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

# List of Abbreviations

| | |
|---|---|
| AUC | area under the curve |
| 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

| | |
|---|---|
| $\boldsymbol{I} = diag(1) \in \mathbb{R}^{s \times s}$ | identity matrix with $s^2$ entries, $s \in \mathbb{N}$ |
| $\mathbf{0} = (0, 0, \ldots, 0)^T \in \mathbb{R}^s$ | $s$-dimensional null vector, $s \in \mathbb{N}$ |
| $\mathbf{1} = (1, 1, \ldots, 1)^T \in \mathbb{R}^s$ | $s$-dimensional one vector, $s \in \mathbb{N}$ |
| $N \in \mathbb{N}$ | number of observed data points |
| $D \in \mathbb{N}$ | number of observed dimensions |
| $d \in \mathbb{N}$ | number of intrinsic dimensions |
| $k \in \mathbb{N}$ | number of neighbors |
| $m \in \mathbb{N}$ | number of prior points |
| $\mathcal{M} \subset \mathbb{R}^D$ | $d$-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 |
| $T_{\boldsymbol{p}}(\mathcal{M})$ | local tangent space at $\boldsymbol{p} \in \mathcal{M}$ |
| $\mathcal{L}(f)$ | Laplace-Beltrami operator in LEM |
| $\boldsymbol{L} \in \mathbb{R}^{N \times N}$ | graph Laplacian in LEM |
| $\mathscr{H}(f)$ | Hessian functional in HLLE |
| $\mathcal{H} \in \mathbb{R}^{N \times N}$ | empirical Hessian approximator in HLLE |
| $\beta > 0$ | confidence parameter in SSLLE |
| $\gamma > 0$ | regularization parameter in SSLLE implementation |

# List of Figures

# 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 $D \gg d$ (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 to 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 ceveloped 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). Yet, 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) propose to 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 varying 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. 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*) between them lying 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 treated as $d$-dimensional, provided 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 $\psi$ itself is not always explicitly retrieved.

Note that, while $D$ is given a priori, the intrinsic dimensionality $d$ is often unknown. $\mathcal{Y}$ 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 is lacking, 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).

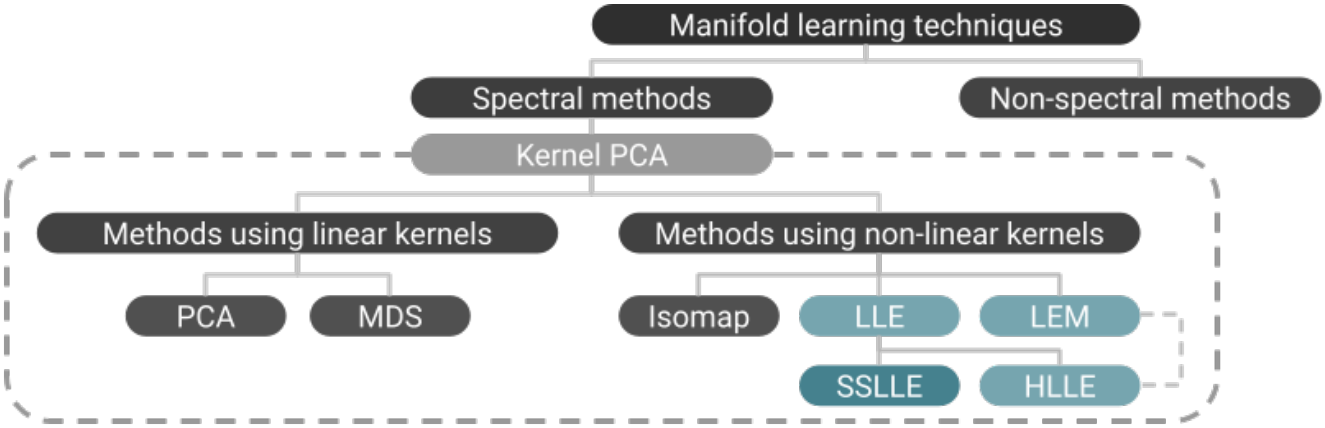


Figure 2: Overview on selected methods of manifold learning. *Source:* own representation, inspired by a similar example in van der Maaten et al. (2009) and re-interpreted with the findings in Bengio et al. (2004).

**Intuition.** 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. 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. To this end, eigenanalysis is performed on the representation matrix, yielding the desired reduction in dimensionality through preserving the most relevant latent dimensions (Schölkopf et al., 1998).

[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.

## 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. Methods using linear kernels, such as standard PCA, suffer from the confinement to linear embedding spaces (van der Maaten et al., 2009). If $\mathcal{X}$ lies on a non-linear manifold, as must be generally assumed, kernelization is best 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.

**Idea.** 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 arbitrary 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 handle (possibly non-convex) manifolds with complicated surfaces. 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 to avoid coarse mappings of the global structure at the expense of local congruence. LGML methods therefore focus on local properties (Cayton, 2005).

**Local neighborhoods.** Graph approximations are constructed from neighborhood relations in the observation space. Neighborhoods are typically taken to be $k$-neighborhoods, i.e., based on a fixed number $k \in \mathbb{N}$ of neighbors. In principle, it is equally possible to restrict neighborhoods to a maximum distance of $\epsilon \in \mathbb{R}^+$ to the centroid. However, $k$-neighborhoods are often more easily specified due to the inherent scale invariance of $k$, and have attracted rather more attention in general research[6] (He et al., 2005). For a formal definition of $k$- and $\epsilon$-neighborhoods, see section A.3. Both notions usually rely on Euclidean distance. In the end, any vicinity condition is admissible so long as it serves to faithfully characterize the manifold surface in a computationally affordable manner (Roweis and Saul, 2000). For the remainder of this report, neighborhoods will be taken to be $k$-neighborhoods. The crucial parameter for LGML is, in fact, neighborhood size. It reflects beliefs about the topological structure of $\mathcal{M}$: smaller neighborhoods correspond to a higher degree of non-linearity, emphasizing local properties more strongly, and vice versa (Sudderth, 2002). Chapter 5 will discuss how the trade-off is addressed in practice.

**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 $\epsilon$-radius). It is easy to see that $k$-neighborhoods are an asymmetric notion and therefore lead to directed graphs. Conversely, the $\epsilon$-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.

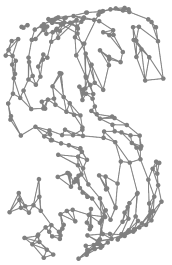

Figure 3: $k$-neighborhood graph for 300 points sampled from the S-curve with $k = 3$. *Source:* own representation.

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

**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 constructed 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.2). The nature of different graph functionals and resulting matrix representations across methods will be discussed in the subsequent chapters.

# 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) and closely relating to the theoretical concepts in HLLE (Donoho and Grimes, 2003). LEM is 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.

**Continuous justification.** Consider the twice differentiable function $f : \mathcal{M} \to \mathbb{R}$ mapping $\boldsymbol{p}, \boldsymbol{q}$ 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 $\ell$, 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 $T_{\boldsymbol{p}}(\mathcal{M})$. Local tangent spaces of $\mathcal{M}$ are $d$-dimensional hyperplanes (Sudderth, 2002), as shown exemplarily by figure 4. If $\boldsymbol{p}$ is identified with the origin of $T_{\boldsymbol{p}}(\mathcal{M})$, the tangent space inherits an orthonormal coordinate system from endowing $T_{\boldsymbol{p}}(\mathcal{M})$ with the inner product of $\mathbb{R}^d$ (Donoho and Grimes, 2003). With this, the distance $|f(\boldsymbol{p}) - f(\boldsymbol{q})|$ of mappings can be expressed as the length of $\int_0^\ell \langle \nabla f(c(t)), c'(t) \rangle dt$. In other words, the geodesic 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.

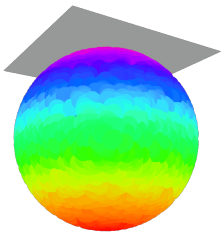

Figure 4: Tangent hyperplane for a point on the two-dimensional unit sphere. *Source:* own representation.

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 (Belkin and Niyogi, 2008). $\|\nabla f\|$ thus 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 minimizing $\int_{\mathcal{M}} \|\nabla f\|^2$. This is just equivalent to minimizing $\int_{\mathcal{M}} \mathcal{L}(f) f$ with the Laplace-Beltrami operator $\mathcal{L}$ (Belkin and Niyogi, 2003). For $\mathcal{L}f = \lambda f$, $f$ is an eigenfunction of $\mathcal{L}$ with $\lambda \in \mathbb{R}$ as its associated eigenvalue. These eigenfunctions are orthogonal and have real eigenvalues, making them natural candidates for 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).

**Finite approximation.** Now the same reasoning can be applied to the neighborhood graph approximation of $\mathcal{M}$. Mapping nearby inputs to nearby is achieved by assigning edge weights[7] $w_{ij} = \exp(\frac{1}{t}\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2), t \in \mathbb{R}$, if $\boldsymbol{x}_i, \boldsymbol{x}_j$ are are connected, and zero otherwise. Clearly, edges between closer points receive larger weights. The *adjacency matrix* $\boldsymbol{D} = (d)_{ij} \in \mathbb{R}^{N\times N}$ takes the row sums of the *weight matrix* $\boldsymbol{W} = (w)_{ij} \in \mathbb{R}^{N\times N}$ on its diagonal. Penalizing output disparities more severely for pairs of nearby points, 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. With $\boldsymbol{L}$ the above can be rewritten as generalized eigenvalue problem, adhering to the LGML algorithmic concept:

$$\min_{\mathcal{Y}} trace(\mathcal{Y}^T \boldsymbol{L} \mathcal{Y}), \quad \text{s.t. } \mathcal{Y}^T \boldsymbol{D} \mathcal{Y} = \boldsymbol{I}, \tag{1}$$

which is solved by eigendecomposition of $\boldsymbol{L}$ (Belkin and Niyogi, 2003). Analogous to the continuous case, the bottom eigenvector with zero eigenvalue is constant and must be discarded[8]. The subsequent $d$ eigenvectors hold the desired low-dimensional embedding coordinates (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 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, again hinting at the important role of neighborhood size.

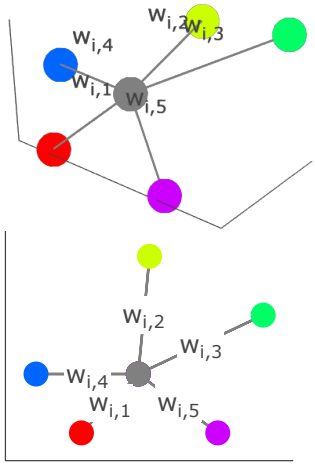


Figure 5: Reconstruction in three (*top*) and two (*bottom*) dimensions. *Source:* own representation.

[7] These weights stem from the heat kernel intimately related to the Laplace-Beltrami operator and ensure positive semi-definiteness of the resulting graph Laplacian (Belkin and Niyogi, 2003).

[8] As a consequence of its definition, $\boldsymbol{L}$ always has at least one eigenpair consisting of a zero eigenvalue and a constant eigenvector. In fact, the multiplicity of the zero eigenvalue corresponds to the number of connected graph components (Marsden, 2013).

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[9] (Roweis and Saul, 2000):

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

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 2 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):

$$\begin{aligned} \min_{\boldsymbol{W}} \varepsilon(\boldsymbol{W}) &= \min_{\boldsymbol{W}} \sum_i \left\| \boldsymbol{x}_i \mathbf{1}^T \boldsymbol{w}_i - \boldsymbol{N}_i \boldsymbol{w}_i \right\|^2 = \min_{\boldsymbol{W}} \sum_i \boldsymbol{w}_i^T (\boldsymbol{x}_i \mathbf{1}^T - \boldsymbol{N}_i)^T (\boldsymbol{x}_i \mathbf{1}^T - \boldsymbol{N}_i) \boldsymbol{w}_i \\ &= \min_{\boldsymbol{W}} \sum_i \boldsymbol{w}_i^T \boldsymbol{G}_i \boldsymbol{w}_i, \quad \text{s.t. } \mathbf{1}^T \boldsymbol{w}_i = 1 \quad \forall i \in \{1, 2, ..., N\}. \end{aligned} \tag{3}$$

By standard use of a Lagrange multiplier, the solution for the above constrained optimization problem collapses to $\boldsymbol{w}_i = \frac{\boldsymbol{G}_i^{-1} \mathbf{1}}{\mathbf{1}^T \boldsymbol{G}_i^{-1} \mathbf{1}}$. 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).

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

$$\begin{aligned} \min_{\mathcal{Y}} \Phi(\mathcal{Y}) = &\min_{\mathcal{Y}} \sum_i \left\| \boldsymbol{y}_i - \sum_j w_{ij} \boldsymbol{y}_j \right\|^2, \\ &\text{s.t. } \frac{1}{N} \sum_i \boldsymbol{y}_i \boldsymbol{y}_i^T = \boldsymbol{I} \quad \text{and} \quad \sum_i \boldsymbol{y}_i = \mathbf{0} \quad \forall i \in \{1, 2, ..., N\}. \end{aligned} \tag{4}$$

The objective can again be 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), yielding:

$$\min_{\tilde{\mathcal{Y}}} \mathit{trace}(\tilde{\mathcal{Y}}^T \boldsymbol{E} \tilde{\mathcal{Y}}), \quad \text{s.t. } \frac{1}{N} \tilde{\mathcal{Y}}^T \tilde{\mathcal{Y}} = \boldsymbol{I} \quad \text{and} \quad \tilde{\mathcal{Y}}^T \mathbf{1} = \mathbf{0}. \tag{5}$$

[9]Note that the weight matrix $\boldsymbol{W}$ is different from the one computed in LEM.

Again, the solution is found by eigenanalysis. 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 (Ghojogh et al., 2020). As mentioned before, the close resemblance to the optimization problem in LEM (equation 1) is not coincidental. 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 $\mathcal{L}$.

### 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 more akin to LLE in an algorithmic sense[10] (Ross, 2008). As opposed to LLE, HLLE is built upon a rigid theoretical foundation. It makes the relatively weak assumptions of local isometry and homeomorphicity to an open, connected subset of $\mathbb{R}^d$, providing veritable convergence guarantees, albeit only for the continuous limit (Donoho and Grimes, 2003).

**Idea.** HLLE considers the same twice-differentiable mapping functions $f : \mathcal{M} \to \mathbb{R}$ as 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[11] function on $\mathcal{M}$, the Hessian vanishes if and only if $f$ is linear (Ross, 2008).

**Continuous justification.** 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 $\boldsymbol{x}^{\mathrm{loc},\,\boldsymbol{p}'}$. Then, the Hessian $\boldsymbol{H}_f^{\mathrm{loc}}(\boldsymbol{p})$ of $f$ at $\boldsymbol{p}$ in tangent coordinates may be expressed as the ordinary Hessian of a function $g : U \to \mathbb{R}$ with $f(\boldsymbol{p}') = g(\boldsymbol{x}^{\mathrm{loc},\,\boldsymbol{p}'})$, $U$ being a neighborhood of zero in $\mathbb{R}^d$ (Donoho and Grimes, 2003):

$$(\boldsymbol{H}_f^{\mathrm{loc}}(\boldsymbol{p}))_{i,j} = \left.\frac{\partial^2 g(\boldsymbol{x}^{\mathrm{loc},\,\boldsymbol{p}'})}{\partial x_i^{\mathrm{loc},\,\boldsymbol{p}'}\partial x_j^{\mathrm{loc},\,\boldsymbol{p}'}}\right|_{\boldsymbol{x}^{\mathrm{loc},\,\boldsymbol{p}'}=0}. \tag{6}$$

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 of linear functions. After discarding the bottom constant function corresponding to a zero eigenvalue, the subsequent $d$ eigenfunctions span the desired low-dimensional embedding space (Donoho and Grimes, 2003). First, however, the dependency on the respective local coordinate systems must be removed by taking the Frobenius norm of the tangent Hessians[12]. Then, $\mathscr{H}(f)$ as a measure for overall curviness of the mapping is given by (Donoho and Grimes, 2003):

$$\mathscr{H}(f) = \int_{\mathcal{M}} \left\|\boldsymbol{H}_f^{\mathrm{loc}}(\boldsymbol{p})\right\|_F^2 d\boldsymbol{p}. \tag{7}$$

[10] 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)).

[11] 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)).

[12] For any alternative coordinate system, $\boldsymbol{H}'$ as obtained by orthogonal transformation of $\boldsymbol{H}$ with a suitable matrix $\boldsymbol{B}$, it must hold that $\|\boldsymbol{H}'\|_F^2 = \left\|\boldsymbol{B}\boldsymbol{H}\boldsymbol{B}^T\right\|_F^2 = \mathit{trace}(\boldsymbol{B}\boldsymbol{H}^T\boldsymbol{B}^T\boldsymbol{B}\boldsymbol{H}\boldsymbol{B}^T) = \mathit{trace}(\boldsymbol{H}^T\boldsymbol{H}) = \|\boldsymbol{H}\|_F^2$, due to the permutation invariance of the trace operator (Ross, 2008).

**Finite approximation.** In analogy to LEM, the functional defined on $\mathcal{M}$ is approximated in an empirical manner; yet, the computations are somewhat more involved. LEM incorporates neighborhood information during weight computation. LLE 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 is neighborhood construction. 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. From these neighborhood matrices the local tangent coordinates are 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 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):

$$\mathcal{H} = \sum_i \sum_j \mathcal{H}_{ij}, \quad \mathcal{H}_{ij} = \sum_\ell \sum_m (\boldsymbol{H}_\ell)_{m,i}(\boldsymbol{H}_\ell)_{m,j}. \tag{8}$$

Eventually, eigenanalysis of $\mathcal{H}$ yields the approximate null space spanned by the $d$ bottom eigenvectors after discarding the constant one. The final step consists of finding a basis for the null space. For this, take $\boldsymbol{Q} \in \mathbb{R}^{N\times d}$ 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 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 only roughly sketched above, it becomes clear that the theoretical guarantees of HLLE come at the expense of rather complex computations[13]. 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 application of HLLE (Cayton (2005), Ye and Zhi (2015)).

## 4.2 Semi-Supervised Locally Linear Embedding (SSLLE)

**Idea.** All of the above methods operate in an 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) propose to anchor the low-dimensional representation in LLE at a number $m \in \mathbb{N}$ of prior points whose coordinates in $\mathbb{R}^d$ are already known. Figure 6 hints at how such prior knowledge, illustrated by points in black, might help to improve the embedding. 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 by information extracted from the pool of unlabeled data, an inherently unsupervised problem is alleviated by specifying part of the solution upfront.

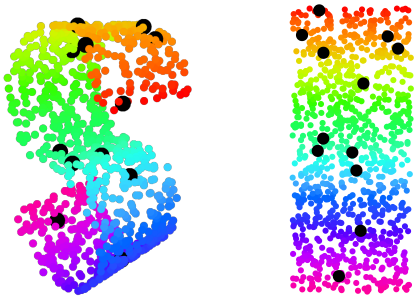

Figure 6: S-curve with 12 randomly sampled prior points. *Left:* prior point locations in three-dimensional observation space. *Right:* locations in true $\mathbb{R}^2$ embedding. *Source:* own representation.

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

**Prior point location.** Obviously, semi-supervision commands the availability of prior information. The practical experiments in chapter 5 will assume a setting where prior information can be inquired from the pool of initially unlabeled observations. A straightforward approach would be to select the prior points in a way that is most informative to the learning algorithm. As this is, of course, not always possible, the sensitivity analysis will examine the impact of such optimal selection versus scenarios of poor landmark choice and sampling uniformly at random:

Figure 7: S-curve with poor (*left*), random (*middle*), and optimal (*right*) coverage by 12 prior points. *Source:* own representation.

**Types of prior information.** Depending on the application, the prior information may be exact or inexact. While the first case greatly simplifies the manifold learning problem, the latter must be treated with some more care, or the semi-supervision will actually be harmful. Besides their location, the reliability of prior points therefore seems crucial to the embedding, which is why the second part of the sensitivity analysis in chapter 5 will assess the robustness of SSLLE to varying levels of label noise.

**Algorithmic impact.** Prior knowledge enters only in the second phase of the LLE algorithm. First, the reconstruction weights are computed as usual (see equation 2). The eigenvalue problem of minimizing embedding cost, by contrast, has a different nature: the matrices $\boldsymbol{E}$ and $\tilde{\mathcal{Y}}$ are partitioned into parts corresponding to known and unknown points, respectively[14]. This leads to a system of linear equations whose solution depends on the purity of prior information. If it is exact, the minimization problem collapses to optimizing over the unknown coordinates, denoted by $\tilde{\mathcal{Y}}_2 \in \mathbb{R}^{m \times d}$ (Yang et al., 2006):

$$\min_{\tilde{\mathcal{Y}}_2} \begin{bmatrix} \tilde{\mathcal{Y}}_1 & \tilde{\mathcal{Y}}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{E}_{11} & \boldsymbol{E}_{12} \\ \boldsymbol{E}_{21} & \boldsymbol{E}_{22} \end{bmatrix} \begin{bmatrix} \tilde{\mathcal{Y}}_1^T \\ \tilde{\mathcal{Y}}_2^T \end{bmatrix} \quad \Leftrightarrow \quad \tilde{\mathcal{Y}}_2^T = \boldsymbol{E}_{22}^{-1} \boldsymbol{E}_{12} \tilde{\mathcal{Y}}_1^T. \tag{9}$$

Equation 9 reflects the importance of prior points noted before: as errors exist only in $\boldsymbol{E}_{12} = \boldsymbol{E}_{21}$ and $\boldsymbol{E}_{22}$, the condition number $\kappa(\boldsymbol{E}_{22})$ of $\boldsymbol{E}_{22}$ multiplied by the relative errors in the off-diagonal blocks acts as an upper bound on the relative embedding error. For a sufficiently large number of observations, it can be shown that $\kappa(\boldsymbol{E}_{22})$ is minimal if the prior points are maximally scattered across the manifold surface, which is intuitively plausible (Yang et al., 2006).

For inexact prior information, the problem is slightly more complicated. The label noise commands the introduction of a regularizing term that penalizes deviations of the assumed coordinates (denoted by the hat symbol) from the true ones. The associated regularization parameter $\beta > 0$ encodes the level of confidence in the prior points (Yang et al., 2006):

$$\begin{aligned} \min_{\tilde{\mathcal{Y}}_1, \tilde{\mathcal{Y}}_2} \begin{bmatrix} \tilde{\mathcal{Y}}_1 & \tilde{\mathcal{Y}}_2 \end{bmatrix} \begin{bmatrix} \boldsymbol{E}_{11} & \boldsymbol{E}_{12} \\ \boldsymbol{E}_{21} & \boldsymbol{E}_{22} \end{bmatrix} \begin{bmatrix} \tilde{\mathcal{Y}}_1^T \\ \tilde{\mathcal{Y}}_2^T \end{bmatrix} + \beta \left\| \tilde{\mathcal{Y}}_1^T - \hat{\tilde{\mathcal{Y}}}_1^T \right\|_F^2 \quad \Leftrightarrow \\ \begin{bmatrix} \boldsymbol{E}_{11} + \beta \boldsymbol{I} & \boldsymbol{E}_{12} \\ \boldsymbol{E}_{21} & \boldsymbol{E}_{22} \end{bmatrix} \begin{bmatrix} \tilde{\mathcal{Y}}_1^T \\ \tilde{\mathcal{Y}}_2^T \end{bmatrix} = \begin{bmatrix} \hat{\tilde{\mathcal{Y}}}_1^T \\ \boldsymbol{0} \end{bmatrix}. \end{aligned} \tag{10}$$

Clearly, for the exact case, $\hat{\tilde{\mathcal{Y}}}_1^T = \tilde{\mathcal{Y}}_1^T$ and equation 10 again reduces to equation 9.

[14] Without loss of generality, the prior points are assumed to be the first $m$ ones.

## 4.3 Particular Challenges

**Amount of and quality of prior knowledge.** As hinted above, the location of prior points must be assumed to have vital impact on the quality of embedding coordinates, and so should their number. Likewise, the reliability of prior information ought to be a critical factor: while the case of exact knowledge is purely beneficial, inexact knowledge will provide less reliable guidance and might even prove harmful. The confidence parameter $\beta$ might also be expected to determine embedding quality. However, it emerges in the practical implementation that results are surprisingly stable for varying levels of $\beta$, which is why the effort has instead been dedicated to address the amount and quality of prior knowledge.

**Choice of intrinsic dimensionality.** Until now, it has been assumed that the intrinsic dimension $d$ of the data is a known parameter. This is obviously not always the case. Some methods offer the advantage of estimating $d$ in a built-in fashion. PCA, for instance, typically shows an indicative gap in its eigenvalue spectrum (Saul et al., 2006). For the methods covered here, no such tell-tale gap exists[15]. 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 the graph approximation, on the other hand, 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 all advantages of locality and non-linearity (de Ridder and Duin, 2002). A tuning step for $k$ has been directly integrated in the SSLLE implementation. For details, please refer to section A.4.

**Robustness of eigendecomposition.** Lastly, the inversion of the Gram matrix required for reconstruction error minimization in (SS)LLE frequently suffers from singularity. This problem has been noted by Roweis and Saul (2000) without offering a specific remedy. The implementation uses a small additive constant $\gamma > 0$ to strengthen the main diagonal, as is standard practice in numerical optimization (Ghojogh et al., 2020). $\gamma$ is computed as the sum over the eigenvalues of the local Gram matrix, multiplied by a tolerance parameter set to 0.01, following a proposal by Grilli (2007). In fact, the embedding is rather sensitive to regularization (de Ridder and Duin, 2002); so, even though it is beyond the scope of this report to specifically address the issue in detail, it would be certainly worthwhile.

# 5 Experimental Results

## 5.1 Experimental Design

### 5.1.1 Sensitivity Analysis

Foo

[15] While Sha and Saul (2005) have drawn a mathematical relation between the eigenspectra in LLE and LEM and intrinsic 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)).

## 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 Formal Definition of Topological Concepts

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[16] 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 $\epsilon$-balls containing all elements within a Euclidean distance of $\epsilon$ 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*[17] (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[18] (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 $\mathbb{R}^d$.

[16] Alternative definitions employ open subsets of $T$, see for example Waldmann (2014).
[17] For locally Euclidean topological spaces it is thus meaningful to speak of elements as points.
[18] $\mathcal{M}$ 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*[19] and *connectedness*[20] (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

$$c(\lambda) = (c_1(\lambda), ..., c_d(\lambda))^T \tag{11}$$

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 $\lambda$ yields *velocity* in $\lambda$:

$$c'(\lambda) = (c_1'(\lambda), ..., c_d'(\lambda))^T. \tag{12}$$

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):

$$d^{\mathcal{M}}(\boldsymbol{p}, \boldsymbol{q}) = \inf_{c \in \mathcal{C}(\boldsymbol{p}, \boldsymbol{q})} L(c). \tag{13}$$

[19] 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).

[20] 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 Formal Definition of Eigenanalysis and Generalized Eigenvalue Problems

**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. $\lambda$ is said to be an eigenvalue to $\boldsymbol{A}$ if there exists $\boldsymbol{v} \in \mathbb{R}^N \setminus \{\boldsymbol{0}\}$ such that $\boldsymbol{A}\boldsymbol{v} = \lambda\boldsymbol{v}$. Then, $\boldsymbol{v}$ is the eigenvector corresponding to the eigenvalue $\lambda$, 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 $\boldsymbol{0}$ upon multiplication from the right: $\{\boldsymbol{v} \in \mathbb{R}^N : \boldsymbol{A}\boldsymbol{v} = \boldsymbol{0}\}$. 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 $\lambda$ 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

$$\boldsymbol{A}\boldsymbol{V} = \boldsymbol{B}\boldsymbol{V}\boldsymbol{\Lambda}, \tag{14}$$

where $\boldsymbol{V} = \begin{bmatrix} \boldsymbol{v}_1 & \boldsymbol{v}_2 & \dots & \boldsymbol{v}_N \end{bmatrix} \in \mathbb{R}^{N \times N}$ is the matrix of eigenvectors of $\boldsymbol{A}$, and $\boldsymbol{\Lambda} = \mathit{diag}(\lambda_1, \lambda_2, \dots, \lambda_N) \in \mathbb{R}^{N \times N}$ is the diagonal matrix of the associated eigenvalues with $\lambda_1 \leq \lambda_2 \leq \dots \leq \lambda_N$. The generalized eigenvalue problem may be stated equivalently as

$$\min_{\boldsymbol{V}} \mathit{trace}(\boldsymbol{V}^T\boldsymbol{A}\boldsymbol{V}), \quad \text{s.t.} \quad \boldsymbol{V}^T\boldsymbol{B}\boldsymbol{V} = \boldsymbol{I}, \tag{15}$$

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

## A.3 Formal Definition of $k$- and $\epsilon$-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 are two principal ways to build a neighborhood around $\boldsymbol{x} \in \mathcal{X}$, both of which usually employ the squared Euclidean norm[21] $\|\cdot\|^2$. 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 $k \in \mathbb{N}$ points[22] with the smallest distance to $\boldsymbol{x}$, such that $\ell = k$ and

$$\mathcal{N}_k(\boldsymbol{x}) = \{\boldsymbol{x}_j \in \mathcal{X} : \|\boldsymbol{x} - \boldsymbol{x}_j\|^2 \leq d_{(k)}\}, \tag{16}$$

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 distance of less than $\epsilon \in \mathbb{R}^+$ to $\boldsymbol{x}$, yielding

$$\mathcal{N}_\epsilon(\boldsymbol{x}) = \{\boldsymbol{x}_j \in \mathcal{X} : \|\boldsymbol{x} - \boldsymbol{x}_j\|^2 < \epsilon\} \tag{17}$$

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

## A.4 Optimal Choice of $k$ in SSLLE Implementation

The tuning procedure for choosing the optimal neighborhood size in the SSLLE implementation builds upon a proposal by Kouropteva et al. (2001). Its key idea is to select $k$ at minimum computational expense, most of which is caused by embedding cost minimization. First, the less costly reconstruction weight computation is carried out for a range of candidate values. The maximum number $k_{\max} \in \mathbb{N}$ to try must still be specified as a hyperparameter, but with arbitrary resources, there is no limit besides the practical $k_{\max} \leq N-1$. This yields an empirical distribution of reconstruction errors over the given range of $1, 2, \ldots, k_{\max}$. The most promising candidates for $k$ emerge as the local minima[23] of the distribution, i.e., whenever the error is lower for a certain $k$ than for the immediate successor and predecessor values. For the selected subset, the actual (expensive) embeddings are calculated and evaluated[24] (Kouropteva et al., 2001).
However, in contrast to what Kouropteva et al. (2001) proposed, embeddings are not evaluated on residual variance. The area under the $R_{NX}$ curve (Kraemer et al., 2019), which is considered a more reliable measure, is used instead. Details on $R_{NX}$ and the corresponding AUC may be found in section A.5.

[21] In principle, alternative metrics are applicable, for instance such that measure angles (Belkin and Niyogi, 2004).
[22] In presence of ties in pairwise distances $k$ may vary across the data, but with zero probability in continuous feature spaces.
[23] While weight reconstruction for a fixed $k$ is a convex optimization problem, finding the minimum error for a range of $k$ values is not.
[24] Again, this is a non-convex, blackbox problem.

## A.5 Area under the $R_{NX}$ Curve

The area under the $R_{NX}$ curve, AUC($R_{NX}$), has been chosen as evaluation criterion to assess embedding quality. It is based on the *co-ranking matrix* of high-dimensional and low-dimensional coordinates and measures the degree of neighborhood preservation during the embedding (Kraemer et al., 2019).

**Co-ranking matrix.** The co-ranking matrix $\boldsymbol{Q} = (q)_{\ell m} \in \mathbb{R}^{N \times N}$ compares neighborhood relations in observation and embedding space. Consider the rank distances matrices $(r)_{ij}^{\text{obs}}, (r)_{ij}^{\text{emb}} \in \mathbb{R}^{N \times N}$, stating for the element in the $i$-th row and $j$-th column that the $j$-th observation ranks $(r)_{ij}$ among the nearest neighbors of the $i$-th observation in the respective space (Lueks et al., 2011). Any suitable distance metric is admissible; here, squared Euclidean distances are used. The co-ranking matrix then counts for each pair of ranks $(\ell, m)$, $\ell, m \in \{1, 2, ..., N\}$, for how many pairs of observations $(i, j)$, $i, j \in \{1, 2, ..., N\}$, the rank in the embedding space distance matrix equals $\ell$ and the rank in the observation space distance matrix equals $m$:

$$q_{\ell m} = \left|\{(i,j) : r_{ij}^{\text{emb}} = \ell \wedge r_{ij}^{\text{obs}} = m\}\right|. \tag{18}$$

Clearly, an ideal embedding would preserve all ranks and only have positive entries on the diagonal, i.e., where $\ell = m$. If, by contrast, most non-zero entries are located on the upper triangle, the embedding has torn apart points that lie close in the observation space, and, vice versa, if it is mostly the lower triangle containing non-zero entries, faraway points have been collapsed together (Lueks et al., 2011).

**Co-ranking-based metrics.** Based on $\boldsymbol{Q}$, various metrics may be derived. The $Q_{NX}$ criterion counts the (normalized) number of points that remain in the $k$-neighborhoods they form part of in $\mathbb{R}^D$, depending on the choice of neighborhood size:

$$Q_{NX}(k) = \frac{1}{kN} \sum_{\ell=1}^{k} \sum_{m=1}^{k} q_{\ell m}. \tag{19}$$

Adjusting for random embeddings and again normalizing to a zero-one range yields the $R_{NX}$ criterion (Kraemer et al., 2019):

$$R_{NX}(k) = \frac{(N-1)Q_{NX}(k) - k}{N - 1 - k}. \tag{20}$$

**Area under the $R_{NX}$ curve.** Plotting $R_{NX}$ versus the number $k$ of neighbors yields the sought-for $R_{NX}$ curve, the area under which serves as a parameter-free measure of embedding quality (Kraemer et al., 2019):

$$\text{AUC}(R_{NX}) = \frac{\sum_{k=1}^{N-2} R_{NX}(k)}{\sum_{k=1}^{N-2} 1/k} \in [0, 1]. \tag{21}$$

As is common with AUC metrics, the maximum value of 1 corresponds to the optimal value, whereas 0 marks an entirely random embedding (Kraemer et al., 2019).

## A.6 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

The entire code base, including the SSLLE implementation as well as the code used to conduct the practical analyses and generate all supporting figures, may be found in the public repository `https://github.com/lisa-wm/manifold-lle`.

# 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.

Grilli, E. (2007). *Automated Local Linear Embedding with an Application to Microarray Data*, PhD thesis, Università di Bologna.

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*.

Kouropteva, O., Okun, O. and Pietikäinen, M. (2001). Selection of the optimal parameter value for the locally linear embedding algorithm, *Proceedings of the 1st International Conference on Fuzzy Systems and Knowledge Discovery*.

Kraemer, G., Reichstein, M. and Mahecha, M. D. (2019). dimred and coranking -— unifying dimensionality reduction in r, *Technical report*.

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*.

Lueks, W., Mokbel, B., Biehl, M. and Hammer, B. (2011). How to evaluate dimensionality reduction? improving the co-ranking matrix, *arXiv: Machine Learning* .

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

Marsden, A. (2013). Eigenvalues of the laplacian and their relationship to the connectedness of a graph.

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.

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*.

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.