Uniform Manifold Approximation and Projection

Peter Juhasz

April 22, 2024

Information

Contact

- name: Peter Juhasz
- email: peter.juhasz@math.au.dk

Agenda

1. Principal Component Analysis: **April 8**
2. t-Distributed Stochastic Neighbor Embedding: **April 15**
3. **Uniform Manifold Approximation & Projection**: **April 22**

Outline

1. Theoretical Background
   - Topology, Manifolds
   - Manifold Approximation
   - Projection
2. Remarks
   - Extensions & Limitations
   - Quiz
3. Examples
   - Interactive Parameter Tuning
   - Scripts

Introduction

Curse of Dimensionality

- increasing dimensions
- exponential growth of data space
- sparse data

Limitations of t-SNE

- time complexity: $O(n^2)$
- global data structure is not captured

Goal

- preserve nonlinear relationships
- preserve global and local information
- higher flexibility
- better scalabity
- robustness to noise

Main Idea

Goal

- embed data points in low-dimensional space
- preserve local and global data structure
- similar data points in high-dimensional space remain close to each other
- distance of clusters of points should be preserved

Main Steps

- assume that the data is uniformly distributed on a high-dimensional manifold
- learn the manifold using Riemannian metrics
- embed the points in a low-dimensional Euclidean space

Table of Contents

1. Theoretical Background
   - Topology, Manifolds
   - Manifold Approximation
   - Projection
2. Remarks
   - Extensions & Limitations
   - Quiz
3. Examples
   - Interactive Parameter Tuning
   - Scripts

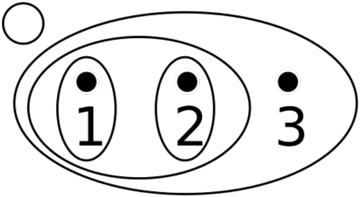
Topological Space

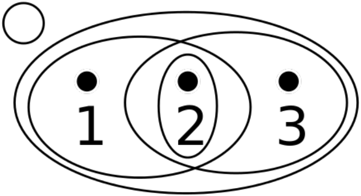
Topological Space

- $(X, \tau) : \tau \subseteq \mathcal{P}(X)$
- $\emptyset \in \tau$, $X \in \tau$
- $U_\alpha \in \tau \implies \bigcup_{\alpha \in I} U_\alpha \in \tau$
- $U_i \in \tau \implies \bigcap_{i=1}^{n} U_i \in \tau$

Examples

- trivial topology
- discrete space
- Euclidean space
- simplicial complex

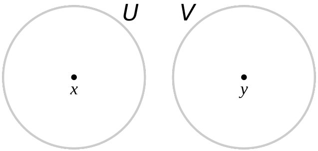


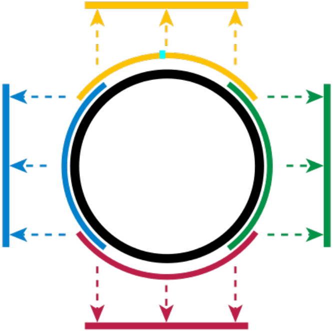


Manifold

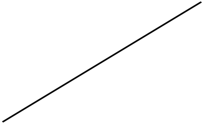
Manifold

- topological space
- second countable
- Hausdorff
- locally homeomorphic to $\mathbb{R}^n$



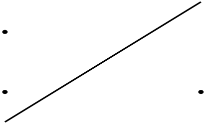


Is this a Manifold?

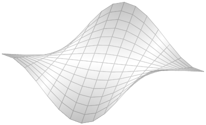


(a) **yes**

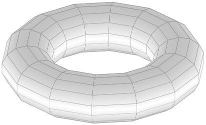
(b) **yes**



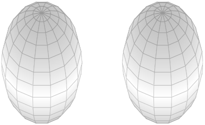
(c) **no**



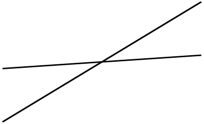
(d) **yes**



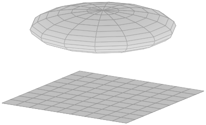
(e) **yes**



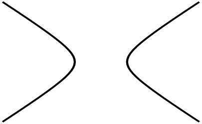
(f) **yes**



(g) **no**



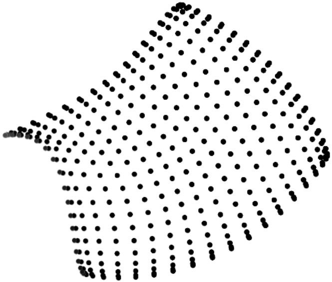
(h) **yes**

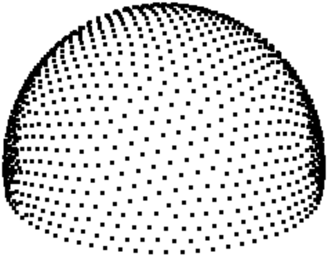


(i) **yes**

UMAP Assumption

Assumption: data is uniformly distributed on a manifold





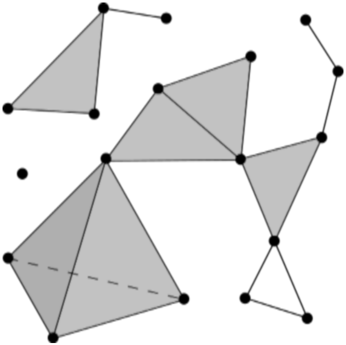
Given the data, how to approximate the manifold?

Simplicial Complex

- simplicial complex: discrete topological space
- idea: approximate the manifold with a simplicial complex

Simplicial Complex

- (V, κ)
- $V \neq \emptyset$, $|V| < \infty$
- $\kappa \subseteq \mathcal{P}(V)$
- $v \in V \implies \{v\} \in \kappa$
- $\tau \in \kappa$, $\sigma \subset \tau \implies \sigma \in \kappa$

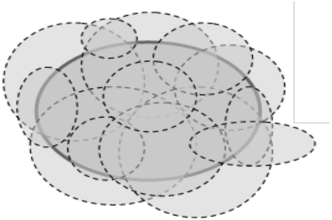


Nerve

How to create a simplicial complex from a manifold?

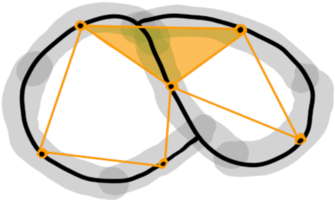
Cover

- $C = \{U_\alpha \subseteq X : \alpha \in A\}$
- $X = \bigcup_{\alpha \in A} U_\alpha$



Nerve

- $\{U_\alpha \subseteq X : \alpha \in A\}$ open cover of X
- $N(U_\alpha)$: simplicial complex
- i-simplices: $\sigma \subseteq A$
- $\mathrm{Supp}(\sigma) := \bigcap_{\alpha \in \sigma} U_\alpha \neq \emptyset$



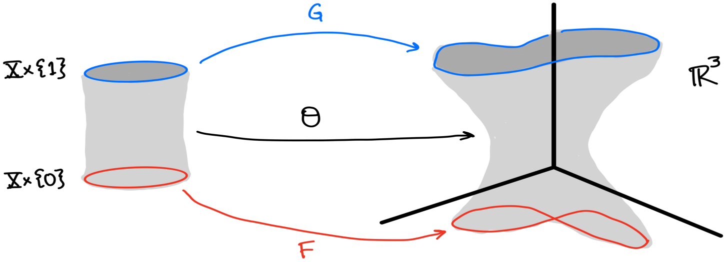
Homotopy Equivalence

Homotopy

- X, Y topological spaces
- $f, g \in C^0 : X \to Y$
- $\Theta \in C^0 : X \times [0,1] \to Y$
- $\Theta(x,0) = f(x)$
  $\Theta(x,1) = g(x)$

Homotopy Equivalence

- X, Y topological spaces
- $f \in C^0 : X \to Y$;
  $g \in C^0 : Y \to X$
- $g \circ f$ homotopic to id_X;
  $f \circ g$ homotopic to id_Y

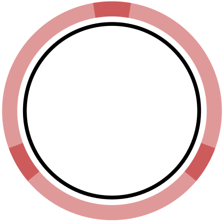


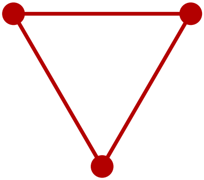
Nerve Theorem

Nerve Theorem

- X topological space
- $\{U_\alpha \subseteq X : \alpha \in A\}$ open cover
- $\sigma \in N(U_\alpha) \implies \mathrm{Supp}(\sigma)$ homotopy equivalent to a point

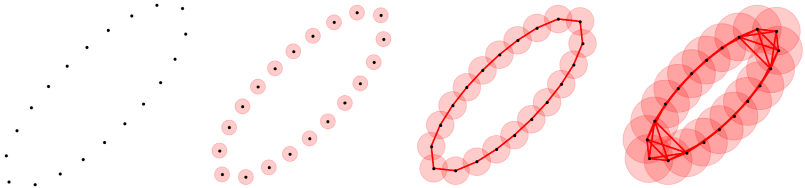
$\implies$ $|N(U_\alpha)|$ homotopy equivalent to X





Back to the Data

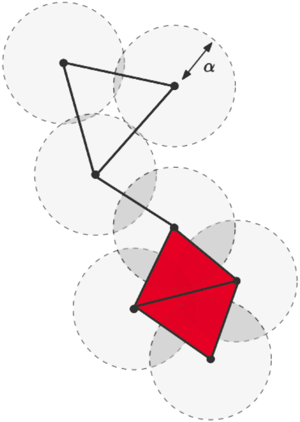
- goal: build a simplicial complex representing the manifold
- idea: cover the manifold with ε-balls
  $B_\varepsilon(p) = \{q \in M : d(p,q) < \varepsilon\}$
- two options: Čech complex, Vietoris–Rips complex



Čech Complex

Čech Complex

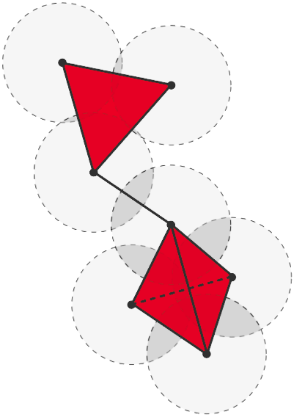
- simplices: set of points such that the covering ε-balls have a nonempty intersection
- $\sigma = \{p_i \in M : \bigcap_i B_\varepsilon(p_i) \neq \emptyset\}$



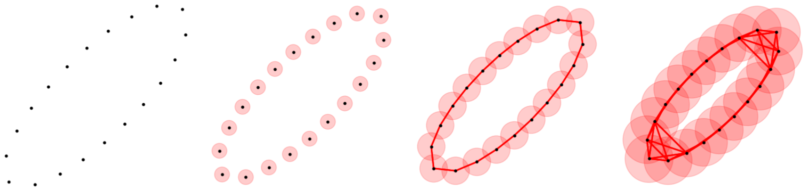
Vietoris–Rips Complex

Vietoris–Rips Complex

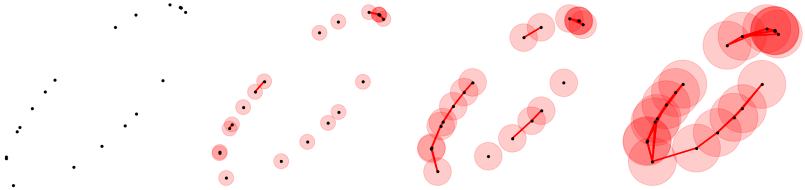
- simplices: set of points such that all pairs are within 2ε distance of each other
- $\sigma = \{p_i, p_j \in M : p_j \in B_{2\varepsilon}(p_i)\}$



Uneven Data Distribution



fine if data is uniformly distributed, but in reality:



Idea: find metric such that the data is uniform

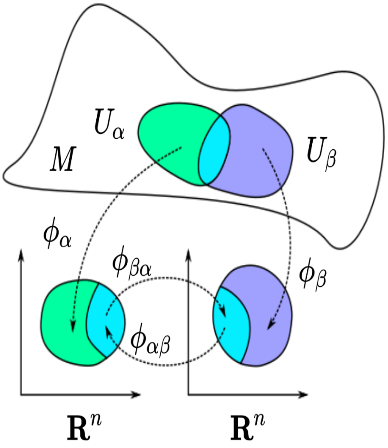
Differentiable Manifolds

Chart

- (U, ϕ); $U \subseteq M$ open
- $\phi : U \to \mathbb{R}^n$
- ϕ homeomorphism

Differentiable Manifold

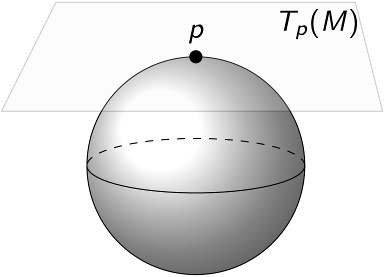
- domain of charts can overlap
- transition functions: maps between overlapping charts
- transition functions must be differentiable



Riemannian Metric

Tangent Space

- $\gamma(t) \in C^0 \colon \mathbb{R} \to M$
- $p \in \gamma(t)$
- tangent vector: $v_p := \dot{\gamma}(p)$
- $T_p M = \mathrm{Span}\left(\{\text{tangent vectors}\}\right)$



Riemannian Metric

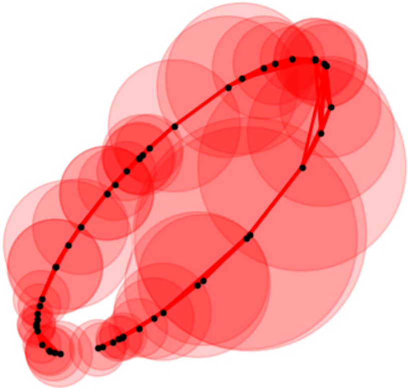
- find a basis for each tangent space
- assign inner product to each tangent space

Theorem

- every differentialble manifold admits a Riemannian metric

Local Notion of Distance

- local notion of distance for each point
- in local metric, unit balls contain k nearest neighbors
- choose a number of neighbors instead of the distance
- k small: local metric, higher variance
  k large: global metric, higher bias



High-Dimensional Distance Metrics

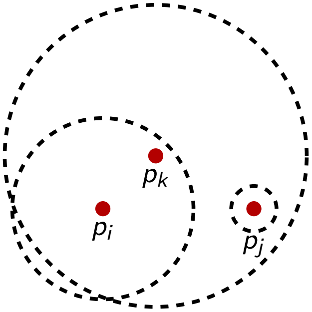
- not only the Euclidean distance can be used (and scaled)
- we can choose different metrics as well

Some Metrics

- Euclidean metric: $d(p_i, p_j) = \sqrt{\sum_{k=1}^{m}(p_{ik} - p_{jk})^2}$
- Chebyshev metric: $d(p_i, p_j) = \max_k |p_{ik} - p_{jk}|$
- Minkowski metric: $d(p_i, p_j) = \left(\sum_{k=1}^{m} |p_{ik} - p_{jk}|^r\right)^{1/r}$
- cosine metric: $d(p_i, p_j) = 1 - \frac{p_i \cdot p_j}{||p_i||_2 \, ||p_j||_2}$
- Mahalanobis metric: $d(p_i, p_j) = \sqrt{(p_i - p_j)^T M (p_i - p_j)}$

Incompatible Local Metrics

Incompatible local metrics



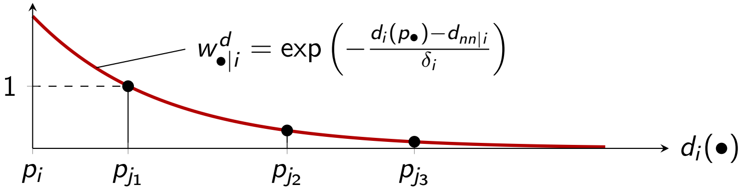
Which edges should be included?

Solution: fuzzy simplices

- based on the local metric at point p_i, assign a fuzzy value $w^d_{\sigma|i}$ to the edges σ
- create fuzzy edges from each point
- take the fuzzy union of all edges (simplicial complexes)

Exponential Kernel

Fuzzy values are determined by the exponential kernel



Local Metric

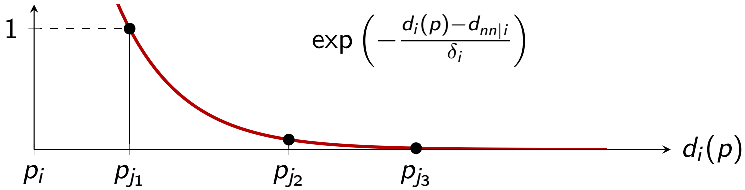
- $d_i(\bullet)$: distance in local metric
- unit ball radius: kernel shifted by distance to nearest neighbor
- local connectedness assumption: no isolated points (nearest neighbor has fuzzy value 1)

Bandwidth

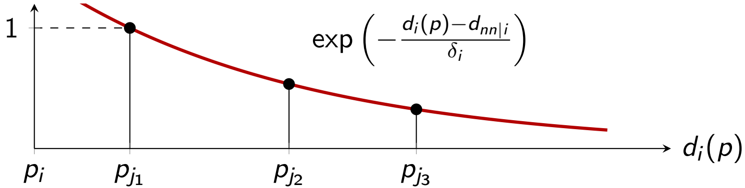
- bandwidth δ_i depends on the point
- higher δ_i: points further away contribute more

Effect of Bandwidth

- lower δ_i: further points have lower fuzzy value



- higher δ_i: further points have higher fuzzy value



Number of Neighbors

- bandwidth is adapted to the density: δ_i is smaller in denser parts of the data space
- δ_i determines the number of neighbors $N_n(p_i)$ of point p_i in the local metric

$$\log_2(N_n(p_i)) := \sum_j w_{j|i}^d$$

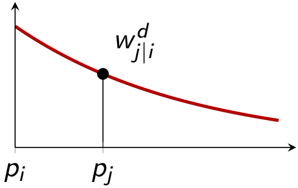
- δ_i is tuned $N_n(p_i)$ matches a predefined value N_n
- fuzzy value of nearest neighbors is always 1
- algorithm for nearest neighbors: **Nearest Neighbor Descent**

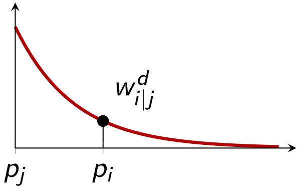
Fuzzy Union

Incompatible local metrics: asymmetrical fuzzy values
Fuzzy union: symmetrize fuzzy values

Example

- $w_{j|i}^d$, $w_{j|i}^d$: fuzzy values of p_j, p_i with respect to the local metric of p_i, p_j
- edges: combine local metrics by $w_{ij}^d := w_{j|i}^d + w_{i|j}^d - w_{j|i}^d \cdot w_{i|j}^d$
- w_{ij}^d: symmetrical; probability that the edge exists from at least in one of the points





Fuzzy Topology

- weight edges with a function of the length in local metric
- fuzzy value: certainty that a point is in a ball of a given radius
- union of fuzzy complexes: simplicial complex
- mathematical foundation: **UMAP Adjunction Theorem**

Exercise – Fuzzy Simplicial Complex

create a fuzzy simplicial complex using the Chebyshev metric

$$X = \begin{bmatrix} 1 & 1 \\ 1 & 1+\ln(2) \\ 1+\ln(4) & 1 \end{bmatrix} \quad \delta = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

1. distance matrix D_X
2. fuzzy values $w^d_{j|i}$ (exponential kernel)
3. fuzzy union w^d_{ij}

$$D_X = \begin{bmatrix} 0 & \ln(2) & \ln(4) \\ \ln(2) & 0 & \ln(4) \\ \ln(4) & \ln(4) & 0 \end{bmatrix} \qquad d_{nn} = \begin{bmatrix} \ln(2) \\ \ln(2) \\ \ln(4) \end{bmatrix}$$

$$w^d_{j|i} = \frac{1}{2} \begin{bmatrix} 0 & 2 & 1 \\ 2 & 0 & 1 \\ 2 & 2 & 0 \end{bmatrix} \qquad w^d_{ij} = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

Interesting: each edge surely exists. But why?

Projection

Goal: embed simplicial complex into a low-dimensional Euclidean space

Tasks	**Known**	**Question**
Approximation	positions	manifold, metric
Projection	manifold, metric	positions

Idea: initialize a fuzzy simplicial complex in the embedding space; minimize cross entropy

Initializing Embedding Positions

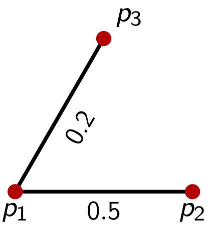
Initialization of Embeddings

- set the dimension of the embedding space
- consider only edges
- create a weighted graph of k nearest neighbors
- initialize the graph using spectral embedding

Spectral Embedding

- weight matrix of edges: $A_{ij} = w_{ij}^{d}$
- diagonal degree matrix: $D_{ii} = \sum_j A_{ij}$
- graph Laplacian: $L = D - A$
- calculate the eigenvalue decomposition of L: $L = U \Lambda U^T$
- consider the eigenvectors corresponding to the **smallest nonzero** eigenvalues

Exercise – Spectral Embedding



$$A = \frac{1}{10}\begin{bmatrix} 0 & 5 & 2 \\ 5 & 0 & 0 \\ 2 & 0 & 0 \end{bmatrix} \qquad D = \frac{1}{10}\begin{bmatrix} 7 & 0 & 0 \\ 0 & 5 & 0 \\ 0 & 0 & 2 \end{bmatrix}$$

$$L = \frac{1}{10}\begin{bmatrix} 7 & -5 & -2 \\ -5 & 5 & 0 \\ -2 & 0 & 2 \end{bmatrix}$$

$$\Lambda \approx \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0.26 & 0 \\ 0 & 0 & 1.14 \end{bmatrix} \qquad U \approx \begin{bmatrix} 1 & -0.32 & -4.68 \\ 1 & -0.68 & 3.68 \\ 1 & 1 & 1 \end{bmatrix}$$

Fuzzy Values in the Embedded Simplicial Complex

Embeddings

- low-dimensional embedding of p_i: q_i
- typically $q_i \in \mathbb{R}^2$ or $\mathbb{R}^3$

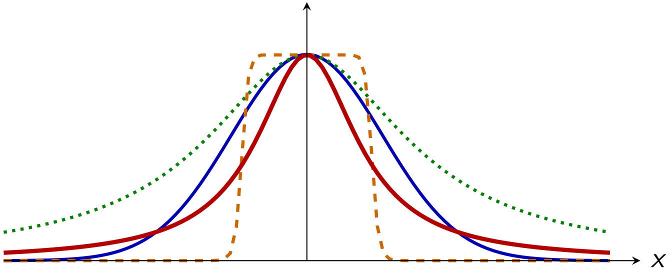
Fuzzy Values

- similarities of embeddings: based on t-distribution

$$w_{ij}^e := w^e(q_i, q_j) := \frac{1}{1 + \alpha||q_j - q_i||_2^{2\beta}} \quad (i \neq j) \qquad w_{ii} := 0$$

- α: lower values increase the spread of embeddings
- β: higher values increase the minimum distance between embeddings

Effect of Parameters



- fuzzy values as a function distance has fat tails
- fuzzy values are higher further away
- embeddings spread out

- Gaussian curve
- base case
- decreased α
- increased β

Objective

Objective

- goal: learn positions of embeddings q_i
- fuzzy values w^e of embeddings q_i should reflect fuzzy values w^d of the data p_i
- minimize "distance" between w^e and w^d

Idea

- consider the cross entropy $H(w^e, w^d)$
- minimize $H(w^e, w^d)$ by adjusting the embeddings

Cross Entropy

Definition

- cross entropy
- measure of dissimilarity between distributions
- expectation of logarithmic probabilities of other distribution:

$$H(P, Q) = \mathbb{E}_P\left[\ln(1/Q)\right] = -\sum_{x \in X} P(x)\ln(Q(x))$$

Relationships

- Kullback-Leibler Divergence: $D_{KL}(P||Q)$
- cross entropy: $H(P, Q) = H(P) + D_{KL}(P||Q)$

Properties

- $H(P, Q) \geq 0; \quad H(P, Q) = 0 \iff P = Q$
- $H(P, Q) \neq H(Q, P)$

Cross Entropy in Our Case

- fuzzy simplicial complex: each edge (simplex) σ is assigned a weight
- Bernoulli distribution: σ exists with probability w_σ
  - w_σ^d in the simplicial complex for the data
  - w_σ^e in the simplicial complex for the embedding

$$H(w^d, w^e) = \sum_{i \neq j} \left(\underbrace{w_{ij}^d \ln \left(\frac{w_{ij}^d}{w_{ij}^e} \right)}_{\substack{\text{term for } i \leftrightarrow j \text{ exists} \\ \text{attractive force}}} + \underbrace{\left(1 - w_{ij}^d\right) \ln \left(\frac{1 - w_{ij}^d}{1 - w_{ij}^e} \right)}_{\substack{\text{term for } i \leftrightarrow j \text{ does not exist} \\ \text{repulsive force}}} \right)$$

- **force-directed graph layout**: minimizing $H(w^d, w^e)$ by adjusting the embeddings

Stochastic Gradient Descent Optimization

Cross Entropy

- minimize $H(w^d, w^e)$
- stochastic gradient descent: iteratively update embeddings
- move similar (dissimilar) points closer together (further apart)

Gradient

- iteratively update embeddings with learning rate α:

$$q_i^{(t+i)} := q_i^{(t)} - \alpha \frac{\partial D_{KL}}{\partial q_i^{(t)}}$$

Simplified Algorithm

- choose an embedding q_i uniformly randomly
- attractive force: choose $q_{j,a}$ from its neighborhood (probability $\sim$ fuzzy value)
- repulsive force: choose $q_{j,r}$ uniformly randomly from points not in the neighborhood
- balance attractive and repulsive forces using cost function

Steps of UMAP

Data Points

- build data matrix
- calculate fuzzy values
- find δ_i for each point
- symmetrize fuzzy values

- X
- $w^d_{j|i} = \exp\left(-(d_{j|i} - d_{nn|i})/\delta_i\right)$
- $\log_2(N_n) = \sum_j w_{j|i}$
- $w^d_{ij} = w_{i|j} + w_{j|i} - w_{i|j} \cdot w_{j|i}$

Embeddings

- initialize embeddings
- calculate fuzzy values

- Y_{init}
- $w^e_{ij} \sim 1/(1 + \alpha ||y_j - y_i||_2^{2\beta})$

Cross Entropy

- consider cross entropy
- stochastic gradient descent

- $H = \sum w^d_{ij} \ln(w^d_{ij}/w^e_{ij})$
  $+(1 - w^d_{ij}) \ln((1 - w^d_{ij})/(1 - w^e_{ij}))$
- $y_i := y_i - \alpha \frac{\partial H}{\partial y_i}$

Main UMAP Parameters

Nearest Neighbors

- k: number of nearest neighbors
- adjusts the bandwidth
- k small: local metric
- k large: global metric

Minimum Distance

- adjusts how close embeddings can be
- low values: clumpier embeddings
- high values: embeddings spread out more

Number of Components

- dimension of embedding space
- 2 or 3: visualization
- > 3: density based clustering

Distance Metric

- metric for high-dimensional space

Table of Contents

1 Theoretical Background
- Topology, Manifolds
- Manifold Approximation
- Projection

2 Remarks
- Extensions & Limitations
- Quiz

3 Examples
- Interactive Parameter Tuning
- Scripts

Some Remarks

Supervised Learning

- create embeddings from training set, then embed new, unseen data points
- labels: separate metric space; use fuzzy intersection to combine complexes

Aligned UMAP

- it is possible to align two UMAP embeddings
- optimize both embeddings in parallel
- apply constraint to shared points

Combining UMAP Models

- if two UMAP models operate on the same data
- use fuzzy topology to combine fuzzy simplicial complexes

Non-Euclidean Embeddings

- it is possible to embed data in non-Euclidean spaces
- set the embedding space dimension
- use a different metric for the embedding space

Limitations

Nonuniform Data

- may not perform well on non-uniform density

Limited Interpretability

- low-dimensional embeddings are hard to interpret

Transformation Bias

- data might not lie on a low-dimensional manifold

Sensitivity

- sensitive to choice of hyperparameters
- interactive tuning is required
- wrong choice may lead to false findings

Quiz – t-SNE, UMAP, or Both?

Which one . . .

- is more scaleable?
- preserves more of the global structure?
- should we consider for larger data sets?
- interprets distances of clusters better?
- is sensitive to the choice of parameters?
- runs in a reproduceable manner?
- uses a force-directed graph layout?
- is more mathematically justified?
- is a nonlinear algorithm?

- UMAP
- UMAP
- UMAP
- UMAP
- both
- t-SNE
- UMAP
- UMAP
- both

Table of Contents

1 Theoretical Background
- Topology, Manifolds
- Manifold Approximation
- Projection

2 Remarks
- Extensions & Limitations
- Quiz

3 Examples
- Interactive Parameter Tuning
- Scripts

Interactive Examples

- Understanding UMAP
- Tensorflow Embedding Projector
- UMAP Explorer
- Visualizing UMAP

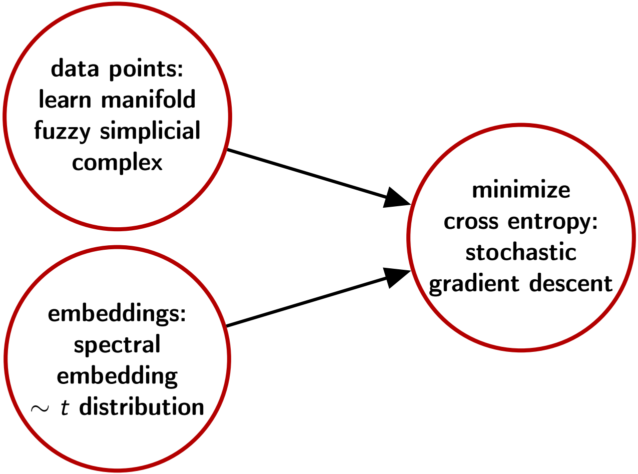
UMAP in Python & R

	Python	R
load library	`import umap.umap_ as umap`	`library(umap)`
load dataset	`data: npt.NDArray = ...`	`data <- ...`
create UMAP object	`model = umap.UMAP( n_neighbors=5, min_dist=0.3, ...)`	
fit model	`embedding = model.fit_transform(data)`	`umap(data)`

R Examples

R Examples

Summary


data points:
learn manifold
fuzzy simplicial
complex
embeddings:
spectral
embedding
$\sim t$ distribution
minimize
cross entropy:
stochastic
gradient descent

Q & A

Resources

- Etienne Becht, Leland McInnes, John Healy, Charles-Antoine Dutertre, Immanuel WH Kwok, Lai Guan Ng, Florent Ginhoux, and Evan W Newell, *Dimensionality reduction for visualizing single-cell data using umap*, Nature biotechnology **37** (2019), no. 1, 38–44.
- Wei Dong, Charikar Moses, and Kai Li, *Efficient k-nearest neighbor graph construction for generic similarity measures*, Proceedings of the 20th international conference on World wide web, 2011, pp. 577–586.
- Alex Diaz-Papkovich, Luke Anderson-Trocmé, and Simon Gravel, *A review of umap in population genetics*, Journal of Human Genetics **66** (2021), no. 1, 85–91.
- Dmitry Kobak and George C Linderman, *Initialization is critical for preserving global data structure in both t-sne and umap*, Nature biotechnology **39** (2021), no. 2, 156–157.
- Manifold, *Manifold — Wikipedia, the free encyclopedia*, 2024, [Online; accessed 15-April-2024].
- Vidit Nanda, *Computational algebraic topology*, Lecture Notes.