Dimensionality reduction with UMAP

Advanced Data Mining seminar

Jakub Bartczuk

March 2020

Overview

Teaser

Reducing dimensionality of COIL20 Dataset

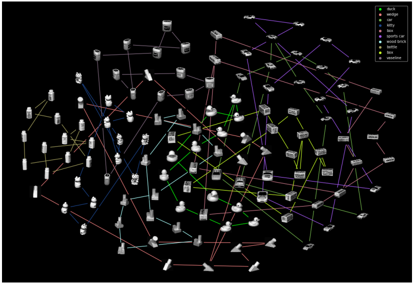


Figure: tSNE

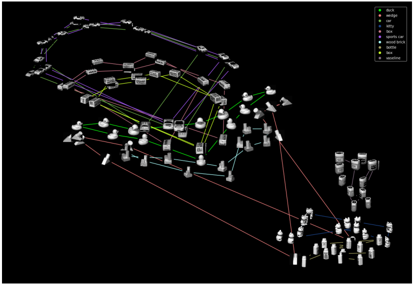
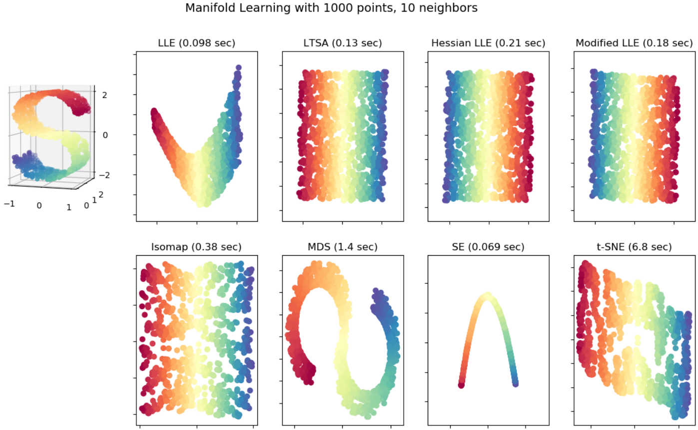


Figure: UMAP

Manifold learning recap

- We want to uncover lower-dimensional structure in high-dimensional space
- Is the structure linear? If yes, use PCA
- What to do if it is not linear?

Manifold learning recap



Classical Multidimensional Scaling

Algorithm: MDS(X, d)

$D_{i,j} = \|x_i - x_j\|^2$
Find y's: $y_i \in \mathbb{R}^d$
$D'_{i,j} = \|y_i - y_j\|^2$
Such that $D' \approx D$
Minimize $R = \sum_{i,j} \sqrt{\|x_i - x_j\|^2 - \|y_i - y_j\|^2}$

Classical MDS properties

- easy to compute - decompose low-rank matrix from D

Classical MDS properties

- easy to compute - decompose low-rank matrix from D
- fast - Euclidean distance matrix just couple of vectorized matrix computations

Classical MDS properties

- easy to compute - decompose low-rank matrix from D
- fast - Euclidean distance matrix just couple of vectorized matrix computations
- treats all distortions alike

Classical MDS properties

- easy to compute - decompose low-rank matrix from D
- fast - Euclidean distance matrix just couple of vectorized matrix computations
- treats all distortions alike

What structure does MDS preserve?

The last property basically means that we preserve local and global structure alike.

- x's are close - they are close in embedding.
- x's are distant - they are equally distant in embedding.

Distances, graphs and matrix decomposition

Local vs global structure

We may not care about preserving exact large distances in embedding space as much as small distances

Idea: Calculate distance along the set 'spanned' by datapoints

Manifold hypothesis

The data lies in lower-dimensional, possibly nonlinear space which is embedded in ambient space

MDS breakdown

1. estimate distances between some pairs of close points
2. make graph (X, E) with edges weighted by distance
3. define D as distance from graph
4. decompose matrix that represents the graph

MDS breakdown

1. estimate distances between some pairs of close points
2. make graph (X, E) with edges weighted by distance
3. define D as distance from graph
4. decompose matrix that represents the graph

Getting manifold learning algorithms from schematic

- Setting $E = \{((x_i, x_j), \|x_i - x_j\|^2) | x_i, x_j \in X\}$ (complete graph) we get classical MDS
- Modifying steps 1-3 will get us Isomap, Hessian Eigenmaps
- tSNE and UMAP will change step 4

Stochastic Neighbor Embedding

Idea: embed into lower-dimensional space, preserve distance statistics

$$q_{j|i} = \frac{f(\|y_i - y_j\|)}{\sum_{k \neq j} f(\|y_i - y_k\|)}, p_{j|i} = \frac{exp(-\|x_i - x_j\|^2 / s_i^2)}{\sum_{k \neq j} exp(-\|x_i - x_k\|^2 / s_i^2)}$$

$$KL(p, q) = \sum_{i,j} p_{j|i} log \frac{p_{j|i}}{q_{j|i}}$$

Notes

Choice of f determines algorithm:

- $f(x) = exp(-x^2)$ - original SNE
- $f(x) = (1 + x^2)^{-1}$ - tSNE (note this is density of Cauchy distribution up to a constant)

Stochastic Neighbor Embedding

Perplexity

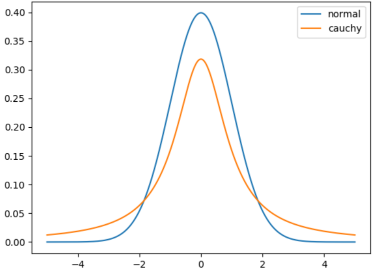
s gives rise to perplexity $P_i = 2^{H(p_i)}$ which controls effective neighborhood size at x_i.

High level algorithm

- for each x_i find s that matches given perplexity
- initialize y_i randomly
- minimize KL with gradient descent w.r.t. y_i

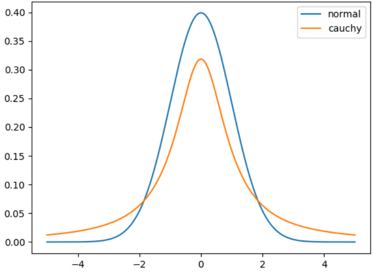
Problems with SNE: Crowding problem

- 'more room' for intermediate distances in higher dimensions



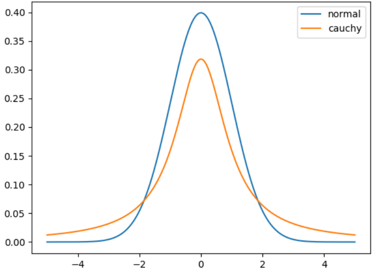
Problems with SNE: Crowding problem

- 'more room' for intermediate distances in higher dimensions
- pairs of points will tend to have similar distance (remember curse of dimensionality for kNN)



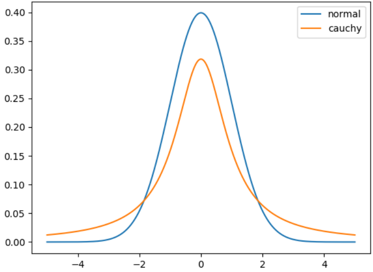
Problems with SNE: Crowding problem

- 'more room' for intermediate distances in higher dimensions
- pairs of points will tend to have similar distance (remember curse of dimensionality for kNN)
- harder to embed them faithfully into lower dimensional space



Problems with SNE: Crowding problem

- 'more room' for intermediate distances in higher dimensions
- pairs of points will tend to have similar distance (remember curse of dimensionality for kNN)
- harder to embed them faithfully into lower dimensional space



- somewhat fixed by using t Distribution which has longer tails (not so concentrated around maximum).
  This problem is not specific to tSNE - it relates to the fact that data may not be sampled uniformly from manifold

Recall UMAP means **Uniform** Manifold Approximation and Projection

Algorithm

1. get input data representation
2. initialize embedding
3. calculate probabilities of points being nearest neighbors
4. optimize embedding

Bird's view of UMAP's theory

Recall UMAP means **Uniform** Manifold Approximation and Projection

Algorithm

1. get input data representation
2. initialize embedding
3. calculate probabilities of points being nearest neighbors
4. optimize embedding

This also describes tSNE, what are the differences?

1. Fuzzy set built using local distance functions (Riemannian metric)
2. kNN graph's Laplacian
3. Probability between points corresponds to likelihood of points being connected in neighborhood graph
4. Cross entropy between fuzzy sets

UMAP vs tSNE in a nutshell

UMAP paper appendix C

	UMAP	tSNE
Initialization	Laplacian decomposition	Random Gaussian
Optimization	SGD + negative sampling	Gradient Descent
$p_{i\|j}$	$exp(-\frac{d(x_i,x_j)-\rho_i}{\sigma_i})$	$\frac{exp(-\|x_i-x_j\|^2/s_i^2)}{\sum_{k\neq j} exp(-\|x_i-x_k\|^2/s_i^2)}$
$q_{i\|j}$	$(1+a\|y_i-y_j\|^{2b}))^{-1}$	$\frac{(1+(\|y_i-y_j\|^2))^{-1}}{\sum_{k\neq j}(1+(\|y_i-y_k\|^2))^{-1}}$
Loss	H(q, p) = H(p) + KL(q, p)	KL(q, p)

Why such p, q?

- ρ_i - Riemannian structure
- σ_i - analogous to s_i
- q - approximation of membership for fuzzy simplicial complex

Riemannian structure

Remember we want to calculate distance **along the manifold**

Curve length in Euclidean space

$L_\gamma = \int_0^1 \|\gamma'(t)\| dt = \int_0^1 \sqrt{\langle \gamma'(t) | \gamma'(t) \rangle} dt$

Riemannian structure

Remember we want to calculate distance **along the manifold**

Curve length in Euclidean space

$L_\gamma = \int_0^1 \|\gamma'(t)\| dt = \int_0^1 \sqrt{\langle \gamma'(t) | \gamma'(t) \rangle} dt$

Definition

Riemannian space has local inner product.
Precisely, it has an inner product on each tangent space that varies smoothly.

Riemannian structure

Remember we want to calculate distance **along the manifold**

Curve length in Euclidean space

$L_\gamma = \int_0^1 \|\gamma'(t)\| dt = \int_0^1 \sqrt{\langle \gamma'(t) | \gamma'(t) \rangle} dt$

Definition

Riemannian space has local inner product.
Precisely, it has an inner product on each tangent space that varies smoothly.

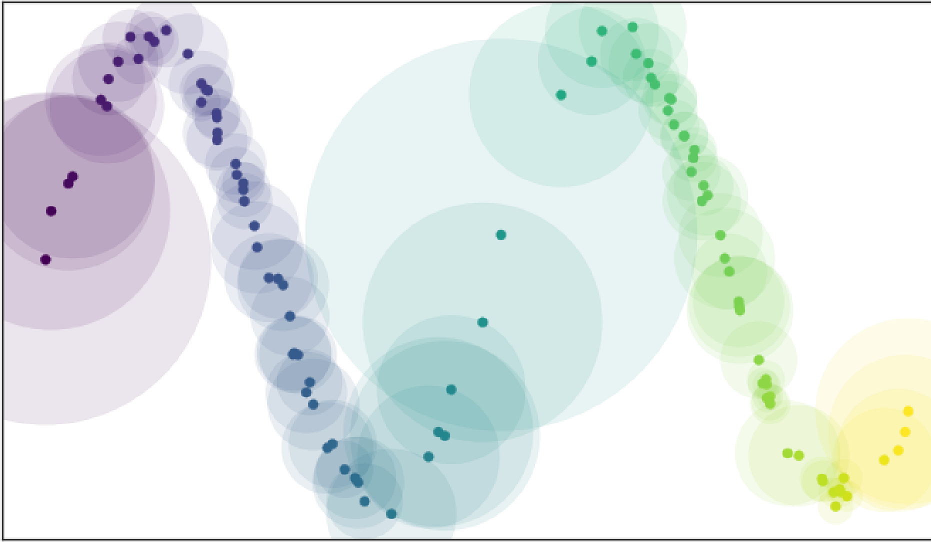
Idea

Estimate Riemannian structure locally with finite samples - set it constant on neighborhoods

Problem

We get incompatible local structures

Riemannian structure



Some Category Theory

Category theory is not a theory per se, it's rather a framework for thinking about mathematics

Definitions

A *category* $\mathscr{C}$ is a collection of objects with *morphisms* that can be composed and the composition satisfies some technical conditions

Think a set of some sets with functions that preserve structure

Definition

A morphism $f : A \to B$ is an *isomorphism* if there is a morphism $g : B \to A$ such that $f \circ g = id_A$ and $g \circ f = id_B$

Some Category Theory

Examples

- category *FinSet* of finite sets and functions between them
- category *Graphs* of graphs and graph homomorphisms
- category *Ab* of Abelian groups and group homomorphisms
- category of metric spaces and contractions

Definition

A mapping $F : \mathscr{C} \to \mathscr{D}$ is a *functor* if $F(f \circ g) = F(f) \circ F(g)$
Warning: technically this is a pair of mappings but we usually omit this

Examples

$F : \mathit{Graphs} \to \mathit{FinSet}\ \ F((V, E)) = V$

Topological preliminaries

Extended pseudometric space

A space is called *pseudometric* if it suffices metric space axioms, but without requiring that $d(x,y) = 0 \implies x = y$
$d : X \times X \to \mathbb{R}^+$

- $d(x,y) = d(y,x)$
- $d(x,y) + d(y,z) \geq d(x,z)$

In *extended pseudometric space* it also can happen that $d(x,y) = \infty$

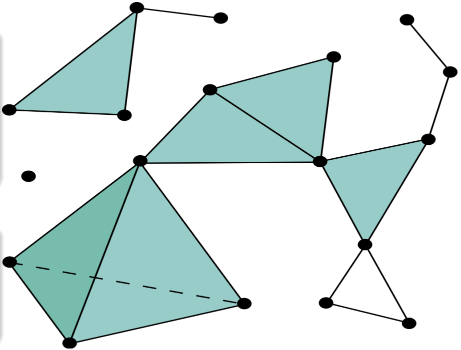
Topological preliminaries

Simplex

A d-dimensional simplex is a set $S \subset \mathbb{R}^n$ such that there are d linearly independent points $S = conv(d)$

Simplicial complex

A set C of simplexes such that if $s, s' \in C \implies s \cap s' \in C$



Topology

Definition

Definition: *nerve* of a family of sets $\mathscr{U} = \{U_i | i \in I\}$ is the set of finite subsets s of I for which $\bigcap_{i \in s} U_i \neq \emptyset$

Definition

$\mathscr{U}$ is a *cover* of X if $\bigcup_i U_i = X$

Nerve theorem

If set $\mathscr{U}$ is an open cover of X then $X \sim N(\mathscr{U})$ (they are homotopy equivalent, this is one notion of equivalence from topology)

Fuzzy sets

Definitions

For fixed set X, $S \subset X$, $m : S \to [0,1]$ (fuzzy membership function)
A pair (S, m) is a *fuzzy set* if $0 \leq m(x) \leq 1$
Fuzzy sets have natural generalizations of set operations:
if $(S, m_S), (R, m_R)$ are fuzzy sets then
$(S \cup R, m_{S \cup R})$ is a fuzzy set
where
$m_{S \cup R}(x) = max(m_S(x), m_R(x))$

Fuzzy sets

Definitions

For fixed set X, $S \subset X$, $m : S \to [0,1]$ (fuzzy membership function)
A pair (S, m) is a *fuzzy set* if $0 \leq m(x) \leq 1$
Fuzzy sets have natural generalizations of set operations:
if $(S, m_S), (R, m_R)$ are fuzzy sets then
$(S \cup R, m_{S \cup R})$ is a fuzzy set
where
$m_{S \cup R}(x) = max(m_S(x), m_R(x))$

Example

$S_1 = ([0,1], m),\ m(x) = [x \in S]x$

Fuzzy sets

Definitions

For fixed set X, $S \subset X$, $m : S \to [0,1]$ (fuzzy membership function)
A pair (S, m) is a *fuzzy set* if $0 \le m(x) \le 1$
Fuzzy sets have natural generalizations of set operations:
if $(S, m_S), (R, m_R)$ are fuzzy sets then
$(S \cup R, m_{S \cup R})$ is a fuzzy set
where
$m_{S \cup R}(x) = max(m_S(x), m_R(x))$

Example

$S_1 = ([0,1], m),\ m(x) = [x \in S]x$

Category of fuzzy sets

- objects: fuzzy sets
- morphisms: functions $f : S \to R$ such that $m_S(x) \le m_R(f(x))$

Fuzzy simplicial sets are generalization of simplicial complexes. This category is denoted by **Fin-sFuzz**.
Category of finite extended pseudometric spaces is denoted by **FinEPMet**

Metric realization of fuzzy simplicial sets

There exist functors
$\mathit{Real} : \mathbf{Fin\text{-}sFuzz} \to \mathbf{FinEPMet}$
$\mathit{Sing} : \mathbf{FinEPMet} \to \mathbf{Fin\text{-}sFuzz}$
That are *adjoint*.
Adjoint pairs of functions estabilish a weak equivalence relation of categories.

Global structure from local structures

Theory

- Construct extended pseudometric spaces locally
- Get fuzzy sets from pseudometric spaces
- Merge local fuzzy sets

Global structure from local structures

Dataset defines extended pseudometrics

$X = \{x_i\}_{i<n}$

ρ_i - distance from x_i to its closest neighbor

$$d_i(x_j, x_k) = \begin{cases} d(x_j, x_k) - \rho_i \text{ if } i = j \text{ or } i = k \\ \infty \text{ otherwise} \end{cases}$$

Global structure from local structures

Dataset defines extended pseudometrics

$X = \{x_i\}_{i<n}$

ρ_i - distance from x_i to its closest neighbor

$$d_i(x_j, x_k) = \begin{cases} d(x_j, x_k) - \rho_i \text{ if } i = j \text{ or } i = k \\ \infty \text{ otherwise} \end{cases}$$

Extended pseudometric spaces → fuzzy simplicial sets

$(X, d_i) \mapsto Sing((X, d_i))$

Global structure from local structures

Dataset defines extended pseudometrics

$X = \{x_i\}_{i<n}$

ρ_i - distance from x_i to its closest neighbor

$$d_i(x_j, x_k) = \begin{cases} d(x_j, x_k) - \rho_i \text{ if } i = j \text{ or } i = k \\ \infty \text{ otherwise} \end{cases}$$

Extended pseudometric spaces → fuzzy simplicial sets

$(X, d_i) \mapsto Sing((X, d_i))$

Local fuzzy sets → global fuzzy set

$\{(X, d_i)\}_{i<n} \mapsto \bigcup_{i<n} Sing((X, d_i))$

UMAP(X, *n-neighbors*, *dim*, *min-dist*, *n-epochs*)

- **forall** $x \in X$: fs_x = LocalFuzzySet(x, *n-neighbors*)
- *top-rep* $= \bigcup_{x \in X} \mathrm{fs}_x$
- Y := SpectralEmbedding(*top-rep*, *dim*)
- Y := OptimizedEmbedding(*top-rep*, *dim*, *min-dist*, *n-epochs*)

Algorithm - details

Simplification

We use only 1-skeleton, weighted neighborhood graph.

⚠ Approximation ⚠

To use gradient descent we need to take a differentiable approximation of fuzzy set membership.

The authors propose $q_{i|j} = (1 + a\|y_i - y_j\|^{2b}))^{-1}$

Probably the most important difference between tSNE (see tSNE paper 6.2)

a, b are set to predefined values or estimated using least squares to approximate

$$\phi(i,j) = \begin{cases} 1 \text{ if } \|y_i - y_j\|_2 \leq \text{min-dist} \\ exp(-\|y_i - y_j\|_2 + \text{min-dist})\text{otherwise} \end{cases}$$

Optimization

Optimize $C(p,q) = H(p) + KL(p,q)$

For p, q use symmetrized $p_{i,j} = p_{i|j} + p_{j|i} - p_{i|j}p_{j|i}$

Computational shortcut

- calculate gradient of error for similar points
- approximate gradient for dissimilar points by negative sampling

Implementation details - Nearest neighbors

- Naive implementation - $O(n^2)$ operations
- UMAP implementation uses approximate kNN
- Nearest Neighbor Descent algorithm is reported to have $O(n^{1.14})$ complexity

Python implementations

tSNE and other algorithms

- scikit-learn implements tSNE, MDS, Isomap, Hessian Eigenmaps, Locally Linear Embedding
- faster implementations available in `megaman` package

UMAP

- original author's package (`pip install umap-learn`)
- GPU accelerated NVidia `rapids (cuML)`

Further reading

- Understanding UMAP - interactive visualizations
- Exploratory Analysis of Interesting Datasets - UMAP's documentation
- UMAP author's google scholar page
- How Exactly UMAP Works - author's other articles mention applications in life sciences
- The Category Theory Behind UMAP
- UMAP - Mathematics and implementational details