Semester Project
year 2014-2015

A Review of Some Manifold Learning Algorithms

Author:
Yoann Trellu
Supervisor:
Prof. Tom Ilmanen

July 20, 2016

ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Contents

1 Introduction **2**

2 Review of Statistical learning **3**
2.1 Supervised Learning 3
2.2 Unsupervised Learning 3
2.2.1 Dimensionality Reduction 4
2.2.2 Geometric Understanding of Real Life Data 5

3 Kernel Methodology **6**
3.1 Kernel PCA 7

4 Laplace Operator **9**
4.1 Motivation 9
4.2 Eigenfunctions and Approximations 12

5 Discrete Laplace Operator **13**
5.1 Motivation for Graphs 13
5.2 Graph Laplacian 13
5.3 Heuristics for the choice of t 14

6 Review of Methods with Manifold Assumption **16**
6.1 Isomap 16
6.2 LLE 17
6.3 Laplacian Eigenmaps 18

7 Conclusion **19**

A Invariance of gradient and divergence under isometries **20**

1 Introduction

This semester project is motivated by papers and conferences given by N. P. Niyogi and M. Belkin on the topic of "Manifold Learning". To briefly summarize their topic, based on the assumption that the data lives on a manifold, they try to include the geometrical information into a statistical learning algorithm for the purpose of dimensionality reduction. Our aim will be three fold:

1. Motivating the addressed problem and the assumptions.
2. Understanding their heavy use of the Laplace-Beltrami operator.
3. Explaining the use of graphs to approximate manifolds.

In order to have a broader view on the topic, we will also take a look at two other methods making the same assumption.

We first review informally the tasks and formulations of statistical learning, and also take a look at the geometry of data sets. We present the *kernel trick* that will help us unify and frame the dimensionality reduction problem. We then take a more formal approach to motivate the use of graphs and of the Laplace-Beltrami operator. Finally we formulate and briefly explain three manifold learning algorithms including Laplacian Eigenmaps.

2 Review of Statistical learning

Statistical learning (or machine learning) aims at understanding and learning from data with an algorithmic point of view. The name is a loose association of statistics, being the general study of data, and learning being the process by which *machines* learn from their environment or input, to perfect the work they were designed for.

Typical tasks are analyzing dependencies, interaction or similarities among data points. Most methods can be classified as either being supervised or unsupervised; they solve different problems depending on the distinction.

2.1 Supervised Learning

Supervised learning receives as input a sequence of ordered pairs, the training data and its label. Knowing the label, we are able to *supervise* the building of the model. Because it is trying to relate a training point to its label, it leads to inference (understanding the relationship between the pair) but also prediction (from *test* data, producing a best guess for the label).

Let $\mathcal{X}$ denote the space where the training data lives and by $\mathcal{L}$ the space of all possible labels. We also denote by $\mathbb{P}$ the probability distribution from which data points $(x, l) \in \mathcal{X} \times \mathcal{L}$ are drawned.

Typically one wants to know the conditional probability of having label l given data x, which can be rewritten as $\mathbb{P}(l|x)$. In fact, it can be shown that attributing labels with respect to the highest achieved probability $\mathbb{P}(\cdot|x)$ is optimal in the sense that it minimizes the error rate $\text{Ave}(I(l \neq \hat{l}))$ – the average of mismatches between estimated labels and true labels – on a *test* data set. Classifying (or attributing labels to) test data in such a way is called the *Bayes Classifier*.

Example 2.1 (Supervised learning)**.** *One wants to explain the salary of a person with attributes such as sex, education in number of years, and nationality. We will then have* $\mathcal{L} = \mathbb{R}_+$ *and* $\mathcal{X} = \{0, 1\} \times \mathbb{N} \times \mathcal{C}$ *with* $\mathcal{C}$ *the set of all possible countries.*

Example 2.2 (Supervised learning)**.** *One could also give the task to a computer of detecting an animal and naming it using only pictures of it. We will then have* $\mathcal{L} = \mathcal{A} \cup \varnothing$ *the set of all possible animals together with the no animal event, and* $\mathcal{X} = [0, 1]^{n^2}$ *the space in which pictures live.*

2.2 Unsupervised Learning

Unsupervised learning on the other hand only receives a sequence of training data as input. we can only hope to achieve a better understanding of patterns within the data set. Using different kinds of measures for similarity between two points, we

can study *clustering*, and using structural assumptions we can study the *geometry* of the data set.

2.2.1 Dimensionality Reduction

Today most interesting data sets live in high dimensions. The main purpose of dimensionality reduction is to ease the computational cost due to *the curse of dimensionality*. The latter regroups every kind of problems arising when the dimension grows to high levels.

From a combinatorial point of view, the number of possible solutions grows exponentially with the dimension. Furthermore, so does the required number of points to sample arbitrary spaces. We also acknowledge complications with distances functions in high dimensional space. When it is possible to reduce dimensions up to order 2 or 3, it also allows us to visualize the data.

We frame the problem as follows: For k given points $\mathbf{x}_1, \cdots, \mathbf{x}_k$ in $\mathbb{R}^n$, we wish to find a "good" representation $\mathbf{y}_1, \cdots, \mathbf{y}_k$ in $\mathbb{R}^m$ with $m \ll n$. Let us denote by $f : \{\mathbf{x}_1, \cdots, \mathbf{x}_k\} \subseteq \mathbb{R}^n \to \mathbb{R}^m$ the function fulfilling the task. The choice of m is not fixed but problem dependent. In fact, tasks of reduction and modeling, should be viewed as a whole since the need of one highly depend on the other.

In order to put dimensionality reduction into perspective, let us point out a strategy for learning from high dimensional data sets :

Scaling The first step differentiates extrinsic properties of the data set with its intrinsic properties. That is, we typically will use rotations, translations and sometimes normalizations to set our problem in a generic way (such as mean-centered, scaled to one, or axed along some canonical basis). We notice that normalizations are not isometric operations, but may be considered more as a data cleaning process.

Structuring We create meaningful relationships between data points such that the structure best reflect the information within the data set. Standard methods use k-nearest neighbors or ϵ-balls to build a graph. This step has to take into account the geometrical assumption on the data set.

Reducing We aim at finding f, the mapping that will *optimally* preserve the structure created while reducing the dimension.

Modeling Using the tractable points given by f, we may use clustering methods or other supervised models to learn, infer or predict from the data.

2.2.2 Geometric Understanding of Real Life Data

Having the task of dimension reduction in mind, we are trying to find new coordinate systems, mappings in general, and other transformations that will keep most of the information, while reducing the number of necessary dimensions to express data points. Studying the *shape* of the data set, the *relative position* of data points or the properties of the underlying space will help us choosing a suitable f.

Example 2.3 (Principal Component Analysis)**.** *Let us look for the direction explaining the maximum variability among a data set. Let us suppose that repeating the process while choosing next directions in orthogonal spaces, we are able to explain 95 % of the variability within the data set, using only 2 orthogonal vectors. By considering a linear structure and two dimensions, we have projected our data set on a plane without losing two much information, relating variance with information in the sense of entropy.*

Linear reduction techniques are well understood and accessible, giving an efficient tool for a first exploratory analysis. To further understand how one can represent the geometry of data sets we need to move beyond linearity.

We can take a step forward by only requiring the assumed underlying space to be *locally* linear. We will see such an example with the LLE method. By generalizing further, requiring every neighborhood of points to be homeomorphic to an n-dimensional euclidean space, we are considering general (topological) manifolds. Describing our sets of points using charts mapping to euclidean space allows for a tractable way of describing data.

While modeling a data set, one is always confronted to the bias-variance trade off. That is, in order to avoid over-fitting, we impose regularity conditions on the model. Thus it can motivate the use of differentiable or smooth manifolds as the support for the data set. In addition let us also assume that the assumed manifold is compact, this will be useful for theoretical results used later and is a reasonable assumption when we consider bounded real data sets.

We have now framed the problem as finding $f : \{\mathbf{x}_1, \cdots, \mathbf{x}_k\} \subseteq \mathcal{M} \subseteq \mathbb{R}^n \to \mathbb{R}^m$ with $\mathcal{M}$ a smooth compact manifold.

Example 2.4 (Moving camera)**.** *Let us consider a camera mounted on rails making a circle around an unmoving object. Every image taken by the camera lives in* $[0,1]^{n\times n}$*, a grey scale image with* n^2 *pixels. We then define* $\mathcal{M}$ *to be the set of all images taken by (smoothly) rotating the camera.* $\mathcal{M}$ *is thus embedded in* $\mathbb{R}^{n^2}$*. By considering k pictures uniformly sampled from this experience, we are looking for* f *from* $\{\mathbf{x}_1, \cdots, \mathbf{x}_k\} \subseteq \mathcal{M} \subseteq \mathbb{R}^{n^2}$ *to* $\mathbb{S}^1 \subseteq \mathbb{R}^2$*.*

Notice how this example can benefit the previous example 2.2. By looking for this kinds of reduction techniques, images of an animal taken with a small angle variation will be mapped close by, to be then classified accordingly by a supervised learning algorithm.

3 Kernel Methodology

In this section we consider the structuring part of the for mentioned strategy, where we make relations between points explicit. Adapting the algorithmic procedure to use only inner products between points offers the advantage of lower computational cost, especially for high dimensional data. Indeed, this approach does not need to consider coordinates of points. Let us then frame the problem and see how we can abstract from the positions of points in space.

Definition (Finitely positive semi-definite functions [2])**.** *A function*

$$\kappa : \mathcal{X} \times \mathcal{X} \longrightarrow \mathbb{R}$$

satisfies the finitely positive semi-definite property if it is a symmetric function, for which the matrices formed by restriction to any finite subset of the space $\mathcal{X}$ are positive semi-definite. We will call such functions kernels*.*

Notice that so far we do not require $\mathcal{X}$ to have any kind of structure. The next theorem uses the kernel function to build a mapping ϕ that will send points from $\mathcal{X}$ to a space with a nice structure (Hilbert space).

Theorem 1 (Characterization of kernels [2])**.** *A function*

$$\kappa : \mathcal{X} \times \mathcal{X} \mapsto \mathbb{R},$$

satisfies the finitely positive semi-definite property if and only if there exist a Hilbert space F as well as a feature map $\phi : \mathcal{X} \mapsto F$ such that

$$\kappa(x, y) = \langle \phi(x), \phi(y) \rangle_F.$$

In other words, having a mapping ϕ from $\mathcal{X}$ to a Hilbert space leads to a kernel but the reverse is also true. We do not include the separability property into the definition of a Hilbert space here. To have this additional property, we need a topological structure on $\mathcal{X}$ as well as continuity for κ.

In practice kernels are best represented and used in matrix form:

Definition (Kernel Matrix)**.** *Given a set of vectors $V = \{v_1, \cdots, v_k\}$ the Kernel matrix is defined as the $k \times k$ matrix $\mathbf{K}$ whose entries are $\mathbf{K}_{ij} = \langle v_i, v_j \rangle$. If we are using a kernel function κ to evaluate the inner products in a feature space with feature map ϕ, the associated Kernel matrix has entries*

$$\mathbf{K}_{ij} = \langle \phi(v_i), \phi(v_j) \rangle = \kappa(v_i, v_j).$$

In [2], they produce a very explicit relations between all steps, using the kernel approach: Having a dataset and assumptions about how to build a kernel, we can then build the kernel matrix. From the kernel matrix we can then apply a learning algorithm to then have a pattern function. In this framework, the kernel matrix acts as a bottleneck for the information that comes from the dataset.

3.1 Kernel PCA

We first review briefly Principal Component Analysis. Given a set $(x_1, \cdots, x_k)$ of points in $\mathcal{X} \subseteq \mathbb{R}^n$ we want to extract principal directions of variance. Computing variances is easier done when the data set is already centered, let us assume the matrix $X = (x_1 \cdots x_k)^\top$ is centered column wise and see why this makes sense for any given directions $w \in \mathbb{S}^{n-1}$.

$$\frac{1}{k}\sum_{i=1}^{k} P_w(x_i) = \frac{1}{k}\sum_{i=1}^{k} x_i \cdot w = \frac{1}{k}(\sum_{j=1}^{n} w_j \underbrace{(x_{1j} + \cdots + x_{kj})}_{=0}) = 0$$

where the map $P_w(x) : \mathbb{S}^{n-1} \times \mathbb{R}^n \mapsto \mathbb{R}$ projects the vector x_i onto the one dimensional subspace defined by the unit direction w (scalar product as a projection). In other words, along any directions the data has mean zero. We will then assume the data is centered as above.

We formulate the problem of extracting the first principal direction $w_{(1)}$ as follows :

$$w_{(1)} = \arg\max_{w \in \mathbb{S}^{n-1}} \frac{1}{k}\sum_{i=1}^{k} (P_w(x_i))^2,$$

which can be rewritten as

$$w_{(1)} = \arg\max_{w \in \mathbb{S}^{n-1}} \frac{1}{k} w^\top X^\top X w$$

where $X^\top X$ becomes the covariance matrix. We notice that by construction, the covariance matrix is symmetric and positive semi definite.

The maximization problem can be solved using a Lagrange multiplier which then leads to the eigenvector having the largest eigenvalue.

The next principal directions $\{w_{(i)}, w_{(i+1)}, \cdots\}$ have to be chosen in the same manner in the subspace perpendicular to the span $\{w_{(1)}, \cdots, w_{(i-1)}\}$ of already found principal directions. They will naturally come from the next eigenvectors.

Using the singular value decomposition (SVD), we link the covariance matrix $C := X^\top X$ and $K := XX^\top$ as follows:

$$X^\top \overset{SVD}{=} U\Lambda^{\frac{1}{2}}V^\top \in \mathbb{R}^{k\times n} \tag{1}$$

with $U = (u_1, \cdots, u_n)$ containing eigenvectors of C and $V = (v_1, \cdots, v_k)$ those of K. Further, the diagonal matrix $\Lambda^{\frac{1}{2}}$ contains as diagonal entries the square roots of eigenvalues - say in decreasing order - shared by the two matrices. In other terms we get the two following decomposition :

$$X^\top X = U\Lambda U^\top \in \mathbb{R}^{n\times n}$$

as well as

$$K = XX^\top = V\Lambda_k V^\top \in \mathbb{R}^{k\times k}$$

where Λ_k agrees with Λ on the first n diagonal components and is extended with $k-n$ zeros, assuming we have more points than dimensions. If the number of dimensions however exceeds the number of points, a similar trick is applied but on the other decomposition. Note that the square roots are well defined since C and K are by construction positive semi definite.

We observe that having only K, we can recover $\hat{X} = V\Lambda^{\frac{1}{2}} \in \mathbb{R}^{k\times n}$. This will be a representation of X that will generate the same K and will be directed according to principal components:

$$\hat{C} = \hat{X}^\top\hat{X} = (V\Lambda^{\frac{1}{2}})^\top V\Lambda^{\frac{1}{2}} = \Lambda^{\frac{1}{2}}V^\top V\Lambda^{\frac{1}{2}} = \Lambda.$$

Finally we also observe that X And $\hat{X}$ are equivalent up to an isomorphism of $\mathbb{R}^n$. This can be seen by looking at equation (1) and noticing that U is an orthonormal matrix.

Next, we assume we are also given a kernel κ. By the above theorem we can define a mapping ϕ and an inner product space F such that for all $x, y \in \mathcal{X}$, we have $\kappa(x,y) = \langle\phi(x), \phi(y)\rangle_F$. Kernel PCA will proceed similarly only with the point images $\phi(x_1), \cdots, \phi(x_k)$. In practice the kernel matrix is sufficient as we have seen. Notice that for $(x_1, \cdots, x_k)$ in $\mathcal{X} \subseteq \mathbb{R}^n$, by taking the Hilbert space as $\mathbb{R}^n$ and the identity for ϕ, we can also build a kernel matrix K. In this case the kernel is $K := XX^\top$. Indeed, this matrix multiplication leads to inner products between the points organized in rows in X.

The dimensionality reduction is then simply carried out by removing columns from $\hat{X}$ starting from the right. Indeed, having eigenvalues $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_{\min(k,n)}$, we express the data set total variance as their sum. By removing columns from the right we explicitly choose to minimize the amount of lost variance.

Now to decide how many columns we should remove, a rule of thumb could be to take the smallest d such that $\sum_{i=1}^{d} \lambda_i / \sum_{j=1}^{\min(k,n)} \lambda_j \geq q$ with a $q \in [0,1]$ some quantile. another rule of thumb, still using eigenvalues, could be to plot them in decreasing order and look for an "elbow" in the downward going line.

It should be mentioned however that an eigenvalue analysis to determine d is not always sufficient. In [7] they give examples where the LLE algorithm cannot rely on this analysis.

4 Laplace Operator

The Laplacian is a linear operator on some function space, say $C^\infty(\Omega)$ for Ω being a plane. In our case we consider the Laplace-Beltrami operator that will then act on a function space such as $W^{2,2}(\mathcal{M})$ (space of twice differentiable real valued functions on $\mathcal{M}$ with first and second derivatives bounded in the $L^2(\mathcal{M})$ norm) with $\mathcal{M}$ a compact riemannian manifold. Coming back to the framework of statistical learning, this $\mathcal{M}$ will be an assumed smooth version of $\mathcal{X}$, the support for the training data. Thus it should also be necessary to consider the weighted Laplace-Beltrami operator since uniformly sampled objects are very unlikely to happen from real experiences. We will however not consider this relaxation. Later on Laplacian will refer to both operators, as one is a generalization of the other.

This operator will generate eigenfunctions and associated eigenvalues. As our main task is to describe the geometry of our support and building a morphism to a low dimensional space, how can the information provided by the operator help us ?

We next see three connected elements of motivation that will shed more light on such a choice. Namely, we mention inverse problem from spectral analysis, invariance under isometries and intuition behind the eigenfunctions.

4.1 Motivation

It is worth mentioning a well known problem in analysis which can be formulated as: Can One Hear the Shape of a Drum? The eponym paper by Mark Kac [1], goes through the problem of inferring the geometry of a plane region, knowing the set of eigenvalues $\{\lambda_i\}_{i\in I}$ coming from the equation

$$\Delta\varphi_k = \lambda_k\varphi_k \qquad \varphi_k|_{\partial\Omega} = 0. \tag{2}$$

To relate the problem to sound, we notice that when we fix a membrane along a (say smooth) boundary - such as the snare drum and its rim - the sound will be explained by the wave equation

$$\frac{\partial^2 F}{\partial t^2} = c^2 \Delta F$$

with $F = F(\vec{x}, t)$, c some physical constant and the laplacian acting only on the position. Now if we consider solutions of the form $F(\vec{x}, t) = U(\vec{x}) \exp(i\omega t)$, we are representing pure tones that can be reproduced by the membrane. Pure tones here mean that the underlying vibration is composed of only one sinusoidal or in other terms one frequency. Substituting U in the equation, we must have

$$c\Delta U + \omega^2 U = 0 \qquad U = 0 \text{ on the boundary.}$$

By redefining λ and including c we come back to equation (2). Under some mild conditions, it can be shown that the set of eigenvalues is discrete. Furthermore, it is possible to relate quantities such as the area, the boundary length, or the genus of Ω using asymptotics from the set of the eigenvalues. However, It has been shown that the spectrum is not sufficient to completely determine the shape of Ω.

Next we display a proof that the Laplacian commutes with isometries. This is a necessary condition, since among all potential operators on the function space, only those will provide information on the intrinsic geometry and not be affected by for example, its position within $\mathbb{R}^m$ for some m if one can find such a mapping. This will also be the occasion to define the Laplace Beltrami operator in more details.

We follow the notation and ideas given in [3].

Proposition 1 (The Laplacian commutes with isometries)**.** *Consider two riemannian manifolds (M, g_M) and (N, g_N) with an isometry $\phi : M \mapsto N$ between them. We then have*

$$\Delta_{g_M}(\varphi \circ \phi) = (\Delta_{g_N} \varphi) \circ \phi$$

for all $\varphi \in C^\infty(N)$.

To make sense of the Laplacian in the context of differential geometry, we define it using more general definitions of the gradient as well as the divergence operator.

Definition (Gradient Operator)**.** *The gradient is the operator*

$$\nabla_{g_M} : C^\infty(M) \mapsto \Gamma_{C^\infty}(TM)$$

such that

$$\langle \nabla_{g_M} \varphi, X \rangle_{g_M} = d\varphi(X) \quad \textit{for all } X \in \Gamma_{C^\infty}(TM) \textit{ and all } \varphi \in C^\infty(M)$$

where $\Gamma_{C^\infty}(TM)$ *is the space of all smooth vector fields on* M*.*

Similarly, we generalize the divergence operator. We notice that it maps vector fields to smooth functions or more abstractly, it maps a *one-form* to a *zero-form*. The central concept used here is the interior multiplication.

Definition (Divergence Operator)**.** *The divergence is the operator*

$$div_{g_M} : \Gamma_{C^\infty}(TM) \mapsto C^\infty(M)$$

such that

$$d(\iota_X \omega_{g_M}) = div_{g_M} X \cdot \omega_{g_M}.$$

The exterior derivative d and interior multiplication ι are defined in the appendix A.

Proof. To prove the proposition we prove that the gradient and the divergence operator are themselves invariant under isometries in the following sense:

$$\phi_\star \nabla_{g_M}(\varphi \circ \phi) = \nabla_{g_N} \varphi \text{ for all } \varphi \in C^\infty(N), \tag{3}$$

$$\mathrm{div}_{g_N}(\phi_\star X) \circ \phi = \mathrm{div}_{g_M}(X) \text{ for all } X \in \Gamma_{C^\infty}(TM). \tag{4}$$

Equation (3) reads: taking the isometric equivalent of φ in $C^\infty(M)$, we apply the gradient and push forward the image, a vector field. This yields the gradient of φ in $C^\infty(N)$. For equation (4), we push forward the vector field X and take its divergence on N. By composing with the isometry we get back the divergence of X on M. Both these equations are developed in appendix A.

To see it implies that the Laplacian commutes with isometries we write

$$\begin{aligned} -\Delta_{g_M}(\varphi \circ \phi) &:= \mathrm{div}_{g_M} \nabla_{g_M}(\varphi \circ \phi) \\ &= \mathrm{div}_{g_M} \phi_\star^{-1} \phi_\star \nabla_{g_M}(\varphi \circ \phi) \\ &= \mathrm{div}_{g_M} \phi_\star^{-1} \nabla_{g_N}(\varphi) \\ &= (\mathrm{div}_{g_N} \nabla_{g_N} \varphi) \circ \phi \\ &= -(\Delta_{g_N} \varphi) \circ \phi. \end{aligned}$$

This uses respectively the definition of the Laplacian, existence of the push forward inverse, equation (3), equation (4), and again the definition of Δ_{g_N}. Notice that we used the definition of the Laplacian which makes it a positive semi definite operator. □

4.2 Eigenfunctions and Approximations

For a compact Riemannian Manifold (M, g), the Laplacian will bring along a countable set of eigenfunctions that will form an orthonormal basis for $L^2(M)$ the space of square integrable functions on M. This result is called *Sturm-Liouville's decomposition.*

For $f \in L^2(M)$, we decompose it as a linear combination of the ordered eigenfunctions:

$$f = \sum_{j=1}^{\infty} a_j \varphi_j$$

where the (φ_j) are ordered with respect to eigenvalues $0 \leq \lambda_1 \leq \cdots \leq \lambda_j \leq \cdots$ and (a_j) are real coefficients.

Now one may consider $f_N = \sum_{j=1}^{N} a_j \varphi_j$, an approximation for f. By the mentioned theorem, f_N will converge to f in $L^2(M)$ norm. We additionally provide two elements that help understand in what sense f_N approximate f:

Let us first look ([8]) at the minimization problem

$$\arg\min_{f \in L^2(M)} \int_M ||\nabla f(x)||^2 \quad \text{such that} \quad ||f||_{L^2(M)} = 1.$$

By the duality between the gradient and the divergence operator, we have $\int_M ||\nabla f(x)||^2 = \int_M (\Delta f) f$. Using the Min-Max theorem, candidates for this minimization problem will be those eigenfunctions of the Laplacian with smallest eigenvalues. In fact what is being minimized here can be seen as a regularization penalty. Since $\nabla f(x)$ gives us the direction of steepest ascent at x as well as by which quantity, we constrain f to vary as little as possible.

A nice picture to have in mind is to consider the normal to a two dimensional surface living in three dimension. If we rewrite this normal using the above approximation we will have more and more details as N increases. The ordering of eigenfunctions can then be viewed intuitively as an ordering from the general form to the details.

The second element uses an analogy with the propagation of heat on a surface. For $\Omega \in \mathbb{R}^2$, we have solutions to the heat equation of the form $u(x, y, t) = \sum_i a_i e^{-\lambda_i t} \varphi_i(x, y)$ with (φ_i, λ_i) the usual ordered eigenvalue pair. Now because of the minus sign in the exponential, the first pair will provide the dominant information about the nature of the propagation since they have the slowest decay. Similarly with sound, first eigenvalues correspond to leading tones.

5 Discrete Laplace Operator

5.1 Motivation for Graphs

Definition (Graph)**.** *We define a graph to be a set $\mathcal{X}$ of vertices together with a set E of edges defined as $E = (\{(u, v) \in \mathcal{X} \times \mathcal{X} : u \neq v\}/\sim)$ with the equivalence relation $(u, v) \sim (v, u)$. A graph can additionally be equiped with weights on every edge, a real number that we denote by $w_{u,v}$.*

This definition commonly refers to a simple undirected weighted graph. The use of graphs comes in handy in our case for the following two reasons:

Locality We can see a graph as the result from a decision process on wether two points are considered to be in the same neighborhood. By fixing an arbitrary rule for decision, edges testify on the locality property between two points.

Often, to describe what the "intrinsic" geometry of a manifold means, we take the example of an ant walking on its surface (it could be a n-dimensional ant !) and walking on it. Similarly, by means of a graph, this analogy translates well in the case of a set of discrete points.

Discrete representation of diffusion In a continuous setting it is possible to study the Bronwian Motion on a manifold by taking the Laplace Beltrami operator as its infinitesimal generator. In a discrete setting, it is possible to relate a version of the graph laplacian - the random walk graph Laplacian - with a random walk on graphs (the process run the graph by randomly choosing an edge at each vertex) [4]. Furthermore, the diffusion process is central to the study of heat.

We mention two ways of building the edge set E out of a metric space $(\mathcal{X}, d)$ with the above motivation. *m-nearest neighbors* consists of an algorithm that defines the set of edges E such that if $(u, v) \in E$ either u is one of the m nearest neighbors of v or v is one of the m nearest neighbors of u.

Another way called *ϵ-balls* defines E as $(u, v) \in E \iff d(u, v) \leq \epsilon$.

5.2 Graph Laplacian

We motivate the structure of the graph laplacian using the heat equation in $\mathbb{R}^m$:

$$\Delta h(x, t) = -\frac{\partial}{\partial t} h(x, t) \quad x \in \mathbb{R}^m,\ t \in \mathbb{R}_+,$$

with initial heat distribution $f(x) = h(x, 0)$. Recall that the minus sign is a correction to make the laplacian a positive semi definite operator. The solution to the heat equation can be written in terms of the heat kernel as follows:

$$\mathbf{H}^t f(x) := h(x,t) = \int_{\mathbb{R}^m} H^t(x,y) f(y) dy.$$

By rewriting the laplacian in terms of $\mathbf{H}^t$ we get

$$\Delta f = -\frac{\partial}{\partial t} h(x,t)\Big|_{x=0} = -\frac{\partial}{\partial t} \mathbf{H}^t f(x)\Big|_{x=0} = \lim_{t\to 0} \frac{1}{t}(f(x) - \mathbf{H}^t f(x)).$$

We can then rewrite the expression using the gaussian kernel and using the fact that it integrates to one:

$$\Delta f(x) = \lim_{t\to 0} \frac{1}{t} \left(f(x) \underbrace{(4\pi t)^{-\frac{m}{2}} \int_{\mathbb{R}^m} e^{-\frac{||x-y||^2}{4t}} dy}_{=1} - (4\pi t)^{-\frac{m}{2}} \int_{\mathbb{R}^m} e^{-\frac{||x-y||^2}{4t}} f(y) dy \right). \tag{5}$$

We can then naively approximate, using a small t, the inner parenthesis in (5) from a discrete set of k points by:

$$L_k^t f(x) := \frac{1}{t} \left(f(x) - \frac{(4\pi t)^{-\frac{m}{2}}}{k} \sum_i e^{-\frac{||x-x_i||^2}{4t}} f(x_i) \right). \tag{6}$$

Let us define $C = \frac{k}{(4\pi t)^{-\frac{m}{2}}}$ and write the equation in matrix form:

$$L_k^t = \frac{C^{-1}}{t} \left(CI - e^{-\frac{||x_j - x_i||^2}{4t}} \right) \propto (CI - W)$$

which can then be applied to $[f(x_1), \cdots, f(x_k)]^\top$.

5.3 Heuristics for the choice of t

Authors of the method do not have to our knowledge a principled manner of choosing t. We try to give some insights on the effects of this parameter.

If we assume the points to be drowned from a m dimensional (smooth, compact) manifold embedded in n dimensional euclidean space, then the heat equation from equation 5 only makes sense as an approximation in a small neighborhood of x. Namely, neighboring points around x should span a near linear m dimensional subspace of $\mathbb{R}^n$.

Notice that in the limit, points far away from x in equation 6 are marginalized (relative to closer neighbors) by a narrow kernel bandwidth. This can be rigorously stated as follows.

Proposition 2 ([8]). *For a compact subset $\Omega \subseteq \mathbb{R}^m$, an open set $B \subseteq \Omega$, a point $p \in B$, and a bounded function on Ω, the quantity*

$$\left| \int_\Omega e^{-\frac{||p-y||^2}{4t}} f(y)dy - \int_B e^{-\frac{||p-y||^2}{4t}} f(y)dy \right|$$

decreases to zero at an exponential rate as t tends to zero.

In other words, by choosing t small enough, we improve the approximation by emphasizing points close enough to be near our linear subspace and by (numerically) discarding points too far away.

It is worth mentioning that the parameter t does not influence the conditioning of the associated eigenvalue problem. Indeed, $(CI - W)$ is a symmetric matrix and as such is well behaved (well-conditioned) with respect to eigenvalue decompositions (see Bauer-Fike Theorem).

However the problem that will arise when t tends to zero is that of connectedness. In most cases a connected manifold and hence a connected graph, is natural (see example 2.4). If we were to find a method that relates the choice of t with the curvature of the manifold, this would not take into account the sampling by k points of this manifold. In the case of non-uniform sampling, low density areas will possibly be disconnected from other points. Thus in practice the connectivity of the graph is favored by the N-nearest neighbors method and t is chosen big enough to prevent (numerically) disconnecting the graph.

Finally, the constant C that was introduced in the naive approximation now needs to take into account the N-nearest neighbors approach. Authors choose to scale the matrix using the degree matrix (degree of vertices on the diagonal). This approach is then closer to the Laplacian matrix from graph theory.

This then leads to

$$\mathcal{L} = D^{-1/2}(D - W)D^{-1/2}$$

where D is the diagonal matrix with degrees of vertices and the weight matrix W is defined as:

$$W_{ij} = \begin{cases} 0 & \text{if } (i,j) \notin E \\ e^{-\frac{||x-x_i||^2}{4t}} & \text{if } (i,j) \in E. \end{cases}$$

The motivation we have seen is one of many kinds of motivations for matrices with similar structure. To name a few other arguments, one could use the mean value property for the heat equation, finite differences on the Laplacian or the underlying generator for a random walk. What is interesting with this specific construction (omitting the nearest neighbors approach) is that it will lead to convergence results with respect to the eigenfunctions. In [5], they provide convergence results for the above as well as a generalization on a manifold M.

6 Review of Methods with Manifold Assumption

We next see three dimensionality reduction methods motivated by geometrical intuition. Typically the authors use terms such as "unfolding". We present them using the same framework; namely we extract the kernel they produce to later use, for example, kernel PCA.

6.1 Isomap

In [6], they rely on the intuition that lengths of geodesics must be preserved during the dimensionality reduction procedure. Thus, the algorithm will try to encode the information on lengths into the kernel.

Two concepts are central in this embedding onto lower space. The first one uses the following theorem:

Theorem 2 ([6])**.** *An Euclidean distance matrix $D \in \mathbb{R}^{k\times k}$ yields a kernel matrix $K := -\frac{1}{2}HDH$, with $H := I - \frac{1}{k}\mathbf{1}\mathbf{1}^\top$.*

This theorem actually has a converse which states that for a given kernel, it will be a Gram matrix of k points with interpoint distances given by D, i.e. $D_{ij} = ||x_i - x_j||^2$.

Now notice that we only get a kernel matrix if D is constructed with euclidean distances. This will almost never be the case and we thus need the second idea. Because the K constructed by the above theorem is not guaranteed to be positive semi definite, we need to find the closest approximation of K in the space of all positive semi definite matrices. We will rely on the next theorem:

Theorem 3 (Optimal projection)**.** *Let us decompose a symmetric matrix A as $A = U\Lambda U^\top$ with $U := [u_1, \cdots, u_k]$ the eigenfunction matrix. Then*

$$\tilde{A} := \sum_{i\,:\lambda_i \geq 0} \lambda_i u_i u_i^\top := U\Lambda_+ U^\top$$

is the best approximation with respect to the Frobenius norm. In other words, $||A - \tilde{A}|| \leq ||A - W||$ for all W in the space of positive semi definite symmetric matrices.

Proof. Notice that the space of all symmetric matrices associated with the scalar product $\langle A\,; B\rangle := tr(A^\top B)$ is a Hilbert space, that we will denote $\mathcal{H}$. Recall that the Frobenius norm is defined as $||\cdot|| = \sqrt{\langle \cdot\,;\cdot\rangle}$ with the above scalar product. Further, the subset of all positive semi definite matrices defines a closed cone that we denote $\mathcal{C} \subset \mathcal{H}$.

To find its polar cone, defined as $\hat{\mathcal{C}} := \{A \in \mathcal{H} \, : \langle A \, ; B \rangle \leq 0 \, , \forall B \in \mathcal{C}\}$, we use the Fan inequality (a refinement of Cauchy-Schwarz in our case):

$$tr(A^\top B) \leq \lambda(A)^\top \lambda(B)$$

Where $\lambda(A)$ is the vector containing eigenvalues of A. By the characterisation of positive and negative definiteness using eigenvalues, this tells us that $\hat{\mathcal{C}} \in \mathcal{H}$ is the subset containing all negative definite matrices. We now write $A = \tilde{A} + \hat{A}$ as the sum of $\tilde{A} := \sum_{i \, : \lambda_i \geq 0} \lambda_i u_i u_i^\top \in \mathcal{C}$ and $\hat{A} := \sum_{j \, : \lambda_j < 0} \lambda_j u_j u_j^\top \in \hat{\mathcal{C}}$.

By Moreau's theorem, $\tilde{A}$ and $\hat{A}$ are the projections of A on their respective cones. □

Now that we have set the background, we display steps to compute geodesics as well as the kernel.

Input k points in an n dimensional Euclidean space (i.e real coordinate space together with an Euclidean distance).

Geodesic Distances Estimation The assumed connected manifold is first represented by a graph G using for example m-nearest neighbors. Geodesics are then estimated by shortest paths within the graph (using for ex. Dijkstra algorithm).

From Distances to Kernel Knowing all distances, we define $D_{ij} := d(x_i, x_j)^2$ and use the construction of theorem 2.

And finally, we project K on the cone of semi definite matrices. This is done using the construction of theorem 3.

Output A kernel matrix K.

6.2 LLE

Locally Linear Embedding [7] computes a low dimensional representation of points with the property that nearby points remain nearby in the representation and in the similar fashion (i.e by preserving distances among neighbors). This idea assumes the underlying manifold is smooth enough and sufficiently sampled allowing each points and its neighbors to be approximately on a linear subspace.

Concretely, after building a graph under some locality rules, we express each point x_i as a suitable convex combination of its direct neighbors. It will precisely be this combination that the algorithm attempts to preserve.

The algorithm can be decomposed into two steps:

Input k points in an n dimensional Euclidean space.

Local geometry of neighborhoods Compute the coefficient matrix W by minimizing $||x_i - \sum_{j \in N(i)} W_{ij} x_j||^2$ with $N(i)$ the set of neighbors of x_i as defined by the graph. As W contains coefficient for a convex combination, its rows must sum to one.

Preserving the convex combination We now look for a $Y \in \mathbb{R}^{k \times n}$ that will minimize $\sum_i ||y_i - \sum_j W_{ij} y_j||^2$. This can also be written as

$$\sum_i (y_i - W_i Y)^\top (y_i - W_i Y) = Y^\top (I - W)^\top (I - W) Y =: Y^\top M Y.$$

Minimizing the last expression with respect to Y is dealt with an eigenvector decomposition of M. Notice that in the task of Kernel PCA, dimensionality reduction is undertook from a variance maximization view point. Therefore we need to transform our minimization problem into a maximization one:

$$\tilde{K} := (\lambda_{max} I - M)$$

where λ_{max} is the maximal eigenvalue of the system $My = \lambda y$. Finally, due to the convex constraint (i.e. rows of W sum to one), the constant vector $\mathbf{1}$ is a solution to the eigenvector problem with associated eigenvalue 0 for M, or λ_{max} in the case of $\tilde{K}$. This eigenvector has to be taken out, otherwise it will lead to degeneracy among data points in Y. Indeed each first coordinate of points would be the same. Therefore we define the kernel to be

$$K := (I - \tfrac{1}{k} \mathbf{1}\mathbf{1}^\top) \tilde{K} (I - \tfrac{1}{k} \mathbf{1}\mathbf{1}^\top)$$

by taking out the (normalized) eigenvector $\frac{1}{\sqrt{k}} \mathbf{1}$.

Notice the similarity with theorem 2 above. This can be explained by the need we have to center our mapped data points $\phi(x_i)$ in the feature space to then use Kernel PCA.

Output A kernel matrix K.

6.3 Laplacian Eigenmaps

Using the theory of sections 4 and 5 we can now present Laplacian Eigenmaps from [8]:

Input k points in an n dimensional Euclidean space.

Building the graph Laplacian We assume a graph G between points constructed using nearest neighbors. In addition, we define W to be the weight matrix (see section 5). We also define D to be the matrix with vertex degrees on its diagonal. We write $L := D - W$ the laplacian of G and $\mathcal{L} := D^{-1/2}LD^{-1/2}$ the normalized laplacian.

Eigenvalue Decomposition Using section 4, we wish to minimize the expression $Y\mathcal{L}Y$ with respect to Y by means of an eigenvalue decomposition. From a linear algebra point of view, we are in the same situation as with LLE (6.2), we therefore proceed analogously:

$$\tilde{K} := (\lambda_{max} I - \mathcal{L})$$

with λ_{max} the maximal eigenvalue of the system $\mathcal{L}y = \lambda y$ and

$$K := (I - \tfrac{1}{k}\mathbf{1}\mathbf{1}^\top)\tilde{K}(I - \tfrac{1}{k}\mathbf{1}\mathbf{1}^\top).$$

Output A kernel matrix K.

7 Conclusion

The three algorithms presented are all, when used with kernel PCA, *spectral embeddings* in the sense that they use an eigenvalue decomposition (spectral techniques) to find a mapping, and that they all attempt to preserve a particular structure in the process. Isomap focus on preserving distances, while Laplacian Eigenmaps and LLE attempt to preserve the neighborhood structure. The Laplacian Eigenmaps technique relies on the Laplace Beltrami operator to justify itself.

A great deal can be said about this operator. For example while the Riemannian metric determines the Laplacian, the converse is also true [9]. We can thus expect a deep connection between the Laplacian and the geometry of (M, g). Furthermore as this operator carries an analog of Fourier analysis on manifolds, it gives tools to change one's point of view as did the frequency domain with respect to the time domain in signal processing.

We did not investigate the weighted Laplace Beltrami operator nor did we look at the methods behavior when data is subject to specific noise, the kind that could violate the smooth manifold assumption. Furthermore, an analysis on what kind of structure one wish to preserve for dimensionality reduction would have also been interesting.

A Invariance of gradient and divergence under isometries

We consider two n-dimensional isometric Riemannian manifolds (M, g_M) and (N, g_N) with associated isometry $\phi : M \mapsto N$. We will need tools such as the pushforward for vector fields as well as the exterior derivative and interior multiplication for general tensor fields.

Definition ([10]). *Consider a smooth bijective function $f : M \mapsto N$. For $X \in \Gamma_{C^\infty}(TM)$, we want to produce $f_\star(X) \in \Gamma_{C^\infty}(TN)$ the push forward of X. Let us define it as follow:*

$$f_\star(X)(q) := df_{f^{-1}(q)}(X(f^{-1}(q))) \in T_qN \quad \forall\, q \in N.$$

Notice that for a smooth bijective function $g : N \mapsto P$, we have $g_\star f_\star X = (g \circ f)_\star(x)$. Let us denote the space of all alternating k forms on M by $\Omega^k(M)$ and by $\Omega^\star(M) := \oplus_{k=0}^n \Omega^k(M)$ the direct sum. We will also need the interior multiplication:

Definition ([3]). *For $\omega \in \Omega^k(M)$ and a smooth vector field X on M we define $\iota_X\omega \in \Omega^{k-1}(M)$ by*

$$(\iota_X\omega)_p(X_1, \cdots, X_{k-1}) = \omega_p(X, X_1, \cdots, X_{k-1})$$

for $p \in M$.

Proposition 3 ([3]).

$$\phi_\star \nabla_{g_M}(\varphi \circ \phi) = \nabla_{g_N}\varphi \text{ for all } \varphi \in C^\infty(N)$$

Proof. We respectively use the isometry property, the gradient definition, the chain rule and the push forward definition.

$$\begin{aligned}
\langle \phi_\star \nabla_{g_M}(\varphi \circ \phi), X\rangle_{g_N} &= \langle \nabla_{g_M}(\varphi \circ \phi), \phi_\star^{-1} X\rangle_{g_M} \\
&= d(\varphi \circ \phi)(\phi_\star^{-1} X) \\
&= d\varphi(d\phi(\phi_\star^{-1} X)) \\
&= d\varphi(X) \qquad\qquad \forall\, X \in \Gamma_{C^\infty}(TN).
\end{aligned}$$

□

For the next proposition we will need to use standard fact from differential geometry as well as the definition for the pull back:

Definition ([3]). *Take $f : M \mapsto N$ a smooth map. We define the pull back of f as the map $f^\star : \Omega^\star(N) \mapsto \Omega^\star(M)$ by:*

1. *$f^\star(g) = g \circ f$ for $f \in \Omega^0(N) = C^\infty(N)$.*
2. *$(f^\star\omega)_x(X_1, \cdots, X_k) = \omega_{f(x)}(f_\star X_1, \cdots, f_\star X_k)$ for $\omega \in \Omega^k(N)$ with $k \geq 1$.*

We then have

$$\phi^\star \omega_{g_N} = \omega_{g_M}$$

which follows from the defining property of riemannian volume forms and the isometry property of ϕ;

$$\iota_{\phi_\star X}((\phi^\star)^{-1}\omega_{g_M}) = (\phi^\star)^{-1}(\iota_X \omega_{g_M})$$

which follows simply by developing the expression; and finally

$$d(\phi^\star(\omega_{g_N})) = \phi^\star(d(\omega_{g_N}))$$

which is one of the defining properties of the exterior differentiation.

Proposition 4 ([3]).

$$div_{g_N}(\phi_\star X) \circ \phi = div_{g_M}(X) \qquad \forall\, X \in \Gamma_{C^\infty}(TM).$$

Proof.

$$\begin{aligned}
\mathrm{div}_{g_N}(\phi_\star X)\, \omega_{g_N} &= d(\iota_{\phi_\star X}\, \omega_{g_N}) \\
&= d(\iota_{\phi_\star X}\, (\phi^\star)^{-1}\omega_{g_M}) \\
&= d((\phi^\star)^{-1}(\iota_X \omega_{g_M})) \\
&= (\phi^\star)^{-1} d(\iota_X \omega_{g_M}) \\
&= \mathrm{div}_{g_M}(X) \circ \phi^{-1}(\phi^\star)^{-1}\omega_{g_M} \\
&= \mathrm{div}_{g_M}(X) \circ \phi^{-1}\omega_{g_N}
\end{aligned}$$

□

References

[1] Mark Kac, Can One Hear the Shape of a Drum ?,The American Mathematical Monthly, Vol. 73, No. 4, Part 2: Papers in Analysis (Apr., 1966), pp. 1-23.

[2] J. Shawe-Taylor, N. Cristianini, Kernel Methods for Pattern Analysis, Cambridge university press, New York, 2004.

[3] Yaiza Canzani, Notes for Analysis on Manifolds via the Laplacian, Harvard University, Fall 2013.

[4] Hein, M., Audibert, J.-Y., and von Luxburg, U. Graph Laplacians and their convergence on random neighborhood graphs. JMLR, 8 1325 1370, 2007.

[5] M. Belkin, P. Niyogi. Convergence of Laplacian Eigenmaps. Preprint, short version NIPS. 2008.

[6] J. B. Tenenbaum, V. deSilva, and J. C. Langford. A global geometric framework for nonlinear dimensionality reduction. Science, 290 2319 2323, 2000.

[7] L. Saul and S. Roweis. Think globally, fit locally: unsupervised learning of low dimensional manifolds. Journal of Machine Learning Research, 4 119 155, 2003.

[8] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. Neural Computation, 15(6) 1373 1396, June 2003.

[9] S. Rosenberg, The Laplacian on a Riemannian Manifold, Cambridge Univ. Press, 1997.

[10] Tom Ilmanen, Notes for Differential Geometry, ETH Zurich, Fall 2014.