UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction

Leland McInnes and John Healy
Tutte Institute for Mathematics and Computing
leland.mcinnes@gmail.com jchealy@gmail.com

February 13, 2018

Abstract

UMAP (Uniform Manifold Approximation and Projection) is a novel manifold learning technique for dimension reduction. UMAP is constructed from a theoretical framework based in Riemannian geometry and algebraic topology. The result is a practical scalable algorithm that applies to real world data. The UMAP algorithm is competitive with t-SNE for visualization quality, and arguably preserves more of the global structure with superior run time performance. Furthermore, UMAP as described has no computational restrictions on embedding dimension, making it viable as a general purpose dimension reduction technique for machine learning.

1 Introduction

Dimension reduction seeks to produce a low dimensional representation of high dimensional data that preserves relevant structure (relevance often being application dependent). Dimension reduction is an important problem in data science for both visualization, and as a potential pre-processing step for machine learning.

As a fundamental technique for both visualization and preprocessing, dimension reduction is being applied in a broadening range of fields and on ever increasing sizes of datasets. It is thus desirable to have an algorithm that is both scalable to massive data and able to cope with the diversity of data available. Dimension reduction algorithms tend to fall into two categories; those that seek to preserve the distance structure within the data or those that favor the preservation of local distances over global distance. Algorithms such as PCA [8], MDS

[9], and Sammon mapping [20] fall into the former category while t-SNE [15] [26], Isomap [24], LargeVis [22], Laplacian eigenmaps [1] [2], diffusion maps [4], NeRV [28], and JSE [12] all fall into the latter category.

UMAP (Uniform Manifold Approximation and Projection) seeks to provide results similar to t-SNE but builds upon mathematical foundations related to the work of Belkin and Niyogi on Laplacian eigenmaps. In particular, we seek to address the issue of uniform distributions on manifolds through a combination of Riemannian geometry and the work of David Spivak [21] in category theoretic approaches to geometric realization of fuzzy simplicial sets.

In this paper we introduce a novel manifold learning technique for dimension reduction. We provide a sound mathematical theory grounding the technique and a practical scalable algorithm that applies to real world data. t-SNE is the current state-of-the-art for dimension reduction for visualization. Our algorithm is competitive with t-SNE for visualization quality and arguably preserves more of the global structure with superior run time performance. Furthermore, UMAP's topological foundations allow it to scale to significantly larger data set sizes than are feasible for t-SNE. Finally, UMAP has no computational restrictions on embedding dimension, making it viable as a general purpose dimension reduction technique for machine learning.

In section 2 we describe the theory underlying the algorithm. In section 3 we provide practical results on real world datasets.

2 The UMAP algorithm

In overview, UMAP uses local manifold approximations and patches together their local fuzzy simplicial set representations. This constructs a topological representation of the high dimensional data. Given a low dimensional representation of the data, a similar process can be used to construct an equivalent topological representation. UMAP then optimizes the layout of the data representation in the low dimensional space, minimizing the cross-entropy between the two topological representations.

The construction of fuzzy topological representations can be broken down into the two problems: approximating a manifold on which the data is assumed to lie; and constructing a fuzzy simplicial set representation of the approximated manifold. In explaining the algorithm we will first discuss the method of approximating the manifold for the source data. Next we will discuss how to construct a fuzzy simplicial set structure from the manifold approximation. We will then

discuss the construction of the fuzzy simplicial set associated to a low dimensional representation (where the manifold is simply $\mathbb{R}^d$), and how to optimize the representation. Finally we will discuss some of the implementation issues.

2.1 Uniform distribution of data on a manifold and geodesic approximation

The first step of our algorithm is to find an estimate of the manifold we assume the data lies on. The manifold may be known apriori (as simply $\mathbb{R}^n$) or may need to be inferred from the data. Suppose the manifold is not known in advance and we wish to approximate geodesic distance on it. Let the input data be $X = \{X_1, \ldots, X_N\}$. As in the work of Belkin and Niyogi on Laplacian eigenmaps [1] [2], for theoretical reasons it is beneficial to assume that the data is uniformly distributed on the manifold. In practice, real world data is rarely so nicely behaved. However, if we assume that the manifold has a Riemannian metric not inherited from the ambient space, we can find a metric such that the data is approximately uniformly distributed with regard to that metric.

Formally, let $\mathcal{M}$ be the manifold we assume the data to lie on, and let g be the Riemannian metric on $\mathcal{M}$. Thus, for each point $p \in \mathcal{M}$ we have g_p, an inner product on the tangent space $T_p\mathcal{M}$.

Lemma 1. *Let $(\mathcal{M}, g)$ be a Riemannian manifold in an ambient $\mathbb{R}^n$, and let $p \in M$ be a point. If g is locally constant about p in an open neighbourhood U such that g is a constant diagonal matrix in ambient coordinates, then in a ball $B \subseteq U$ centered at p with volume $\frac{\pi^{n/2}}{\Gamma(n/2+1)}$ with respect to g, the geodesic distance from p to any point $q \in B$ is $\frac{1}{r} d_{\mathbb{R}^n}(p, q)$, where r is the radius of the ball in the ambient space and $d_{\mathbb{R}^n}$ is the existing metric on the ambient space.*

Proof. Let $x^1, \ldots, x^n$ be the coordinate system for the ambient space. A ball B in $\mathcal{M}$ under Riemannian metric g has volume given by

$$\int_B \sqrt{\det(g)} dx^1 \wedge \cdots \wedge dx^n.$$

If B is contained in U, then g is constant in B and hence $\sqrt{\det(g)}$ is constant and can be brought outside the integral. Thus, the volume of B is

$$\sqrt{det(g)} \int_B dx^1 \wedge \ldots \wedge dx^n = \sqrt{det(g)} \frac{\pi^{n/2} r^n}{\Gamma(n/2+1)},$$

where r is the radius of the ball in the ambient $\mathbb{R}^n$. If we fix the volume of the ball to be $\frac{\pi^{n/2}}{\Gamma(n/2+1)}$ we arrive at the requirement that

$$\det(g) = \frac{1}{r^{2n}}.$$

Now, since g is assumed to be diagonal with constant entries we can solve for g itself as

$$g_{ij} = \begin{cases} \frac{1}{r^2} \text{ if } i = j, \\ 0 \text{ otherwise} \end{cases}. \tag{1}$$

The geodesic distance on $\mathcal{M}$ under g from p to q (where $p, q \in B$) is defined as

$$\inf_{c \in C} \int_a^b \sqrt{g(\dot{c}(t), \dot{c}(t))} dt,$$

where C is the class of smooth curves c on $\mathcal{M}$ such that $c(a) = p$ and $c(b) = q$, and $\dot{c}$ denotes the first derivative of c on $\mathcal{M}$. Given that g is as defined in (1) we see that this can be simplified to

$$\begin{aligned} &\frac{1}{r} \inf_{c \in C} \int_a^b \sqrt{\langle \dot{c}(t), \dot{c}(t) \rangle} dt \\ =&\frac{1}{r} \inf_{c \in C} \int_a^b \|\dot{c}(t)\| dt \\ =&\frac{1}{r} d_{\mathbb{R}^n}(p, q) \end{aligned} \tag{2}$$

□

If we assume the data to be uniformly distributed on $\mathcal{M}$ (with respect to g) then any ball of fixed volume should contain approximately the same number of points of X regardless of where on the manifold it is centered. Conversely, a ball centered at X_i that contains exactly the k-nearest-neighbors of X_i should have fixed volume regardless of the choice of $X_i \in X$. Under Lemma 1 it follows that we can approximate geodesic distance from X_i to its neighbors by normalising distances with respect to the distance to the k^{th} nearest neighbor of X_i.

In essence, by creating a custom distance for each X_i, we can ensure the validity of the assumption of uniform distribution on the manifold assumption. The cost is that we now have an independent notion of distance for each and

every X_i, and these notions of distance may not be compatible. That is, we have a family of discrete metric spaces (one for each X_i) that we wish to merge into a consistent global structure. This can be done in a natural way by converting the metric spaces into fuzzy simplicial sets.

2.2 Fuzzy topological representation

We will convert to fuzzy topological representations as means to merge the incompatible local views of the data. The topological structure of choice is that of simplicial sets. For more details on simplicial sets we refer the reader to [7] and [6]. Our approach draws heavily upon the work of David Spivak in [21], and many of the definitions and theorems below are drawn from those notes.

Definition 1. *The category $\mathbf{\Delta}$ has as objects the finite order sets $[n] = \{1, \ldots, n\}$, with morphims given by (non-strictly) order-preserving maps.*

Definition 2. *A* simplicial set *is a functor from $\mathbf{\Delta}^{op}$ to **Sets**, the category of sets.*

Simplicial sets provide a combinatorial approach to the study of topological spaces. In contrast, we are dealing with metric spaces, and require a similar structure that carries with it metric information. Fortunately the complete theory for this has already been developed by Spivak in [21]. Specifically, he extends the classical theory of singular sets and topological realization (from which the combinatorial definitions of simplicial sets were originally derived) to fuzzy singular sets and metric realization. We will briefly detail the necessary terminology and theory below, following Spivak.

Let I be the unit interval $(0, 1] \subseteq \mathbb{R}$ with topology given by intervals of the form $(0, a)$ for $a \in (0, 1]$. The category of open sets (with morphisms given by inclusions) can be imbued with a Grothendieck topology in the natural way for any poset category.

Definition 3. *A presheaf $\mathscr{P}$ on I is a functor from I^{op} to **Sets**. A* fuzzy set *is a presheaf on I such that all maps $\mathscr{P}(a \leq b)$ are injections.*

Presheaves on I form a category with morphisms given by natural transformations. We can thus form a category of fuzzy sets by simply restricting to those presheaves that are fuzzy sets. We note that such presheaves are trivially sheaves under the Grothendieck topology on I. A section $\mathscr{P}([0, a))$ can be thought of as the set of all elements with membership strength at least a. We can now define the category of fuzzy sets.

Definition 4. *The category* ***Fuzz*** *of* fuzzy sets *is the full subcategory of sheaves on I spanned by fuzzy sets.*

Defining fuzzy simplicial sets is simply a matter of considering presheaves of $\mathbf{\Delta}$ valued in the category of fuzzy sets rather than the category of sets.

Definition 5. *The category of* fuzzy simplicial sets ***sFuzz*** *is the category with objects given by functors from $\mathbf{\Delta}^{op}$ to* ***Fuzz****, and morphisms given by natural transformations.*

Alternatively, a fuzzy simplicial set can be viewed as a sheaf over $\mathbf{\Delta} \times I$, where $\mathbf{\Delta}$ is given the trivial topology and $\mathbf{\Delta} \times I$ has the product topology. We will use $\Delta^n_{<a}$ to denote the sheaf given by the representable functor of the object $([n], (0, a))$. The importance of this fuzzy (sheafified) version of simplicial sets is their relationship to metric spaces. We begin by considering the larger category of extended-pseudo-metric spaces.

Definition 6. *An* extended-pseudo-metric space (X, d) *is a set X and a map $d : X \times X \to \mathbb{R}_{\geq 0} \cup \{\infty\}$ such that*

1. $d(x, y) \geqslant 0$*, and $x = y$ implies $d(x, y) = 0$;*
2. $d(x, y) = d(y, x)$*; and*
3. $d(x, z) \leqslant d(x, y) + d(y, z)$.

The category of extended-pseudo-metric spaces ***EPMet*** *has as objects extended-pseudo-metric spaces and non-expansive maps as morphisms. We denote the subcategory of finite extended-pseudo-metric spaces* ***FinEPMet****.*

The choice of non-expansive maps in Definition 6 is due to Spivak, but we note that it closely mirrors the work of Carlsson and Memoli in [3] on topological methods for clustering as applied to finite metric spaces. This choice is significant since pure isometries are too strict and do not provide large enough Hom-sets.

In [21] Spivak constructs a pair of adjoint functors, Real and Sing between the categories **sFuzz** and **EPMet**. These functors are the natural extension of the classical realization and singular set functors from algebraic topology (see [7] or [16] for example). We are only interested in finite metric spaces, and thus use the analogous adjoint pair FinReal and FinSing. Formally we define the finite realization functor as follows:

Definition 7. *Define the functor* $\mathsf{FinReal} : \mathbf{sFuzz} \to \mathbf{FinEPMet}$ *by setting*

$$\mathsf{FinReal}(\Delta^n_{<a}) \triangleq (\{x_1, x_2, \ldots, x_n\}, d_a),$$

where

$$d_a(x_i, x_j) = \begin{cases} -\log(a) & \text{if } i \neq j, \\ 0 & \text{otherwise} \end{cases}.$$

and then defining

$$\mathsf{FinReal}(X) \triangleq \operatorname*{colim}_{\Delta^n_{<a} \to X} \mathsf{FinReal}(\Delta^n_{<a}).$$

A morphism $(\sigma, \leq) : ([n], ([0, a)) \to ([m], ([0, b))$ only exists for $a \leq b$, and in that case we can define

$$\mathsf{FinReal}((\sigma, \leq)) : \mathsf{FinReal}(\Delta^n_{<a}) \to \mathsf{FinReal}(\Delta^m_{<b})$$

to be the map

$$(\{x_1, x_2, \ldots, x_n\}, d_a) \mapsto (\{x_{\sigma(1)}, x_{\sigma(2)}, \ldots, x_{\sigma(n)}\}, d_b),$$

which is non-expansive since $a \leq b$ implies $d_a \geq d_b$.

Since FinReal preserves colimits it admits a right adjoint, the fuzzy singular set functor FinSing. To define the fuzzy singular set functor we require some further notation. Given a fuzzy simplicial set X let $X^n_{<a}$ be the set $X([n], (0, a))$. We can then define the fuzzy singular set functor in terms of the action of its image on $\mathbf{\Delta} \times I$.

Definition 8. *Define the functor* $\mathsf{FinSing} : \mathbf{FinEPMet} \to \mathbf{sFuzz}$ *by*

$$\mathsf{FinSing}(Y)^n_{<a} \triangleq \hom_{\mathbf{FinEPMet}}(\mathsf{FinReal}(\Delta^n_{<a}), Y).$$

With the necessary theoretical background in place, the means to handle the family of incompatible metric spaces described above becomes clear. Each metric space in the family can be translated into a fuzzy simplicial set via the fuzzy singular set functor, distilling the topological information while still retaining metric information in the fuzzy structure. Ironing out the incompatibilities of the resulting family of fuzzy simplicial sets can be done by simply taking a (fuzzy) union across the entire family. The result is a single fuzzy simplicial set which captures the relevant topological and underlying metric structure of the manifold $\mathcal{M}$.

It should be noted, however, that the fuzzy singular set functor applies to extended-pseudo-metric spaces, which are a relaxation of traditional metric spaces. The results of Lemma 1 only provide accurate approximations geodesic distance local to X_i for distances measured from X_i – the geodesic distances between other pairs of points within the neighborhood of X_i are not well defined. In deference to this uncertainty we define distances between X_j and X_k in the extended-pseudo metric space local to X_i (where $i \neq j$ and $i \neq k$) to be infinite (local neighborhoods of X_j and X_k will provide suitable approximations).

For real data it is safe to assume that the manifold $\mathcal{M}$ is locally connected. In practice this can be realized by measuring distance in the extended-pseudo-metric space local to X_i as geodesic distance *beyond* the nearest neighbor of X_i. Since this sets the distance to the nearest neighbor to be equal to 0; this is only possible in the more relaxed setting of extended-pseudo-metric spaces. It ensures, however, that each 0-simplex is the face of some 1-simplex with fuzzy membership strength 1, meaning that the resulting topological structure derived from the manifold is locally connected. We note that this has a similar practical effect to the truncated similarity approach of Lee and Verleysen [13], but derives naturally from the assumption of local connectivity of the manifold.

Combining all of the above we can define the fuzzy topological representation of a dataset.

Definition 9. *Let $X = \{X_1, \ldots, X_N\}$ be a dataset in $\mathbb{R}^n$. Let $\{(X, d_i)\}_{i=1\ldots N}$ be a family of extended-pseudo-metric spaces with common carrier set X such that*

$$d_i(X_j, X_k) = \begin{cases} d_{\mathcal{M}}(X_j, X_k) - \rho & \text{if } i = j \text{ or } i = k, \\ \infty & \text{otherwise .} \end{cases}$$

where ρ is the distance to the nearest neighbor of X_i and $d_{\mathcal{M}}$ is geodesic distance on the manifold $\mathcal{M}$, either known apriori, or approximated as per lemma 1.

The fuzzy topological representation of X is

$$\bigcup_{i=1}^{n} \mathsf{FinSing}((X, d_i)).$$

The (fuzzy set) union provides the means to merge together the different metric spaces. This provides a single fuzzy simplicial set as the global representation of the manifold formed by patching together the many local representations.

Given the ability to construct such topological structures, either from a known manifold, or by learning the metric structure of the manifold, we can perform

dimension reduction by simply finding low dimensional representations that closely match the topological structure of the source data. We now consider the task of finding such a low dimensional representation.

2.3 Optimizing a low dimensional representation

Let $Y = \{Y_1, \ldots, Y_N\} \subseteq \mathbb{R}^d$ be a low dimensional ($d \ll n$) representation of X such that Y_i represents the source data point X_i. In contrast to the source data where we want to estimate a manifold on which the data is uniformly distributed, we know the manifold for Y is $\mathbb{R}^d$ itself. Therefore we know the manifold and manifold metric apriori, and can compute the fuzzy topological representation directly. Of note, we still want to incorporate the distance to the nearest neighbor as per the local connectedness requirement. This can be achieved by supplying a parameter that defines the expected distance between nearest neighbors in the embedded space.

Given fuzzy simplicial set representations of X and Y, a means of comparison is required. If we consider only the 1-skeleton of the fuzzy simplicial sets we can describe each as a fuzzy graph, or, more specifically, a fuzzy set of edges. To compare two fuzzy sets we will make use of fuzzy set cross entropy. For these purposes we will revert to classical fuzzy set notation. That is, a fuzzy set is given by a reference set A and a membership strength function $\mu : A \to [0, 1]$. Comparable fuzzy sets have the same reference set. Given a sheaf representation $\mathscr{P}$ we can translate to classical fuzzy sets by setting $A = \bigcup_{a \in (0,1]} \mathscr{P}([0, a))$ and $\mu(x) = \sup\{a \in (0, 1] \mid x \in \mathscr{P}([0, a))\}$.

Definition 10. *The cross entropy C of two fuzzy sets (A, μ) and (A, ν) is defined as*

$$C((A, \mu), (A, \nu)) \triangleq \sum_{a \in A} \mu(a) \log\left(\frac{\mu(a)}{\nu(a)}\right) + (1 - \mu(a)) \log\left(\frac{1 - \mu(a)}{1 - \nu(a)}\right).$$

Similar to t-SNE we can optimize the embedding Y with respect to fuzzy set cross entropy C by using stochastic gradient descent. However, this requires a differentiable fuzzy singular set functor. If the expected minimum distance between points is zero the fuzzy singular set functor is differentiable for these purposes, however for any non-zero value we need to make a differentiable approximation (chosen from a suitable family of differentiable functions).

This completes the algorithm: by using manifold approximation and patching together local fuzzy simplicial set representations we construct a topological

representation of the high dimensional data. We then optimize the layout of data in a low dimensional space to minimize the error between the two topological representations.

2.4 Implementation

Practical implementation of this algorithm requires k-nearest-neighbor calculation and efficient optimization via stochastic gradient descent.

Efficient approximate k-nearest-neighbor computation can be achieved via the Nearest-Neighbor-Descent algorithm of Dong et al. [5]. The error intrinsic in a dimension reduction technique means that such approximation is more than adequate for these purposes.

In optimizing the embedding under the provided objective function, we follow work of Tang et al. [22]; making use of probabilistic edge sampling and negative sampling [17]. This provides a very efficient approximate stochastic gradient descent algorithm since there is no normalization requirement. Furthermore, since the normalized Laplacian of the fuzzy graph representation of the input data is a discrete approximation of the Laplace-Betrami operator of the manifold (see [1] and [2]), we can provide a suitable initialization for stochastic gradient descent by using the eigenvectors of the normalized Laplacian.

Combining these techniques results in highly efficient embeddings, which we will discuss in the next section. A reference implementation can be found at `https://github.com/lmcinnes/umap`.

3 Experimental results

While the strong mathematical foundations of UMAP were the motivation for its development, it must ultimately be judged by its practical efficacy. In this section we examine the fidelity and performance of low dimensional embeddings of multiple diverse real world data sets under UMAP. The following datasets were considered:

COIL 20 [18] A set of 1440 greyscale images consisting of 20 objects under 72 different rotations spanning 360 degrees. Each image is a 128x128 image which we treat as a single 16384 dimensional vector for the purposes computing distance between images.

COIL 100 [19] A set of 7200 colour images consisting of 100 objects under 72 different rotations spanning 360 degrees. Each image consists of 3 128x128 in-

tensity matrices (one for each color channel). We treat this as a single 49152 dimensional vector for the purposes of computing distance between images.
Statlog (Shuttle) [14] is a NASA dataset consisting of various data associated to the positions of radiators in the space shuttle, including a timestamp. The dataset has 58000 points in a 9 dimensional feature space.
MNIST [11] is a dataset of 28x28 pixel grayscale images of handwritten digits. There are 10 digit classes (0 through 9) and 70000 total images. This is treated as 70000 different 784 dimensional vectors.
F-MNIST [29] or Fashion MNIST is a dataset of 28x28 pixel grayscale images of fashion items (clothing, footwear and bags). There are 10 classes and 70000 total images. As with MNIST this is treated as 70000 different 784 dimensional vectors.
GoogleNews word vectors [17] is a dataset of 3 million words and phrases derived from a sample of Google News documents and embedded into a 300 dimensional space via word2vec.

For all the datasets except GoogleNews we use Euclidean distance between vectors. For GoogleNews, as per [17], we use cosine distance (or angular distance in t-SNE which does support non-metric distances).

3.1 Qualitative analysis

The current state of the art for dimension reduction for visualisation purposes is the t-SNE algorithm of Hinton and Van der Maaten [27] (and variations thereupon). In comparison to previous techniques, including PCA [8], multidimensional scaling [9], and Isomap [23], t-SNE offers a dramatic improvement in finding and preserving local structure in the data. This makes t-SNE the benchmark against which any dimension reduction technique must be compared.

We claim that the quality of embeddings produced by UMAP is comparable to t-SNE when reducing to two or three dimensions. For example, Figure 1 shows both UMAP and t-SNE embeddings of the COIL20, MNIST, Fashion MNIST, and Google News datasets. While the precise embeddings are different, UMAP distinguishes the same structures as t-SNE.

It can be argued that UMAP has captured more of the global and topological structure of the datasets than t-SNE. More of the loops in the COIL20 dataset are kept intact, including the intertwined loops. Similarly the global relationships among different digits in the MNIST digits dataset are more clearly captured with 1 (red) and 0 (dark red) at far corners of the embedding space, and

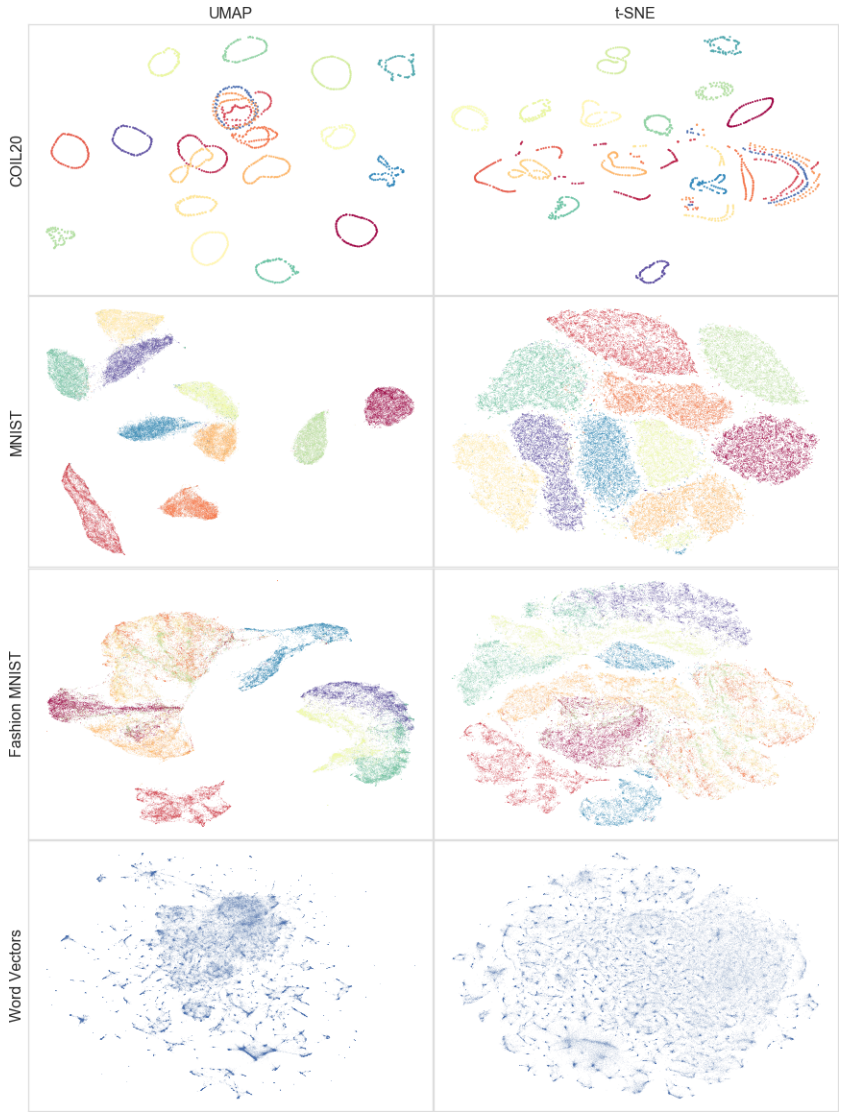

Figure 1: Comparison of UMAP and t-SNE embeddings for a number of real world datasets. More of the loops in the COIL20 dataset are kept intact, including the intertwined loops by UMAP. Similarly the global relationships among different digits in the MNIST digits dataset are more clearly captured with 1 (red) and 0 (dark red) at far corners of the embedding space, and 4,7,9 (yellow, sea-green, and violet) and 3,5,8 (orange, chartreuse, and blue) separated as distinct clumps of similar digits. In the Fashion MNIST dataset the distinction between clothing (dark red, yellow, orange, vermilion) and footwear (chartreuse, sea-green, and violet) is made more clear.

4,7,9 (yellow, sea-green, and violet) and 3,5,8 (orange, chartreuse, and blue) separated as distinct clumps of similar digits. In the Fashion MNIST dataset the distinction between clothing (dark red, yellow, orange, vermilion) and footwear (chartreuse, sea-green, and violet) is made more clear. Finally, while both t-SNE and UMAP capture groups of similar word vectors, the UMAP embedding arguably evidences a clearer global structure among the various word clusters.

3.2 Performance and Scaling

For performance comparisons we chose to compare with MulticoreTSNE [25], which we believe to be the fastest extant implementation of t-SNE at this time, even when run in single core mode. It should be noted that MulticoreTSNE is a heavily optimized implementation written in C++ based on Van der Maaten's `bhtsne` [26] code. In contrast our UMAP implementation was written in Python (making use of the numba [10] library for performance). MulticoreTSNE was run in single threaded mode to make fair comparisons to our single threaded UMAP implementation.

Benchmarks against the various real world datasets were performed on a Macbook Pro with a 3.1 GHz Intel Core i7 and 8GB of RAM. Scaling benchmarks on the Google News dataset were performed on a server with Intel Xeon E5-2697v4 processors and 512GB of RAM due to memory constraints on loading the full size dataset.

Table 1: Runtime of UMAP and t-SNE on various datasets

dataset	**dataset size**	**t-SNE**	**UMAP**
COIL20	1440x16384	20s	**7s**
COIL100	72000x49152	683s	**121s**
Shuttle	58000x9	741s	**140s**
MNIST	70000x784	1337s	**98s**
F-MNIST	70000x784	906s	**78s**
GoogleNews	200000x300	16214s	**821s**

As can be seen in Table 1, t-SNE scales with both dataset size and dataset dimension. In contrast, scaling of our UMAP implementation is largely dominated by dataset size. It is also worth noting that while Barnes-Hut t-SNE is reliant on quad-trees or oct-trees in low dimensional embedding space, the UMAP implementation has no such restrictions, and thus scales easily with respect to embed-

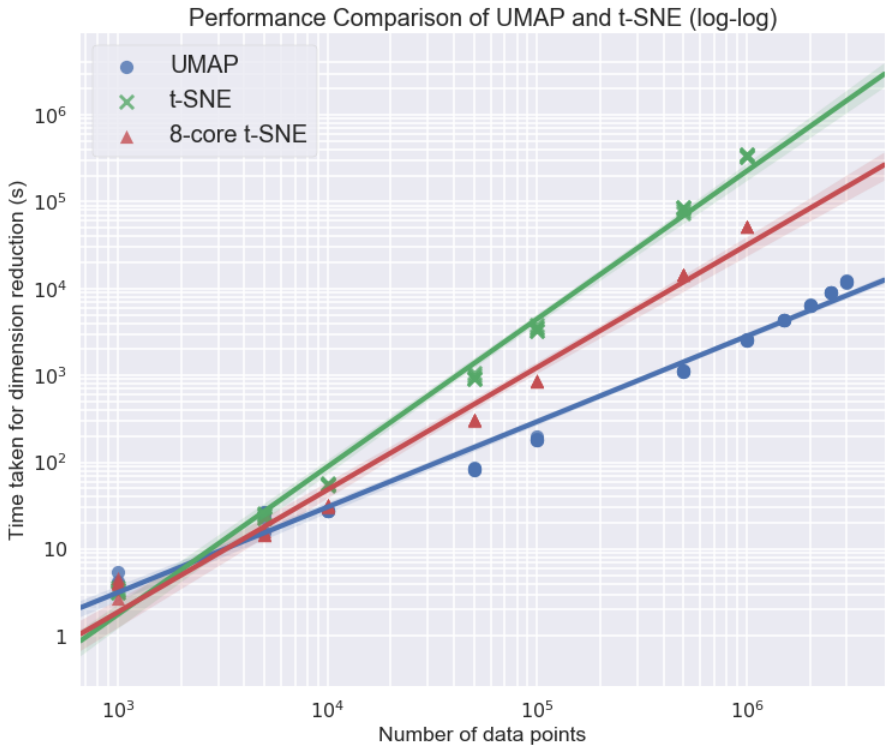

Figure 2: Runtime performance scaling of t-SNE and UMAP on various sized sub-samples of the full Google News dataset. The lower t-SNE line is the wall clock runtime for Multicore t-SNE using 8 cores.

ding dimension. This allows UMAP to be used as a general purpose dimension reduction technique rather than merely as a visualization technique.

As a more direct comparison of runtime scaling performance with respect to dataset size, the GoogleNews dataset was sub-sampled at varying dataset sizes. The results, as depicted in Figure 2, show that UMAP has superior asymptotic scaling performance, and on large data performs roughly an order of magnitude faster than t-SNE even on multiple cores. The UMAP embedding of the full GoogleNews dataset of 3 million word vectors, as seen in Figure 3, was completed in around 200 minutes, as compared with several days required for t-SNE, even using multiple cores.

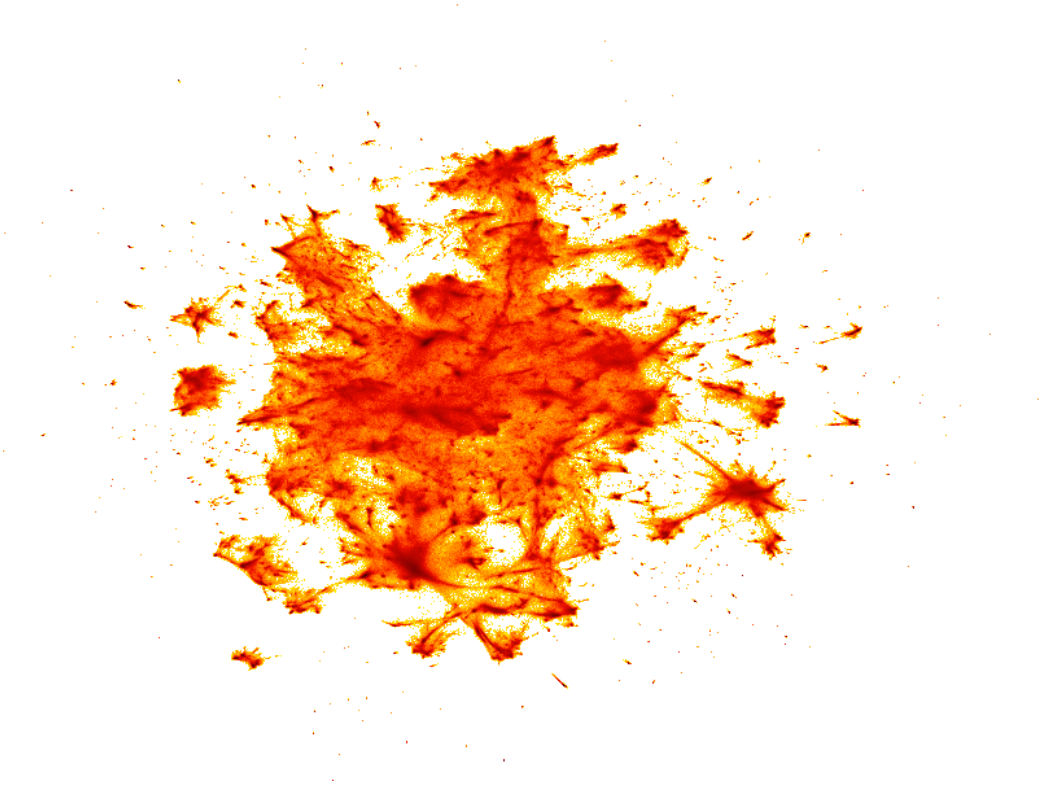

Figure 3: Visualization of the full 3 million word vectors from the GoogleNews dataset as embedded by UMAP.

4 Conclusions

We have developed a general purpose dimension reduction technique that is grounded in strong mathematical foundations. The algorithm is demonstrably faster than t-SNE and provides better scaling. This allows us to generate high quality embeddings of larger data sets than had been previously attainable.

References

[1] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in neural information processing systems*, pages 585–591, 2002.

[2] Mikhail Belkin and Partha Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003.

[3] Gunnar Carlsson and Facundo Mémoli. Classifying clustering schemes. *Foundations of Computational Mathematics*, 13(2):221–252, 2013.

[4] Ronald R Coifman and Stéphane Lafon. Diffusion maps. *Applied and computational harmonic analysis*, 21(1):5–30, 2006.

[5] Wei Dong, Charikar Moses, and Kai Li. Efficient k-nearest neighbor graph construction for generic similarity measures. In *Proceedings of the 20th International Conference on World Wide Web*, WWW '11, pages 577–586, New York, NY, USA, 2011. ACM.

[6] Greg Friedman et al. Survey article: an elementary illustrated introduction to simplicial sets. *Rocky Mountain Journal of Mathematics*, 42(2):353–423, 2012.

[7] Paul G Goerss and John F Jardine. *Simplicial homotopy theory*. Springer Science & Business Media, 2009.

[8] Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.

[9] J. B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, Mar 1964.

[10] Siu Kwan Lam, Antoine Pitrou, and Stanley Seibert. Numba: A llvm-based python jit compiler. In *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, LLVM '15, pages 7:1–7:6, New York, NY, USA, 2015. ACM.

[11] Yann Lecun and Corinna Cortes. The MNIST database of handwritten digits.

[12] John A Lee, Emilie Renard, Guillaume Bernard, Pierre Dupont, and Michel Verleysen. Type 1 and 2 mixtures of kullback–leibler divergences as cost functions in dimensionality reduction based on similarity preservation. *Neurocomputing*, 112:92–108, 2013.

[13] John A Lee and Michel Verleysen. Shift-invariant similarities circumvent distance concentration in stochastic neighbor embedding and variants. *Procedia Computer Science*, 4:538–547, 2011.

[14] M. Lichman. UCI machine learning repository, 2013.

[15] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

[16] J Peter May. *Simplicial objects in algebraic topology*, volume 11. University of Chicago Press, 1992.

[17] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[18] Sameer A. Nene, Shree K. Nayar, and Hiroshi Murase. Columbia object image library (coil-20. Technical report, 1996.

[19] Sameer A. Nene, Shree K. Nayar, and Hiroshi Murase. object image library (coil-100. Technical report, 1996.

[20] John W Sammon. A nonlinear mapping for data structure analysis. *IEEE Transactions on computers*, 100(5):401–409, 1969.

[21] David I Spivak. Metric realization of fuzzy simplicial sets. *Self published notes*.

[22] Jian Tang, Jingzhou Liu, Ming Zhang, and Qiaozhu Mei. Visualizing large-scale and high-dimensional data. In *Proceedings of the 25th International Conference on World Wide Web*, pages 287–297. International World Wide Web Conferences Steering Committee, 2016.

[23] Joshua B. Tenenbaum. Mapping a manifold of perceptual observations. In M. I. Jordan, M. J. Kearns, and S. A. Solla, editors, *Advances in Neural Information Processing Systems 10*, pages 682–688. MIT Press, 1998.

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

[25] Dmitry Ulyanov. Multicore-tsne. `https://github.com/DmitryUlyanov/Multicore-TSNE`, 2016.

[26] Laurens Van Der Maaten. Accelerating t-sne using tree-based algorithms. *Journal of machine learning research*, 15(1):3221–3245, 2014.

[27] Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-SNE. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[28] Jarkko Venna, Jaakko Peltonen, Kristian Nybo, Helena Aidos, and Samuel Kaski. Information retrieval perspective to nonlinear dimensionality reduction for data visualization. *Journal of Machine Learning Research*, 11(Feb):451–490, 2010.

[29] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.