Diffusion geodesic embedding

Tom Marty
Polytechnique Montréal

Guillaume Huguet
Université de Montréal

Abstract

Dimensionality reduction techniques are often used to visualize the underlying geometry of a high-dimensional dataset. These methods usually rely on specific similarity measures. In this project, we approximate the geodesic distance on the underlying manifold to create an embedding. Our method called Diffusion geodesic follows the same outline as Isomap, where we first approximate the geodesic distance and use MDS to create the embedding. However, our approximation is very different as it relies on a diffusion process over a graph instead of the computation of the shortest path. We compare our model with popular algorithms such as PHATE, UMAP, and Isomap on toy datasets and RNA-seq dataset.

1 Introduction

In this project, we are interested in unsupervised data dimensionality reduction methods. In practice, datasets are usually in a very high-dimensional space. For example, in single-cell RNA sequencing, each cell is encoded given its genes expression. Hence, the dimension is the number of genes, which can be a few thousand. To understand the structure of the observations, or the possible existence of clusters (cells that are similar), or even the dynamic of the cell differentiation, it is necessary to use dimensionality reduction techniques. Usually, given N observations $\{x_i\}_{i=1}^N$ where $x_i \in \mathbb{R}^d$, these methods define a notion of similarity between points, and an embedding that preserves this similarity. For instance, in Diffusion map [1], the notion of similarity is the diffusion distance, which is preserved by their embedding.

In this project, we assume that the dataset is a sample from an underlying manifold, and we make any assumption on the sampling distribution. We present Diffusion Geodesic an approximation to a distance that is equivalent to the geodesic distance on the manifold. From this approximation, we create an embedding by using Multidimensional Scaling.

Our distance is closely related to many notions from the course. Indeed, it is defined by a diffusion process on a complete graph, which can also be seen as comparing a heat filter applied to different Dirac distributions centered on the vertices of the graph. It is also similar to Diffusion map and PHATE, while providing a different embedding.

Our goal is to define a meaningful distance, that would provide a good embedding. On a more personal level, our goal is to gain a better understanding of manifold learning techniques, both theoretically and practically. We also wanted to work with a complex dataset, which is why chose a cell embedding task.

We structure the project as follows. In Sec. 2, we present relevant methods on dimensionality reduction. We present our embedding method in Sec. 3.1 as well as related discussion. In Sec. 4, we present our experiments on toy datasets and RNA-seq dataset. Then we will conclude in Sec. 5.

2 Previous work

In this section, we briefly present common unsupervised dimensionality reduction techniques. Each of these can be used for visualization, by letting the target dimension be 2-D or 3-D. We note that these methods can also be used for classification, by finding an embedding that separates between classes. In this project, we focus on visualization. Here, we first present three linear methods, and we follow with the non-linear ones.

Principal Components Analysis (PCA) [2] is a linear dimensionality reduction method, its goal is to define a subspace that maximizes the variance of the dataset. A variation of PCA is Independent Component Analysis (ICA) [3], in which the features of the subspace are independent. Another popular linear method is Multidimensional Scaling (MDS) [4]. Given a distance matrix, MDS finds an embedding such that the euclidean distance in that embedding is similar to the initial one. When the distance matrix is euclidean, MDS is equivalent to PCA. These methods are very popular and well understood. However, one major drawback is that they are linear methods. In practice, the mapping from a high-dimensional manifold to a lower-dimensional space is often non-linear. Hence, they frequently fail to retrieve the structure of the manifold. For example, on the swill roll dataset, two points can be close in the euclidean space, while being very far over the manifold. These previous methods won't be able to make this distinction.

Isomap [5], tries to tackle this issue by preserving the local structure of the dataset. It is based on MDS, where the distance matrix is an approximation of the geodesic distance over the manifold. This approximation can be made using Dijkstra's algorithm [6] or Floyd–Warshall algorithm [7]. A major weakness of this method is its sensibility to the choice of the k nearest neighbor while constructing the k-nearest neighbor's graph. If k is too large it can result in "short-circuit errors" which will yield an incorrect estimation of the geodesic and a poor embedding [8]. When successful, Isomap is a great tool to represent a dataset as a continuous manifold. In this paper, we will mainly compare our method with Isomap, since we also define an embedding based on an approximation of the geodesic distance.

Another common non-linear method is t-SNE [9], which defines the similarity between data points by a joint probability measure. The embedding is defined by minimizing the Kullback-Leibler divergence between the joint distribution from the embedding and the one from the dataset. One downside of this method is the lack of global structure, which is partially solved by UMAP [10]. Both are very good at finding clusters in the dataset, however, the distances between clusters are not always meaningful (especially true for t-SNE).

The two last methods we consider are based on the idea of diffusion on a manifold. In Diffusion map [1], the similarity between points is defined by the diffusion distance, which is a weighted L^2 norm between the propagation of the heat induced by these points. Intuitively, two points are close if, starting at these points, the heat will diffuse similarly. An embedding is defined via the eigenvector of the diffusion operator, and it is shown that the euclidean distance in the embedding preserves the diffusion distance. Building upon similar ideas, PHATE [11] build a localized diffusion distance combined with MDS to create an embedding. In PHATE, the authors find an optimal time scale t using the von Neumann entropy, whereas our method combines multiple time scales of the diffusion operator. Similar to Isomap, these methods will embed a continuous manifold, and are less successful at identifying clusters (as opposed to t-SNE and UMAP).

In the next section, we propose a new dimensionality reduction technique. It follows similar steps than Isomap, and it is also closely related to PHATE. In particular, we propose an approximation of the geodesic based on the heat kernel, and we use MDS for the embedding.

3 Diffusion geodesic distance

In this section, we present our new dimensionality reduction method called Diffusion geodesic. We will first establish the theoretical basis of our approach in Sec. 3.1, then in Sec. 3.2, we will detail our approximation to the geodesic, and the resulting embedding. In section 3.3, we provide intuitions on the hyper-parameters. Finally, in Sec. 3.4, we will discuss several points concerning the choice of kernel, and the efficiency of our implementation.

3.1 Theoretical results

We start by stating an important result from [12], this result justifies our approximation of the geodesic distance. We consider a closed Riemannian manifold $(\mathcal{M}, d)$, where d is the geodesic distance, and we note h_t the heat kernel on the manifold. We note that for a point $x \in \mathcal{M}$, the heat kernel $h_t(x, \cdot)$ induces a measure, i.e. how the heat has propagated on the manifold at time t. The diffusion ground distance between $x, y \in \mathcal{M}$ is based on the L^1 norm between the measures induced by the heat kernel at x and y evaluated at different scales.

Definition 1. The diffusion ground distance between $x, y \in \mathcal{M}$ is defined as

$$D_\alpha(x, y) := \sum_{k \geq 0} 2^{-k\alpha} ||h_{2^{-k}}(x, \cdot) - h_{2^{-k}}(y, \cdot)||_1,$$

for $\alpha \in (0, 1/2)$, the scale parameter $k \geq 0$, and h_t the heat kernel on $\mathcal{M}$.

Next we state an interesting result from [12], this result links the diffusion ground distance and the geodesic.

Theorem 1. *Let $(\mathcal{M}, d)$ a closed Riemannian manifold, and d the geodesic distance. Let $\alpha \in (0, 1/2)$, the distance D_α is equivalent to $d^{2\alpha}$.*

Proof. See Theorem 2 and the conclusion of section 3.3 in [12]. □

We emphasize that this result is about equivalence and not equality. Two distances d_1 and d_2 are equivalent if, for each $x \in \mathcal{M}$, there exists two constants $c, C > 0$ such that $cd_1(x, y) \leq d_2(x, y) \leq Cd_1(x, y)$, for all $y \in \mathcal{M}$. We use the notation $d_1 \simeq d_2$ for two equivalent distances.

In practice, we cannot evaluate the heat kernel on a manifold, however results from [1] provide an approximation using an anisotropic kernel. For a positive and symmetric kernel $\tilde{k}_\epsilon(x, y) := h(||x - y||_2^2/\epsilon)$, we define the kernel

$$k_\epsilon(x, y) := \frac{\tilde{k}_\epsilon(x, y)}{\tilde{q}(x)\tilde{q}(y)} \text{ , where } \tilde{q}(x) := \int \tilde{k}_\epsilon(x, y)\mu(y)dy,$$

and μ is a density on $\mathcal{M}$ (the Borel sets are generated by the open ball w.r.t. d). Therefore, we can define the anisotropic transition kernel

$$p_\epsilon(x, y) := \frac{k_\epsilon(x, y)}{q(x)} \text{ , where } q(x) := \int k_\epsilon(x, y)\mu(y)dy.$$

As it is defined, the transition kernel P_ϵ encodes the local geometry around points of the manifold. It is shown that the operator $P_\epsilon^{t/\epsilon}$ will converge to the heat operator as ϵ goes to zero (Prop.3 [1]). We note that, if we assume a uniform distribution, the *isotropic* kernel $\tilde{k}_\epsilon(x, y)/\tilde{q}(x)$ will also converge to the heat operator.

3.2 Geodesic approximation and embedding

Since we only have access to a finite set of points, we can only approximate the kernel p_ϵ. Consider a dataset $\mathsf{X} = \{x_i\}_{i=1}^N$, where $x_i \in \mathbb{R}^d$, we build a graph with the affinity matrix defined by $(\mathbf{K}_\epsilon)_{ij} := \exp(-||x_i - x_j||_2^2/\epsilon)$, and we consider the density normalized matrix $\mathbf{M}_\epsilon := \mathbf{Q}^{-1}\mathbf{K}_\epsilon\mathbf{Q}^{-1}$, where $\mathbf{Q}_{ii} := \sum_j (\mathbf{K}_\epsilon)_{ij}$. Lastly, a Markov diffusion operator is defined by

$$\mathbf{P}_\epsilon := \mathbf{D}^{-1}\mathbf{M}_\epsilon \text{ , where } \mathbf{D}_{ii} := \sum_{j=1}^N (\mathbf{M}_\epsilon)_{ij}.$$

Here, both $\mathbf{Q}$ and $\mathbf{D}$ are diagonal matrices. We remark that the stationary distribution associated to the Markov operator $\mathbf{P}_\epsilon$ is $\boldsymbol{\pi}$, where

$$\boldsymbol{\pi}_i = \frac{\mathbf{D}_{ii}}{\sum_{j=1}^N \mathbf{D}_{jj}},$$

since $\mathbf{P}_\epsilon$ is $\boldsymbol{\pi}$-reversible. Finally, we note that the matrix $\mathbf{P}_\epsilon$ follows a similar construction as the kernel p_ϵ, essentially the integrals are approximated by sums. We refer to [1] for more information about the convergence of the matrix operator $\mathbf{P}_\epsilon$ to the operator P_ϵ.

Until now, we have defined a matrix $\mathbf{P}_\epsilon$ that approximates the operator P_ϵ, which in turn converges to the heat operator. Now, we define an approximation of the diffusion ground distance, based on the matrix $\mathbf{P}_\epsilon$. We use the notation $(\mathbf{P}_\epsilon)_{i:}$ to represent the i-th row of $\mathbf{P}_\epsilon$.

Definition 2. We define the Diffusion geodesic between $x_i, x_j \in \mathsf{X}$ as

$$G(x_i, x_j) := \sum_{k=0}^{K} 2^{-(K-k)/2} ||(\mathbf{P}_\epsilon)_{i:}^{2^{(K-k)}} - (\mathbf{P}_\epsilon)_{j:}^{2^{(K-k)}}||_1 + 2^{-(K+1)/2} ||\boldsymbol{\pi}_i - \boldsymbol{\pi}_j||_1.$$

The idea behind this definition is that for K large enough, $(\mathbf{P}_\epsilon)_{i:}^{2^{(K+1)}}$ approximates $h_{-\infty}(x_i, \cdot)$ the stationary heat kernel on $\mathcal{M}$. We can then only consider the K first time scales and the stationary distribution $\boldsymbol{\pi}$.

Embedding Now that we defined a meaningful distance matrix on our manifold, one can decide to extract a 2D-embedding from the distance matrix using multidimensional-scaling. One advantage of our embedding is that it requires fewer assumptions on the data to generate a potentially relevant embedding, as opposed to other modern approaches. For example, unlike UMAP, we do not assume that the data is uniformly distributed on a Riemannian manifold.

3.3 Parameters and implementation

Our model relies on two hyper-parameters; the bandwidth ϵ and the maximum dyadic scale K. Both parameters impact directly the computational cost and precision with respect to the diffusion ground distance presented in Def. 1.

The parameter ϵ defines the width of the Gaussian kernel. On a manifold, this parameter controls the weight given to the neighbors of a point. As ϵ increases, more weight is given to the distant points, it can be seen as considering more neighbors in a KNN graph. This parameter acts exactly like the variance for a Gaussian distribution.

Regarding the value of the maximum dyadic time scale K, the higher it is, the better the approximation of the diffusion ground distance is. However, as the computational cost grows of $(\mathbf{P}_\epsilon)_{i:}^{2^{(K-k)}}$ exponentially with the time scale K, the choice of K becomes a computational trade-off *that relies on our tolerance to approximations*. It could be possible to control this approximation by considering the mixing time of the diffusion operator, i.e. the number of steps required to be arbitrarily close to the stationary distribution. As we saw in this course, there exists a lower bound on the mixing time for a specific random walk, this bound could provide a heuristic in our case. Intuitively, we know that for a more complex manifold, the sequence of diffusion operators might take longer to converge to its stationary distribution π. In that case, the computation of the distance G should consider terms of higher time scale, in order to give a better approximation of the diffusion ground distance. On the other hand, for simpler manifolds, fewer time scales will be required to get a relatively good approximation of the diffusion ground distance.

Curse of dimensionality Our method is affected by the curse of dimensionality since the diffusion kernel is based on the euclidean distance. When dimension increases, the space volume increases exponentially such that data becomes sparse, and data points distant from each other. Hence, capturing a meaningful local measure of the volume $\tilde{q}(x)$ requires to increase the value of the width of the Gaussian kernel ϵ depending on the input data dimension. These two considerations define a Cornelian trade-off in which we need to assign a value that satisfies both conflicting conditions. We observed that for higher dimensional dataset we needed a large ϵ in order to capture relations between the points, and to find a meaningful embedding. Another way to achieve this is by taking a small ϵ, resulting in a sparse transition matrix, and using more transition scales. Intuitively, both the bandwidth ϵ and the time t, control the transition probability of farther neighbors. By increasing t, more steps are taken in a random walk, and by increasing ϵ more weight is given to distant points.

3.4 Discussion

Impact of input distribution One notable characteristic of our embedding is that it doesn't require any assumption on the input data distribution thanks to the double normalization introduced before [1]. This is an interesting feature since most datasets are obtained from non-uniform distributions. Hence, it will be interesting to compare the robustness of different embeddings to non-uniformly distributed input data. This is what we do in Sec. 4.1 by comparing embeddings obtained using either an anisotropic or an isotropic kernel.

KNN graph We note that we could also define the diffusion kernel based on the Laplacian of a k-nearest neighbors graph. However, this method would still suffer from the curse of dimensionality, since the KNN graph is constructed using the euclidean distance. Moreover, because the diffusion matrix would be very sparse, we would need to compute many time scales in order to understand the global geometry of the manifold. Hence, even though the matrix is sparse, it could be even less computationally efficient. This is partially why we preferred to use a complete graph, for which we need to fine-tune the bandwidth parameter ϵ, instead of the number of nearest neighbors.

Condensation algorithm We also tried to combine our approximation with the Condensation algorithm [13]. This algorithm provides a multiscale representation of a dataset. Indeed, we consider an initial dataset X_0 for which we evaluate the diffusion operator P_0, and we recursively update the dataset as $\mathsf{X}_{t+1} := \mathbf{P}_t \mathsf{X}_t$. In our case, we were interested in the resulting family of diffusion operators $\{\mathbf{P}_i\}_{i=0}^K$, which we used to compute the Diffusion geodesic distance instead of powers of the same diffusion operator. We present some embeddings in the appendix Fig. 10. We did not observe much improvement, further work needs to be done to define a scheduling of ϵ, and to speed up the algorithm.

Performance If at first sight, our method seems not to be as efficient as other dimension reduction methods in terms of computing time, it is important to remember that several improvements could have been done regarding the code optimization. For instance, as our solution mainly relies on matrix calculation, our approach could be highly parallelizable, resulting in significantly faster computing time. We could also use a Chebyshev approximation of the Heat filter, applied to a delta distribution centered at a given vertex. The main drawback of our method is that it relies on MDS, so it is necessary to store a $N \times N$ distance matrix. However, this is also true for methods such as Isomap and PHATE.

4 Experiments

In this section, we present all the different experiments we made on our Diffusion geodesic embedding method. First, we will compare our method with different dimension reduction methods such as t-SNE or UMAP, and validate theoretical properties of our embedding. Finally, we will discuss the impact of kernel choice and hyper-parameters on datasets of increasing dimensions.

4.1 Toy examples

We first use our method on toy datasets, for which we are able to visually judge the quality of the embedding. We consider the *Blobs*, which consists of five 10 dimensional Gaussian distributions. We use the *Iris* dataset that contains observations on four features of three types of iris. We also use the popular *swiss roll* and the *sphere*, where points that are close on the manifold share the same color. For each dataset, we implemented t-SNE, Isomap, MDS, UMAP, Diffusion map, PHATE, and our proposed Diffusion geodesic. Further details on the implementation are available in Appendix A. The results are shown in Fig. 1.

Like we previously mentioned, we observe that t-SNE and UMAP successfully identify the clusters, but are less efficient to understand the underlying manifold, e.g. the representation of the sphere is poor. The embeddings from MDS and Isomap seem to be good representations, both in therm of clustering and visualization of the manifold. Diffusion map, PHATE, and our method provide rather similar embeddings. Interestingly, our method and Isomap have different behavior on the Iris and Blobs datasets, albeit both relying on an approximation of the geodesic.

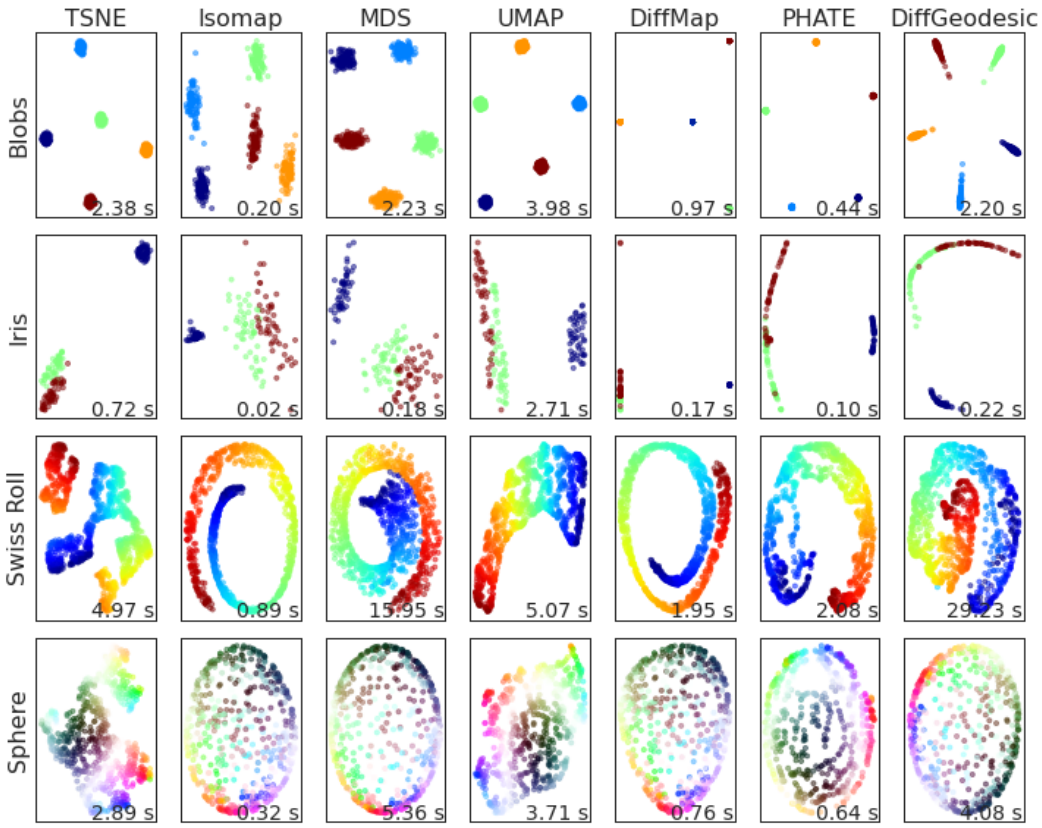

Figure 1: Embedding of selected methods on toy datasets

Anisotropic or isotropic kernel We study the influence of the choice of the kernel. In particular, in Fig. 2 we produced the embedding using the anisotropic and isotropic kernel. We remark that, apart from the Iris dataset (Fig. 8), we cannot always retrieve a proper embedding by using the isotropic kernel. This observation is consistent with the remarks in [1]. The isotropic kernel can provide an approximation to the heat kernel if the underlying distribution is uniform. When this assumption is not satisfied, the impact of the density on the quality of the embedding can be important (see section 3.2 [1]). For instance, in this experiment, we generated several embeddings of four different datasets (Blobs, Iris, Swiss Roll and Sphere) using either an anisotropic kernel or an isotropic kernel. While the embedding of the Iris dataset seems not to be really impacted by the nature of the kernel, we lost almost all the information about the underlying structure on every other dataset.

Bandwidth parameter For certain datasets, the choice of the bandwidth parameter ϵ is crucial to learn a representation of the manifold. Indeed, we find that for observations in a high dimensional space, it is necessary to increase the bandwidth parameter. This is precisely due to the curse of dimensionality explained previously. The maximum dimension of the previous dataset was 10 (the Blobs), in Fig. 3, we present the influence of ϵ on the embedding of the *Digits* datasets (8×8 pictures of the 10 digits), for which the observations are in a 64 dimensional space.

The previous experiment clearly exposed a dichotomy between theoretical results and practical implementation. Indeed, theoretically, we should favor a very low ϵ and a large number of scales. Such a parametrization would result in a more accurate approximation of the heat operator. However, from our experiments, we observe that allowing more information from distant neighbors (i.e. increasing ϵ), prevails over the theoretical necessity of approximating the heat kernel, and yields a more expressive embedding. Increasing ϵ has also its own limitation, in Fig. 9 we present various embeddings of the swiss roll for very large ϵ. We think that the degradation in quality of the embedding is due to two things. First, the approximation of the heat kernel must be worse when we increase the bandwidth parameter ϵ, hence the approximation of the geodesic is worsened. Second, as

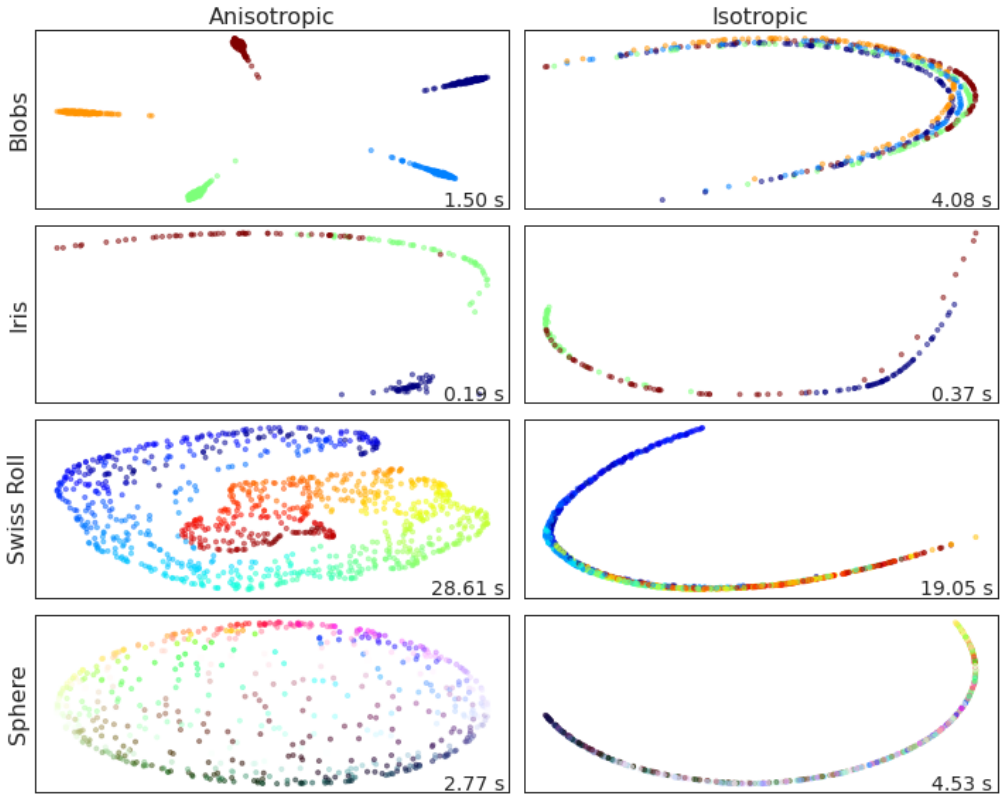

Figure 2: Anisotropic and Isotropic kernel for Diffusion geodesic

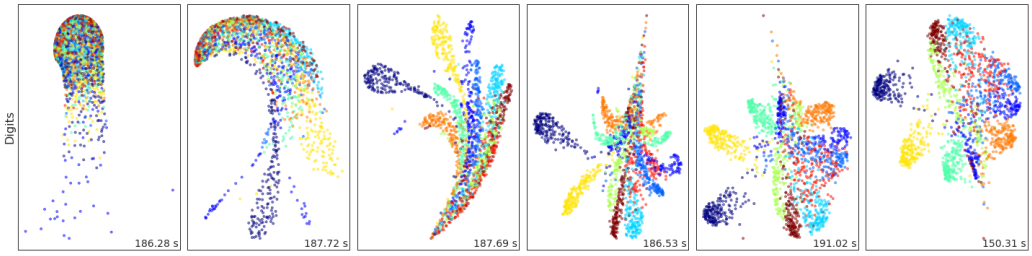

Figure 3: Embeddings of the Digits for $\epsilon \in \{50, 100, 150, 200, 250, 300\}$

ϵ increases, the weights of distant neighbors increases, and the distinction between local and global geometry becomes ambiguous, resulting in a poor representation of the manifold.

4.2 Cell embedding

Furthermore, we wanted to evaluate the performance of our method on a dataset with higher dimensions and complex intrinsic geometry. Hence, we focused on the Cells dataset, presented in [11] and in Krishnaswamy Lab's workshops [1], a dataset with 9750 entries of dimension 16507 showing different cells' RNA sequencing at successive moments of their differentiation. Among these genes, some of them known as housekeeping genes are expressed everywhere, yet in different intensities. Some others are cell-type specific, and are more likely to be present in a small region of the manifold. What is really interesting in this dataset, apart from its high dimension, is the inherent dynamic of the differentiation phase. The label of each entry does not represent a specific class as usual, but different period of their growth. We thus expect to see a time dynamic in the embedding. This can be verified on Fig. 4, where the first axis resembles time, and the second expose an increasing variance (suggesting cellular heterogeneity).

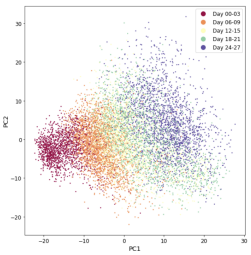

Figure 4: 2D-PCA on Cells dataset

Identification of different growing periods and cells subgroups We compared three methods including our method on a truncated subset obtained by keeping the 20 first dimensions of a PCA on the original dataset. While we used the 9750 entries for PHATE and UMAP, we had to reduce the dataset to 2000 points to obtain results in a reasonable amount of time with our method.

The first evaluation metric was to analyze how well a method could retrieve the cell differentiation dynamic, and cell clusters. In practice, these kinds of results would allow an expert to potentially establish the cell lineage, and identify key-moments of differentiation during the growth.

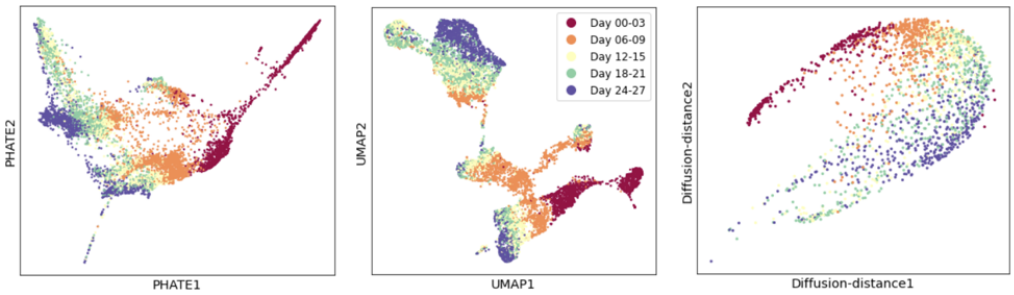

Figure 5: Comparison of different embeddings

In Fig. 5, we see that neighboring clusters in the UMAP embedding are the ones at successive growth periods, letting us think that we could run through the embedding and identify key-moments of differentiation which ultimately gave rise to different cell subgroups identified in each blue cluster.

As expected, our method has more difficulty separating classes into clearly identified clusters. While PHATE performs better at this task, the best result in terms of class separation goes to UMAP that provides a representation that retains the global structure and with the least overlapping of classes. This phenomenon is one downside of methods such as Diffusion distance which defines the similarity between data points using a specific distance on the manifold. For instance, two type-specific cells, (let say a cell of the retina and a bone marrow cell) could be far from each other (ie. genetically specialized) while being at the same growth period. The embedding based on our computed diffusion distance is more likely to keep them away from each other, resulting in a poor method to identify cells of the same growth period. On the other hand, PHATE and our method successfully represent the cell dynamic on the manifold.

Analysis of genes expression along the embedding Recall that a gene expression represents one dimension of the input data, in a second time, we plotted the expression of three specific genes (namely one housekeeping gene: ACTB, and two cell-specific genes: SOX10, HAND1) for the three previously defined embeddings. The results are given in Fig. 6. We then wanted to analyze the

[1] `https://www.krishnaswamylab.org/workshop`

capacity of each embedding to keep track of zones of high expression. Being able to isolate these specific genes is a good indicator of the capacity of an embedding to identify all different cell-types.

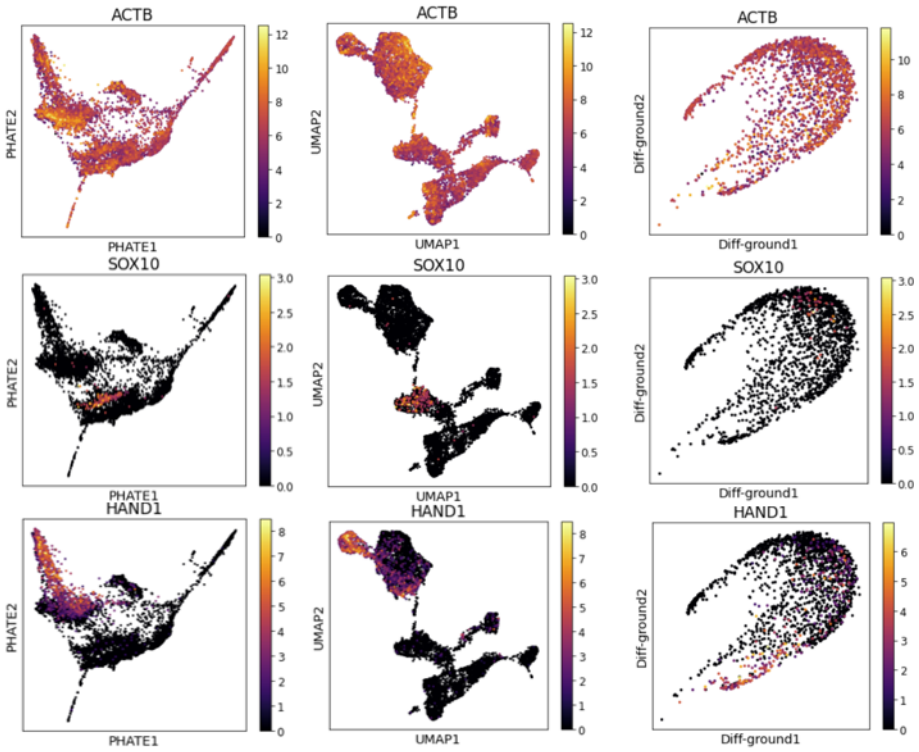

Figure 6: genes expression for 3 specific genes on 3 dimension reduction method : PHATE, UMAP and Diffusion distance

Once again, our method does not perform as well as PHATE or UMAP to identify specific type of cells, whether for housekeeping genes or cell-type specific genes. While the first two embeddings provide a localized representation of cells with equivalent levels of gene expression, our method fails to spatially contain these cells in a localized area.

Study of the importance of the bandwidth ϵ, depending on the dataset dimension For this experiment, we started by precomputing a PCA with dimension 1000 on the cells dataset in order to focus on the sub-space where the highest variance in the data is observed.

We generated new truncated datasets by selecting only some of the first PCA_{dims} dimensions out of an initial PCA with dimension 1000. In Fig. 7, we plotted several embeddings for multiple couples (PCA_{dims}, ϵ) In this experiment, we can again clearly see the effect of the *curse of dimensionality*. As long as dimension increases, the bandwidth ϵ needs to be adjusted (i.e. increased) to consider points further apart than before in the computation of the diffusion matrix. For too small values of ϵ, almost no neighbors are taken into account in the computation of the matrix $\mathcal{M}_\epsilon$, making any points indistinguishable from one another, and resulting in a non-expressive embedding.

5 Conclusion

In conclusion, we defined a new method for dimension reduction based on the approximation of the geodesic distance that requires fewer assumptions on the input data than other commonly used embedding methods. Our method succeeds to provide a relevant embedding on several datasets, while being able to keep track of the underlying structure of the manifold. However, while being close in its design from PHATE, our method suffer from a higher computational cost due to successive matrix multiplication, and has more difficulties to clearly identify group of interest in high dimensional manifold.

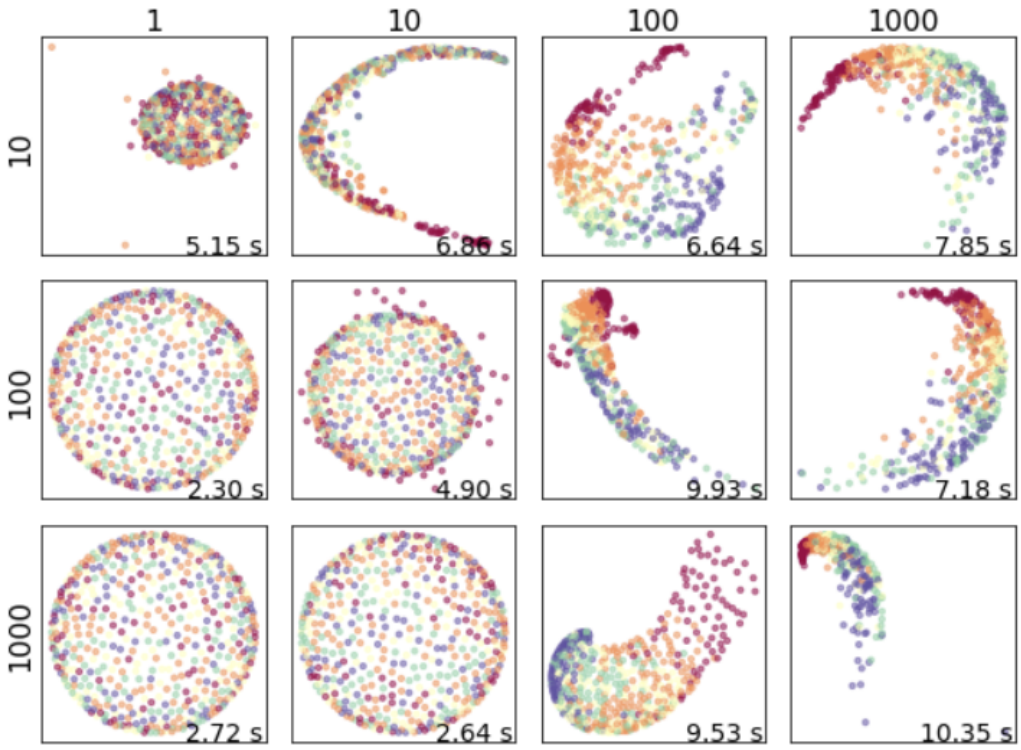

Figure 7: Embeddings on subspace of dimension $PCA_{dims} \in \{10, 100, 1000\}$ obtained with different $\epsilon \in \{1, 10, 100, 1000\}$

5.1 Contribution

For this project, I worked with Guillaume Huguet. For my part, I first worked on the elaboration of an analogy between our diffusion operator and the lower bound on the mixing time during a random walks on a specific weighted graph. Unfortunately, we have not succeeded in establishing a clear mathematical relationship between the 2 concepts.

I also implemented a first version in python of Diffusion Distance, which underwent several improvements as the project progressed. Finally, I worked on the analysis of the cells dataset and the influence of hyperparameters on the quality of the embeddings obtained.

I learned a lot from Guillaume's experience in research who took the lead in the theoretical foundations of the project and guided me in this first research work. we hope to be able to pursue this project thereafter.

References

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

[2] Ian T Jolliffe. Principal components in regression analysis. In *Principal component analysis*, pages 129–155. Springer, 1986.

[3] Pierre Comon. Independent component analysis, a new concept? *Signal processing*, 36(3):287–314, 1994.

[4] Michael AA Cox and Trevor F Cox. Multidimensional scaling. In *Handbook of data visualization*, pages 315–347. Springer, 2008.

[5] Joshua B Tenenbaum, Vin De Silva, and John C Langford. A global geometric framework for nonlinear dimensionality reduction. *science*, 290(5500):2319–2323, 2000.

[6] Edsger W Dijkstra et al. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[7] Robert W Floyd. Algorithm 97: shortest path. *Communications of the ACM*, 5(6):345, 1962.

[8] Mukund Balasubramanian, Eric L Schwartz, Joshua B Tenenbaum, Vin de Silva, and John C Langford. The isomap algorithm and topological stability. *Science*, 295(5552):7–7, 2002.

[9] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[10] Leland McInnes, John Healy, and James Melville. Umap: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018.

[11] 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, et al. Visualizing structure and transitions in high-dimensional biological data. *Nature biotechnology*, 37(12):1482–1492, 2019.

[12] William Leeb and Ronald Coifman. Hölder–Lipschitz Norms and Their Duals on Spaces with Semigroups, with Applications to Earth Mover's Distance. *Journal of Fourier Analysis and Applications*, 22(4):910–953, August 2016.

[13] Nathan Brugnone, Alex Gonopolskiy, Mark W Moyle, Manik Kuchroo, David van Dijk, Kevin R Moon, Daniel Colon-Ramos, Guy Wolf, Matthew J Hirn, and Smita Krishnaswamy. Coarse graining of data via inhomogeneous diffusion condensation. In *2019 IEEE International Conference on Big Data (Big Data)*, pages 2624–2633. IEEE, 2019.

[14] Tim Sainburg, Leland McInnes, and Timothy Q Gentner. Parametric UMAP Embeddings for Representation and Semisupervised Learning. *Neural Computation*, 33(11):2881–2907, 2021.

A Appendix

A.1 Implementation details

We use the implementation from Scikit Learn for TSNE, PCA, Isomap, and MDS. For UMAP we used [14], for PHATE we used the implementation [11], and *Pydiffmap* for Diffusion map.

Toy examples For the toy example, we used the same parameters for all the datasets. For t-SNE we set the perplexity to 50. For Isomap, we set the number of neighbors to 30. For UMAP, we set the number of neighbors to 30, and the threshold distance to be 0.30. For Diffusion map, we considered 2000 neighbors, the *bgh generous* for the bandwidth scheduling, and we set *alpha* to 1. For PHATE, we used 5 nearest neighbors. Lastly, for our method, we used the anisotropic kernel and the bandwidth parameter set to 10, and $K = 2$.

A.2 Additional results

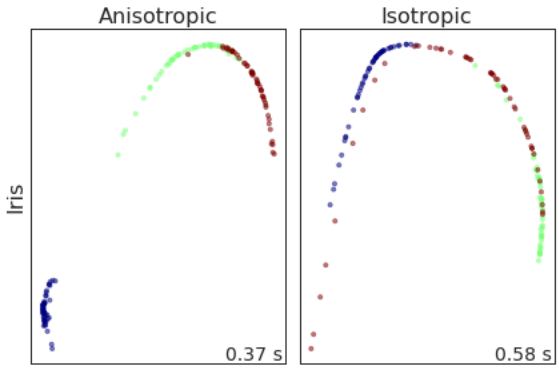

Figure 8: Diffusion geodesic with Anisotropic or Isotropic kernel on the Iris dataset

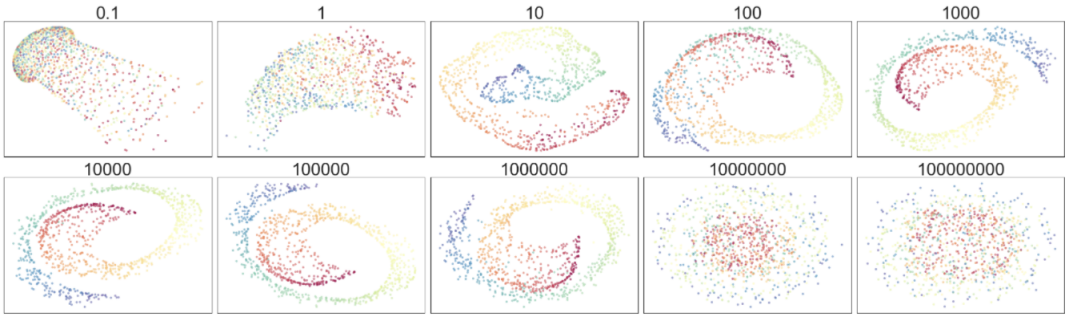

Figure 9: Diffusion distance embedding on swissroll dataset for $\epsilon \in \{0, 1, 1, \ldots, 10^8\}$

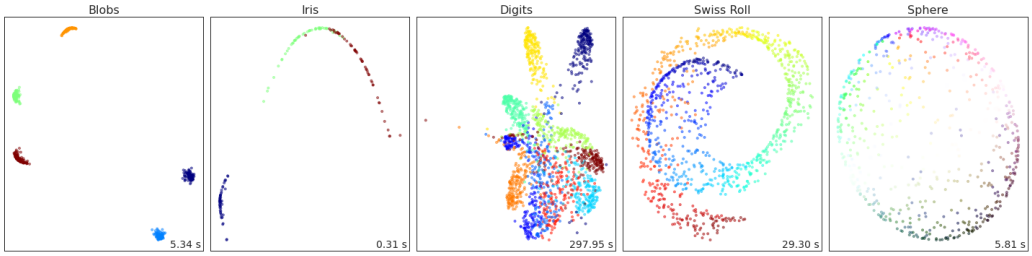

Figure 10: Embeddings using 3 condensation steps with $\epsilon = 250$