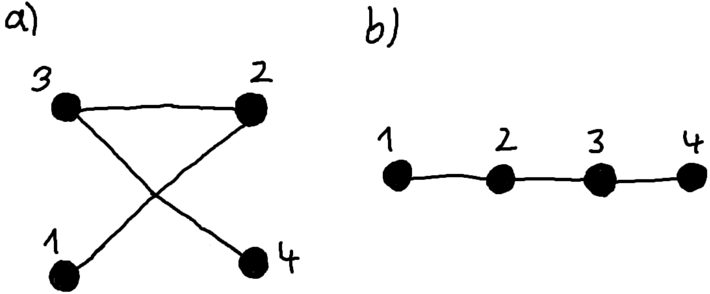
Graph Layouts and Contrastive Learning using Neighbour Embedding Algorithms

Hertie Institute for Artificial Intelligence in Brain Health, Berens Lab
supervised by Dr. Dmitry Kobak

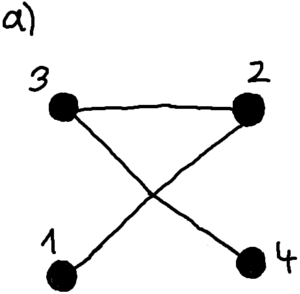
presented by Alica Guzmán

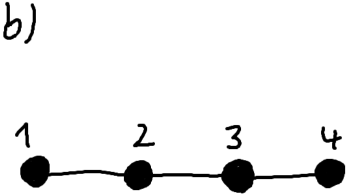
23-02-2023

2D viszualizations of a graph



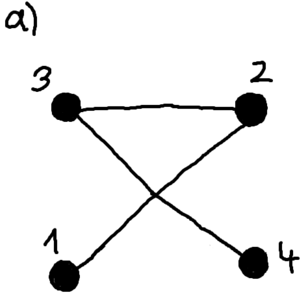
2D viszualizations of a graph

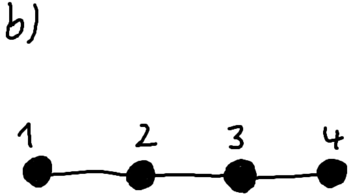




$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

2D viszualizations of a graph



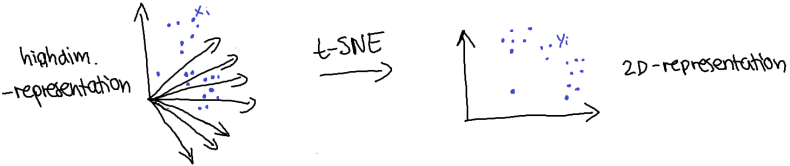


$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

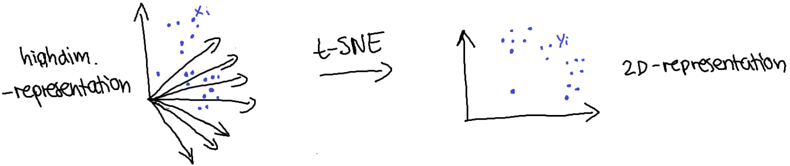
low-dimensional coordinates $\mathbf{y}_i$ with $i = 1, \ldots, 4$ (2-dim.)

Introduction to t-SNE

t-SNE



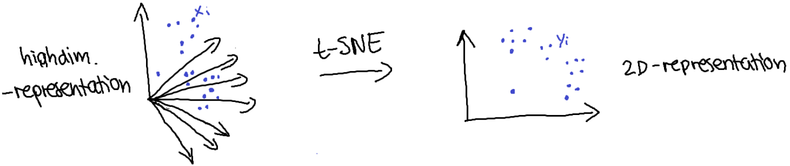
t-SNE



$p_{ij}(||x_i - x_j||, \sigma_i) = \frac{p_{i|j}+p_{j|i}}{2N}$

$p_{j|i} \sim \exp\left(\frac{-||x_i - x_j||^2}{2\sigma_i^2}\right)$

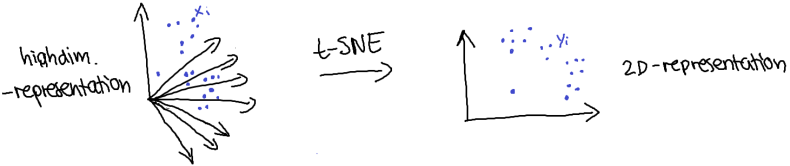
t-SNE



$$p_{ij}(||x_i - x_j||, \sigma_i) = \frac{p_{i|j} + p_{j|i}}{2N}$$

$$p_{j|i} \sim \exp\left(\frac{-||x_i - x_j||^2}{2\sigma_i^2}\right) \text{ approx. } \begin{cases} \frac{1}{k} & \text{if j kNN of i} \\ 0 & \text{otherwise.} \end{cases}$$

t-SNE

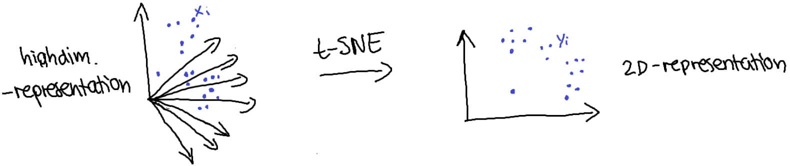


$$p_{ij}(||x_i - x_j||, \sigma_i) = \frac{p_{i|j} + p_{j|i}}{2N}$$

$$p_{j|i} \sim \exp\left(\frac{-||x_i - x_j||^2}{2\sigma_i^2}\right) \text{ approx. } \begin{cases} \frac{1}{k} & \text{if j kNN of i} \\ 0 & \text{otherwise.} \end{cases}$$

$$q_{ij}(||y_i - y_j||) \sim \frac{1}{1 + ||y_i - y_j||^2} = w_{ij}$$

t-SNE



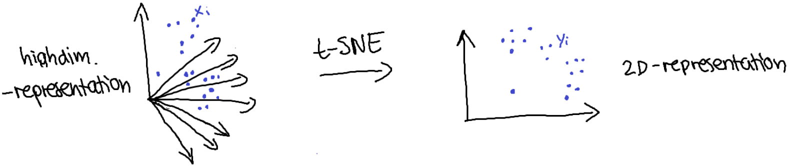
$p_{ij}(||x_i - x_j||, \sigma_i) = \frac{p_{i|j}+p_{j|i}}{2N}$

$p_{j|i} \sim \exp\left(\frac{-||x_i - x_j||^2}{2\sigma_i^2}\right)$ approx. $\begin{cases} \frac{1}{k} & \text{if j kNN of i} \\ 0 & \text{otherwise.} \end{cases}$

$q_{ij}(||y_i - y_j||) \sim \frac{1}{1+||y_i-y_j||^2} = w_{ij}$

$$C_{KL} = \sum_i \sum_j p_{ij} \log\left(\frac{p_{ij}}{q_{ij}}\right).$$

t-SNE



$p_{ij}(||x_i - x_j||, \sigma_i) = \frac{p_{i|j}+p_{j|i}}{2N}$

$q_{ij}(||y_i - y_j||) \sim \frac{1}{1+||y_i-y_j||^2} = w_{ij}$

$p_{j|i} \sim \exp\left(\frac{-||x_i-x_j||^2}{2\sigma_i^2}\right)$ approx. $\begin{cases} \frac{1}{k} & \text{if j kNN of i} \\ 0 & \text{otherwise.} \end{cases}$

$$C_{KL} = \sum_i \sum_j p_{ij} \log\left(\frac{p_{ij}}{q_{ij}}\right).$$

$$\frac{\partial C_{KL}}{\partial y_i} \sim \underbrace{\sum_j p_{ij} N w_{ij}(y_i - y_j)}_{\text{attractive forces}} - \underbrace{\frac{N}{\sum_{k\neq l} w_{kl}} \sum_j w_{ij}^2 (y_i - y_j)}_{\text{repulsive forces}}.$$

(Maaten and Hinton 2008, Böhm, Berens and Kobak 2022)

Graph t-SNE

We do not have x_i, $i = 1, \ldots, N$.

Graph t-SNE

We do not have x_{i}, $i = 1, \ldots, N$.
$\rightarrow$ Extract the high-dimensional simmilarities/affinities p_{ij} from the adjacency $\mathbf{A}$:

Graph t-SNE

We do not have x_{i}, $i = 1, \ldots, N$.
$\rightarrow$ Extract the high-dimensional simmilarities/affinities p_{ij} from the adjacency $\mathbf{A}$:

- $\mathbf{A}$ is a matrix containing all the graph neighbours

Graph t-SNE

We do not have x_{i}, $i = 1, \ldots, N$.
$\rightarrow$ Extract the high-dimensional simmilarities/affinities p_{ij} from the adjacency $\mathbf{A}$:

- $\mathbf{A}$ is a matrix containing all the graph neighbours
- obtain $p_{\mathrm{j|i}}$ by row normalization

Graph t-SNE

We do not have x_i, $i = 1, \ldots, N$.
$\rightarrow$ Extract the high-dimensional simmilarities/affinities p_{ij} from the adjacency $\mathbf{A}$:

- $\mathbf{A}$ is a matrix containing all the graph neighbours
- obtain $p_{j|i}$ by row normalization
- symmetrize and normalize to obtain p_{ij}

Graph t-SNE

We do not have x_i, $i = 1, \ldots, N$.
$\rightarrow$ Extract the high-dimensional simmilarities/affinities p_{ij} from the adjacency $\mathbf{A}$:

- $\mathbf{A}$ is a matrix containing all the graph neighbours
- obtain $p_{j|i}$ by row normalization
- symmetrize and normalize to obtain p_{ij}

Same procedure...

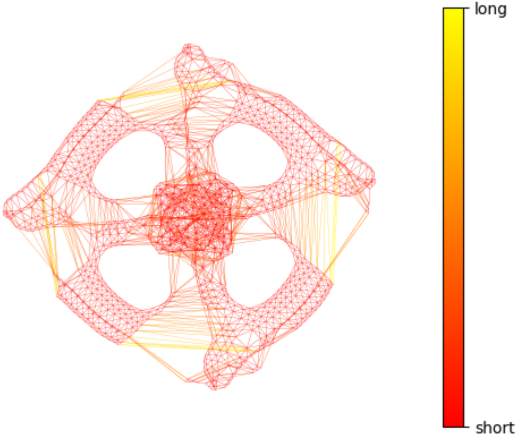
Results

Graph t-SNE I

dwt_1005 (from UF Sparse Matrix Collection): A structural dataset containing 1005 nodes and 8621 edges

Graph t-SNE I

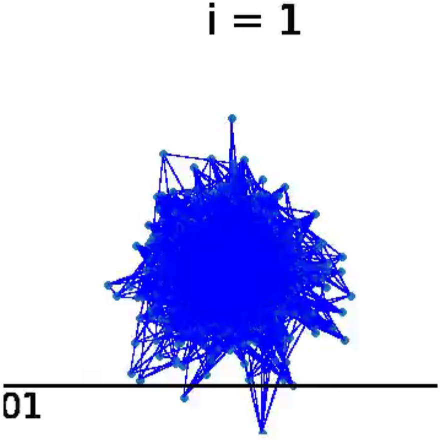
dwt_1005 (from UF Sparse Matrix Collection): A structural dataset containing 1005 nodes and 8621 edges



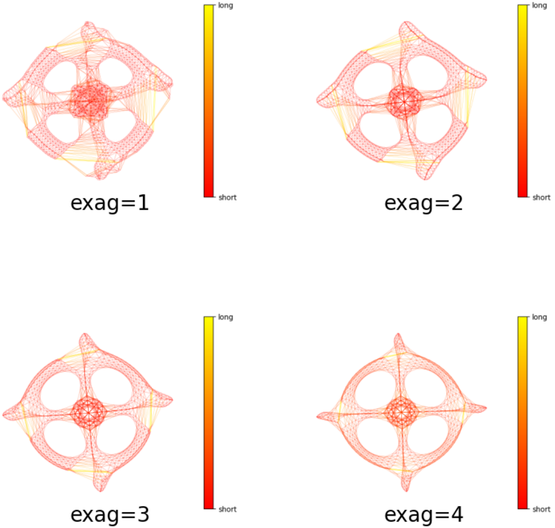
Standard t-SNE on dwt_1005. kNN recall = 0.79

Graph t-SNE I - Animation

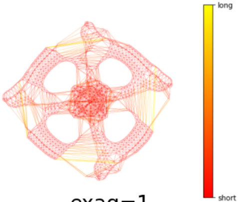
show here animation



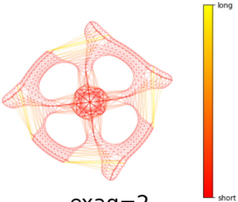
Graph t-SNE I - Exaggeration



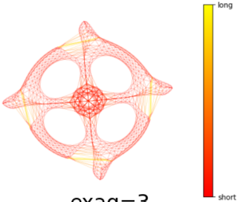
Graph t-SNE I - Exaggeration



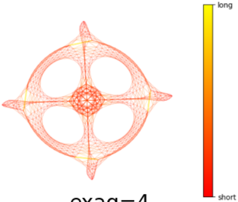
exag=1



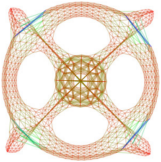
exag=2



exag=3



exag=4



dwt_1005 graph layout by tsNET* (Kruiger et al. 2017)

Graph t-SNE II

- Amazon Computer (from McAuley et al. 2015): A real world dataset containing 13752 nodes and 491722 edges
    - segment of Amazon Co-Purchase Graph

Graph t-SNE II

- Amazon Computer (from McAuley et al. 2015): A real world dataset containing 13752 nodes and 491722 edges
  - segment of Amazon Co-Purchase Graph
  - nodes indicate items

Graph t-SNE II

- Amazon Computer (from McAuley et al. 2015): A real world dataset containing 13752 nodes and 491722 edges
  - segment of Amazon Co-Purchase Graph
  - nodes indicate items
  - edges indicate which items are frequently bought together

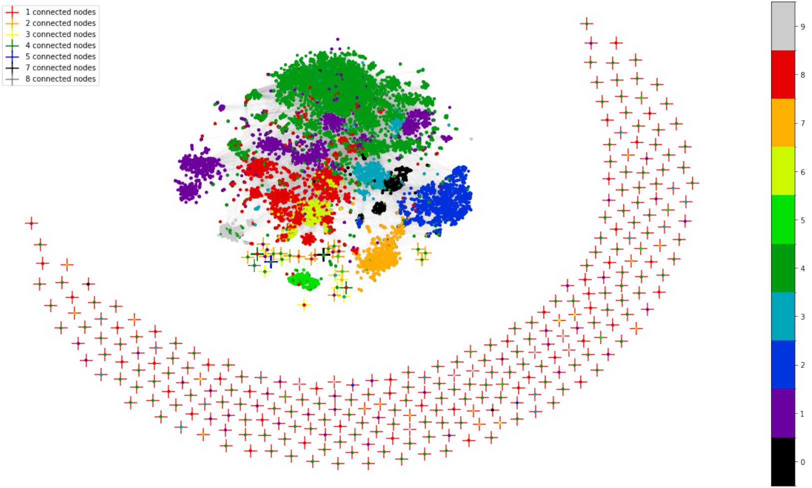
Graph t-SNE II

- Amazon Computer (from McAuley et al. 2015): A real world dataset containing 13752 nodes and 491722 edges
  - segment of Amazon Co-Purchase Graph
  - nodes indicate items
  - edges indicate which items are frequently bought together
  - labeled by 10 subcategories

Graph t-SNE II

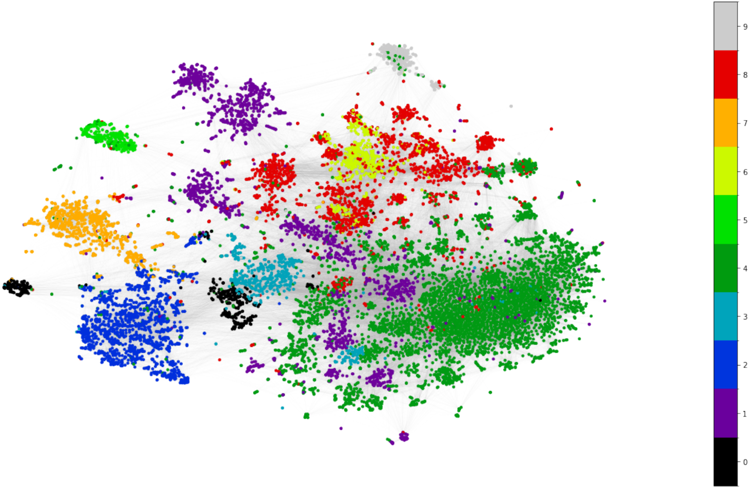
- Amazon Computer (from McAuley et al. 2015): A real world dataset containing 13752 nodes and 491722 edges
  - segment of Amazon Co-Purchase Graph
  - nodes indicate items
  - edges indicate which items are frequently bought together
  - labeled by 10 subcategories
  - contains 767 node features, which are bag-of-words encoded product reviews

Graph t-SNE II



Standard t-SNE on Amazon Computer. Crosses indicate connected components. Coloured by Labels. kNN accuracy = 0.89

Graph t-SNE II



Standard t-SNE on the biggest connected component of Amazon Computer. Coloured by Labels. kNN accuracy = 0.93

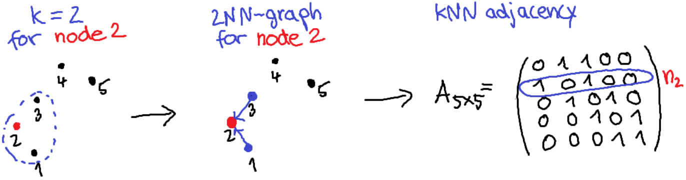
Feature t-SNE

What is the input?

Feature t-SNE

What is the input?

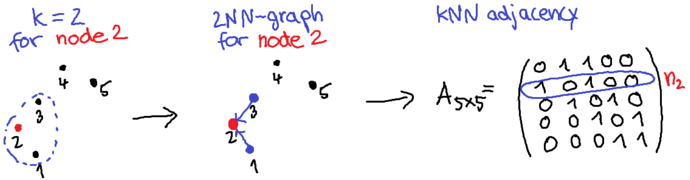
Constuct kNN Graph:



1. Find k nearest neighbours (default $k = 90$) in the feature space and construct a kNN adjacency matrix.

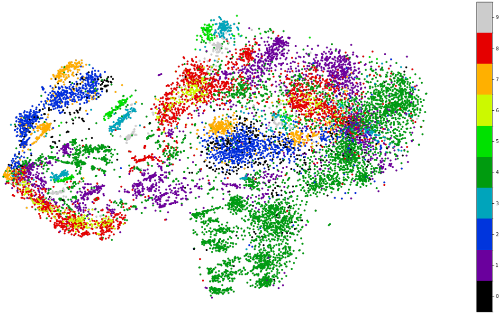
Feature t-SNE

What is the input?
Constuct kNN Graph:



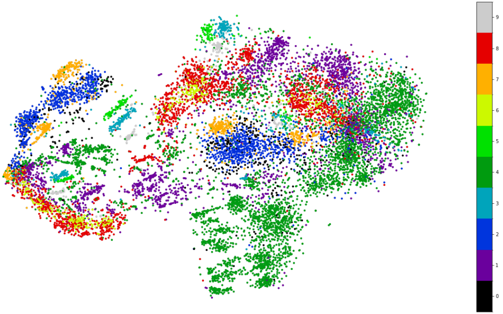
1. Find k nearest neighbours (default $k = 90$) in the feature space and construct a kNN adjacency matrix.
2. Construct an affinity matrix from the kNN adjacency.

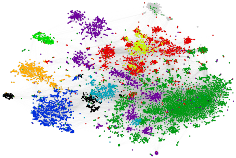
Feature t-SNE



Standard t-SNE on features of biggest connected component of Amazon Computer. Coloured by Labels. kNN accuracy = 0.79

Feature t-SNE



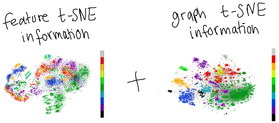


Graph t-SNE. kNN accuracy = 0.93

Standard t-SNE on features of biggest connected component of Amazon Computer. Coloured by Labels. kNN accuracy = 0.79

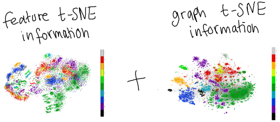
Contrastive Graph t-SNE Embedding - Concept

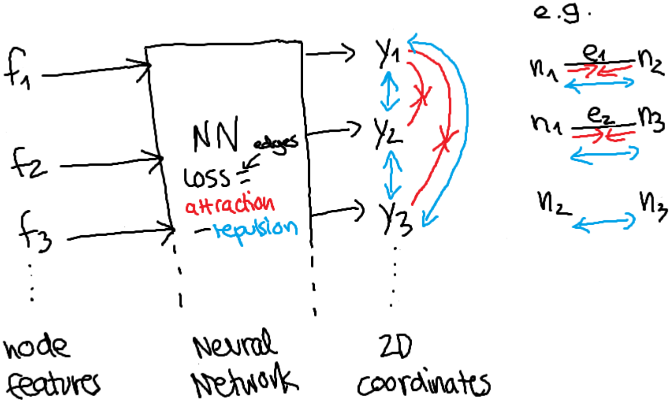
Idea: Combine



Contrastive Graph t-SNE Embedding - Concept

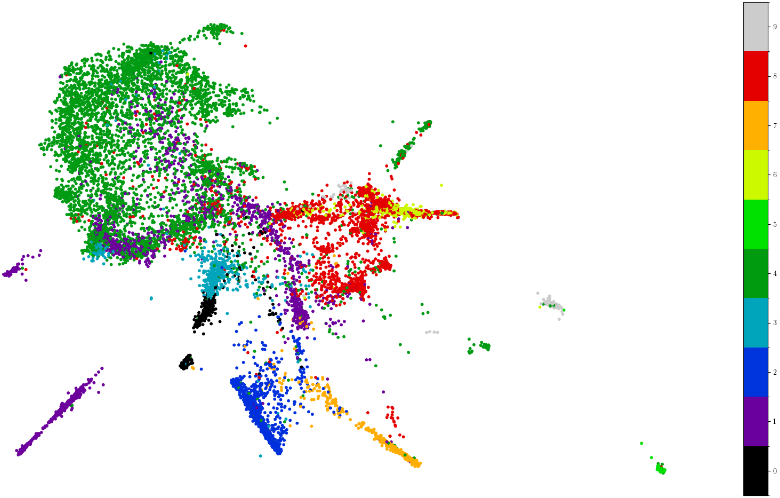
Idea: Combine





framework based on Damrich et al. 2022

Contrastive t-SNE Embedding



Contrastive t-SNE Embedding of biggest connected component of Amazon Computer. Coloured by label. kNN accuray = 0.84

Conclusion and Outlook

Conclusion:

Conclusion and Outlook

Conclusion:

- graph embeddings based on edge information, feature information and both using t-SNE

Conclusion and Outlook

Conclusion:

- graph embeddings based on edge information, feature information and both using t-SNE
- our method is easy to implement

Conclusion and Outlook

Conclusion:

- graph embeddings based on edge information, feature information and both using t-SNE
- our method is easy to implement
- no computation of shortest path distances needed

Conclusion and Outlook

Conclusion:

- graph embeddings based on edge information, feature information and both using t-SNE
- our method is easy to implement
- no computation of shortest path distances needed

Outlook:

Conclusion and Outlook

Conclusion:

- graph embeddings based on edge information, feature information and both using t-SNE
- our method is easy to implement
- no computation of shortest path distances needed

Outlook:

- Contrastive t-SNE Embedding promising for datasets where features encode useful neighbourhood information e.g. citation networks

Conclusion and Outlook

Conclusion:

- graph embeddings based on edge information, feature information and both using t-SNE
- our method is easy to implement
- no computation of shortest path distances needed

Outlook:

- Contrastive t-SNE Embedding promising for datasets where features encode useful neighbourhood information e.g. citation networks
- investigate Contrastive t-SNE Graph Embedding framework with different settings

Thank you!