Non-linear DR Visualization using t-SNE

CK Raju PhD

October 29, 2023

Quotes

- Through t-SNE's magic, dimensions transform,

Quotes

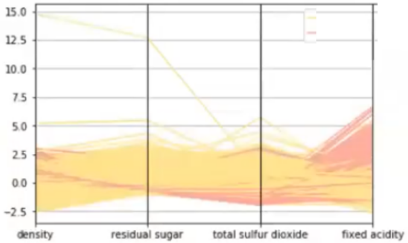
- Through t-SNE's magic, dimensions transform,
- High-dimensional worlds in visual grace, reborn.

Recap

DR and Visualization

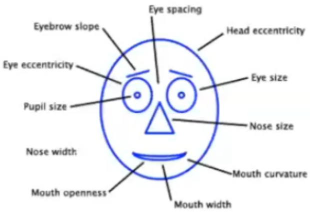
Direct Visualisation

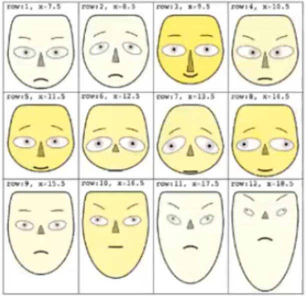
- Parallel Co-ordinates
  - Each vertical line is a dimension
  - A data item is connected by line segments
  - Large number of samples clutters the visualization



Direct Visualisation

- Parallel Co-ordinates
  - Each vertical line is a dimension
  - A data item is connected by line segments
  - Large number of samples clutters the visualization
- Chernoff Faces [Herman Chernoff: 1973]





Other Methods for Direct Visualisation

- A few approaches have been proposed to improve upon Chernoff Faces
- Empathic Visualization
  - Improve visual similarity of similar vectors
- Metaphoric 3D glyphs
  - Use metaphors other than faces (say Trees)

Image: The Empathic Visualization Algo: Chernoff Faces Revisited
http://www0.cs.ucl.ac.uk/staff/a.loizides/218.pdf

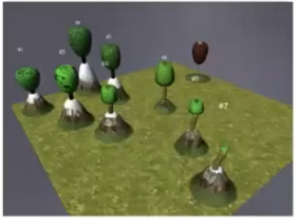
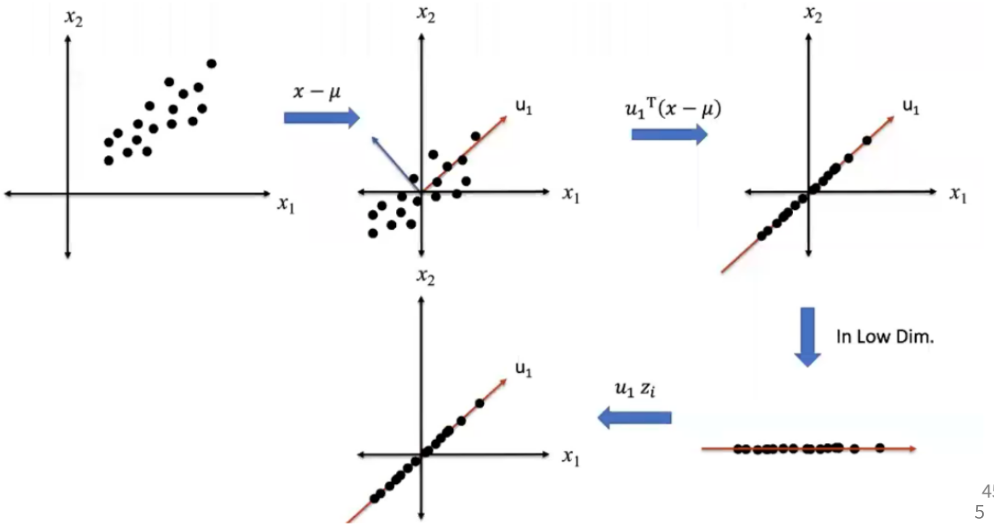
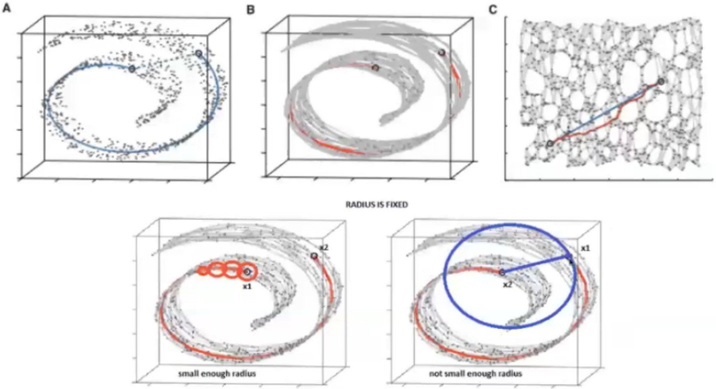


Image: "Beyond Chernoff faces: Multivariate visualization with metaphoric 3D glyphs", Marco Lardelli

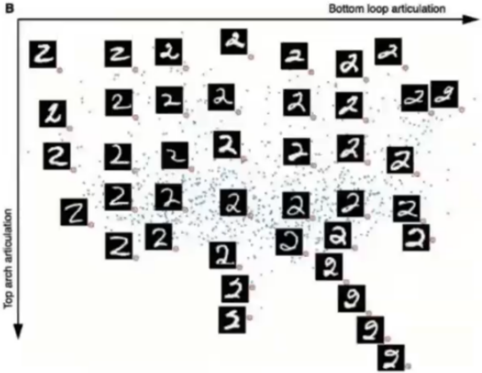
Dimensionality Reduction and Inverse Mapping



ISOMAP - Effect of Radius in Computation



Example - Handwritten Letters



Neighborhood Embedding

Prioritizing the Near and Dear

Neighborhood Embedding

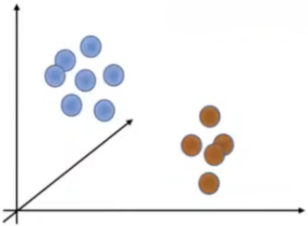
- **Neighborhood**: A collection of samples that are close to a given sample. Defined by:
  - A radius
  - Number of samples

Neighborhood Embedding

- **Neighborhood**: A collection of samples that are close to a given sample. Defined by:
  - A radius
  - Number of samples
- **Embedding**: Representation of samples in a low-dimensional subspace

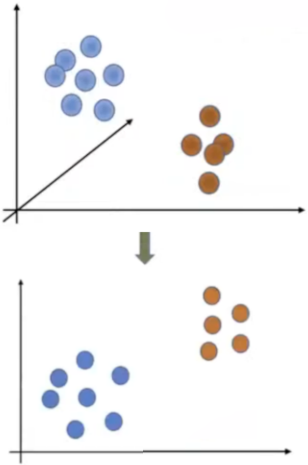
Neighborhood Embedding

- **Neighborhood**: A collection of samples that are close to a given sample. Defined by:
  - A radius
  - Number of samples
- **Embedding**: Representation of samples in a low-dimensional subspace



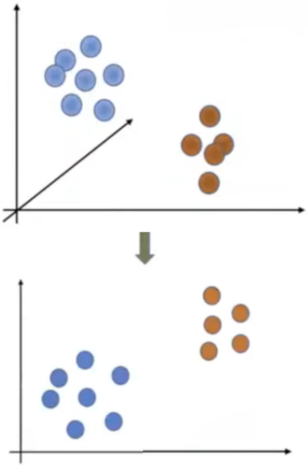
Neighborhood Embedding

- **Neighborhood**: A collection of samples that are close to a given sample. Defined by:
  - A radius
  - Number of samples
- **Embedding**: Representation of samples in a low-dimensional subspace



Neighborhood Embedding

- **Neighborhood**: A collection of samples that are close to a given sample. Defined by:
  - A radius
  - Number of samples
- **Embedding**: Representation of samples in a low-dimensional subspace
- **Goal**: Preserve the structure of local neighborhood



The Process

- Define a Metric to Preserve: $d(X)$
- Compute the metric for each neighborhood: $d(X_i)$

The Process

- Define a Metric to Preserve: $d(X)$
- Compute the metric for each neighborhood: $d(X_i)$
- Find an embedding X_i' that minimizes:

$$\text{Cost} = \sum_i |d(X_i) - d(X_i')|$$

The Process

- Define a Metric to Preserve: $d(X)$
- Compute the metric for each neighborhood: $d(X_i)$
- Find an embedding X_i' that minimizes:

$$\text{Cost} = \sum_i |d(X_i) - d(X_i')|$$

- Approach:
  1. Randomly Initialize X_i'.
  2. Compute $d(X_i')$.
  3. Move X_i' in a direction that reduces Cost.
  4. Repeat 3 and 4 until convergence.

The Process

- Define a Metric to Preserve: $\boldsymbol{d(X)}$
- Compute the metric for each neighborhood: $\boldsymbol{d(X_i)}$
- Find an embedding $\boldsymbol{X_i}'$ that minimizes:

$$\text{Cost} = \sum_i |\boldsymbol{d}(\boldsymbol{X}_i) - \boldsymbol{d}(\boldsymbol{X}_i')|$$

- Approach:
  1. Randomly Initialize $\boldsymbol{X_i}'$.
  2. Compute $\boldsymbol{d}(\boldsymbol{X}_i')$.
  3. Move $\mathbf{X}_i$' in a direction that reduces Cost.
  4. Repeat 3 and 4 until convergence.

Optimization

What Metric to Minimize

- Distance between points in a neighborhood

$$d_{ij} = d(X_i, X_j) = Eucl.Dist(X_i, X_j)$$

$$Cost = \sum_{i} \sum_{j \in N(X_i)} |d_{ij} - d'_{ij}|$$

What Metric to Minimize

- Distance between points in a neighborhood

$$d_{ij} = d(X_i, X_j) = Eucl.\,Dist(X_i, X_j)$$

$$Cost = \sum_{i} \sum_{j \in N(X_i)} |d_{ij} - d'_{ij}|$$

- Similarity between points in a neighborhood
  - Nearby Points have high similarity; Far-away points have lower similarity
  - Use s_{ij} instead of d_{ij}

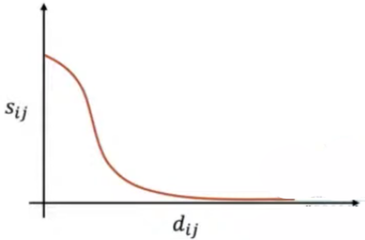
What Metric to Minimize

- Distance between points in a neighborhood

$$d_{ij} = d(X_i, X_j) = Eucl.Dist(X_i, X_j)$$

$$Cost = \sum_{i} \sum_{j \in N(X_i)} |d_{ij} - d'_{ij}|$$

- Similarity between points in a neighborhood
  - Nearby Points have high similarity; Far-away points have lower similarity
  - Use s_{ij} instead of d_{ij}



What Metric to Minimize

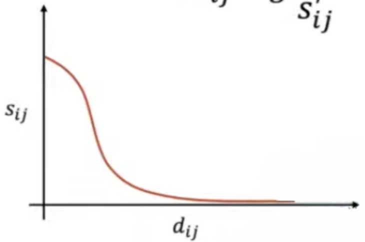
- Distance between points in a neighborhood

$$d_{ij} = d(X_i, X_j) = Eucl.\,Dist(X_i, X_j)$$

$$Cost = \sum_{i} \sum_{j \in N(X_i)} \boxed{|d_{ij} - d'_{ij}|}$$

$$s_{ij} \log \frac{s_{ij}}{s'_{ij}}$$

- Similarity between points in a neighborhood
  - Nearby Points have high similarity; Far-away points have lower similarity
  - Use s_{ij} instead of d_{ij}



What Metric to Minimize

- Distance between points in a neighborhood

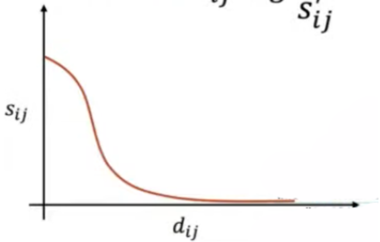
$$d_{ij} = d(X_i, X_j) = Eucl.Dist(X_i, X_j)$$

$$Cost = \sum_{i} \sum_{j \in N(X_i)} \boxed{|d_{ij} - d'_{ij}|}$$

SNE

$$s_{ij} \log \frac{s_{ij}}{s'_{ij}}$$

- Similarity between points in a neighborhood
  - Nearby Points have high similarity; Far-away points have lower similarity
  - Use s_{ij} instead of d_{ij}



PCA vs SNE

PCA on MNIST *(0-9)*

SNE on MNIST *(0-5)*

..in short

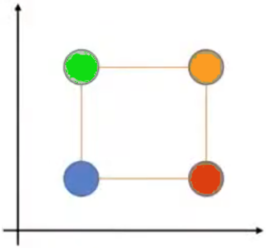
- preserve similarity in lower dimensional representation
- known as stochastic neighborhood embedding
- unsupervised technique, improvement over PCA

The t-SNE Algorithm

Computing the Embedding

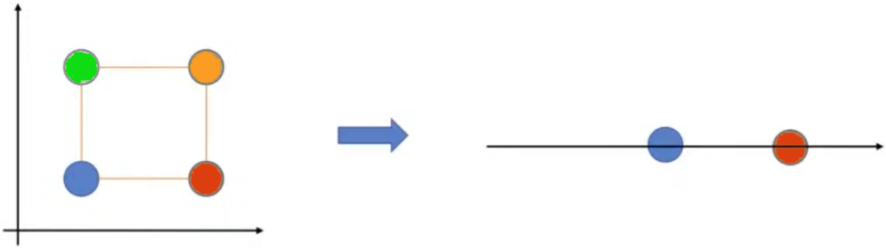
SNE - Properties/Issues

- Much better embedding than PCA
- Problem of Crowding



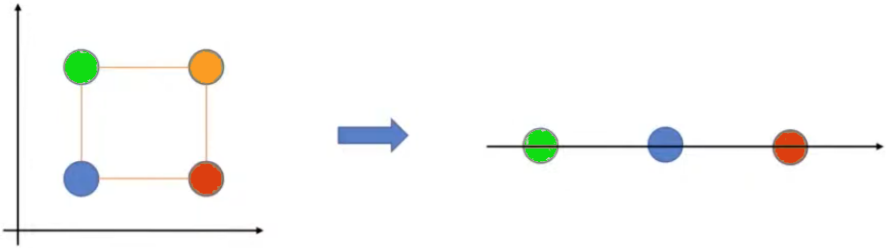
SNE - Properties/Issues

- Much better embedding than PCA
- Problem of Crowding



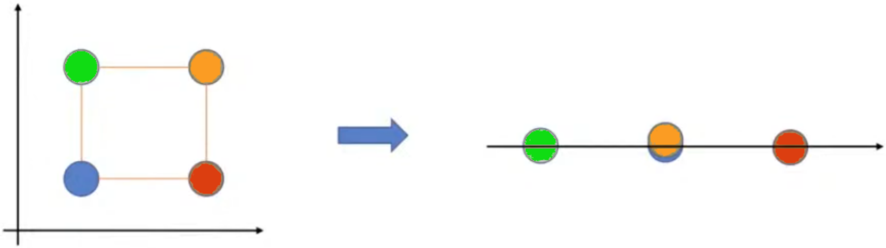
SNE - Properties/Issues

- Much better embedding than PCA
- Problem of Crowding



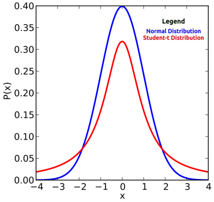
SNE - Properties/Issues

- Much better embedding than PCA
- Problem of Crowding



t-SNE

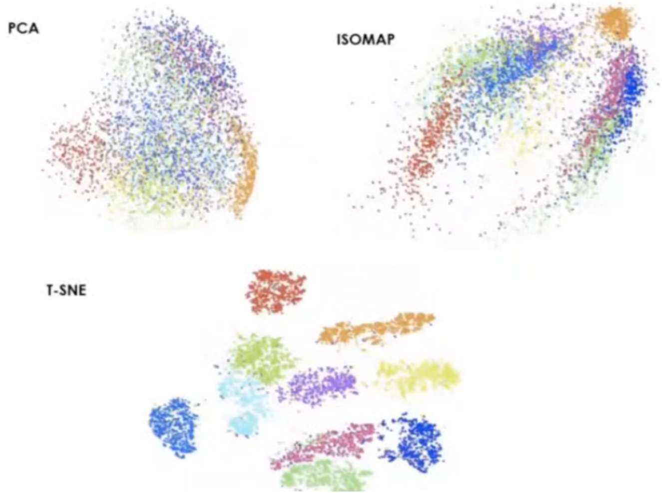
- Solution
  - Distant points drift farther apart
  - More flexibility in point distribution
- How to do this ?
  - Student-t distribution - computes S'_{ij}
  - Rest of the solution is same as SNE



Parameters of t-SNE

- Perplexity
  - Number of effective neighbors to consider
- Iterations
  - Steps after which we stop updating the embedding
- Step-size
  - Size of each update
- Initialization
  - Different initialisations lead to different visualisations

Comparison



Examples of t-SNE

Inferences from t-SNE Visualization

Example - Blog distill.pub

Link to t-SNE Visualization

Using and Reading t-SNE Results

- Running t-SNE

Using and Reading t-SNE Results

- Running t-SNE
  - Always run multiple trials

Using and Reading t-SNE Results

- Running t-SNE
  - Always run multiple trials
  - Use appropriate perplexity

Using and Reading t-SNE Results

- Running t-SNE
  - Always run multiple trials
  - Use appropriate perplexity
  - Let the samples stabilize (iterations)

Using and Reading t-SNE Results

- Running t-SNE
  - Always run multiple trials
  - Use appropriate perplexity
  - Let the samples stabilize (iterations)
- Reading t-SNE

Using and Reading t-SNE Results

- Running t-SNE
  - Always run multiple trials
  - Use appropriate perplexity
  - Let the samples stabilize (iterations)
- Reading t-SNE
  - Do not give importance to distances between points far-away

Using and Reading t-SNE Results

- Running t-SNE
  - Always run multiple trials
  - Use appropriate perplexity
  - Let the samples stabilize (iterations)
- Reading t-SNE
  - Do not give importance to distances between points far-away
  - Do not give significance to density of clusters

Using and Reading t-SNE Results

- Running t-SNE
  - Always run multiple trials
  - Use appropriate perplexity
  - Let the samples stabilize (iterations)
- Reading t-SNE
  - Do not give importance to distances between points far-away
  - Do not give significance to density of clusters
  - Do not infer anything from a single output