t-Distributed Stochastic Neighbor Embedding

Peter Juhasz

April 15, 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 and Projection: **April 22**

Outline

Introduction

Curse of Dimensionality

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

Limitations of Traditional Techniques

- only global structure is considered
- nonlinear relationships are not captured

Goal

- reduce number of features
- nonlinear dimensionality reduction
- preserve global and local information
- visualization, interpretation

Main Idea

Goal

- embed data points in low-dimensional space
- fine-grained relationships: preserve local structure
- similar data points in high-dimensional space remain close to each other with high probability

Main Steps

- construct probability distribution over pairs of high-dimensional points
- define similar probability distribution over pairs of low-dimensional points

Similarities

Euclidean distance

- data matrix: $X = \begin{bmatrix} \mathrm{x}_1 & \cdots & \mathrm{x}_n \end{bmatrix}^T$
- calculate Euclidean distance for each pair: $||\mathrm{x}_i - \mathrm{x}_j||$

Conditional Probabilities

- similarity of x_j to x_i = conditional probability $p_{j|i}$
- use Gaussian kernel to define probabilities $p_{j|i}$:

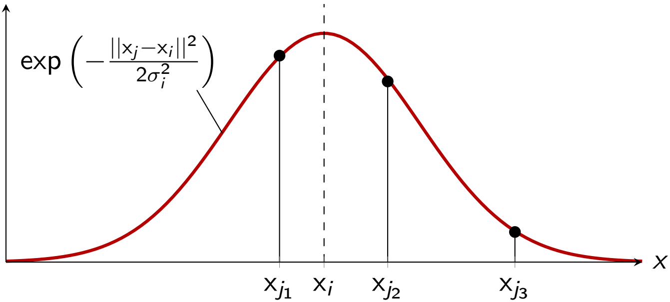
$$p_{j|i} \sim \exp\left(-\frac{||\mathrm{x}_j - \mathrm{x}_i||^2}{2\sigma_i^2}\right) \quad (i \neq j) \qquad P_{i|i} := 0$$

- $p_{j|i}$ = probability that x_i would pick x_j as its neighbor
- normalization: $p_{j|i}$ must be normalized for each data point i
- not symmetric: $p_{j|i} \neq p_{i|j}$

Symmetrization

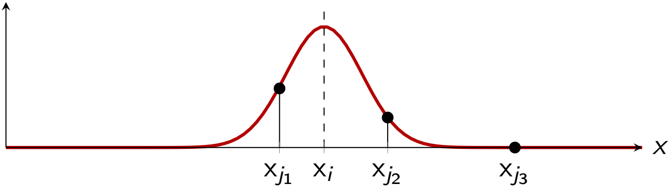
- symmetrize probabilities: $p_{ij} = (p_{j|i} + p_{i|j})/2N \qquad p_{ij} = p_{ji}$

Gaussian Kernel

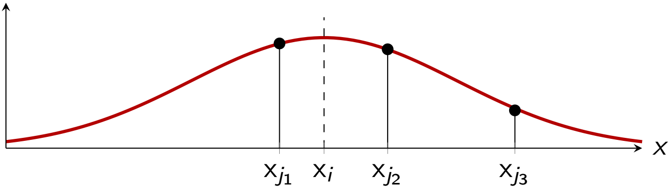


- σ_i depends on the point x_i
- higher σ_i: points further away contribute more
- lower σ_i: points further away contribute less

Effect of Bandwidth



- lower σ_i: points further away contribute less



- higher σ_i: points further away contribute more

Perplexity

- bandwidth is adapted to the density: σ_i is smaller in denser parts of the data space
- Shannon entropy of $p_{j|i}$

$$H_i := \mathbb{E}\underbrace{\left[\log_2\left(\frac{1}{p_{j|i}}\right)\right]}_{\text{surprise}} = -\sum_j p_{j|i}(\mathrm{x}_j)\log_2 p_{j|i}(\mathrm{x}_j)$$

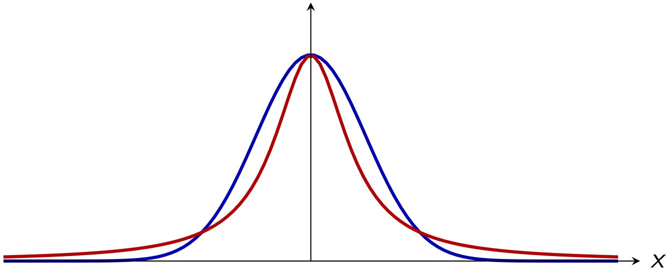
- Perplexity of x_i:

$$\mathrm{Perp}(\mathrm{x}_i) := 2^{H_i}$$

- σ_i is tuned so that perplexity matches a predefined value R
- bisection method: find σ_i with searching the root of $\mathrm{Perp}(\mathrm{x}_i) - R = 0$

Remark: Student t-distribution

- 1 degree of freedom: Cauchy distribution
- very fat tails



Low-Dimensional Embeddings

Embeddings

- low-dimensional embeddings of X: $Y := \begin{bmatrix} \mathrm{y}_1 & \cdots & \mathrm{y}_n \end{bmatrix}^T$
- typically $Y \in \mathbb{R}^{N \times 2}$ or $\mathbb{R}^{N \times 3}$

Similarities

- similarities of embeddings: t-distribution

$$q_{ij} := Q\left(||\mathrm{y}_j - \mathrm{y}_i||\right) \sim \frac{1}{\pi} \frac{1}{1 + ||\mathrm{y}_j - \mathrm{y}_i||^2} \quad (i \neq j) \qquad Q_{ii} := 0$$

- q_{ij} must be normalized
- t-distribution (Cauchy distribution): heavy tails

Back to the Objective

Objective

- goal: learn embeddings Y
- embedding similarities q_{ij} reflect original similarities p_{ij}
- minimize "distance" between P and Q

Idea

- minimize the Kullback-Leibler divergence D_{KL} of P, Q
- heavy tails in $Q \implies$ embeddings of dissimilar points in X can be far apart in Y

Remark: Kullback-Leibler Divergence

Definition

- relative entropy = Kullback-Leibler divergence
- measure of dissimilarity between distributions
- expectation of (base 2 or base e) logarithmic difference

$$D_{KL}(P||Q) = -\sum_{x \in X} P(x) \ln\left(\frac{Q(x)}{P(x)}\right)$$

Properties

- $D_{KL} \geq 0$
- $D_{KL}(P||Q) = 0 \Longleftrightarrow P = Q$
- $D_{KL}(P||Q) \neq D_{KL}(Q||P)$
- absolute continuity: $Q(x) = 0 \Longrightarrow P(x) = 0 \ (\forall x \in X)$

Minimization of Kullback-Leibler Divergence

Kullback-Leibler Divergence

- aim: minimize $D_{KL}\left(P||Q\right)$ by adjusting Y

Gradient Descent

- initialize embeddings
  - random initialization
  - principal component analysis
- iteratively update embeddings with learning rate α:

$$\frac{\partial D_{KL}}{\partial \mathrm{y}_i} = 4 \sum_{i \neq j} \frac{(p_{ij} - q_{ij})(\mathrm{y}_i - \mathrm{y}_j)}{1 + ||\mathrm{y}_j - \mathrm{y}_i||^2} \qquad \mathrm{y}_i := \mathrm{y}_i - \alpha \frac{\partial D_{KL}}{\partial \mathrm{y}_i}$$

Steps of t-SNE

Data Point Similarities

- build data matrix — X
- calculate, normalize $p_{j|i}$ — $p_{j|i} \sim \exp\left(-||x_j - x_i||^2/(2\sigma_i^2)\right)$
- find σ_i for each point — $R = 2^{-\sum_j p_{j|i}(x_j)\log_2 p_{j|i}(x_j)}$
- symmetrize similarities — $p_{ij} = (p_{i|j} + p_{j|i})/(2N)$

Embedding Similarities

- initialize embeddings — Y_{init}
- calculate, normalize Q — $q_{ij} \sim 1/(1 + ||y_j - y_i||^2)$

Kullback–Leibler Divergence

- consider D_{KL} — $D_{KL}(P||Q) = \sum p_{ij} \ln\left(p_{ij}/q_{ij}\right)$
- calculate gradient — $\frac{\partial D_{KL}}{\partial y_i} = 4\sum_{i\neq j} \frac{(p_{ij}-q_{ij})(y_i-y_j)}{1+||y_j-y_i||^2}$
- update embeddings — $y_i := y_i - \alpha \frac{\partial D_{KL}}{\partial y_i}$

Exercise – One Iteration of t-SNE

$$X = \begin{bmatrix} 1 & 2 \\ 1 & 2+\sqrt{\ln(3)} \\ 1+\sqrt{\ln(2)} & 2 \end{bmatrix} \quad Y_{\text{init}} = \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} \quad \sigma = \frac{\sqrt{2}}{2}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad \alpha = 1$$

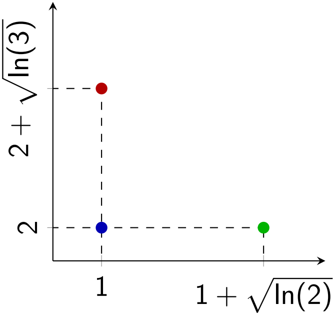
$$D_X = \begin{bmatrix} 0 & \sqrt{\ln(3)} & \sqrt{\ln(2)} \\ \sqrt{\ln(3)} & 0 & \sqrt{\ln(6)} \\ \sqrt{\ln(2)} & \sqrt{\ln(6)} & 0 \end{bmatrix} \qquad D_Y = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 0 & 1 \\ 2 & 1 & 0 \end{bmatrix}$$

$$p_{j|i} = \begin{bmatrix} 0 & 2/5 & 3/5 \\ 2/3 & 0 & 1/3 \\ 3/4 & 1/4 & 0 \end{bmatrix} \quad p_{ij} = \begin{bmatrix} 0 & 8/45 & 9/40 \\ 8/45 & 0 & 7/72 \\ 9/40 & 7/72 & 0 \end{bmatrix} \quad q_{ij} = \frac{1}{24}\begin{bmatrix} 0 & 5 & 2 \\ 5 & 0 & 5 \\ 2 & 5 & 0 \end{bmatrix}$$

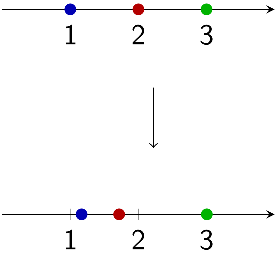
$$D_{KL}(P||Q) = \sum_{i \neq j} p_{ij} \ln\left(\frac{p_{ij}}{q_{ij}}\right) = 0.2424 \quad \frac{\partial D_{KL}}{\partial y_i} = \begin{bmatrix} 0.1656 \\ -0.2833 \\ 0.0044 \end{bmatrix} \quad Y_{\text{upd}} = \begin{bmatrix} 1.1656 \\ 1.7167 \\ 3.0044 \end{bmatrix}$$

Exercise – One Iteration of t-SNE

Data Points



Embeddings



Parameter Tuning

Interactive Examples

Limitations

Curse of Dimensionality

- Gaussian kernel uses Euclidean distance
- other distance metrics may be used (UMAP)

Sensitivity

- sensitive to parametrization
- sensitive to initialization of embeddings
- interactive parameter tuning required
- non-deterministic results

Complexity

- pairwise similarities: computationally expensive
- time complexity: $O(n^2)$
  space complexity: $O(n^2)$

False Findings

- finds clusters in nonclustered data
- hard to interpret results

Optimizations, Variants

Barnes-Hut Approximation

- approximate long-range similarities
- replace group of distant points with center of mass
- reduced time complexity: $O(n^2) \to O(n \log n)$

Momentum Gradient Descent

- note momentum (previous step directions)
- update = weighted sum of current and previous gradients

Early Exaggeration

- goal: avoid local minima
- increase p_{ij} for the first few iterations
- points close to each other move together

Similarity Cutoff

- neglect similarities if $||x_j - x_i|| > 3\sigma_i$

Quiz — True or False?

- The data matrix X must be normalized
- Similarities $p_{j|i}$ of X — when normalized — are characterized by a Gaussian density function
- Increasing perplexity leads to preserving more the local structure, leading to higher variance, lower bias
- Outliers are assigned to the nearest cluster
- t-SNE would work if the similarities of the embeddings Q were Gaussian
- Distribution of tossing a coin has a higher Shannon–entropy than rolling a die

- **False**
- **False**
- **False**
- **False**
- **True**
- **False**

Steps of t-SNE in R

- Load library
- Load dataset
- Remove duplicates
- Perform t-SNE
- Interpret / visualize results

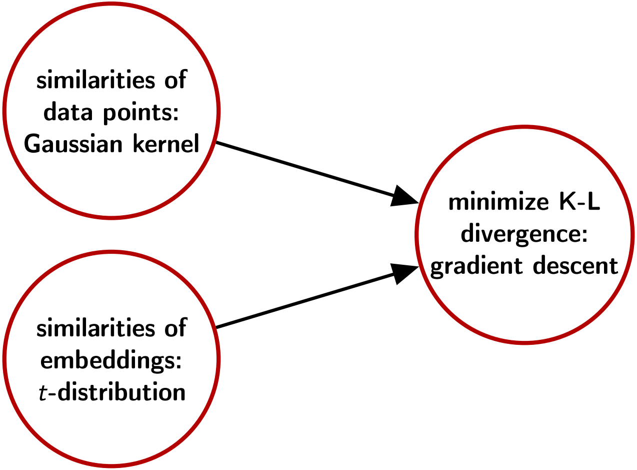
- library(Rtsne)
- data(iris) or read.csv()
- uniq <- unique(iris)
- tsne <- Rtsne(uniq[-5])
- plot <- data.frame(...)
  ggplot2::ggplot(...)

R Examples

R Examples

Summary

- goal: reduce dimensionality + preserve local structures



Q & A

Resources

- Geoffrey E Hinton and Sam Roweis, *Stochastic neighbor embedding*, Advances in neural information processing systems **15** (2002).
- 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.
- George C Linderman and Stefan Steinerberger, *Clustering with t-sne, provably*, SIAM journal on mathematics of data science **1** (2019), no. 2, 313–332.
- Laurens van der Maaten and Geoffrey Hinton, *Visualizing data using t-sne*, Journal of machine learning research **9** (2008), no. Nov, 2579–2605.