}\}, \tag{1}$$

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

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

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

Both k and ϵ are hyperparameters that must be specified up-front. Their choice 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 4.3 will discuss this trade-off, which is also addressed in the practical implementation (section 5), in more detail. In this, the remainder of the report will focus on the k-neighborhood notion as it is typically more easily specified due to its inherent scale invariance and has attracted rather more attention in general research[10].

Neighborhood graphs. $\mathcal{M}$ can now be characterized by a *neighborhood graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, still 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, depending on the neighborhood definition.

It is easy to see that k-neighborhoods are an asymmetric notion; for one point to be among another's k nearest neighbors the reverse need not be true. k-neighborhoods therefore lead to directed graphs. Conversely, the ϵ-distance boundary holds in both directions and produces undirected graphs (He et al., 2005). Figure 5 shows how a neighborhood graph may be used as an approximation for the example of the S-curve manifold. It was built using k-neighborhoods with $k = 3$. For a densely sampled set of points, the graph representation should yield a fairly good approximation of the manifold surface.

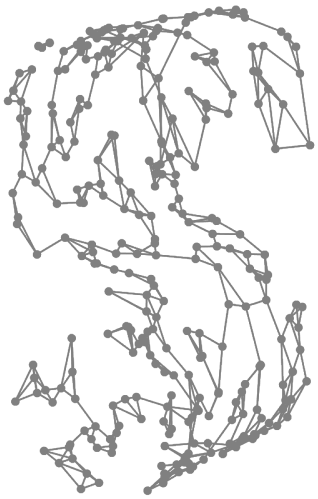

Figure 5: k-neighborhood graph for 300 points sampled from the S-curve with $k = 3$. *Source:* own representation.

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

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

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

3.2.3 Eigenanalysis

Eventually, spectral manifold learning boils down to an eigenanalysis of the matrix $\boldsymbol{M}$ believed to hold information about the intrinsic manifold structure. As explained in chapter 3.1, PCA finds the principal eigenvectors of empirical covariance, thereby defining a low-dimensional subspace containing most of the data-inherent variability. The very same idea applies when diagonalizing the more general matrix corresponding to the non-linear feature map: the top (or bottom[11]) d eigenvectors of M span a subspace into which the data may be projected under minimal loss of information. More precisely, the representation of $\mathcal{X}$ by the d selected eigenvectors of M is loss-optimal with respect to the least-squares error (Schölkopf et al., 1998). Figure **??** depicts the idea of eigenanalysis in a schematic way.

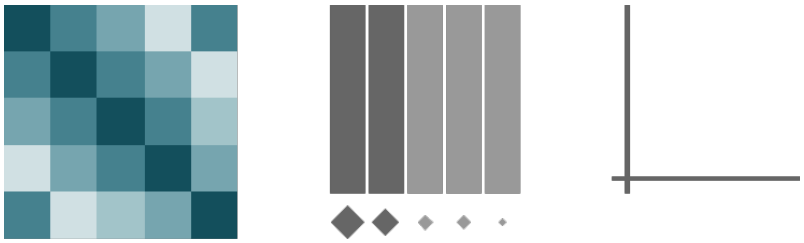

Figure 6: Conceptual idea of eigenanalysis. Eigenvectors of a matrix (*left*) point in the direction of greatest variability (*middle*), the degree of which is measured by the associated eigenvalues depicted as rhombi. Retaining the thus determined principal d eigenvectors allows to span a linear subspace 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

$$\boldsymbol{AV} = \boldsymbol{BV\Lambda}, \tag{3}$$

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

$$\min_{\boldsymbol{V}} trace(\boldsymbol{V}^T \boldsymbol{AV}), \quad \text{s.t.} \quad \boldsymbol{V}^T \boldsymbol{BV} = \boldsymbol{I}, \tag{4}$$

and translated to the first form by means of a Lagrangian multiplier (Ghojogh et al., 2019). It must be noted that solving eigenvalue problems becomes computationally challenging rather quickly, an issue on which chapter 4.3 also briefly comments with regard to the methods discussed here

[11] This differs across methods and shall be made clear later.

(Börm and Mehl, 2012).

Building upon the concepts of neighborhood graph approximation and subsequent eigenanalysis, the following chapter will now present in detail how LEM, LLE and HLLE approach the manifold learning task, and, eventually, how SSLLE seeks to improve the low-dimensional embedding through anchoring with prior points.

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 manifolds, 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.

Laplace-Beltrami operator. Consider the twice differentiable function $f : \mathcal{M} \to \mathbb{R}$ mapping two points $\boldsymbol{p}, \boldsymbol{q} \in \mathcal{M}$ to $f(\boldsymbol{p})$ and $f(\boldsymbol{q})$, respectively. On $\mathcal{M}$ these points are connected by a length-parametrized curve $c(t)$. Denote the geodesic distance between $\boldsymbol{p}$ and $\boldsymbol{q}$ by ℓ, such that $\boldsymbol{p} = c(0)$ and $\boldsymbol{q} = c(\ell)$.

Gradients of f with respect to $\boldsymbol{p}$ are defined in the local tangent space $T_{\boldsymbol{p}}(\mathcal{M})$ spanned by vectors tangent to $\mathcal{M}$ at $\boldsymbol{p}$. As $\mathcal{M}$ is embedded in $\mathbb{R}^D$, its tangent spaces are Euclidean and of dimension d, i.e., d-dimensional hyperplanes (Sudderth, 2002) as depicted in figure 7. If $\boldsymbol{p}$ is identified with the origin of $T_{\boldsymbol{p}}(\mathcal{M})$, the tangent space inherits an orthonormal coordinate system obtained from endowing $T_{\boldsymbol{p}}(\mathcal{M})$ with the inner product from $\mathbb{R}^d$ (Donoho and Grimes, 2003). With this, the distance $|f(\boldsymbol{p}) - f(\boldsymbol{q})|$ of mappings can be expressed as the length of the integral $\int_0^\ell \langle \nabla f(c(t)), c'(t) \rangle dt$. In other words, the geodesic curve connecting $\boldsymbol{p}$ and $\boldsymbol{q}$ is projected onto $T_{\boldsymbol{p}}(\mathcal{M})$, and the length of this projection depends on the gradient of f and the curve velocity.

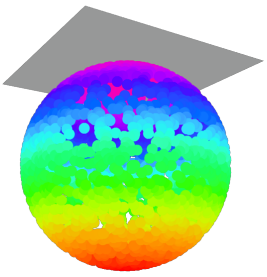

Figure 7: Tangent hyperplane for an exemplary point on the two-dimensional unit sphere manifold, embedded in $\mathbb{R}^3$. *Source:* own representation.

Exploiting the Schwartz equality and relations formally proved by Belkin and Niyogi (2008), it can be shown that $|f(\boldsymbol{p}) - f(\boldsymbol{q})| \leq \|\nabla f(\boldsymbol{p})\| \cdot \|\boldsymbol{p} - \boldsymbol{q}\| + o$, where o marks a term of vanishing size. As the distance between $\boldsymbol{p}$ and $\boldsymbol{q}$ is a datum, $\|\nabla f\|$ controls how far apart points are mapped on the real line. Consequently, the goal is to find a mapping that, on average, preserves locality by posing a second-order penalty on $\|\nabla f\|$ and minimizing $\int_{\mathcal{M}} \|\nabla f\|^2$. This is just equal to minimizing $\int_{\mathcal{M}} \mathcal{L}(f) f$ with the Laplace-Beltrami operator $\mathcal{L}$ (Belkin and Niyogi, 2003). Applying the operator $\mathcal{L}$ to f results in a function from the same function space as f, and for $\mathcal{L}f = \lambda f$, f is an eigenfunction of $\mathcal{L}$ with $\lambda \in \mathbb{R}$ as its associated eigenvalue. Crucially, these eigenfunctions are orthogonal for $\mathcal{L}$ and their eigenvalues are real, meaning they are natural candidates for forming

a functional basis (Levy, 2006). The optimal embedding map is then given by the d principal eigenfunctions of $\mathcal{L}$ after removing the bottom one which would map $\mathcal{M}$ to a single point (Belkin and Niyogi, 2003).

Graph Laplacian. Now the same reasoning can be applied to the neighborhood graph approximation of $\mathcal{M}$. Recall the desideratum of mapping nearby inputs to nearby outputs. LEM achieves this by assigning edge weights[12]

$$w_{ij} = \begin{cases} \exp(\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{t}), t \in \mathbb{R}, & \text{if } \boldsymbol{x}_i, \boldsymbol{x}_j \text{ are connected,} \\ 0 & \text{otherwise,} \end{cases}$$

forming a weight matrix $\boldsymbol{W} = (w)_{ij} \in \mathbb{R}^{N \times N}$. Clearly, edges between closer points receive larger weights. A second matrix $\boldsymbol{D} = (d)_{ij} \in \mathbb{R}^{N \times N}$ takes the row sums of $\boldsymbol{W}$ on its diagonals. Penalizing output disparities more severely for pairs of nearby points, i.e., pairs with a large weight coefficient, the smoothness requirement may be stated as follows:

$$\begin{aligned} \min_{\mathcal{Y}} \sum_{i,j} \|\boldsymbol{y}_i - \boldsymbol{y}_j\|^2 w_{ij} &= \min_{\mathcal{Y}} \sum_{i,j} \boldsymbol{y}_i^T \boldsymbol{y}_i w_{ij} + \boldsymbol{y}_j^T \boldsymbol{y}_j w_{ij} - 2\boldsymbol{y}_i^T \boldsymbol{y}_j w_{ij} \\ &= \min_{\mathcal{Y}} \sum_{i} \boldsymbol{y}_i^T \boldsymbol{y}_i d_{ii} + \sum_{j} \boldsymbol{y}_j^T \boldsymbol{y}_j d_{jj} - 2 \sum_{i,j} \boldsymbol{y}_i^T \boldsymbol{y}_j w_{ij}. \end{aligned}$$

Now, define the graph Laplacian as $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{W} \in \mathbb{R}^{N \times N}$, thereby coercing all information about the graph structure into a single matrix representation[13]. With $\boldsymbol{L}$ the above can be rewritten as

$$\min_{\mathcal{Y}} trace(\mathcal{Y}^T \boldsymbol{L} \mathcal{Y}), \quad \text{s.t. } \mathcal{Y}^T \boldsymbol{D} \mathcal{Y} = \boldsymbol{I}, \tag{5}$$

which has precisely the form of the generalized eigenvalue problem in equation 3 and is therefore solved by eigendecomposition of $\boldsymbol{L}$ (Belkin and Niyogi, 2003). As in the continuous case, the bottom eigenvector with zero eigenvalue is constant and must be discarded. The subsequent d eigenvectors yield the desired low-dimensional embedding (Levy, 2006).

4.1.2 Locally Linear Embedding (LLE)

In proposing LEM, Belkin and Niyogi (2003) also demonstrated how the somewhat earlier LLE algorithm may be reinterpreted within the LEM framework: it can be shown to approximate the graph Laplacian under certain conditions and thus asymptotically approach the Laplace-Beltrami operator. More recent research, however, suggests that these conditions might be more restrictive than previously assumed. In particular, convergence appears to depend on the choice of a regularization parameter required in the case of $D < k$ (Wu and Wu, 2018).

Idea. The initial proposal by Roweis and Saul (2000), ignorant to these findings, was made with a different, and rather heuristically motivated, intuition. LLE relies on a simple yet powerful idea. Each point $\boldsymbol{x}_i$ in the D-dimensional input space is expressed as a convex combination of its neighbors, such that the weighting coefficients of this reconstruction essentially represent the edge weights of the neighborhood graph around $\boldsymbol{x}_i$. These (generalized) barycentric coordinates

[12] These weights stem from the heat kernel intimately related to the Laplace-Beltrami operator and ensure positive semi-definiteness of the resulting graph Laplacian. As an alternative, Belkin and Niyogi (2003) propose a simpler kernel that is equal to 1 for connected nodes and 0 otherwise.

[13] This corresponds to the general matrix $\boldsymbol{M}$ introduced in chapter 3.2.1; the deviating notation shall merely emphasize the special role of $\boldsymbol{M}$ as the Laplacian.

now bear a crucial property: they are invariant to rotation, rescaling and translation of the neighborhood, and thus topological properties that equally hold in the low-dimensional embedding space. In other words, the same weights that serve to reconstruct $\boldsymbol{x}_i$ in $\mathbb{R}^D$ should do so in $\mathbb{R}^d$ (Roweis and Saul, 2000). Obviously, this belief is only justified if $\mathcal{M}$ is indeed locally linear and the graph edges run along the manifold surface rather than short-circuiting it, hinting at the important role of neighborhood size which will be discussed in chapter 4.3.

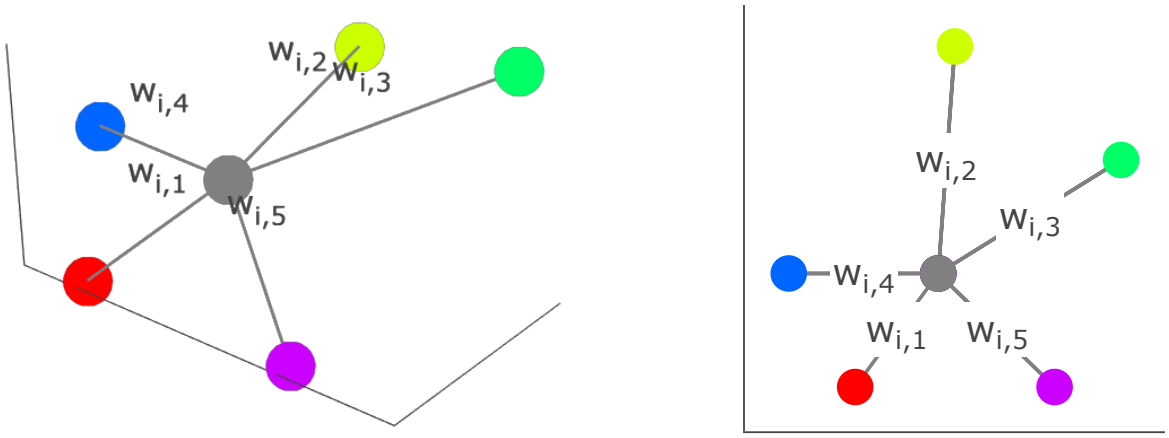

Figure 8: Reconstruction in three (*left*) and two (*right*) dimensions, employing the same reconstruction weights. *Source:* own representation.

Algorithmically, LLE performs two subsequent steps (Roweis and Saul, 2000):

1. Compute the reconstruction weights in $\mathbb{R}^D$ minimizing reconstruction loss.
2. Compute the embedding coordinates in $\mathbb{R}^d$ minimizing embedding loss.

Reconstruction loss minimization. Reconstruction errors are measured by a quadratic loss function. Optimization of the objective is subject to a sum-one constraint for the weights of each point. A second constraint, zero weights for non-neighboring points, is implicitly enforced during construction of the neighborhood graph, where edges are only drawn to vertices belonging to $\boldsymbol{x}_i$'s neighborhood (Ghojogh et al., 2020). The resulting optimization problem is convex and has a unique closed-form solution[14] (Roweis and Saul, 2000):

$$\begin{aligned} \min_{\boldsymbol{W}} \varepsilon(\boldsymbol{W}) &= \min_{\boldsymbol{W}} \sum_i \left\| \boldsymbol{x}_i - \sum_j w_{ij} \boldsymbol{x}_j \right\|^2 \\ &= \min_{\boldsymbol{W}} \sum_i \left\| \boldsymbol{x}_i - \boldsymbol{N}_i \boldsymbol{w}_i \right\|^2, \\ \text{s.t. } & \mathbf{1}^T \boldsymbol{w}_i = 1 \quad \forall i \in \{1, 2, ..., N\}. \end{aligned} \tag{6}$$

Here, $\boldsymbol{N}_i \in \mathbb{R}^{D \times k}$ denotes the matrix of feature vectors of $\boldsymbol{x}_i$'s neighbors and $\boldsymbol{w}_i = \sum_j w_{ij} \in \mathbb{R}^k$. Equation 6 can be re-arranged by use of the sum-one constraint and simplified by introduction of the gram, or local covariance, matrix $\boldsymbol{G}_i$ (Saul and Roweis, 2001):

[14] Note that the weight matrix $\boldsymbol{W}$ is different from the one computed in LEM, but the same symbol is used as not to overload notation.

$$\begin{aligned}\min_{\boldsymbol{W}} \varepsilon(\boldsymbol{W}) &= \min_{\boldsymbol{W}} \sum_i \left\| \boldsymbol{x}_i \mathbf{1}^T \boldsymbol{w}_i - \boldsymbol{N}_i \boldsymbol{w}_i \right\|^2 \\ &= \min_{\boldsymbol{W}} \sum_i \boldsymbol{w}_i^T (\boldsymbol{x}_i \mathbf{1}^T - \boldsymbol{N}_i)^T (\boldsymbol{x}_i \mathbf{1}^T - \boldsymbol{N}_i) \boldsymbol{w}_i \\ &= \min_{\boldsymbol{W}} \sum_i \boldsymbol{w}_i^T \boldsymbol{G}_i \boldsymbol{w}_i, \\ \text{s.t. } & \mathbf{1}^T \boldsymbol{w}_i = 1 \quad \forall i \in \{1, 2, ..., N\}.\end{aligned} \tag{7}$$

By standard use of a Lagrange multiplier, the solution for the above constrained optimization problem collapses to

$$\boldsymbol{w}_i = \frac{\boldsymbol{G}_i^{-1} \mathbf{1}}{\mathbf{1}^T \boldsymbol{G}_i^{-1} \mathbf{1}}. \tag{8}$$

Solving the reconstruction problem therefore requires N matrix inversions, which may prove problematic if the gram matrices do not achieve full rank. In the case of $D < k$, $\boldsymbol{G}_i$ is indeed singular and must be robustified by adding a small numerical constant to its diagonal (Ghojogh et al., 2020). Chapter 4.3 will discuss how this regularization is applied.

Embedding loss minimization. The second optimization problem minimizes the embedding cost arising from mapping neighborhood geometries, derived in the form of reconstruction weights, into the d-dimensional subspace. Keeping the weight coefficients fixed, the aim is to find such embedding coordinates that best preserve the vicinity structures and adhere to the constraints of summing to zero (i.e., being centered around the origin) and having unit covariance (Roweis and Saul, 2000):

$$\begin{aligned}\min_{\mathcal{Y}} \Phi(\mathcal{Y}) &= \min_{\mathcal{Y}} \sum_i \left\| \boldsymbol{y}_i - \sum_j w_{ij} \boldsymbol{y}_j \right\|^2, \\ &\text{s.t. } \frac{1}{N} \sum_i \boldsymbol{y}_i \boldsymbol{y}_i^T = \boldsymbol{I} \quad \text{and} \quad \sum_i \boldsymbol{y}_i = \mathbf{0} \quad \forall i \in \{1, 2, ..., N\}.\end{aligned} \tag{9}$$

The objective can be equivalently stated as an eigenvalue problem. For this purpose, define $\boldsymbol{E} = (\boldsymbol{I} - \boldsymbol{W})^T (\boldsymbol{I} - \boldsymbol{W})$ and set $\tilde{\mathcal{Y}} = \mathcal{Y}^T$ (Cayton, 2005), such that:

$$\min_{\tilde{\mathcal{Y}}} \mathit{trace}(\tilde{\mathcal{Y}}^T \boldsymbol{E} \tilde{\mathcal{Y}}), \quad \text{s.t. } \frac{1}{N} \tilde{\mathcal{Y}}^T \tilde{\mathcal{Y}} = \boldsymbol{I} \quad \text{and} \quad \tilde{\mathcal{Y}}^T \mathbf{1} = \mathbf{0}. \tag{10}$$

Again, the solution is found by eigenanalysis of the matrix encoding the intrinsic manifold structures. Note that the first constraint carries a factor $1/N$ as originally proposed. In fact, any such quadratic form, provided its right hand side is of full rank, would suffice to ensure the embedding vectors actually span a d-dimensional space (Burges, 2010). The additional sum-zero condition is implicitly met by discarding the constant eigenvector associated with the bottom, zero eigenvalue and taking the subsequent d eigenvectors to embed the data in $\mathbb{R}^d$ (Ghojogh et al., 2020).
As mentioned before, the close resemblance to the optimization problem in LEM (equation 5) is not coincidental. Recall that LEM builds upon eigenfunctions of the Laplace-Beltrami operator $\mathcal{L}$. Belkin and Niyogi (2003) show that LLE approximates the eigenfunctions of the iterated form $\frac{1}{2}\mathcal{L}^2$, which are identical to those of $\mathcal{L}$. Therefore, with a somewhat different intuition, LLE essentially arrives at the same conclusion.

4.1.3 Hessian Locally Linear Embedding (HLLE)

Lastly, HLLE (Donoho and Grimes, 2003) pursues an approach toward LGML that straddles the two former techniques: it borrows heavily from the idea behind LEM but is akin to LLE in an algorithmic sense[15] (Ross, 2008). Proposed under the title *Hessian eigenmaps*, it is therefore also referred to as Hessian LLE (Donoho and Grimes, 2003).
As opposed to LLE, HLLE is built upon a rigid theoretical foundation. It makes the assumptions of local isometry and homeomorphicity to an open, connected subset of $\mathbb{R}^d$ (see chapter 3.2.1), which is a much less restrictive demand than Isomap's global isometry and parameter space convexity (Donoho and Grimes, 2003). With this, HLLE provides veritable convergence guarantees for a wide range of cases, albeit only for the continuum limit and not in finite-sample situations (Cayton, 2005).

Idea. HLLE considers the twice-differentiable mapping functions $f : \mathcal{M} \to \mathbb{R}$ employed in LEM. Recall that LEM defines the gradient of f with respect to local tangent spaces $T_{\boldsymbol{p}}(\mathcal{M})$ at $\boldsymbol{p} \in \mathcal{M}$ as a notion of smoothness. Similarly, HLLE computes the Hessian to measure curviness of f (Donoho and Grimes, 2003). One advantage of this modification is that, while the Laplacian equals zero for any harmonic[16] function on $\mathcal{M}$, the Hessian vanishes if and only if f is linear (Ross, 2008).

Hessian functional. Consider $\boldsymbol{p} \in \mathcal{M}$ and its k-neighborhood $\mathcal{N}_k(\boldsymbol{p})$, each of whose members has a unique closest point on $T_{\boldsymbol{p}}(\mathcal{M})$ via the smooth mapping f. Identifying $f(\boldsymbol{p})$ with $\mathbf{0} \in \mathbb{R}^d$ yields a system of local coordinates on $T_{\boldsymbol{p}}(\mathcal{M})$ that depends on this particular choice of the origin. For a neighbor $\boldsymbol{p}'$ of $\boldsymbol{p}$, let these local coordinates be denoted by $\boldsymbol{x}^{\text{loc},\,\boldsymbol{p}'}$. Then, the Hessian $\boldsymbol{H}_f^{\text{loc}}(\boldsymbol{p})$ of f at $\boldsymbol{p}$ in tangent coordinates may be expressed as the ordinary Hessian of a function $g : U \to \mathbb{R}$ with $f(\boldsymbol{p}') = g(\boldsymbol{x}^{\text{loc},\,\boldsymbol{p}'})$, U being a neighborhood of zero in $\mathbb{R}^d$ (Donoho and Grimes, 2003):

$$(\boldsymbol{H}_f^{\text{loc}}(\boldsymbol{p}))_{i,j} = \frac{\partial^2 g(\boldsymbol{x}^{\text{loc},\,\boldsymbol{p}'})}{\partial x_i^{\text{loc},\,\boldsymbol{p}'} \partial x_j^{\text{loc},\,\boldsymbol{p}'}}\bigg|_{\boldsymbol{x}^{\text{loc},\,\boldsymbol{p}'}=0}. \tag{11}$$

From these point-wise tangent Hessians it is now possible to construct a quadratic functional $\mathscr{H}(f)$ on the entire manifold, analogous to the Laplace-Beltrami operator in LEM. The crucial property of $\mathscr{H}(f)$ is given by the fact that, if $\mathcal{M}$ is truly locally homeomorphic to an open, connected subset of $\mathbb{R}^d$, $\mathscr{H}(f)$ has a $(d+1)$-dimensional null space. After discarding the constant function corresponding to the bottom zero eigenvalue, the subsequent d eigenfunctions again span the desired low-dimensional embedding space (Donoho and Grimes, 2003).
First, however, the dependency on the respective local coordinate systems must be removed. This is achieved by taking the Frobenius norm of the tangent Hessians: for any alternative coordinate system $\boldsymbol{H}'$ obtained by orthogonal transformation of $\boldsymbol{H}$ with a suitable matrix $\boldsymbol{B}$, it must hold that $\|\boldsymbol{H}'\|_F^2 = \|\boldsymbol{B}\boldsymbol{H}\boldsymbol{B}^T\|_F^2 = \mathit{trace}(\boldsymbol{B}\boldsymbol{H}^T\boldsymbol{B}^T\boldsymbol{B}\boldsymbol{H}\boldsymbol{B}^T) = \mathit{trace}(\boldsymbol{H}^T\boldsymbol{H}) = \|\boldsymbol{H}\|_F^2$, due to the permutation invariance of the trace operator (Ross, 2008). Thus, $\mathscr{H}(f)$ as a measure for overall curviness of the mapping is given by (Donoho and Grimes, 2003):

$$\mathscr{H}(f) = \int_{\mathcal{M}} \|\boldsymbol{H}_f^{\text{loc}}(\boldsymbol{p})\|_F^2 \, d\boldsymbol{p}. \tag{12}$$

HLLE algorithm. In analogy to LEM, the functional defined on $\mathcal{M}$ is approximated in an empirical manner; yet, the computations performed in HLLE are somewhat more involved. LEM incorporates neighborhood information as pairwise distances entering weight computation. LLE

[15] HLLE is also closely related to another technique beyond the scope of this report, namely *local tangent space alignment (LTSA)* (see, for example, Ting and Jordan (2018)).

[16] An example is indeed given by the coordinate functions; however, other functions that are clearly non-linear have the harmonic property (see, for example, Axler et al. (2001)).

and HLLE take a more explicit look at locally linear patches on the manifold surface and attempt to map these to the low-dimensional space (Cayton, 2005). As before, the first step consists of finding the k nearest neighbors to $\boldsymbol{x}_i \in \mathcal{X}$ for $i \in \{1, 2, ..., N\}$, followed by Let $\boldsymbol{N}_i \in \mathbb{R}^{D \times k}$ again denote the matrix of feature vectors of $\boldsymbol{x}_i$'s neighbors, this time centered with respect to the mean over all members of the neighborhood. From these neighborhood matrices the local tangent coordinates are then estimated by means of N singular value decompositions $\boldsymbol{N}_i = \boldsymbol{U_i D_i V_i}^T$ (Ross, 2008). In effect, this amounts to finding the basis of $T_{\boldsymbol{x}_i}(\mathcal{M})$ by performing PCA on the local covariance matrix at $\boldsymbol{x}_i$ and retaining the d principal eigenvectors. Now a matrix $\boldsymbol{Z_i}$, whose columns contain all cross products of $\boldsymbol{U}_i$ up to order d, is constructed at each point and coerced into orthonormal form. Extracting the transpose of the last $\frac{d(d+1)}{2}$ columns of $\boldsymbol{Z_i}$ yields the local Hessian approximator $\boldsymbol{H}_i$ as the least-squares estimate of a local quadratic polynomial regression in the neighborhood of $\boldsymbol{x}_i$ (van der Maaten et al. (2009), Ting and Jordan (2018)). The empirical Hessian functional $\mathcal{H}$ is obtained as a quadratic form of the local Hessian approximators (Donoho and Grimes, 2003):

$$\mathcal{H}_{ij} = \sum_{\ell} \sum_{m} (\boldsymbol{H}_\ell)_{m,i} (\boldsymbol{H}_\ell)_{m,j}. \tag{13}$$

Eventually, eigenanalysis of $\mathcal{H}$ yields the approximate null space spanned by the d bottom eigenvectors after discarding the constant one. The final step required to find the desired embedding coordinates consists of finding a basis for the null space. For this, take the matrix $\boldsymbol{Q} \in \mathbb{R}^{N \times d}$ containing the d non-constant eigenvectors and find a second matrix $\boldsymbol{R}$ such that the columns of $\boldsymbol{QR}$ restricted to a fixed local neighborhood are orthonormal. The d-dimensional embedding coordinates are then given by $\boldsymbol{Q}^T \boldsymbol{R}^T$ (Ye and Zhi, 2015). As an alternative, Ross (2008) proposes to replace the last step by simply scaling $\boldsymbol{Q}$ with $\sqrt{N}$.

Tracing the steps described above, it becomes clear that the theoretical guarantees of HLLE come at the expense of rather complex computations that have been sketched only roughly here[17]. At the same time, its implementation employs numerous approximations calling the merit of theoretical convergence into question. It is perhaps this approximate yet computationally challenging design, along with the fact that the other methods are more easily accessible by intuition, that has acted as a limiting factor on the practical applications of HLLE (Cayton (2005), Ye and Zhi (2015)).

4.2 Semi-Supervised Locally Linear Embedding (SSLLE)

4.2.1 Employment of Prior Information

All of the above methods operate in a fully unsupervised manner, relying solely on the D-dimensional coordinates of the observation space. The endeavor of dimensionality reduction will thus sometimes fail to produce a meaningful embedding. Yang et al. (2006) therefore propose to anchor the low-dimensional representation at a number of prior points whose coordinates in $\mathbb{R}^d$ are already known. Obviously, this semi-supervision commands the availability of prior information. It is, however, easy to conceive practical applications where prior knowledge is indeed at hand or may be obtained at little cost. For instance, consider the image classification task stated as an introductory example. While the learning algorithm must regard all of the numerous dimensions spanned by (high-resolution) image data on pixel level, a human annotator may quickly spot the typically much fewer latent sources causing variability, such as rotation or zooming. Other applications like, say, textual analysis might not reveal this information at first sight, but can benefit from pre-labeled data made available by the advance of open-source research.
What Yang et al. (2006) dub semi-supervised LLE is actually somewhat different from the idea typically employed in semi-supervised learning. Rather than supporting a supervised learning task

[17]For a more in-depth analysis see, for example, Ting and Jordan (2018).

by information extracted from the pool of unlabeled data, an inherently unsupervised problem is alleviated by specifying part of the solution upfront. Arguably, SSLLE might therefore be viewed in the context of active learning, where the model is allowed to query labels in a sequential manner that seeks to maximize utility at minimal expense [18]. The practical experiments in chapter 5 will thus assume a setting where prior information can be inquired from the pool of initially unlabeled observations. A straightforward approach is then to select the prior points in a way that is most informative to the SSLLE learning algorithm.

4.2.2 Finding Prior Points

Prior points: take minmax approach from sparse MDS (Sparse multidimensional scaling using landmark points Vin de Silva and Joshua B. Tenenbaum 2004). Easy and deterministic after choosing seed value. Instead Euclidean distances, though, take geodesics as estimated in isomap. can we view the prior info as some kind of active learning? like we choose some points to label in a hopefully cleverish way and then hand them to you (e.g., to look at some pictures instead of all droelf thousand)

4.2.3 SSLLE Algorithm

- What is different wrt standard LLE?

4.3 Particular Challenges

A number of computational and design-related challenges arise from this procedure that must be faced in implementation.

Choice of intrinsic dimensionality. Until now, it has been assumed, rather implicitly, that the intrinsic dimension d of the data is a known parameter. This is obviously not always the case in practical applications. Some methods offer the advantage of estimating d in a built-in fashion. PCA, MDS and Isomap, for instance, typically show an indicative gap in their eigenvalue spectrum, distinctly pointing out the dimensions with the largest share of variability (Saul et al., 2006). For LLE, LEM and HLLE, no such tell-tale gap exists. While Sha and Saul (2005) have indeed drawn a mathematical relation between the respective eigenspectra in LLE and LEM and intrinsic data dimensionality, they immediately discarded this finding for practical applications due to large computational overhead and lack of reliability in finite-sample situations. There have been various other proposals to tackle the problem of dimensionality estimation (for an extensive discussion, see for example Wissel (2017)). However, as the focus of this report lies on a semi-supervised method of manifold learning, it is mainly concerned with situations where prior knowledge of coordinates, and of d in particular, is actually available.

Choice of neighborhood size. Choosing the size of neighborhoods for graph approximation does pose a challenge. It is a standard hyperparameter optimization problem in which a trade-off between locality and overall approximation must be balanced. If neighborhoods are too small, the model will not be able to learn the global manifold structure; with overly large neighborhoods, it will forgo the advantages of locality and non-linearity and essentially behave like PCA (de Ridder and Duin, 2002).
Describe applied approach

Robustness of eigendecomposition. Mainly problem in LLE (?)

Computational cost. Text

[18] For an extensive overview on active learning see, for example, Settles (2009).

- Number of neighbors (diss grilli (referenced in lle manual) discusses regression model, ghojogh propose different things)
- Intrinsic dimensionality
- Singularity of gram matrix (LLE-specific?!)
- Large data (landmarks)

Comment on difficulty of finding neighbors in high dimensions
What about using RF proximities for neighbor search? Unsupervised RF works with simulating new data from the estimated dist of the present ones, see e.g. https://horvath.genetics.ucla.edu/html/RFclus https://arxiv.org/pdf/2004.02121.pdf

4.4 Comparative Remarks

HLLE has convergence guarantees - not so LLE, LEM (sudderth)

5 Experiment Results

5.1 Experimental Design

5.1.1 Software Implementation

5.1.2 Evaluation Framework

5.2 Application to Synthetic Data

5.2.1 Data

5.2.2 Results

5.3 Sensitivity Analysis

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[19] 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*[20] (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[21] (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 $\mathbb{R}^d$.

[19] Alternative definitions employ open subsets of T, see for example Waldmann (2014).
[20] For locally Euclidean topological spaces it is thus meaningful to speak of elements as points.
[21] $\mathcal{M}$ 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. One last aspect shall be briefly touched upon, namely how to handle distances on general manifolds where Euclidean metrics are not meaningful. Rather than measuring "shortcuts" between points across $\mathbb{R}^D$ it makes intuitive sense to constrain distances to the manifold surface. In order to enable the construction of such a metric, manifolds must fulfill two additional properties: *smoothness*[22] and *connectedness*[23] (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

$$c(\lambda) = (c_1(\lambda), ..., c_d(\lambda))^T \tag{14}$$

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:

$$c'(\lambda) = (c_1'(\lambda), ..., c_d'(\lambda))^T. \tag{15}$$

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 $\boldsymbol{q}$:

$$d^{\mathcal{M}}(\boldsymbol{p}, \boldsymbol{q}) = \inf_{c \in \mathcal{C}(\boldsymbol{p}, \boldsymbol{q})} L(c). \tag{16}$$

Intuitively, geodesic distance can be identified with Euclidean distance in Euclidean spaces where shortest curves are just straight lines (Ma and Fu, 2011).

A.2 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.

[22] 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).

[23] 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).

B Electronic Appendix

Data, code and figures are provided in electronic form.

References

Axler, S., Bourdon, P. and Ramey, W. (2001). *Harmonic Function Theory*, 2 edn, Springer.

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.

Belkin, M. and Niyogi, P. (2008). Towards a theoretical foundation for laplacian-based manifold methods, *Journal of Computer and System Sciences* **74**(8): 1289–1308.

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.

Burges, C. J. (2010). Geometric methods for feature extraction and dimensional reduction - a guided tour, *in* O. Maimon and L. Rokach (eds), *Data Mining and Knowledge Discovery Methods*, Springer US, pp. 53–82.

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., Ghodsi, A., Karray, F. and Crowley, M. (2020). Locally linear embedding and its variants: Tutorial and survey.

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.

Levy, B. (2006). Laplace-beltrami eigenfunctions towards an algorithm that "understands" geometry, *Proceedings of the IEEE International Conference on Shape Modeling and Applications*.

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.

Ross, I. (2008). *Nonlinear Dimensionality Reduction Methods in Climate Data Analysis*, PhD thesis, University of Bristol.

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

Saul, L. K. and Roweis, S. T. (2001). An introduction to locally linear embedding, *Journal of Machine Learning Research* **7**.

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.

Settles, B. (2009). Active learning literature survey, *Technical Report 1648*, University of Wisconsin – Madison.

Sha, F. and Saul, L. K. (2005). Analysis and extension of spectral methods for nonlinear dimensionality reduction, *Proceedings of the 22nd International Conference on Machine Learning*.

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.

Ting, D. and Jordan, M. I. (2018). On nonlinear dimensionality reduction, linear smoothing and autoencoding, *arXiv: Machine Learning* .

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*.

Williams, C. K. (2002). On a connection between kernel pca and metric multidimensional scaling, *Machine Learning* **46**: 11–19.

Wissel, D. R. (2017). *Intrinsic Dimension Estimation using Simplex Volumes*, PhD thesis, Rheinische Friedrich-Wilhelms-Universität Bonn.

Wu, H.-T. and Wu, N. (2018). Think globally, fit locally under the manifold setup: Asymptotic analysis of locally linear embedding, *Ann. Stat* **246**(6B): 3805–3837.

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

Ye, Q. and Zhi, W. (2015). Discrete hessian eigenmaps for dimensionality reduction, *Journal of Computational and Applied Mathematics* **278**: 197–212.

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.