School of Informatics

Informatics Research Review
Manifold learning: Techniques for non-linear dimensionality reduction

[redacted]

January 2021

Abstract

Manifold learning is an approach to non-linear dimensionality reduction that attempts to find a mapping from the high-dimensional feature space to the lower-dimensional space of underlying parameters. The past two decades has seen a dramatic improvement in the speed and effectiveness of manifold learning algorithms, leading to impactful research in fields as diverse as genomics, physics, materials science and medicine. In this review we begin by introducing the concept of manifold learning. We then survey and critique several of the most popular manifold learning algorithms and, by contrasting their behaviour, provide practical advice to the data scientist for algorithm selection.

Date: Friday 22$^{\text{nd}}$ January, 2021

Supervisor: [redacted]

1 Introduction

We live in a time of exploding demand for fast, powerful analysis of high-dimensional data, whether it be in uncovering geometric patterns in complex networks (social, computer, and transportation) [1], in determining camera orientation from multi-million dimensional image vectors [2], or in identifying disease-associated loci in million-long DNA sequences [3]. But despite the backdrop of burgeoning applications, working directly with high-dimensional data continues to present a challenge for data analysis: it may be computationally intractable, difficult to visualise, highly sparse, and the abundance of feature variables can often conceal the presence of a simpler underlying structure [4]. These challenges make it vital to be able to summarise such a dataset into a lower-dimensional representation before further analysis, a process known as dimensionality reduction (DR).

A rich history of research in machine learning and applied mathematics has given rise to dozens of distinct algorithms for DR, each with their own idiosyncrasies and application niches. Traditionally the most popular choices have been linear DR algorithms such as Principal Component Analysis (PCA) and Metric Multidimensional Scaling (MDS); owing their popularity to their ease of implementation, interpretability and computational speed [5]. However, these linear DR algorithms have one major deficiency which limits their applicability: the dimensionality reduction depends only on global properties of the data, making them blind to local geometry.[1]

In many practical problems, such as genomics and computer vision, the local geometry of data is highly informative and ought to be preserved by dimensionality reduction [6, 7]. In these contexts it is vital to use non-linear DR algorithms. Popular strategies for non-linear DR exploit the fact that the dimensionality of a dataset is often only artificially high; physical and geometric constraints mean that the high-dimensional features are expressible as a function of only a few underlying parameters. Geometrically this means that data are often constrained to lie on a low-dimensional manifold embedded in a high-dimensional feature space, an assumption known as the manifold hypothesis. Manifold learning, a non-linear dimensionality reduction techniques, is the process of identifying the map from this feature space to the space of underlying manifold parameters.

Although many algorithms for manifold learning have been developed, it is possible to roughly classify these based on their theoretical approach as either Manifold Projection Methods, Eigenmap Methods, or Topological Embedding Methods. In this review we will discuss the most popular manifold learning algorithms from each of these categories. The range of manifold learning algorithms that we discuss will not be entirely exhaustive, since several of these algorithms have multiple, closely related variants. We therefore focus predominantly on the original, unsupervised versions of these algorithms applied to continuous data. Other techniques for non-linear dimensionality reduction exist,[2] but we exclude detailed discussions of algorithms which do not make explicit use of the manifold hypothesis in their approach. Also note that several manifold learning algorithms build upon the linear dimensionality reduction techniques of PCA and MDS. We refer the interested reader to Cunningham and Ghahramani [5] for a more detailed review of these, and other, linear methods. For a review on the more general use of dimensionality reduction techniques in data analysis see Sorzano *et al.* [11].

We begin our review in section 2 with an overview of the mathematics, intuition and assumptions which underpin all manifold learning algorithms. In section 3 we then critically evaluate the algorithms from each of the three main theoretical approaches, examining how they compare in terms of underlying theory, computational complexity, domain of applicability, and quality of

[1] For example, note that with a linear method it is impossible to unambiguously uncover the structure of a helix, circle, or indeed *any* non-linear geometry, in a dataset.

[2] Prominent examples include the techniques of Self-Organizing Map [8], Generative Topological Mapping [9], and Stochastic Neighbour Embedding [10]

the resulting dimensionality reduction. Throughout this critical analysis we inform and advise the data scientist looking to implement a manifold-learning algorithm: which option should they choose? Does the best choice depend on the details of the task/dataset? On the available computational resources? On the convenience of implementation?

2 Preliminaries

2.1 Manifolds

At an intuitive level, a d-dimensional manifold can be considered to be a space which locally resembles d-dimensional Euclidean space, but whose global structure could be very different (for instance a helix, a torus, a sphere). The only restriction we impose on this global geometry is that it must be able to be *charted back* to d-dimensional Euclidean space.[3] We call this charted representation of the manifold the parameter space. Formulating these ideas more precisely,

Definition 1. A d-dimensional manifold $\mathcal{M}$ is a topological space that is locally homeomorphic with $\mathbb{R}^d$. That is, for each $x \in \mathcal{M}$, there exists an open neighborhood around x, N_x, and a homeomorphism[4] $f : N_x \to \mathbb{R}^d$. These neighborhoods are referred to as coordinate patches, and the map is referred to a a coordinate chart. The image of the coordinate charts is referred to as the parameter space.

All manifold learning algorithms impose smoothness requirements on the manifold, and this constrains the class of manifolds we consider to be differentiable manifolds:

Definition 2. A d-dimensional differentiable manifold is a d-dimensional manifold whose coordinate charts are differentiable with a differentiable inverse.

In practice we usually deal with manifolds which lie embedded in a higher dimensional space than their intrinsic dimension. In fact, often the embedding dimension may be truly vast. Consider for example video footage of a grandfather clock. The embedding dimension of the data is equal to the number of pixels (several million) but the intrinsic dimension is only one-dimensional, described by the angle of swing of the pendulum.

2.2 Manifold Learning

The *manifold hypothesis* is the heuristic observation that high-dimensional data often lie in lower-dimensional embedded manifolds which arise as a natural consequence of geometric or physical constraints on the data [12]. The existence of embedded manifolds implies that the intrinsic dimensionality of the data is often much lower than the feature space. Practical examples of embedded manifolds abound in high-dimensional data from images, speech, genomes, and other sources [7, 13, 14]. It is often pertinent to identify this manifold - either for the purposes of direct data visualization or for finding useful features for machine learning algorithms. We now proceed to formalise this task, known as the manifold learning problem (MLP):

[3] Consider for example the surface of the Earth. This surface can clearly be charted back to two-dimensional Euclidean space. You can even buy these charts in a shop - we call them an atlas. The Earth's surface is therefore an example of a two-dimensional manifold.

[4] Recall that a *homeomorphism* is a continuous function whose inverse is also a continuous function.

The Manifold Learning Problem (MLP): Given as input a set $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_k$ of k points in $\mathbb{R}^{\mathcal{D}}$ that lie on a d-dimensional manifold $\mathcal{M}$, with $d < \mathcal{D}$, then, assuming the manifold's parameter space can be mapped by a single coordinate chart,[5] $f : \mathcal{M} \to \mathbb{R}^d$, identify the set $\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_k$ of k points in $\mathbb{R}^d$, known as *the embedding representation*, such that $\mathbf{y}_i = f(\mathbf{x}_i), \ \forall i \in \{1, 2, ..., k\}$.

Over the past two decades many distinct algorithms have been devised to tackle the MLP. The main reason for the propensity of different approaches is that MLP is in fact a mathematically ill-posed problem - we do not know apriori what the correct low-dimensional representation of our data is, or whether this representation is unique. This inherent ambiguity has made it difficulty to construct a gold-standard metric on which to evaluate algorithm performance. A further complication is that real data may be noisy, often displaced away from the underlying manifold, or highly clustered, and this has given rise to different algorithmic approaches which make different underlying assumptions about the noise and sampling properties of the data.

It is from this background of challenges that three distinct approaches to MLP have emerged - Manifold Projection Methods, Eigenmap Methods, and Topological Embedding Methods - with each strategy giving rise to different computational efficiencies, theoretical guarantees, free parameter choices, and noise robustness. We now proceed to outline the theory that underpins these distinct approaches, critically examining and contrasting the most popular algorithms.

3 Manifold Learning Algorithms

3.1 Manifold Projection Methods

Isomap

We begin our exposition of manifold learning algorithms with Isomap, a type of manifold projection method. Introduced by Tenebaum *et al.* [16] in 2000, Isomap was one of the first manifold learning algorithms and has since become one of the best known and most widely applied - seeing numerous applications in biomedicine, climate physics, traffic networking and many others [14, 17, 18, 19].

The Isomap algorithm is underpinned by a linear dimensionality reduction technique known as classical Multidimensional Scaling (cMDS). Recall that in cMDS the task is: given as input a matrix $D \in \mathbb{R}^{n \times n}$, attempt to construct the set of n points $\{\mathbf{y}_i\}_{i=1}^{n}$ (where $\mathbf{y}_i \in \mathbb{R}^m$ and $m \leq n$) whose inter-point Euclidean distances match the entries of D as closely as possible.[6] An underlying assumption made by cMDS is that the data are confined to a linear subspace of $\mathbb{R}^n$, but this assumption is invalid whenever data lie on a non-linear manifold. Isomap manages to overcome this hurdle by encoding information of the non-linear nature of the manifold in the dissimilarity matrix D. Specifically this is done by constructing D from estimates of inter-point geodesic distances, i.e. shortest-path distances along the manifold, before applying cMDS. The precise algorithic procedure is as follows,

[5] All the manifold learning algorithms we discuss make this assumption. In practice it is not very limiting since all smooth compact manifolds can be described by a single coordinate chart [15].

[6] For example if the entries of D are the pairwise distances between n cities then cMDS attempts to reconstruct, up to an arbitrary translation and rotation, the original coordinate locations of the n cities. In the process the algorithm also automatically provides an estimate of the dimensionality, m, of the Euclidean space in which the cities lie. For further details on cMDS and other linear dimensionality reduction algorithms see [5].

Isomap
input: $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_k \in \mathbb{R}^{\mathcal{D}}$, K

1. Compute the K-nearest-neighbours to each data point $\mathbf{x}_i$ based on Euclidean distances, from this construct the K-nearest-neighbour weighted graph with edge weights $W_{ij} = ||\mathbf{x}_i - \mathbf{x}_j||$ for any two connected points $\mathbf{x}_i$ and $\mathbf{x}_j$.

2. Compute the shortest path difference between all pairs of points on the graph using Floyd or Dijkstra's algorithm. Store these path differences as entries in the dissimilarity matrix $D \in \mathbb{R}^{k \times k}$.

3. The embedding representation $Y = [\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_k]^\top$ is obtained by applying cMDS to the dissimilarity matrix, i.e. $Y = \text{cMDS}(D)$.

There are two defining features of Isomap which set it apart from most other manifold learning algorithms. Firstly, as its name suggests, it assumes that there exists an isometric (i.e. distance preserving) chart from the parameter space to the high-dimensional space, and attempts to find this embedding. Secondly, a convenient feature is that Isomap automatically provides an estimate of the dimensionality of the underlying manifold, whereas in most manifold learning algorithms the required embedding dimension is an additional hyperparameter.

In the original Tenebaum *et al.* paper the authors argue that a non-linear manifold learning algorithm such as Isomap is required because linear methods are fundamentally inadequate at capturing geometric structure in a wide class of datasets, citing examples from computer vision, acoustics, astronomy and Earth sciences. The virtues of the Isomap algorithm are then quantitatively justified by showing that, for a selection of real and synthetic data sets, residual variance in the dimensionally-reduced data is consistently less with Isomap than for linear methods. Following on from this the authors also show a practical example of how Isomap, when applied to high-dimensional image datasets of face-poses, recovers the correct, three-dimensional embedding, with the individual dimensions being highly interpretable as measures of left-right/up-down pose and lighting direction.

Although the above results successfully demonstrate several benefits of Isomap, the analysis is somewhat incomplete; most notably there no discussion of computational efficiency, noise robustness, algorithm stability, and there is also no side-by-side visual comparision of embeddings produced by linear methods and Isomaps. Furthermore, the datasets used in the paper are limited to examples from computer vision and the swiss-roll[7], and so are by no means all-encompassing.

Many of these unanswered questions have since been addressed elsewhere. For instance, on the theoretical front, a deeper understanding of Isomap's apparent success came when Silva *et al.* proved that, for noiseless data, Isomap is guaranteed to asymptotically[8] recover the true dimensionality and geometry of any convex manifold [20]. However, with regards to algorithm stability, Balasubramanian *et al.* show that in fact Isomap can suffer from a topological instability: if K is too large, the K-nearest-neighbour graph has 'short-circuit errors' which leads to a misrepresentation of the underlying manifold structure [21]. For similar reasons, Isomap is also highly sensitive to noise, with its performance depending strongly on careful data pre-processing [21]. Although a visual comparison of Isomap with PCA is lacking from the original paper, more recent review papers have reliably demonstrated superior performance of manifold learning algorithms, including Isomap, over linear dimensionality reduction techniques [14, 22].

Several modified versions of Isomap have been developed which address the original algorithm's

[7] A standard swiss-roll is defined by parameteric equations $x = u\cos(u)$, $z = u\sin(u)$, $y = v$ with $1.5\pi \leq u \leq 4.5\pi$, $0 \leq v \leq 20$.

[8] i.e. as the size of the input data set tends to infinity, assuming we draw samples from over the entire manifold.

limitations. The standard Isomap algorithm, with a computational complexity $\mathcal{O}(N^2 \log N)$, does not scale well to large datasets. To address this issue of computational complexity Silva *et al.* developed a modified version of Isomap, known as Landmark Isomap, which uses $n \ll N$ 'landmark' datapoints to greatly reduce the dimensionality of the dissimilarity matrix and hence speed-up the final MDS stage, reducing the time complexity to $\mathcal{O}(N \log N)$, albeit at the expense of reducing the accuracy of the obtained manifold [20]. In that paper the authors also address another limitation of the standard Isomap - the restrictive assumption of assuming the existence of an isometric map. They developed a modified version of Isomap, known as C-Isomap, which requires the map only to be angle-preserving, rather than both angle and distance preserving.[9] And finally, to address the issue of noise-sensitivity, a closely related algorithm, known as Locally Linear Embeddings (LLE), was developed [23]. LLE is one of the many so-called eigenmap methods for manifold learning, which we shall now discuss.

3.2 Eigenmap Methods

Shortly after the original Isomap paper, a flurry of research led to the development of several faster, more noise-robust algorithms - the so-called *eigenmap methods*. The most well-known and widespread eigenmap methods include Locally Linear Embeddings (LLE) [23], Local Tangent Space Alignment (LTSA) [24], Hessian Eigenmaps (HE) [25], Laplacian Eigenmaps (LE) [26], and Diffusion Maps (DM) [27]. Although each different in their details, these algorithms obtain an embedding representation by following the same three generic steps, which we call the 'eigenmap strategy':

Eigenmap Strategy
input: $\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_k \in \mathbb{R}^{\mathcal{D}}$, K or r

1. Assign neighbours to each data point $\mathbf{x}_i$ based on Euclidean distances. Denote this collection of neighbours as $N(i)$. Neighbours can be defined in one of two ways either (i) all data points within a radius r ball around $\mathbf{x}_i$, or (ii) the K-nearest-neighbours to $\mathbf{x}_i$. This arbitrary choice, as well as the values of K and r, are free parameters common to every eigenmap algorithm.

2. For each local neighbourhood $N(i)$ construct a corresponding weight matrix W_i. In general this weight matrix will be a function of coordinates of each of the neighbours of $\mathbf{x}_i$, as well as possibly one or more free parameters, details of the construction of W_i being different for each algorithm.

3. The embedding representation $Y = [\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_k]^\top$ is obtained as the matrix which minimises some convex function $\Phi(Y; W_1, ..., W_k)$, subject to specific normalization constraints on Y. Details of the form of Φ, as well as the normalizing constraints, are different for each algorithm. Often this optimization problem can be reformulated as an eignenvalue problem, hence the paronymous term 'eigenmap'.

In the appendix to this report we explain in detail how the eigenmap strategy is followed for the Laplacian Eigenmaps algorithm. The interested reader can find analogous descriptions of each of the other eigenmap algorithms in [28].

Comparing Eigenmap Methods

Eigenmap methods, by virtue of being faster and more robust than the standard Isomap algorithm, are scalable to large, real world datasets, and have found numerous applications in

[9] Maps which only preserve angles are known as *conformal*. Hence the term C-Isomap.

bio-informatics, computational chemistry, and medicine amongst others [13, 29, 30].

The algorithms all differ in their theoretical approach, and so make different accuracy and computational trade-offs [22]. For instance, LLE performs well on sparse, non-uniformly sampled datasets but is not very robust to noise. LSTA improves on LLE by being less sensitive to hyperparameter choice, but at the cost of making more restrictive assumptions on the manifold structure. HE handles non-covex geometry well, albeit at the expense of computational speed. LE is noise-robust and works well on sparse data, but depends strongly on the requirement that the data be uniformly sampled. DM improves on LE by handling non-uniformly sampled data, although at the expense of increased computational cost and high dependence on careful hyperparameter tuning. In addition to these differences, the algorithms also make different assumptions about the type of manifold embedding; see Table 1 in section 3.3 for detailed comparisons.

A common strength of the eigenmap papers is that they all contain a detailed theoretical justification for the chosen approach, as well as each showing several examples of the approach applied to synthetic datasets. However, the degree to which the approach is then benchmarked, theoretically analysed, and contrasted with existing algorithms, varies.

The Laplacian Eigenmap (LE) paper particularly excels in that, in addition to tests on synthetic datasets, it is one of the few papers (alongside LLE, LTSA) to test the algorithm on real datasets from computer vision, linguistics, and speech - showing that, on these datasets, the resulting dimensionality reduction leads to interpretable results which uncover underlying patterns in the data not detected by PCA. It is also one of the few papers (alongside DM) to explore the performance sensitivity of the algorithm to its free parameters. However, unlike LE, LTSA is the only paper which runs experiments testing noise robustness, whereas HE is the only paper to visualise how its embedding compares with other manifold learning algorithms - in particular, showing how it is superior to LE and Isomap for a certain type of non-convex[10] manifold.

Despite their various merits, a common deficiency of the papers is the omission of a proof of a convergence guarantee, and so the central question - whether the algorithm is able to uncover the true underlying low-dimensional structure - was left unanswered until the theoretical work of Goldberg *et al.* [28]. There the authors prove that, for an arbitrary manifold, there is in fact *no guarantee* that the output of any eigenmap method (up to an arbitrary affine transformation) should resemble the original sample. The class of manifolds on which this guarantee fails is wide, including non-isometrically embedded manifolds and real world data. They even proceed to show various specific examples of such failures, for instance showing how, somewhat bizarrely, LE completely fails to recover the structure of a rectangular manifold if the aspect ratio is greater than two (to correct for this undesirable behaviour, Gerber *et al.* have since developed a more robust version of LE, although this is at the price of increased computational expense, thereby making it unsuitable for large datasets [31]). The cautionary message here is that eigenmap methods, although fast, can in some circumstances yield inaccurate embeddings.

Yet despite this warning, eigenmap methods are generally found to be superior to Isomap in practical settings. For example, in [13] the authors contrasted the performance of several common dimensionality reduction algorithms (including PCA, MDS, Isomap, LE, and LLE) by evaluating the performance of a cancer-detection binary classifier algorithm on eleven different biomedical datasets of cancerous and non-cancerous genomic sequences. They found that, on average, eigenmap methods marginally outperform Isomap (and Isomap outperformed all linear methods). This is in agreement with a similar finding by Wittman [22], where the author performs a wide-ranging practical comparison of eight popular manifold learning algorithms,

[10] A simple way to make a convex manifold non-convex is to 'punch a hole in it'. Indeed, the HE authors use exactly this approach - taking a swiss-roll manifold and then removing datapoints along a rectangular strip to make it non-convex.

applied to over a dozen different synthetic manifold geometries. The key result is that although no algorithm universally outperforms the others, there always exists, for the most commonly encountered manifold geometries, at least one eigenmap method which outperforms Isomap in both speed and embedding quality. Specifically Wittman concludes that if the data are non-convex and densely sampled then HE performs best, if the data are noisy or uniformly sampled then LE is the preferred choice, otherwise LLE typically yields the most accurate embedding.[11]

3.3 Topological Embedding Methods

Topological Data Analysis (TDA) is a recently emergent field in data analysis which applies tools and concepts from the mathematical branch of algebraic topology to perform a rigorous study of shape and structure in datasets. In the short timespan since its inception, it has led to a multitude of applications in fields of medicine, physical-chemistry, cosmology, and many others [32, 33, 34, 35]. The message is simple - geometric structure in data can be insightful.

Recently, McInnes *et al.* showed how tools from TDA can be applied to directly tackle the manifold learning problem - with an algorithm known as Uniform Manifold Approximation and Projection (UMAP) [36]. We now proceed to discuss in broad terms the operational principle behind this algorithm, what makes it unique, and critically analyse its performance within the context of other manifold learning algorithms. Note that our level of exposition will assume no prior familiarity with TDA and will therefore necessarily omit several technical points. For a more in-depth introduction to TDA for the data scientist see [37], and for a more general introduction to algebraic topology see [38].

UMAP Algorithm

At the high level, UMAP works by constructing a topological representation[12] of high-dimensional data and finding the lower-dimensional embedding (of specified dimension d) whose topological representation is most similar to that of the original dataset. Crucial to the operation of UMAP is the assumption that the high-dimensional data lie on a lower dimensional manifold $\mathcal{M}$. UMAP can therefore be seen as a manifold learning algorithm that seeks to find a coordinate chart $f : \mathcal{M} \to \mathbb{R}^d$ which most faithfully preserves topological structure.

UMAP, despite arising from very different theoretical foundations to Isomap and Eigenmap methods, shares several key similarities in its computational implementation. Specifically, it is also a weighted-graph based algorithm whose first two steps are identical to those of the Eigenmap strategy - only in the case of UMAP the set of weight matrices have been carefully chosen based on a theoretical motivation to encode topological information. The most significant area in which UMAP differs is in the computation of the embedding representation. Here the embedding dimension must be specified in advance, and the embedding representation is then found by performing stochastic gradient descent (SGD) on a force-directed graph drawing of the K-nearest-neighbour-graph of the embedded coordinates, where the cost function being minimised by SGD is the cross-entropy between the topological representations of the embedding and the original dataset. The optimisation problem is non-convex, and so UMAP is a stochastic algorithm - yielding a slightly different embedding for the same input data each time it is run. We refer the reader to [36] for additional mathematical details and computational details of the algorithm.

[11] But note that LTSA was not studied. There is evidence that LTSA may produce a superior embedding [24].

[12] The type of topological representation chosen is known as a *fuzzy simplicial set*.

Algorithm	Embedding Type	Embedding Dimension	Free Parameters	Computational Complexity	Available Variants
Isomap [16]	Globally Isometric	Learnt	1	$\mathcal{O}(N^2 \log N)$	U/SS/S [39, 40]
LE [26]	Unknown	Specified*	3	$\mathcal{O}(N^2)$	U/SS/S [41, 42]
DM [27]	Unknown	Specified*	4	$\mathcal{O}(N^3)$	U/SS [43]
LLE [23]	Conformal	Specified	2	$\mathcal{O}(N^2)$	U/SS/S [44]
LTSA [?]	Locally Isometric	Specified*	2	$\mathcal{O}(N^2)$	U/SS/S [45]
HE [25]	Locally Isometric	Specified*	2	$\mathcal{O}(N^2)$	U/SS/S [46]
UMAP [36]	Fuzzy Topological	Specified	4	$\mathcal{O}(N^{1.14})$	U/SS/S [47]

(*) although the embedding dimension is a free parameter and must be specified, the algorithm provides a heuristic method for estimating it.

Table 1: Comparison of the most popular manifold learning algorithms based on several of their key attributes. *Embedding Type* refers to the type of chart map that is learnt by the algorithm (Isometric - conserves distances, Conformal - conserves angles but not lengths, Fuzzy Topological - conserves topological properties). For simplicity, *Computational Complexity* is shown only with respect to the input data size N, not any other algorithm free parameters. U, SS and S are short for Unsupervised/Semi-Supervised/Supervised respectively.

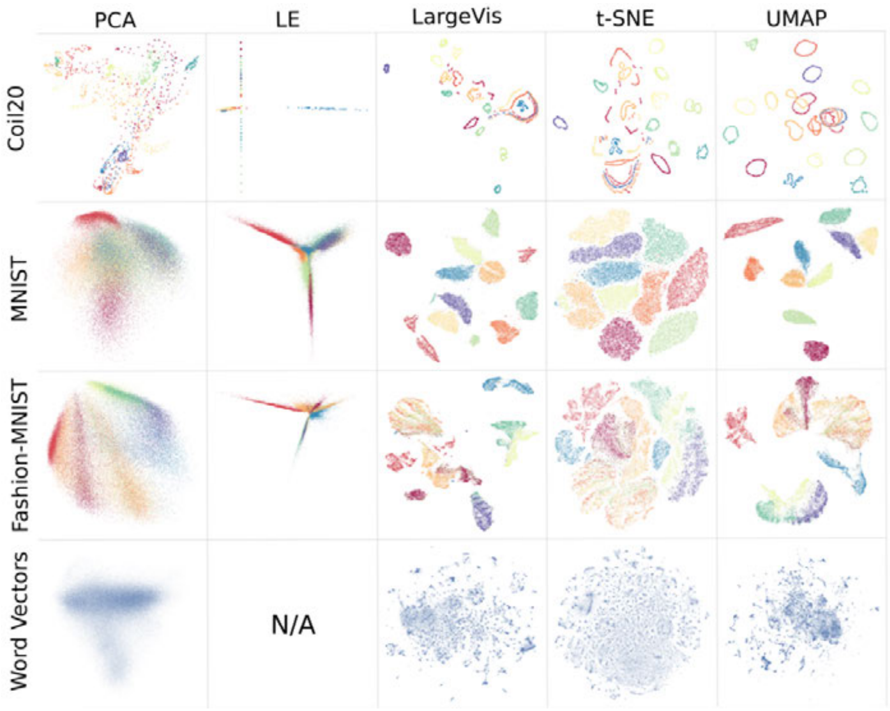

Figure 1: A comparison of several dimension reduction algorithms on Coil20, MNIST, Fashion-MNIST, and Word Vectors datasets. Note in particular how UMAP does the best of preserving the topological structure in Coil20, and generally preserves both the global structure (that is well represented by e.g. Laplacian Eigenmaps) and also the fine local structure (well preserved by LargeVis and t-SNE). Figure adapted from [36].

A unique strength of the UMAP paper is that the authors, in addition to visualizing the effects of changing the algorithm's hyperparameters, also conduct an expansive range of benchmark tests on real world datasets (from fields of computer vision, bioinformatics, space engineering, cellular biology, linguistics, and others). These tests are then used to facilitate a detailed comparison of the practical efficacy of UMAP with several other dimensionality reduction algorithms.[13] In particular, UMAP is shown to be the fastest of the algorithms on all but the smallest of datasets, whilst being particularly computationally efficient for higher dimensional embeddings.[14] Furthermore, the authors use embedding visualisation to show that UMAP is unique among the dimensionality reduction algorithms by faithfully preserving both large-scale structure and local fine structure. The superiority of UMAP's embedding is then shown quantitatively by training a kNN classifier on the embedding spaces produced by the various algorithms, and finding that the resulting classifier accuracy is consistently highest for UMAP. This superior embedding performance of UMAP has subsequently been independently verified by Moon *et al.* who showed that UMAP outperformed Diffusion Maps, Isomap, and LLE on a quantitative metric for visualization performance on biological datasets [49].

Although the original UMAP paper was limited to the discussion of an unsupervised learning algorithm, this has since been extended to semi-supervised and supervised domains with parametric-UMAP [47]. Both UMAP and parametric-UMAP have since seen numerous applications in machine learning, materials science, and bioinformatics among others [50, 51, 52, 53, 54]. Notably, it has found widespread use in population genetics to study population structure, enabling state-of-the-art visualizations of ancestral components and geographic patterns in human genetic datasets [6].

Despite its popularity and effectiveness, most algorithms must make trade-offs, and UMAP is no exception. Notable issues with UMAP are that random noise does not always look random, and that hyperparameter fine-tuning is essential for accurate results. Another major limitation is in its interpretability. UMAP's objective is to preserve topological structure but, as a fact any undergraduate student of topology would be eager to share, a doughnut is topologically equivalent to a teacup, and so blindly preserving topology can sometimes lead to unintuitive results. For UMAP this means that the cluster sizes, as well as the distances between clusters, may be meaningless. In fact, if preserving global structure is of primary interest, then it has been shown that PHATE [55], a manifold learning-inspired algorithm which combines concepts from diffusion maps and MDS for fast dimensionality reduction whilst preserving global structure, generally has superior performance to UMAP [49].

4 Summary

Many areas of modern science rely on exploratory data analysis and visualization of vast, high-dimensional datasets. These datasets can be simplified by recognising that physical and geometric constraints often result in data lying on low-dimensional manifolds embedded in the feature space. In this review we have critiqued and compared the most popular and historically significant algorithms which attempt to learn this underlying manifold structure.

Identifying the underlying manifold is a mathematically ill-posed problem and so many different algorithms, each with different theoretical assumptions and procedures, have been developed. Isomap, introduced in 2000, was the first manifold learning algorithm to improve significantly on linear dimensionality reduction methods, and still continues to find applications in fields of

[13] Specifically PCA, LE, t-SNE, and LargeVis [48]

[14] For instance, the authors showed that UMAP could operate directly on a 1.8 million dimensional dataset. This is in contrast to other manifold learning algorithms which would require dimensionality reduction preprocessing (e.g. with PCA) for such a high-dimensional dataset.

medicine and traffic control to this day [14, 18]. The algorithm has several appealing properties, namely a convergence guarantee and an automatic determination of the manifold dimension. However, the computational inefficiency of Isomap, combined with its sensitivity to noise, has made it unsuitable for the many of the modern challenges in big data.

To address these fundamental limitations, several faster, more noise-robust algorithms were subsequently developed: Locally Linear Embeddings (2000), Local Tangent Space Alignment (2002), Hessian Eignmaps (2003), Laplacian Eigenmaps (2003), and Diffusion Maps (2006). Through extensive review studies it has been shown that none of these algorithms is universally superior to the others. Rather, the choice of which algorithm to use depends heavily on the form of the dataset - whether it is densely sampled or non-convex (use Hessian Eigenmaps), is noisy or uniformly sampled (use Laplacian Eigenmaps), or noise-free (use LLE or LTSA). What is remarkable is that each of these algorithms, although developed from distinct theoretical motivations, can be analysed under a common framework known as the eigenmap method. A disadvantage of all eigenmap methods is that, unlike Isomap, they have no convergence guarantee. Additionally, many of these algorithms have several free parameters which must be carefully fine-tuned for optimal results. Finally, although typically faster than Isomap, all eigenmap algorithms are at least $\mathcal{O}(N^2)$ time complexity, and remain inappropriate for handling the largest of datasets, such as those which appear in geonomics; in these cases other non-linear dimensionality reduction algorithms, namely t-SNE, LargeVis, and UMAP, are preferred.

Most recently, progress in designing computationally efficient and scalable manifold learning algorithms has come from applying tools from Topological Data Analysis (TDA). In particular, UMAP, first introduced in 2018, is a manifold learning algorithm which achieves its embedding by attempting to preserve the topological structure of the dataset under dimensionality reduction. It is faster than all other manifold learning algorithms on all but the smallest of datasets ($N \lesssim 500$), and is unique amongst the manifold learning algorithms in that it faithfully preserves both global scale and local fine scale structure [36]. UMAP also ranks higher than any other manifold learning algorithm on several quantitative tests for embedding quality [36, 49], making UMAP a good candidate as a first algorithm of choice for manifold learning.

5 Conclusion

The past two decades has seen enormous progress in the development of manifold learning algorithms, and they have since found a multitude of applications in machine learning, genomics, cell-cytometry, physics and elsewhere [31, 56, 53]. The flurry of recent advances means that today the data scientist can choose from nearly a dozen manifold learning algorithms, all with different behaviours when it comes to embedding type, theoretical guarantees, robustness to noise and computational efficiency. Yet this diversity of approaches means that the decision of which manifold learning algorithm to choose is not always transparent.

In this review we have showed that manifold learning algorithms based on topological embeddings (namely UMAP and parametric-UMAP) generally stand out as superior due to their scalability and high embedding quality. But what is deemed desirable in an embedding is highly application specific, and all manifold learning algorithms are highly sensitive to the choice of free parameters. The importance of thorough experimentation and evaluation should therefore not be undervalued, since techniques and parameters which are most appropriate on one dataset might not be transferable to another. This review has contrasted and critiqued the major differences amongst the popular manifold learning algorithms, thereby providing the data scientist with the necessary prerequisite knowledge to aid in this evaluation process.

References

[1] Samuraí Brito, L. R. da Silva, and Constantino Tsallis. Role of dimensionality in complex networks. *Scientific Reports*, 6(1):27992, June 2016. Number: 1 Publisher: Nature Publishing Group.

[2] Keyang Cheng, Muhammad Saddam Khokhar, Misbah Ayoub, and Zakria Jamali. Nonlinear dimensionality reduction in robot vision for industrial monitoring process via deep three dimensional Spearman correlation analysis (D3D-SCA). *Multimedia Tools and Applications*, October 2020.

[3] Milos Hauskrecht, Richard Pelikan, Michal Valko, and James Lyons-Weiler. Feature Selection and Dimensionality Reduction in Genomics and Proteomics. In Werner Dubitzky, Martin Granzow, and Daniel Berrar, editors, *Fundamentals of Data Mining in Genomics and Proteomics*, pages 149–172. Springer US, Boston, MA, 2007.

[4] Iain M. Johnstone and D. Michael Titterington. Statistical challenges of high-dimensional data. *Philosophical transactions. Series A, Mathematical, physical, and engineering sciences*, 367(1906):4237–4253, November 2009.

[5] John P. Cunningham and Zoubin Ghahramani. Linear dimensionality reduction: survey, insights, and generalizations. *The Journal of Machine Learning Research*, 16(1):2859–2900, January 2015.

[6] Alex Diaz-Papkovich, Luke Anderson-Trocmé, and Simon Gravel. A review of UMAP in population genetics. *Journal of Human Genetics*, 66(1):85–91, January 2021. Number: 1 Publisher: Nature Publishing Group.

[7] Jun-Yan Zhu, Philipp Krähenbühl, Eli Shechtman, and Alexei A. Efros. Generative Visual Manipulation on the Natural Image Manifold. In Bastian Leibe, Jiri Matas, Nicu Sebe, and Max Welling, editors, *Computer Vision – ECCV 2016*, volume 9909, pages 597–613. Springer International Publishing, Cham, 2016. Series Title: Lecture Notes in Computer Science.

[8] T. Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, September 1990. Conference Name: Proceedings of the IEEE.

[9] Christopher M. Bishop, Markus Svensén, and Christopher K. I. Williams. GTM: The Generative Topographic Mapping. *Neural Computation*, 10(1):215–234, January 1998.

[10] Geoffrey Hinton and Sam Roweis. Stochastic Neighbor Embedding. *Advances in Neural Information Processing Systems*, 15:8.

[11] C. O. S. Sorzano, J. Vargas, and A. Pascual Montano. A survey of dimensionality reduction techniques. *arXiv:1403.2877 [cs, q-bio, stat]*, March 2014. arXiv: 1403.2877.

[12] Charles Fefferman, Sanjoy Mitter, and Hariharan Narayanan. Testing the manifold hypothesis. *Journal of the American Mathematical Society*, 29(4):983–1049, 2016.

[13] George Lee, Carlos Rodriguez, and Anant Madabhushi. Investigating the Efficacy of Nonlinear Dimensionality Reduction Schemes in Classifying Gene- and Protein-Expression Studies. *IEEE/ACM transactions on computational biology and bioinformatics / IEEE, ACM*, 5(3):368–384, 2008.

[14] Hernán Stamati, Cecilia Clementi, and Lydia E. Kavraki. Application of Non-Linear Dimensionality Reduction to Characterize the Conformational Landscape of Small Peptides. *Proteins*, 78(2):223–235, February 2010.

[15] I Introduction to Manifolds. In William M. Boothby, editor, *Pure and Applied Mathematics*, volume 120, pages 1–19. Elsevier, January 1986.

[16] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319–2323, December 2000. Publisher: American Association for the Advancement of Science Section: Report.

[17] A. Hannachi and A. G. Turner. Isomap nonlinear dimensionality reduction and bimodality of Asian monsoon convection. *Geophysical Research Letters*, 40(8):1653–1658, 2013. _eprint: https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1002/grl.50351.

[18] Qingchao Liu, Yingfeng Cai, Haobin Jiang, Jian Lu, and Long Chen. Traffic state prediction using ISOMAP manifold learning. *Physica A: Statistical Mechanics and its Applications*, 506:532–541, September 2018.

[19] Peng Dai, Femida Gwadry-Sridhar, Michael Bauer, and Michael Borrie. A hybrid manifold learning algorithm for the diagnosis and prognostication of Alzheimer's disease. *AMIA Annual Symposium Proceedings*, 2015:475–483, November 2015.

[20] Vin Silva and Joshua Tenenbaum. Global Versus Local Methods in Nonlinear Dimensionality Reduction. *Advances in Neural Information Processing Systems*, 15:721–728, 2002.

[21] American Association for the Advancement of Science. The Isomap Algorithm and Topological Stability. *Science*, 295(5552):9–9, January 2002. Publisher: American Association for the Advancement of Science.

[22] Todd Wittman. Manifold Learning Techniques: So which is the best. *Geometric Data Analysis, University of Minnesota*, 2005.

[23] Sam T. Roweis and Lawrence K. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 290(5500):2323–2326, December 2000. Publisher: American Association for the Advancement of Science Section: Report.

[24] Zhenyue Zhang and Hongyuan Zha. Principal Manifolds and Nonlinear Dimensionality Reduction via Tangent Space Alignment. *SIAM Journal on Scientific Computing*, 26(1):313–338, January 2005.

[25] David L. Donoho and Carrie Grimes. Hessian eigenmaps: Locally linear embedding techniques for high-dimensional data. *Proceedings of the National Academy of Sciences*, 100(10):5591–5596, May 2003. ISBN: 9781031596106 Publisher: National Academy of Sciences Section: Physical Sciences.

[26] M. Belkin and P. Niyogi. Laplacian Eigenmaps for Dimensionality Reduction and Data Representation. *Neural Computation*, 2003.

[27] Ronald R. Coifman and Stéphane Lafon. Diffusion maps. *Applied and Computational Harmonic Analysis*, 21(1):5–30, July 2006.

[28] Yair Goldberg, Alon Zakai, Dan Kushnir, and Ya'acov Ritov. Manifold Learning: The Price of Normalization. *The Journal of Machine Learning Research*, 9:1909–1939, June 2008.

[29] Z. Trstanova, B. Leimkuhler, and T. Lelièvre. Local and global perspectives on diffusion maps in the analysis of molecular systems. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 476(2233):20190036, January 2020. Publisher: Royal Society.

[30] Mingai Li, Xinyong Luo, Jinfu Yang, and Yanjun Sun. Applying a Locally Linear Embedding Algorithm for Feature Extraction and Visualization of MI-EEG. *Journal of Sensors*, 2016:1–9, 2016.

[31] Samuel Gerber, Tolga Tasdizen, and Ross Whitaker. Robust non-linear dimensionality reduction using successive 1-dimensional Laplacian Eigenmaps. *Proceedings of the 24th international conference on Machine learning - ICML &#39;07.*

[32] Li Li, Wei-Yi Cheng, Benjamin S. Glicksberg, Omri Gottesman, Ronald Tamler, Rong Chen, Erwin P. Bottinger, and Joel T. Dudley. Identification of type 2 diabetes subgroups through topological analysis of patient similarity. *Science Translational Medicine*, 7(311):311ra174–311ra174, October 2015. Publisher: American Association for the Advancement of Science Section: Research Article.

[33] P. Y. Lum, G. Singh, A. Lehman, T. Ishkanov, M. Vejdemo-Johansson, M. Alagappan, J. Carlsson, and G. Carlsson. Extracting insights from the shape of complex data using topology. *Scientific Reports*, 3(1):1236, February 2013. Number: 1 Publisher: Nature Publishing Group.

[34] Brenda Y. Torres, Jose Henrique M. Oliveira, Ann Thomas Tate, Poonam Rath, Katherine Cumnock, and David S. Schneider. Tracking Resilience to Infections by Mapping Disease Space. *PLOS Biology*, 14(4):e1002436, April 2016. Publisher: Public Library of Science.

[35] K. Madhobi, M. Kamruzzaman, A. Kalyanaraman, E. Lofgren, R. Moehring, , and B. Krishnamoorthy. A visual analytics framework for analysis of patient trajectories. *10th ACM Conference on Bioinformatics, Computational Biology, and Health Informatics (BCB '19)*, January 2019.

[36] Leland McInnes, John Healy, and James Melville. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. *arXiv:1802.03426 [cs, stat]*, September 2020. arXiv: 1802.03426.

[37] Frédéric Chazal and Bertrand Michel. An introduction to Topological Data Analysis: fundamental and practical aspects for data scientists. *arXiv:1710.04019 [cs, math, stat]*, October 2017. arXiv: 1710.04019.

[38] J. Peter May. *A concise course in algebraic topology.* Chicago lectures in mathematics. University of Chicago Press, Chicago, 1999.

[39] Xin Geng, De-Chuan Zhan, and Zhi-Hua Zhou. Supervised nonlinear dimensionality reduction for visualization and classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 35(6):1098–1107, December 2005. Conference Name: IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics).

[40] Yan Zhang, Zhao Zhang, Jie Qin, Li Zhang, Bing Li, and Fanzhang Li. Semi-supervised local multi-manifold Isomap by linear embedding for feature extraction. *Pattern Recognition*, 76:662–678, April 2018.

[41] Feng Zheng, Na Chen, and Luoqing Li. Semi-supervised Laplacian eigenmaps for dimensionality reduction. In *2008 International Conference on Wavelet Analysis and Pattern Recognition*, pages 843–849, Hong Kong, China, August 2008. IEEE.

[42] B. Raducanu and F. Dornaika. A supervised non-linear dimensionality reduction approach for manifold learning. *Pattern Recognition*, 45(6):2432–2444, June 2012.

[43] Feng Zheng, Ling Shao, and Zhan Song. Eigen-space learning using semi-supervised diffusion maps for human action recognition. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, CIVR '10, pages 151–157, New York, NY, USA, July 2010. Association for Computing Machinery.

[44] Benyamin Ghojogh, Ali Ghodsi, Fakhri Karray, and Mark Crowley. Locally Linear Embedding and its Variants: Tutorial and Survey. *arXiv:2011.10925 [cs, stat]*, November 2020. arXiv: 2011.10925.

[45] Hongyu Li, Li Teng, Wenbin Chen, and I-Fan Shen. Supervised Learning on Local Tangent Space. In Jun Wang, Xiaofeng Liao, and Zhang Yi, editors, *Advances in Neural Networks – ISNN 2005*, Lecture Notes in Computer Science, pages 546–551, Berlin, Heidelberg, 2005. Springer.

[46] Lianbo Zhang, Dapeng Tao, and Weifeng Liu. Supervised Hessian Eigenmap for dimensionality reduction. *2015 IEEE 16th International Conference on Communication Technology (ICCT)*, 2015.

[47] Tim Sainburg, Leland McInnes, and Timothy Q. Gentner. Parametric UMAP: learning embeddings with deep neural networks for representation and semi-supervised learning. *arXiv:2009.12981 [cs, q-bio, stat]*, September 2020. arXiv: 2009.12981.

[48] Jian Tang, Jingzhou Liu, Ming Zhang, and Qiaozhu Mei. Visualizing Large-scale and High-dimensional Data. *arXiv:1602.00370 [cs]*, April 2016. arXiv: 1602.00370.

[49] Kevin R. Moon, David van Dijk, Zheng Wang, Scott Gigante, Daniel B. Burkhardt, William S. Chen, Kristina Yim, Antonia van den Elzen, Matthew J. Hirn, Ronald R. Coifman, Natalia B. Ivanova, Guy Wolf, and Smita Krishnaswamy. Visualizing Structure and Transitions in High-Dimensional Biological Data. *Nature biotechnology*, 37(12):1482–1492, December 2019.

[50] Tomas Mikolov, Ilya Sutskever, Kai Chen, G. S. Corrado, and J. Dean. Distributed Representations of Words and Phrases and their Compositionality. *NIPS*, 2013.

[51] Carlos Escolano, Marta R. Costa-jussà, and José A. R. Fonollosa. (Self-Attentive) Autoencoder-based Universal Language Representation for Machine Translation. *arXiv:1810.06351 [cs]*, October 2018. arXiv: 1810.06351.

[52] Xin Li, Ondrej E. Dyck, Mark P. Oxley, Andrew R. Lupini, Leland McInnes, John Healy, Stephen Jesse, and Sergei V. Kalinin. Manifold learning of four-dimensional scanning transmission electron microscopy. *npj Computational Materials*, 5(1):1–8, January 2019. Number: 1 Publisher: Nature Publishing Group.

[53] Junyue Cao, Malte Spielmann, Xiaojie Qiu, Xingfan Huang, Daniel M. Ibrahim, Andrew J. Hill, Fan Zhang, Stefan Mundlos, Lena Christiansen, Frank J. Steemers, Cole Trapnell, and Jay Shendure. The single-cell transcriptional landscape of mammalian organogenesis. *Nature*, 566(7745):496–502, February 2019. Number: 7745 Publisher: Nature Publishing Group.

[54] Dmitry Kobak and George C. Linderman. UMAP does not preserve global structure any better than t-SNE when using the same initialization. *bioRxiv*, page 2019.12.19.877522, December 2019. Publisher: Cold Spring Harbor Laboratory Section: Contradictory Results.

[55] Kevin R. Moon, David van Dijk, Zheng Wang, Scott Gigante, Daniel B. Burkhardt, William S. Chen, Kristina Yim, Antonia van den Elzen, Matthew J. Hirn, Ronald R. Coifman, Natalia B. Ivanova, Guy Wolf, and Smita Krishnaswamy. Visualizing Structure and Transitions for Biological Data Exploration. *Nature*, page 120378, April 2019. Publisher: Cold Spring Harbor Laboratory Section: New Results.

[56] Kevin R. Moon, David van Dijk, Zheng Wang, William Chen, Matthew J. Hirn, Ronald R. Coifman, Natalia B. Ivanova, Guy Wolf, and Smita Krishnaswamy. PHATE: A Dimensionality Reduction Method for Visualizing Trajectory Structures in High-Dimensional Biological Data. *bioRxiv*, page 120378, March 2017. Publisher: Cold Spring Harbor Laboratory Section: New Results.

A Laplacian Eigenmaps

To elucidate the eigenmap strategy we explore in detail how it is followed for one particular eigenmap algorithm, Laplacian Eigenmaps. Note that step one of the eigenmap strategy is common to all of the algorithms, so we shall begin at step two by discussing the choice of weight matrix.

Let W_{ij} denote the j^{th} entry of the weight matrix W_i. The intuition behind Laplacian Eigenmaps is that we choose a form of a weight matrix W_{ij} that measures the degree of local similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$. More precisely we choose

$$W_{ij} = \begin{cases} e^{-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{2\sigma^2}} & \text{if } \mathbf{x}_j \in N(i) \\ 0 & \text{otherwise} \end{cases}$$

where σ is a free parameter to be set manually. This choice of weight matrix is known as the Gaussian heat kernel, and it can be shown to result in several convenient properties for the algorithm - a complete theoretical justification is given in [26].

If $\mathbf{x}_i$ and $\mathbf{x}_j$ lie close together on the manifold we see that they have a high degree of similarity as measured by W_{ij}. In a faithful embedding, we intuitively expect that the corresponding points $\mathbf{y}_i$ and $\mathbf{y}_j$ in the embedding representation should try maintain this similarity as much as possible. In the Laplacian Eigenmaps algorithm this intuition is the motivation for defining the following convex cost function

$$\Phi = \sum_{ij} W_{ij} ||\mathbf{y}_i - \mathbf{y}_j||^2$$

which we minimise with respect to $Y = [\mathbf{y}_1, \mathbf{y}_2, ..., \mathbf{y}_k]^\top$ to extract the embedding. Note, however, that naively minimising Φ results in $\mathbf{y}_1 = \mathbf{y}_2 = ... = \mathbf{y}_k = \mathbf{0}$; a useless embedding. To avoid this trivial solution we enforce the normalization constraints $\mathbf{y}_i^\top D \mathbf{y}_i = 1$, $\forall i \in \{1, 2, ..., k\}$ where D is the diagonal matrix with elements $D_{ii} = \sum_j W_{ij}$. In this case the optimization problem, including constraints, can be reformulated as an eigenvalue problem[15] which can be solved with sparse vector routines in $\mathcal{O}(N^2)$ time complexity.

[15] Specifically, $L\mathbf{y} = \lambda D\mathbf{y}$ where $L := D - W$ is known as the Laplacian matrix. See [26] for mathematical details.