Differential Operators and Nonlinear Dimension Reduction

Julian Christopher

January 27, 2021

1 Introduction

When considering data from real world sources, we wish to collect as much data as possible both in terms of the number of data points collected and the number of features measured about each point; when we collect measurements of about the features of our data points, we do not know whether some features are related to one another and what effect they may have on the outcome. In the case of data with a very large number of features it becomes more and more likely that the number of“intrinsic parameters”,aggregates of measured features with independent effects on the outcome , is far less than the number of parameters collected; this is equivalent to saying that the data all lie on some parameterizable subset of the input space $\mathbb{R}^D$. In PCA we assume that these intrinsic parameters are linear combinations of the feature space, this is a useful notion as linear relationships are common in nature, but is clearly ill suited to the case where the data do not lie on a linear subspace of $\mathbb{R}^D$. The next level of generality, and the approach we will be considering here, is to assume that the features lie on a submanifold of $\mathbb{R}^D$; this confers the obvious advantage that many nonlinear relationships between features will be considered.

How to find and do analysis on such a submanifold is the focus of the papers Hein [2005] and McInnes, Healy and Melvile [2018] that we will be taking relevant constructions from in this exposition. We will begin by discussing the methods of finding the submanifold used in the ISOMAP and UMAP algorithms. ISOMAP uses an isometric embedding while UMAP takes advantage of the freedom to define a custom metric to make the submanifold friendlier to analysis; we are mainly interested in UMAP in this report, ISOMAP is explored for contrast. We then explore methods of Kernel Density Estimation (KDE) on manifolds, a method of estimating the probability density function that generated our data. Finally we will define the divergence, gradient and Laplacian operators on the discrete data sets that lie on our manifold, and discuss how these definitions relate to the continuous operators that share their names.

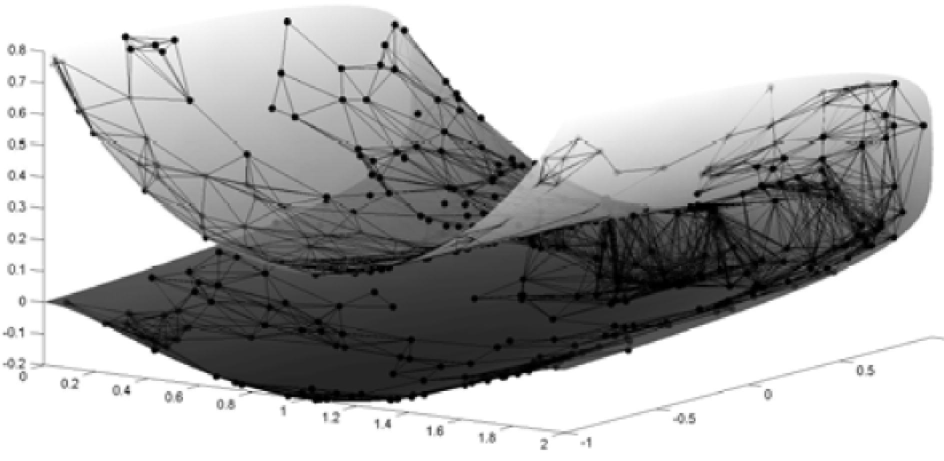

Figure 1: Graph on submanifold, image from Hein [2005]

2 The Weighted Graph Determined by UMAP

A first step in reducing our space to a submanifold is to define a weighted graph G that will form a "skeleton" for M; the approach used in ISOMAP and UMAP is to construct a weighted k-nearest neighbours graph. In both ISOMAP and UMAP we assume that, for x_i close to x_j on $\mathcal{M}$, $d_{\mathbb{R}^D}(x_i, x_j) \approx d_M(x_i, x_j)$. The spirit of this approach is well captured in figure 1 from Hein [2005].

ISOMAP simply assigns edge weights to the graph with the euclidean distance metric, $w_{ij} = d_{\mathbb{R}^N}(v_i, v_j)$; this creates the skeleton of an isometric embedding such that $w_{ij} \approx d_{\mathcal{M}}(x_i, x_j)$. The advantage of this isometric embedding is that it is highly intuitive.

With UMAP we spread the data points uniformly on M with respect to the metric $d_{\mathcal{M}}$. The approach to local distance taken in the seminal UMAP paper McInnes, Healey and Melville [2018] is based on category theoretic arguments that are beyond the scope of this discussion, and it is possible to construct and use the UMAP weighted graph with only a low resolution understanding of the approach like that presented on the UMAP documentation website [8]. The detailed construction can be found in section 2 of McInnes, Healy and Melville [2018]. For each $x \in \{x_i\}$ let $N_k(x) := \{k \text{ nearest neighbours of } x_i\}$; we can think of the k-nearest neighbours to x as belonging to a neighbourhood N of x with some probability $p_x : N_k(x) \to [1, 0)$, with the nearest neighbour x_1 having probability $p_x(x_1) = 1$ and the others decreasing in probability with distance from x. For data points y not in N_k, $p_x(y) = 0$. This concept is illustrated nicely in figure 2 from the UMAP documentation website.

To construct the $UMAP$ weighted graph G, let

$$\rho_i = \min\{d_{\mathbb{R}^d}(x_i, y) : y \in N_k(x_i)\},$$

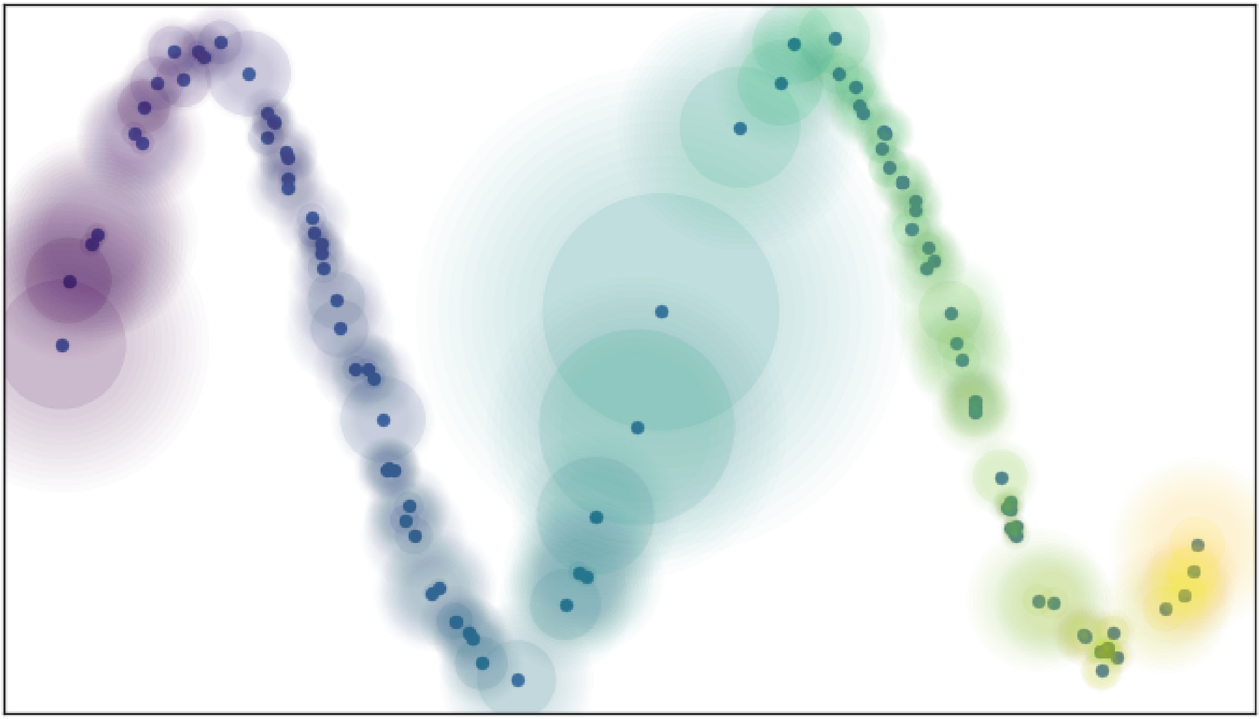

Figure 2: UMAP probabilistic topology, image from [8]

and let σ_i be such that

$$\sum_{j=1}^{k} \exp\left(\frac{-\max_{y \in N_k(x_i)}(0, d(x_i, y) - \rho_i)}{\sigma_i}\right) = \log_2(k).$$

We can now construct the graph $G = (V, E)$. The vertex set V is simply the data points $\{x_i\}_{i=1}^{N}$. Let A be the adjacency matrix for the directed graph $\tilde{G}$ with edge weights

$$w(x_i, y) = \exp\left(\frac{-\max_{y \in N_k(x_i)}(0, d(x_i, y) - \rho_i)}{\sigma_i}\right).$$

If we take this to be our final construction we would end up with $d(x_i, x_j) \neq d(x_j, x_i)$; to fix this problem, let A be the adjacency matrix of $\tilde{G}$, let $B = A + A^T - A \circ A^T$ (where $\circ : M^{N^2} \times M^{N^2} \to M^{N^2}$ is componentwise multiplication). This adjacency matrix is symmetric and so represents a weighted, undirected graph G, we will take this to be the skeleton graph of our manifold. The topology implied by this graph makes sure that

1. $\mathcal{M}$ is locally connected and

2. the points $\{x_i\}$ are roughly evenly distributed on $\mathcal{M}$.

3 Degree Functions and Kernel Density Estimation

Kernel density estimation makes the assumption that we will find more data points near the sampled data points; we then estimate the probability distribution function on all of $\mathcal{M}$ by

$$p(x) \approx \frac{1}{m_0 h^m N} \sum_{i=1}^{n} k\left(\frac{||x - x_i||_{\mathbb{R}^N}}{h}\right),\ x \in \mathcal{M}; \tag{1}$$

where $m_0 = \int_{\mathbb{R}}^{m} K(||z||)dz$, $K : \mathbb{R}_{>0} \to \mathbb{R}_{<0}$ is a function with exponential decay, and h is a hyperparameter called the *bandwidth*. A higher bandwidth gives a smoother probability distribution function. For illustration, if we take the Gaussian kernel

$$p(x) = \frac{1}{m_0 n} \sum_{i=1}^{n} \frac{1}{(n\pi h^2)^{\frac{1}{2}}} e^{-\frac{||x-x_i||^2}{2h^2}}$$

we are placing a Gaussian curve over each x_i in our data set and the bandwidth h is the variance of the Gaussian, so a smaller h means sharper peaks around the data points. For a weighted, undirected graph, define the *degree function* of a vertex v_i by $\mathsf{d}(v_i) = \sum_{j=1}^{n} w_{ij}$, the sum of weights of all edges (assume edges that don't exist have weight 0). Consider the kernel function

$$k_h(x_i, x_j) = \frac{1}{h^m} k(||x_i - x_j||^2/h^2).$$

Let $G_{k,h}$ be a weighted, undirected graph with edge weights $k_h(x_i, x_j) = w_{ij}$. We can expand the degree function from $\{x_i\}$ to all of $\mathcal{M}$: let

$$\mathsf{d}_{h,n} = \sum_{i=1}^{n} k_h(i(x), i(x_j))$$

where $i : M \to \mathbb{R}^D$ is an isometric inclusion map. We can see that the degree function is a kernel density estimator. Hein [2005], Prop. 2.31, shows that if we place certain assumptions on k then as $h \to 0$ and $nh^m \to \infty$

$$\lim_{n\to\infty} \mathsf{d}_{h,n}(x) = m_0 p(x).$$

The assumptions on k are given in **Assumption 2.25** in Hein [2005] and are as follows:

- $k : \mathbb{R}_{>0} \to \mathbb{R}$ is measurable, non-negative and non-increasing;
- $k \in C^2(\mathbb{R}_0)$;
- k, $\left|\frac{\partial k}{\partial x}\right|$, and $\left|\frac{\partial^2 k}{\partial x^2}\right|$ have exponential decay;
- $k(0) = 0$;
- there exists an $r_k > 0$ such that $k(x) \geq \frac{||k||_\infty}{2}$ for $x \in (0, r_k]$;

- k has compact support.

Note that the weighted graph produced by the UMAP algorithm satisfies the conditions on k and is therefore a good candidate for this conception of KDE; we have assumed this function is zero beyond the k^th neighbour of x_i for all i and thus this kernel has compact support. The weights of the ISOMAP graph do not allow for a kernel function satisfying these assumptions.

4 Differential operators on weighted graphs

4.1 Gradient, divergence and Laplacian of an unweighted, directed graph

The graph Laplacian on an unweighted, directed graph is a construction used in classical graph theory. I will give a short introduction to how the graph Laplacian is constructed that is inspired 5.6 of Strang [2016], and a very illuminating discussion thread that I found at *https://math.stackexchange.com/questions/1657788/discrete-laplacian.*

Consider $G = (V, E)$ a directed graph with $|E| = m$ and $|V| = n$. Let A be the incidence matrix of G, that is, $a_{ij} = 1$ if the i^{th} edge is directed towards the j^{th} vertex and $a_{ij} = -1$ if the i^{th} edge is directed away from the j^{th} vertex. We can think of A as an analogue of the gradient of a function f on the vertex set V of G, in that application of A to V gives gives as it's output in f as you move away from a vertex v_i along edge e_{ij}. More concretely, let A have n^2 rows such that row $ni + j$ corresponds to edge e_{ij}, if the edge does not exist then we will have a row of zeros; in practice the zero rows are omitted, but this representation helps with indexing. For $f : V \to \mathbb{R}$, let $f(V)$ be a column vector such that $f(V)_i = f(v_i)$, then

$$Af(V)_{ni+j} = \text{sign}(e_{ij})(f(v_j) - f(v_i));$$

The transpose of A is the analogue of the divergence operator. Let $u : E \to \mathbb{R}$, and let $u(E)$ be a vector such that $u(E)_{ni+j} = u(e_{ij})$; then

$$A^T u(E)_i = \sum_{j=1}^{n} -\text{sign}(e_{ij})u(e_{ij}),$$

the difference in flow in and flow out for v_i under u.

We can use our graph gradient divergence to define the Laplacian of a graph as $A^T A$. The analogue is both in terms of function and definition as the Laplacian from differential calculus can be defined as $\text{div}\nabla f = \sum_{i=1}^{n} \frac{\partial^2 f}{\partial x_i^2}$. We can calculate that

$$A^T Af(v)_i = \sum_{j=1, j\neq i}^{n} (f(v_i) - f(v_j))$$

So $A^T A$ applied to $f(V)$ gives the total change in f between a node and it's nearest neighbours, similar to how in calculus $\text{div}\nabla f(x)$ gives the average change in f over the unit ball around x.

4.2 The difference operator and Laplacian of a graph

We now adapt these structures to our weighted graphs as in section 2 of Hein[2005]. We define a difference operator on a function $f : V \to \mathbb{R}$ by

$$d_\gamma f(e_{ij}) = \gamma(w_{ij})(f(v_j) - f(v_i))$$

for all $e_{ij} \in E$,where $\gamma : \mathbb{R}_{>0} \to \mathbb{R}_{>0}$; the function γ allows us to decide how we would like differences in the function to interact with the weights of our graph; for example, if the weights correspond to isometric distance on $\mathcal{M}$ we may wish to choose $\gamma(w_{ij}) = \frac{1}{w_{ij}}$. Notice that d_γ can be formulated as an incidence matrix like in the previous section; this could be thought of as the incidence matrix of a graph $\tilde{G}$ with the same vertices and edges as G but with the weights adjusted for the purpose of measuring the gradient of f.

To adapt the divergence to a weighted graph we can use the adjoint of d_γ: define

$$(d^* u_\gamma)(v_i) = \frac{1}{2\chi(v_i)} \left(\frac{1}{n} \sum_{j=1}^{n} \gamma(w_{ji}) u_{ji} \phi(w_{ji}) - \frac{1}{n} \sum_{j=1}^{n} \gamma(w_{ij}) u_{ij} \phi(w_{ij}) \right),$$

where $\phi : \mathbb{R}_{>0} \to \mathbb{R}_{>0}$; this formula is derived in Hein [2005] as an adjoint operator to d_γ using a Hilbert space construction that highlights a nice symmetry between these definitions and those in the previous section.

We can now define the Laplacian:

Definition 4.1. *The graph Laplacian is defined as $\Delta = d^* d$, in components*

$$\Delta f(v_l) = \frac{1}{2n\chi(v_l)} \left[f(v_l) \sum_{i=1}^{n} \left(\gamma(w_{il})^2 \phi(w_{il}) + \gamma(w_{li})^2 \phi(w_{li}) \right) - f(v_i) \left(\gamma(w_{il})^2 \phi(w_{il}) + \gamma(w_{li})^2 \phi(w_{li}) \right) \right];$$

again $\chi : \mathbb{R}_{>0} \to \mathbb{R}_{>0}$ is an artefact of the Hilbert space argument. In this form, we can see the interpretation of the Laplacian matrix as the average of the change in f between v_i and it's nearest neighbours holds.

In the case of our skeleton graph where w_{ij} has an interpretation as a distance $w_{ij} = w_{ji}$, i.e. we have an undirected graph; Hein begins with directed graphs in his paper and I have included them in my discussion because I think that these forms are illustrative of the connection between the graph operations and those from calculus. In the case of the *undirected graph Laplacian* we have

$$\Delta f(v_l) = \frac{1}{n\chi(v_l)} \left[f(v_l) \sum_{i=1}^{n} \gamma(w_{il})^2 \phi(w_i l) - \sum_{i=1}^{n} f(v_i) \gamma(w_{il})^2 \phi(w_{il}) \right]$$

An important special case in this analysis is that of the *normalized graph Laplacian*

$$(\Delta_{norm} f)(x) = f(x) - \frac{1}{d_{h,n}} \frac{1}{n} \sum_{j=1}^{n} w_{ij} f(v_j)$$

In extending the Laplace operator we can use a kernel with the same assumptions as in the degree function to extend the normalized Laplace operator to all of $\mathcal{M}$

$$(\Delta_{h,n} f)(x) = \frac{1}{h^2}\left(f(x) - \frac{1}{d_{h,n}}\frac{1}{n}\sum_{j=1}^{n} k_h(i(x), i(x_j)) f(x_j)\right).$$

By theorem 2.37 of Hein, f $h \to 0$ and $nh^{m+4} \to \infty$, then

$$\lim_{n\to\infty} (\Delta_{h,n} f)(x) = -\frac{C}{m_0}(\Delta_2 f)(x) \textit{ in probability},$$

where $C = \int_{\mathbb{R}^m} k(||y||^2) y_1^2 dy$, and Δ_2 is a version of the Laplace-Beltrami operator, a generalization of the Laplacian to differentiable manifolds. The definition of Δ_2 is given in section 2.2 of Hein[2005], and there is an excellent discussion of the Laplace-Beltrami operator in Jürgen [2011].

5 Next steps

The exploration in this report has defined some objects that will be useful in designing sophisticated nonlinear dimension reduction schemes; the next step will be to use them. A known applications of the graph Laplacian is smoothness regularization, where a learning algorithm can be designed to favour smoother functions. It would be interesting to attempt to implement this regularization, possibly in conjunction with UMAP.

There is also much more room for exploration of the connections between the discrete an continuous versions of the operators we have discussed here. The fact that the gradient, divergence and Laplacian are such versatile tools in multivariate calculus suggests that adapting their use to manifold methods in machine learning will be fertile ground for discovery.

References

[1] Berry, Tyrus and Saur, Timothy [2016] *Density Estimation on Manifolds With Boundary*, Computational Statistics and Data Analysis 107(2017) 1-17.

[2] Bishop, Christopher M. [2006] *Pattern Recognition and Machine Learning*, Springer Science+Business Media, Inc. ISBN 9780387310732.

[3] Hein, Matthias [2017] *Geometrical Aspects of Statistical Learning Theory,* Dissertation, Universität Darmstadt.

[4] Jost, Jürgen [2011] *Riemannian Geometry and Geometric Analysis*, Springer Berlin Heidleberg. ISBN 9783642212970.

[5] McInnes, L. Healy, J. and Melville, J. [2018] *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*, Journal of Open Source Software. 3(29):861.

[6] Tenenbaum, J. de Silvia, V. Langford, J. [2000] *A Global Framework for Nonlinear Dimensionality Reduction*, Science, New Series, Vol. 290, No. 5500(Dec. 22, 2000), pp.2319-2323

[7] Strang, Gilbert. [2014] *Differential Equations and Linear Algebra*, Wellesley-Cambridge Press. ISBN 9780980232790.

[8] Umap Documentation Website *https://umap-learn.readthedocs.io/en/latest/how_umap_works.html*, accessed April 2020.