Seminar Report

Applying Semi-Supervised Locally Linear Embedding

Department of Statistics
Ludwig-Maximilians-Universität München

By Lisa Wimmer
Under the supervision of Jann Goschenhofer, Ph.D.
Munich, month dayth, 2021

Abstract

Storyline

- Goal: present SS-LLE as a local, graph-based manifold learning method incorporating prior knowledge
- Step 0: define basic mathematical concepts required to understand argumentation (plus notation)
- Step 1: introduce idea of **isometry** (most basic: MDS)
- Step 2: introduce idea of **graph-based** models
  - Achieve non-linearity
  - Common structure: build graph → derive matrix as quadratic form over graph function → derive embedding from eigenvalue problem
  - Most basic: ISOMAP (global, dense, convex)
- Step 3: introduce idea of **locality**
  - Relax global to local isometry
  - Find sparse rather than dense matrices
  - **Laplacian eigenmaps** as concept in which the others can be generalized
    - Define weighting scheme for neighborhood
    - Use Laplacian to derive matrix
    - Solve sparse eigenvalue problem
- Step 4: introduce **local linearity**
  - **LLE**
    - Obtain weights via linear reconstructions
    - Can be shown to approximate graph Laplacian (Belkin & Niyogi (2006))
  - **Hessian LLE**
    - Replace Laplacian by Hessian
- Step 5: introduce **prior knowledge**
  - **SS-LLE**
  - Improve results by pre-specifying some manifold coordinates

Extended Abstract

The goal of this report is to lay out the theoretical framework behind the manifold learning technique of *semi-supervised locally linear embedding (SS-LLE)*, as proposed by Yang et al. (2006), and to put it to implementation for data sampled from manifolds.
Manifold learning in general is concerned with dimensionality reduction. As data analysis employs increasingly high-dimensional data, it is frequently necessary to scale down the number of features to ensure models work as desired and remain interpretable. Dimensionality reduction is justified by the assumption that data observed in D dimensions often truly lie on a d-dimensional manifold (d-manifold), i.e., the d-dimensional generalization of a curved surface, embedded in $\mathbb{R}^D$ (with $d \ll D$). As an example for this phenomenon one might consider image data showing objects in different poses. While images are typically stored in high-dimensional pixel representations, intuitively, it is in fact a very small number of features causing the variation in the data.
A crucial property of d-manifolds embedded in $\mathbb{R}^D$ is their local topological equivalence to $\mathbb{R}^d$. This locally Euclidean behavior is exemplified by a sphere embedded in $\mathbb{R}^3$: although the sphere as a whole is entirely non-linear, on sufficiently small patches of its surface it behaves just like a flat plane in $\mathbb{R}^2$. It is precisely this fact that allows manifold coordinates to be mapped to $\mathbb{R}^d$ in a reduction of dimensionality. The goal is now to learn this mapping in an unsupervised manner. Mapping manifold coordinates to $\mathbb{R}^d$ is in general not equivalent to simple projection onto the d-dimensional coordinate hyperplanes. Instead, models must learn the intrinsic neighborhood structure of the manifold to establish a notion of "nearness" between points. As the sphere example demonstrates, standard distance metrics do not apply here since points on general manifolds are connected by curved paths rather than straight lines.
Some manifold learning techniques try to retain global isometry by mapping pairwise distances to $\mathbb{R}^d$. For instance, *multi-dimensional scaling (MDS)* does so using Euclidean distances, thereby limiting the manifolds it can learn to linear ones, while *ISOMAP* generalizes this approach to non-linear manifolds by applying geodesic distances. Research indicates, however, that for non-convex manifolds it is more effective to preserve local structures only. Otherwise, solutions are prone to shortcuts, i.e., placing points close in $\mathbb{R}^D$ next to each other when they lie in fact on quite different parts of the manifold. In order to avoid such miscalculations, sparse techniques focus on merely local neighborhood structures, modeled through graph representations. The information from these graphs is then condensed into a sparse matrix. *Laplacian eigenmaps*, a method to which SS-LLE can be generalized, employs the graph Laplacian to this end. It then solves an eigenvalue problem that yields the orthogonal eigenvectors on which the d-dimensional coordinates are located. While Laplacian eigenmaps do well in preserving locality, they are less adept at determining local linearity. This shortcoming is mitigated by *locally linear embedding (LLE)*, the fundamental technique SS-LLE hails from, and *Hessian LLE (H-LLE)*, a further variant of LLE which uses the Hessian en lieu of the Laplacian.
The original LLE approximates the Laplacian by a weight matrix obtained through linearly reconstructing points from their neighbors. Since these weights are believed to reflect the intrinsic geometry of the manifold, they are topological properties

and as such invariant to rotations, rescalings and translations. By consequence, the same weights that reconstruct an individual point in D dimensions should do so in d dimensions. LLE first computes the optimal weights for reconstructing points in $\mathbb{R}^D$ by solving a least-squares problem and then finds the $\mathbb{R}^d$ coordinates that best preserve these weights by solving a sparse eigenvalue problem. Convexity of both sub-problems ensures globality of local optima.
This approach has been shown to successfully retrieve manifold structures in different applications. However, its fully unsupervised functionality offers a drawback: it may fail to find a low-dimensional embedding that has an actual reflection in the real-life setting. Such situations might require the specification of some pre-labeled instances. Also, it may simply be the case that manual analysis of a subset of the data is available at low cost.
When prior knowledge is at hand it is only natural to use it. Therefore, Yang et al. (2006) proposed SS-LLE as an extension to LLE that is able to harvest prior information. Both exact and inexact knowledge, the latter regularized with an uncertainty coefficient, are applicable. The information is incorporated in the second step of the algorithm by fixing some of the sought-for coordinates in advance. Perhaps unsurprisingly, Yang et al. (2006) find that careful selection of the prior points to be maximally scattered across the manifold surface works better than random sampling. Indeed, the presented results indicate considerable success of their technique.
It is the aim of this report to (1) reproduce these results, thereby creating an open-source implementation of SS-LLE, and (2) to apply SS-LLE to further manifold learning tasks for a more thorough assessment of its performance.

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 operations, interpretation of results, and geometric idiosyncrasies. Computational cost must be considered but is becoming less of an issue with technological evolution (Leist et al., 2009). By contrast, the demand for explainable results (for reasons of, say, safety or ethics) is rather intensified by the advance of complex technology. Alas, interpretation in more than a few dimensions is virtually inaccessible to humans (Doshi-Velez and Kim, 2017). The geometric aspect is often addressed as *curse of dimensionality*, a term subsuming various phenomena of high-dimensional spaces. It is generally not straightforward to infer properties of objects in these spaces as geometric intuition developed in two or three dimensions can be misleading. Crucially, the exponential increase of spatial volume induces sparsity. Consequences of this behavior are, among others, a sharp incline in the number of points required to sample the feature space and a loss in meaningfulness of distances. Many learners, however, rely on these concepts[1] and see their functionality deteriorate (Verleysen and Francois, 2005).
These challenges make the case for *dimensionality reduction*, that is, the endeavor of compressing problem dimensionality to a manageable size. Far from undue simplification, dimensionality reduction is justified by the idea that the latent data-generating process is indeed of much lower dimension than is observed. Consider, for example, image data showing objects in different poses. Such data are typically stored in high-dimensional pixel representations. It is, however, reasonable to suppose that variation in these images is in fact caused by a small number of latent features. More formally, the data are assumed to lie on a d-dimensional *manifold* embedded in the D-dimensional observation space, with $d \ll D$ (Cayton, 2005).
A crucial property of d-manifolds, i.e., the d-dimensional generalization of a curved surface, embedded in $\mathbb{R}^D$, is their local topological equivalence to $\mathbb{R}^d$ (Ma and Fu, 2011). It is precisely this fact that allows manifold coordinates to be mapped to $\mathbb{R}^d$ in a reduction of dimensionality[2]. The goal is now to learn this mapping in an unsupervised manner (Cayton, 2005). Mapping manifold coordinates to $\mathbb{R}^d$ is in general not equivalent to simple projection onto the d-dimensional coordinate hyperplanes. Instead, models must learn the intrinsic neighborhood structure of the manifold to establish a notion of "nearness" between points. Standard distance metrics do not apply here as points on general manifolds are connected by curved paths rather than straight lines (Ma and Fu, 2011).
Various approaches have been proposed to retrieve points' manifold coordinates. A taxonomy may, for example, be found in van der Maaten et al. (2009). Many rely on spectral techniques, trying to find a matrix representation of the data whose principal eigenvectors are used to span a d-dimensional subspace. One group of

[1] For instance, consider support vector machines and k-nearest neighbors, both of which rely on distances, or tuning, which often requires extensive sampling of the hyperparameter space.
[2] The most intuitive example of this is probably the representation of the Earth, which is a two-dimensional manifold enclosed in three-dimensional space, on two-dimensional maps.

spectral methods attempts to retain global isometry by mapping pairwise distances to $\mathbb{R}^d$. Among them, some are based on Euclidean distances and thus confined to learning linear embeddings (such as *principal component analysis (PCA)* or *multi-dimensional scaling (MDS)*). Since linearity is a strong assumption that will not hold for general manifolds, non-linear techniques are more widely applicable. For example, *ISOMAP* achieves non-linearity by applying geodesic distances (van der Maaten et al., 2009). Research indicates, however, that for non-convex manifolds it is more effective to preserve local structures only. Otherwise, solutions are prone to shortcuts, i.e., placing points close in $\mathbb{R}^D$ next to each other when they lie in fact on quite different parts of the manifold (Belkin and Niyogi, 2003). In order to avoid such miscalculations, sparse techniques focus on merely local neighborhood structures, modeled through graph representations. The information from these graphs is then condensed into a sparse matrix (van der Maaten et al., 2009). *Laplacian eigenmaps* employ the graph Laplacian to this end (Belkin and Niyogi, 2003). While Laplacian eigenmaps do well in preserving locality, they are less adept at determining local linearity, a property that comes with manifolds' being locally Euclidean (Ross, 2008) find better source!!.

This shortcoming is mitigated by locally linear methods. One such technique is *locally linear embedding (LLE)*, proposed by Roweis and Saul (2000). While it originated from a different perspective, it can be shown to be a variant of Laplacian eigenmaps (Belkin and Niyogi, 2003). LLE is based on the idea that neighborhoods on the manifold are locally linear and so points can be linearly reconstructed from their neighbors. Since the corresponding weights are believed to reflect the intrinsic geometry of the manifold, they are topological properties and as such invariant to rotations, rescalings, and translations. By consequence, the same weights that reconstruct an individual point in D dimensions should do so in d dimensions. LLE thus maps vicinity structures to the d-dimensional subspace and finds the coordinates that preserve them best. This requires solving the least-squares problem of minimizing reconstruction error and then the sparse eigenvalue problem of minimizing embedding cost. Convexity of both sub-problems ensures globality of local optima (Roweis and Saul, 2000).

This approach has been shown to successfully retrieve manifold structures in different applications (evidence!!!). However, its fully unsupervised functionality offers a drawback: it may fail to find a low-dimensional embedding that has an actual reflection in the real-life setting. Such situations might require the specification of some pre-labeled instances. Also, it may simply be the case that manual analysis of a subset of the data is available at low cost (Yang et al., 2006). When prior knowledge is at hand it is only natural to use it. Therefore, Yang et al. (2006) proposed *semi-supervised locally linear embedding (SS-LLE)*, an extension to LLE that is able to harvest prior information.

Indeed, the presented results indicate considerable success of their technique. It is the aim of this report to (1) reproduce these results, thereby creating an open-source implementation of SS-LLE, and (2) to apply SS-LLE to further manifold learning tasks for a more thorough assessment of its performance. The rest of the report is organized as follows: chapter 2 provides a mathematical framework where fundamental concepts are briefly introduced; chapter 3 explains the idea of local graph-based manifold learning; chapter 4 presents SS-LLE in detail; chapter 5 discusses the results of the conducted experiments; and chapter 6 draws final conclusions.

2 Basic Manifold Theory

2.1 Concepts in Manifold Learning

This chapter introduces the main geometric concepts considered necessary to provide a solid understanding of SS-LLE[3]. It must be noted that everything discussed here is presented through the lens of machine learning, deliberately forsaking the generality inherent to topology. Therefore, assuming features can be represented by coordinates in D-dimensional Euclidean space, all concepts are examined with regard to their meaning in $\mathbb{R}^D$. Dimensionality reduction techniques take the data observed in $\mathbb{R}^D$ to actually lie in a d-dimensional topological space that is not necessarily Euclidean but exhibits some specific properties.

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[4] on T, the following properties must hold (Brown, 2006):

1. If $\mathcal{T}$ is a neighborhood of t, then $t \in \mathcal{T}$.
2. If $\mathcal{T}$ is a subset of T containing a neighborhood of t, then $\mathcal{T}$ is a neighborhood of t.
3. The intersection of two neighborhoods of t is again a neighborhood of t.
4. 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. The ultimate goal is again the interpretation of the data in a Euclidean space, albeit one with lower dimensionality, where such concepts are meaningful. The next step is thus to study how a (potentially highly non-linear) topological space might relate to $\mathbb{R}^d$.

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 is homeomorphic to an open subset of $\mathbb{R}^d$, S is *locally Eu-*

[3]Obviously, the list of concepts discussed is by no means extensive. Theory is presented much more in detail (and mathematical rigor) in, for example, good book.
[4]Alternative definitions employ open subsets of T, see for example Waldmann (2014).

clidean[5]. In local neighborhoods S then behaves like $\mathbb{R}^d$, which is conceivably a desirable property in the context of manifold learning (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[6] (also: d-manifold) it must meet the following conditions (Waldmann, 2014):

1. $\mathcal{M}$ is Hausdorff.
2. $\mathcal{M}$ is second-countable.
3. $\mathcal{M}$ is locally homeomorphic to $\mathbb{R}^d$.

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).
Manifolds can now be *embedded* in Euclidean space. Consider $\mathbb{R}^K \supset \mathbb{R}^d$[7]. $\mathbb{R}^d$ is endowed with the so-called *subspace topology* that results from intersecting open subsets of $\mathbb{R}^K$ with $\mathbb{R}^d$ (for $\mathbb{R}^2$, these are ϵ-circles obtained by intersecting $\mathbb{R}^3$-ϵ-balls with the coordinate planes). For a manifold $\mathcal{M}$, to be embedded in $\mathbb{R}^K$ means that $\mathcal{M}$ is enclosed by $\mathbb{R}^K$ but locally homeomorphic to $\mathbb{R}^d$, thereby inheriting the metric subspace topology from $\mathbb{R}^d$ (Waldmann, 2014). It can be shown that $K = 2d + 1$ is sufficient to create an embedding, but K may be smaller (Ma and Fu, 2011).
This now has important consequences for manifold learning: data lying on a d-dimensional manifold $\mathcal{M}$ embedded in $\mathbb{R}^D$ are observed as D-dimensional points but may locally be treated like points from $\mathbb{R}^d$.

Figure 1 shows the well-known *S-curve* manifold embedded in $\mathbb{R}^3$. Clearly, the S-curve as a whole is far from linear, but local patches on its surface behave like planes from $\mathbb{R}^2$. So the S-curve is two-dimensional and feature dimensionality can in effect be compressed from $\mathbb{R}^3$ to $\mathbb{R}^2$. The challenge is now to unravel the manifold in a way that preserves its structure to maximum extent. Obviously, a simple projection to $\mathbb{R}^2$ will not accomplish this task. Instead, manifold learning must capture the intrinsic neighborhood structures and map these to $\mathbb{R}^2$, which, in this case, can be imagined as a "flattening-out" of the S-curve.

Figure 1: 10,000 points sampled from the S-curve manifold. *Source:* own representation, inspired by implementation in `Python`'s `scikit-learn` library (Pedregosa et al., 2011).

[5]For locally Euclidean topological spaces it is thus meaningful to speak of elements as points.
[6]$\mathcal{M}$ is again a shorthand, omitting the explicit notation of the corresponding topology.
[7]Here, K is used to denote a general higher-dimensional space (a manifold embedded in $\mathbb{R}^K$ is also embedded in $\mathbb{R}^{K+1}$, and so on, as homeomorphisms are transitive (Waldmann, 2014)). This is deliberate to distinguish it from the more specific notation D indicating the number of observed features.

Geodesics. One last aspect remains open, namely how to handle distances on manifolds. Figure 1 illustrates that standard Euclidean distances are not meaningful here: rather than measuring "shortcuts" between points across $\mathbb{R}^3$ (where, for instance, points in the red upper area would be considered quite close to points in the cyan mid area), 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: being *Riemannian* and being *connected* (Ma and Fu, 2011). The Riemannian property is based on differentiability and ensures that concepts of curvature, length and angle remain meaningful (Ma and Fu, 2011).[8] Connectedness means that no separation $\{U, V\}$ of a manifold $\mathcal{M}$ exists with open, non-empty and disjoint $U, V \subset \mathcal{M}$, $\mathcal{M} = U \cup V$. For manifolds, connectedness is immediately equivalent to path-connectedness. Informally stated, any two points on a connected manifold can be linked by a path (McCleary, 2006).
For connected Riemannian manifolds it is now possible to define a distance metric, or *geodesic distance*. Geodesic distance is the length of the shortest curve (*geodesic*) on $\mathcal{M}$ between two points on $\mathcal{M}$, as measured by arc-length[9] (such a curve must exist due to connectedness). Intuitively, geodesic distance can be identified with Euclidean distance in Euclidean spaces where shortest curves are but straight lines (Ma and Fu, 2011).

2.2 Formal Goal of Manifold Learning

Building on the above concepts, the goal of manifold learning shall be formalized in a way that will be referenced throughout the remainder of this report and is inspired by Saul et al. (2006).

Given. Data $\boldsymbol{X} = (\boldsymbol{x}_1, \boldsymbol{x}_2, ..., \boldsymbol{x}_N)$, with $\boldsymbol{x}_i \in \mathbb{R}^D \; \forall i \in \{1, 2, ..., N\}$ and $N, D \in \mathbb{N}$. $\boldsymbol{X}$ thus consists of N real-valued data vectors observed in D dimensions. The true data-generating process is taken to have dimensionality $d \in \mathbb{N}$, such that $\boldsymbol{X}$ is in fact a sample from a d-dimensional, Riemannian and connected manifold: $\boldsymbol{X} \sim \mathcal{M} \subset \mathbb{R}^D$. For manifold learning methods to yield satisfying results, $\mathcal{M}$ is always assumed to be sampled well by $\boldsymbol{X}$.

Goal. Find an appropriate d-dimensional representation of the data, i.e., compute $\boldsymbol{Y} = (\boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_N)$, with $\boldsymbol{y}_i \in \mathbb{R}^d \; \forall i \in \{1, 2, ..., N\}$. Appropriateness of $\boldsymbol{Y}$ is determined by the extent to which the d-dimensional representation preserves intrinsic structures on $\mathcal{M}$.

Note that, while D is specified a priori, the *intrinsic dimensionality* d is typically unknown and must be estimated during learning. Since there are not always (finite-sample) guarantees for finding the true value of d, it must be assumed that the estimation may differ from d. In order to emphasize this potential gap, the dimensionality of the low-dimensional representation will henceforth be denoted by $m \in \mathbb{N}$, such that $\boldsymbol{Y} \in (\mathbb{R}^m)^N$.

[8] A detailed derivation is beyond the scope of this report and may be found, for example, in Mukherjee (2015).
[9] Geodesics are but a peripheral note here; for a precise definition see for example Ma and Fu (2011).

3 Local Graph-Based Manifold Learning

3.1 Principles of Local Graph-Based Manifold Learning

3.1.1 General Concept

(Local) graph-based techniques of manifold learning arise from varying geometric intuitions and computational approaches. Interestingly, they still share a common structure that allows for interpretation in a general framework. The first main block of this framework produces a matrix representation of the D-dimensional data which is then mapped to d dimensions in a subsequent step by means of spectral decomposition, such that the data are expressed via coordinates on the principal eigenvectors of the matrix (Saul et al., 2006). In fact, this procedure is not unique to graph-based methods but characterizes all spectral methods (Cayton, 2005). Precisely how the matrix is constructed, however, determines the kind of intrinsic structure that can be learned and preserved. As argued before, non-linearity is a desirable property that techniques based on linear data structures[10] fail to achieve. Graph-based methods derive their matrix from weighted neighborhood graphs. In effect, these graphs are a discretized approximation of the underlying manifold $\mathcal{M}$ (still assuming $\mathcal{M}$ is sampled well by $\boldsymbol{X}$), and as such able to capture highly non-linear structures. To this end, vicinities are expressed via graph representations (Saul et al., 2006). Local graph-based methods additionally restrict neighborhoods to local ones and thereby avoid shortcuts: for manifolds with strongly non-convex surfaces, global solutions often prove too coarse. The consequence of such locality is sparsity of graph matrices (Belkin and Niyogi, 2003).
Summing up the above, local graph-based manifold learning can be schematized as follows:

1. Compute neighborhoods of input data.
2. Construct a sparse weighted graph from these neighborhood relations.
3. Condense the graph information in a matrix.
4. Learn an embedding from the eigenvectors of this matrix.

Crucially, the complex manifold learning problem can be addressed by a sequence of tractable optimization steps (Saul et al., 2006). The following chapters explain these steps in some more detail.

3.1.2 Neighborhood Graphs

Approximating the intrinsic structure of $\mathcal{M}$ by a neighborhood graph requires the determination of a *neighborhood* around each $\boldsymbol{x} \in \boldsymbol{X}$. A neighborhood of $\boldsymbol{x}$ is but a subset of $\boldsymbol{X}$ containing another, open subset of $\boldsymbol{X}$ 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 now locally applies to manifolds as a direct consequence of their locally Euclidean behavior (Ma and Fu,

[10] For instance, principal component analysis (PCA) decomposes the covariance matrix, whereas multi-dimensional scaling (MDS) is based on the Gram matrix. Both approaches essentially perform a rotation of the data and a linear projection to the subspace of maximum variance, yielding the same results (Cayton, 2005).

2011). Neighborhood relations in local graph-based manifold learning may therefore established in the Euclidean observation space and are as such of linear nature.
There are two principal ways to build a neighborhood around $\boldsymbol{x} \in \boldsymbol{X}$, both of which usually employ squared Euclidean distances[11], denoted by $\|\cdot\|^2$. Let $\mathcal{N} : \boldsymbol{X} \to \boldsymbol{X}^\ell, \boldsymbol{x} \mapsto \mathcal{N}(\boldsymbol{x})$ 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 with the smallest distance to $\boldsymbol{x}$, such that $\ell = k$ and $\mathcal{N}_k(\boldsymbol{x}) = \{\boldsymbol{x}_j \in \boldsymbol{X} : \|\boldsymbol{x} - \boldsymbol{x}_j\|^2 \leq \gamma\}$, with $\gamma \in \mathbb{R}$ being the k-percentile of ordered pairwise distances. Alternatively, the neighborhood may be constructed by collecting all points that have a maximum distance of $\epsilon \in \mathbb{R}$ to $\boldsymbol{x}$, no matter how many neighbors this will result in, yielding $\mathcal{N}_\epsilon(\boldsymbol{x}) = \{\boldsymbol{x}_j \in \boldsymbol{X} : \|\boldsymbol{x} - \boldsymbol{x}_j\|^2 \leq \epsilon\}$ and $\ell = |\mathcal{N}_\epsilon(\boldsymbol{x})|$ (He et al., 2005).

The thus defined neighborhoods can now be described by a *neighborhood graph* $\mathcal{G}$, where input points form vertices and edges indicate neighborhood relations. 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. Building upon k-neighborhoods therefore leads to directed graphs (He et al., 2005). An example for such a directed graph is given by figure 2, showing 2-neighborhoods for seven fictional data points. Conversely, the ϵ-distance boundary holds in both directions and produces undirected graphs (He et al., 2005).

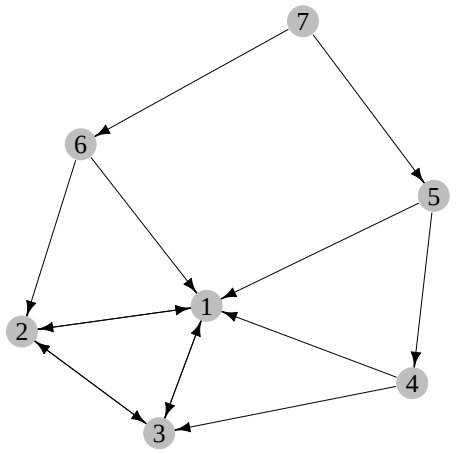

Figure 2: Exemplary neighborhood graph for seven points where outgoing arrows point to members of the vertex's respective k-neighborhood, $k = 2$. *Source:* own representation.

3.1.3 Matrix Representation of Neighborhood Graphs

- Degree and adjacency matrices
- Laplacian operators

3.1.4 Solving Eigenwertproblems

- Eigenvectors, eigenvalues
- Spectral decomposition

3.2 Techniques of Local Graph-Based Manifold Learning

3.2.1 Laplacian Eigenmaps

- Notion of locality

[11] In principle, alternative distance measures are equally applicable.

- Laplacian eigenmaps

3.2.2 Locally Linear Embedding (LLE)

- Notion of local linearity
- Approximation of graph Laplacian

3.2.3 Hessian Locally Linear Embedding (H-LLE)

- Hessian instead of Laplacian (eigenmaps)
- Hessian instead of LS fit (LLE)

4 Semi-Supervised Locally Linear Embedding (SS-LLE)

4.1 Employment of Prior Information

- Why use labels in the first place?
- How will that help?
- How do we even find prior points?
- Exact vs inexact knowledge

4.2 SS-LLE Algorithm

- What is different wrt standard LLE?

4.3 Strengths and Drawbacks of SS-LLE

Potential shortcoming: what if manifold is not well-sampled? Not a problem with synthetic data, but IRL. But probably problematic with all manifold approaches
Also: generalization to new points (w/o recomputing everything) neighborhood-preserving propositions

5 Experiment Results

5.1 Experimental Design

5.1.1 Software Implementation

5.1.2 Evaluation Framework

5.2 Replication of Original Results on the Incomplete Tire

5.2.1 Incomplete Tire Data

5.2.2 Choice of Hyperparameter and Prior Point Configuration

5.3 Application on Swiss Roll and Guglhupf Data

5.3.1 Swiss Roll and Guglhupf Data

5.3.2 Choice of Hyperparameter and Prior Point Configuration

5.4 Results and Discussion

6 Conclusion

Lorem ipsum

A Appendix

Lorem ipsum

B Electronic Appendix

Data, code and figures are provided in electronic form.

References

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

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

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

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

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

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

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

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

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

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. and Duchesnay, E. (2011). Scikit-learn: Machine learning in Python, *Journal of Machine Learning Research* **12**: 2825–2830.

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., 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.

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.

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

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.