t-Distributed Stochastic Neighbor Embedding (t-SNE)

Slides: Selina Carter

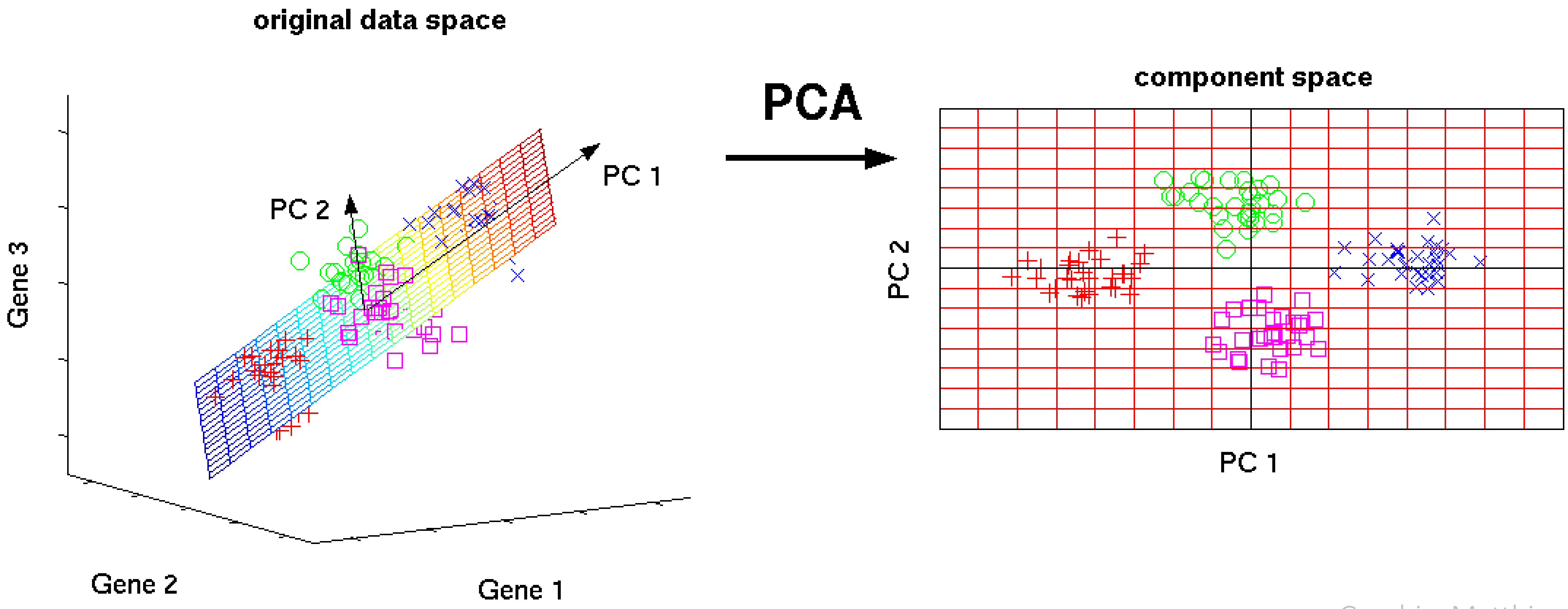
April 19, 2022

Outline

1. Motivation
2. What is “manifold learning”
3. SNE algorithm
   + Examples
4. t-SNE algorithm
   + Examples

Motivation

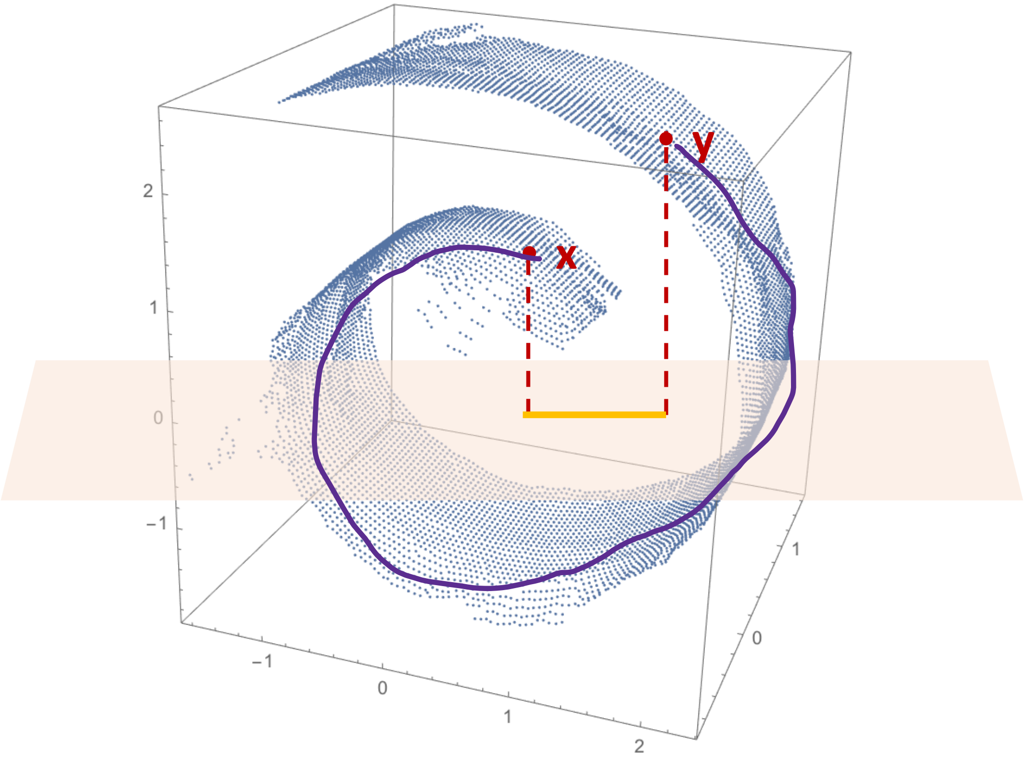
- PCA maps data in high dimensions to a plane

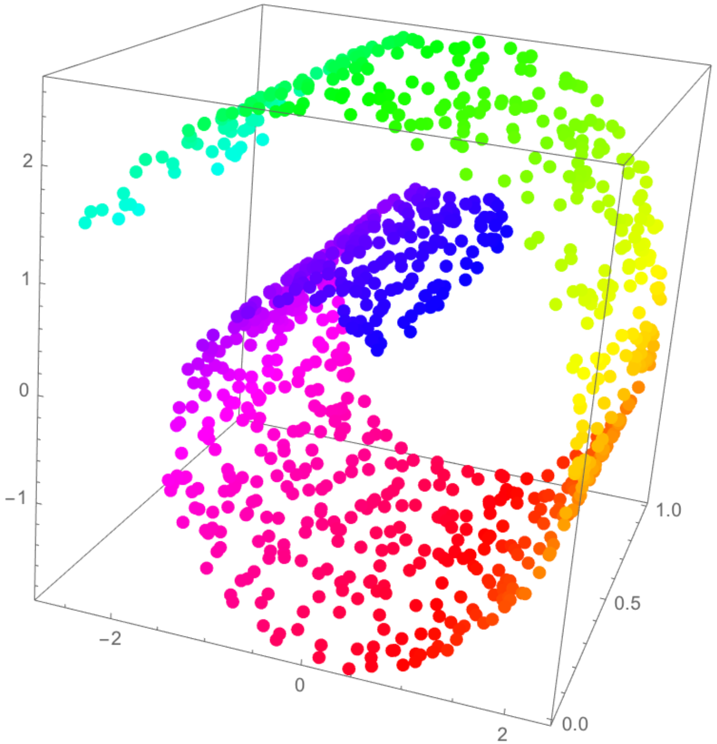


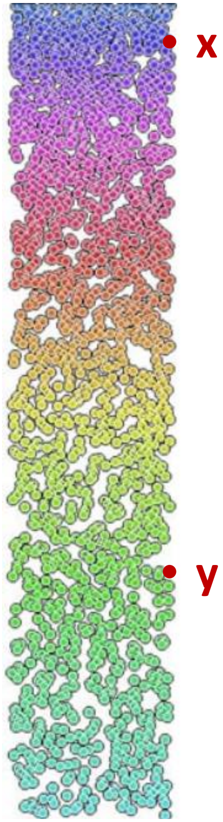
Graphic: Matthias Sholz

Motivation

- But what if your data isn't nicely represented by a plane?





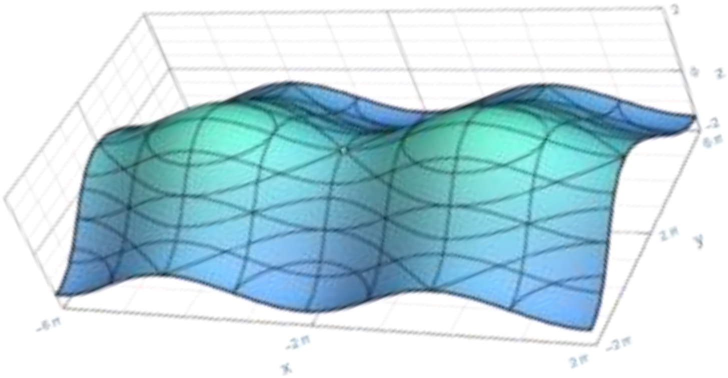


Graphics: Wolfram, Gashler et al 2011

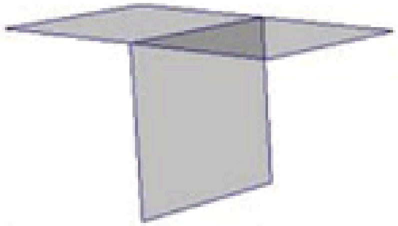
This is called “manifold learning”

- **Manifold**: basically, a type of curve or surface (locally resembles a Euclidean space)
  - In our case, we’re talking about k-manifolds that are subspaces of a Euclidean space

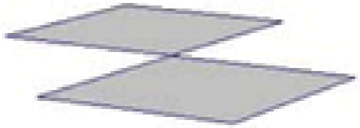
- Simple example: 2-manifold (a surface)



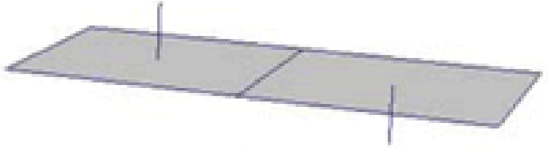
- Not manifolds:



Three or more faces share an edge.



Two or more faces share a single vertex but no edge.



Adjacent faces have opposite normals.

Graphics: Wikipedia , Autodesk

Manifold learning is not new

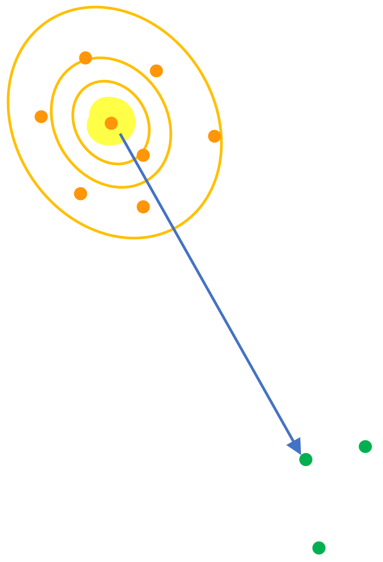
- Linear manifolds (subspaces)
  - Principal Component Analysis (PCA) (Pearson, 1901; Hotelling, 1933)
  - (Classical) Multidimensional Scaling (MDS) (Torgerson, 1952)

- Non-linear manifolds
  - Sammon mapping (Sammon, 1969)
  - Curvilinear components analysis (CCA) (Demartines and Herault, 1997)
  - Isomap (Tenenbaum et al., 2000)
  - Locally Linear Embedding (LLE) (Roweis and Saul, 2000)
  - Stochastic Neighbor Embedding (SNE) (Hinton and Roweis, 2002)
  - Laplacian Eigenmaps (Belkin and Niyogi, 2002)
  - Maximum Variance Unfolding (MVU) (Weinberger et al., 2004)
  - t-Distributed Stochastic Neighbor Embedding (t-SNE) (Van der Maaten and Hinton, 2008)

Stochastic neighbor embedding (SNE)

- Big picture:
  - **Dimensionality reduction**: describe distances between objects in a high-D space by placing the objects in a low-D space
    - "**embed**" objects originally in high-D space into a lower-D space

- Toolkit:
  - Probabilistic approach: there is no "one" mapping
  - Gaussian distribution in high-D and low-D space
  - Gradient descent



SNE algorithm (part 1)

1. Define a “distance metric” between points *i* and *j* in original (high-dimensional) space $(\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathbb{R}^N)$:

$$d_{ij}^2 := \frac{\left|\left|\boldsymbol{x}_i - \boldsymbol{x}_j\right|\right|^2}{2\sigma_i^2}, \qquad \boldsymbol{x}_i, \boldsymbol{x}_j \in \mathbb{R}^N, \sigma_i \in \mathbb{R}$$

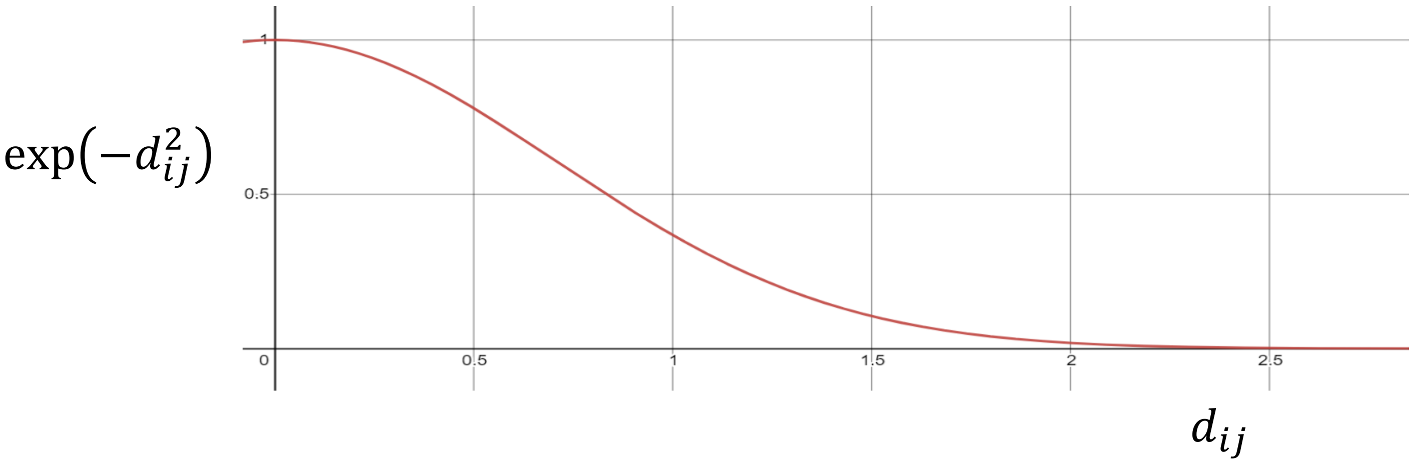
Note:
$d_{ij} \neq d_{ji}$

2. For each object *i* and each potential neighbor *j*, compute the “similarity of $\boldsymbol{x}_j$ to $\boldsymbol{x}_i$”:

   i.e., conditional probability $p_{j|i}$ that *i* picks *j* as its neighbor:

$$p_{j|i} := \frac{\exp\left(-d_{ij}^2\right)}{\sum_{k \neq i} \exp\left(-d_{ik}^2\right)}, \qquad p_{i|i} = 0$$

Note:
$p_{j|i} \neq p_{i|j}$



SNE algorithm (part 2)

3. Now in low-dimensions ($\mathbb{R}^m, m \ll N$): imagine low-dimensional images $\boldsymbol{y}_i \in \mathbb{R}^m$ of each point *i*.

The conditional probability $q_{j|i}$ that *i* picks *j* as its neighbor is:

$$q_{j|i} \coloneqq \frac{\exp\left(-\left|\left|\boldsymbol{y}_i - \boldsymbol{y}_j\right|\right|^2\right)}{\sum_{k \neq i} \exp\left(-\left|\left|\boldsymbol{y}_i - \boldsymbol{y}_k\right|\right|^2\right)}, \qquad q_{i|i} = 0$$

Note:
$q_{j|i} \neq q_{i|j}$

4. **Goal**: choose mappings $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\}$ so that $p_{j|i} = q_{j|i}$

➡ For each object *i*, minimize cost function: sum of Kullback-Leibler divergences between $p_{j|i}$ and $q_{j|i}$:

$$C \coloneqq \sum_{i=1}^{N} \sum_{j=1}^{N} p_{j|i} \log\left(\frac{p_{j|i}}{q_{j|i}}\right) = \sum_{i=1}^{N} KL(P_i || Q_i)$$

Look closer at the cost function

$$C := \sum_{i=1}^{N} \sum_{j=1}^{N} p_{j|i} \log\left(\frac{p_{j|i}}{q_{j|i}}\right) = \sum_{i=1}^{N} KL(P_i || Q_i)$$

- With respect to *what* are we minimizing?

$$\min_{\mathbf{y}_1,\dots\mathbf{y}_{\mathrm{n}\in\mathbb{R}^m}} C = \min_{\mathbf{y}_1,\dots\mathbf{y}_{\mathrm{n}\in\mathbb{R}^m}} \sum_{i=1}^{N} KL(P_i || Q_i)$$

- How do you minimize it?
  - Naive: ~~minimize~~ $\cancel{\frac{p_{j|i}}{q_{j|i}}}$
  - You have a "budget" for $q_{j|i}$: $\sum_{j=1}^{N} q_{j|i} = 1$
  - Result: choose $\mathbf{y}_1, \dots \mathbf{y}_{\mathrm{n}\in\mathbb{R}^m}$ such that $q_{j|i} \approx p_{j|i}$
- Note: KL divergence is not symmetric in p and q
  - Cost(small $q_{j|i}$, large $p_{j|i}$) $>$ Cost(large $q_{j|i}$, small $p_{j|i}$)

SNE algorithm (review)

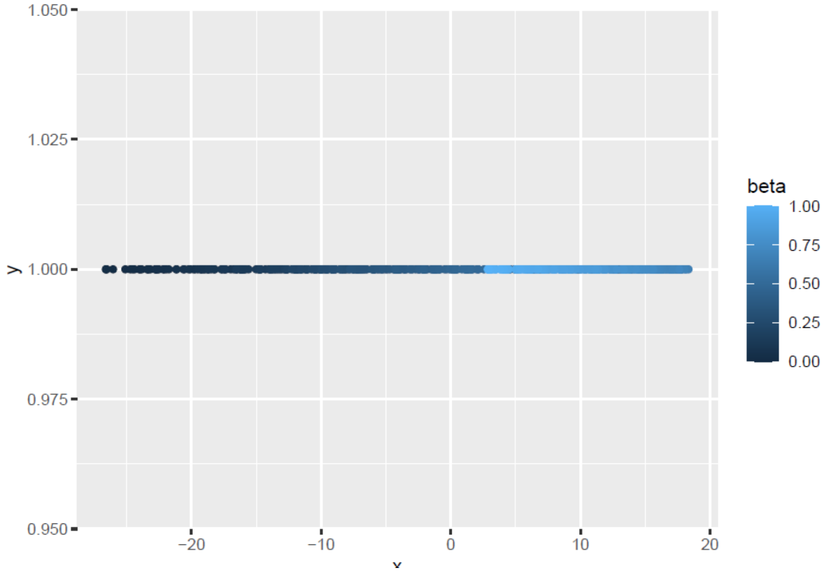
- **Fixed things:**
  1. Data: $\boldsymbol{x}_1, \dots, \boldsymbol{x}_n \in \mathbb{R}^N$
  2. Choose size of lower dimension ($\mathbb{R}^m, m \ll N$)
  3. Distance metric d_{ij}^2 in $\mathbb{R}^N$
  $$d_{ij}^2 \coloneqq \frac{\left|\left|\boldsymbol{x}_i - \boldsymbol{x}_j\right|\right|^2}{2\sigma_i^2}, \qquad \sigma_i \in \mathbb{R}$$
  4. In $\mathbb{R}^N$, compute probability *i* picks *j* as its neighbor:
  $$p_{j|i} \coloneqq \frac{\exp\left(-d_{ij}^2\right)}{\sum_{k \neq i} \exp\left(-d_{ik}^2\right)}$$
  5. Given the images $\mathbf{y}_1, \dots, \boldsymbol{y}_n \in \mathbb{R}^m$,
  $$q_{j|i} \coloneqq \frac{\exp\left(-\left|\left|\boldsymbol{y}_i - \boldsymbol{y}_j\right|\right|^2\right)}{\sum_{k \neq i} \exp\left(-\left|\left|\boldsymbol{y}_i - \boldsymbol{y}_k\right|\right|^2\right)}$$
- **Task**: choose $\mathbf{y}_1, \dots \mathbf{y}_{\mathrm{n} \in \mathbb{R}^m}$ such that $q_{ij} \approx p_{ij}$

  ➡ Use gradient descent: $\mathcal{Y}^{(t)} = \mathcal{Y}^{(t-1)} + \eta \frac{\delta C}{\delta \mathcal{Y}} + \alpha(t)\left(\mathcal{Y}^{(t-1)} - \mathcal{Y}^{(t-2)}\right)$

Example 1: SNE

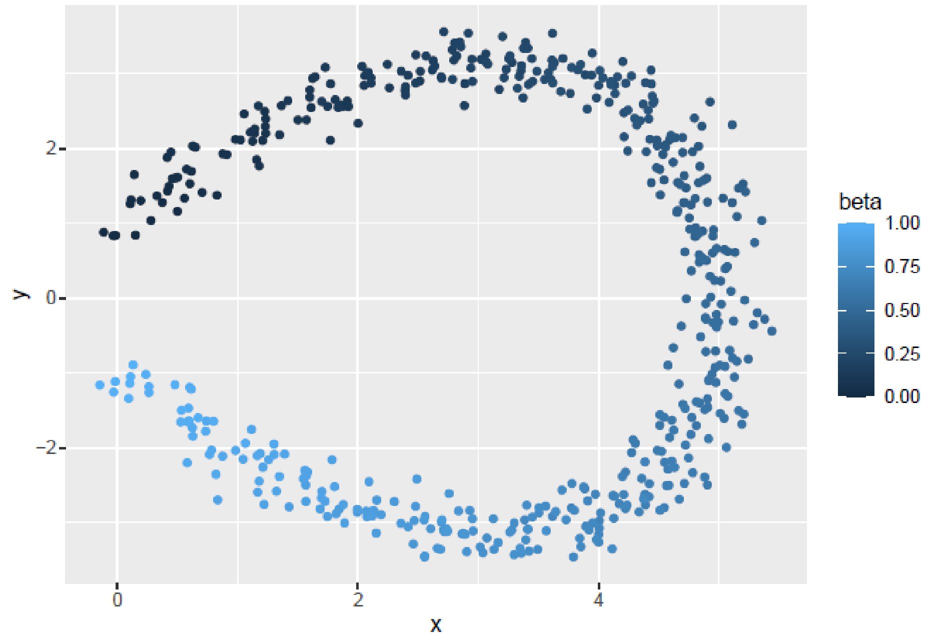
- Horseshoe shape:
  1. Data: $\boldsymbol{x}_1, \dots, \boldsymbol{x}_{500} \in \mathbb{R}^{10}$
  2. Lower dimension: $\mathbb{R}^1$ and $\mathbb{R}^2$

Mapping to $\mathbb{R}^1$



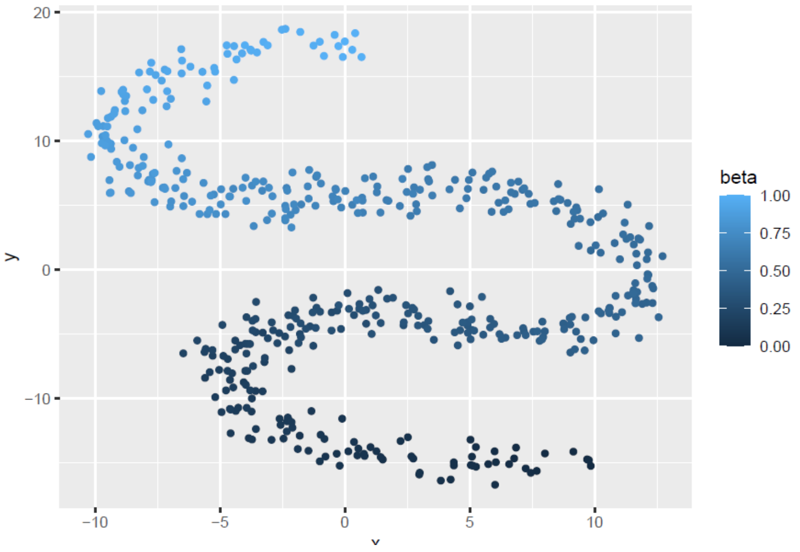
SNE to $\mathbb{R}^1$

Original data
(horseshoe jittered in $\mathbb{R}^{10}$)



SNE to $\mathbb{R}^2$

Mapping to $\mathbb{R}^2$

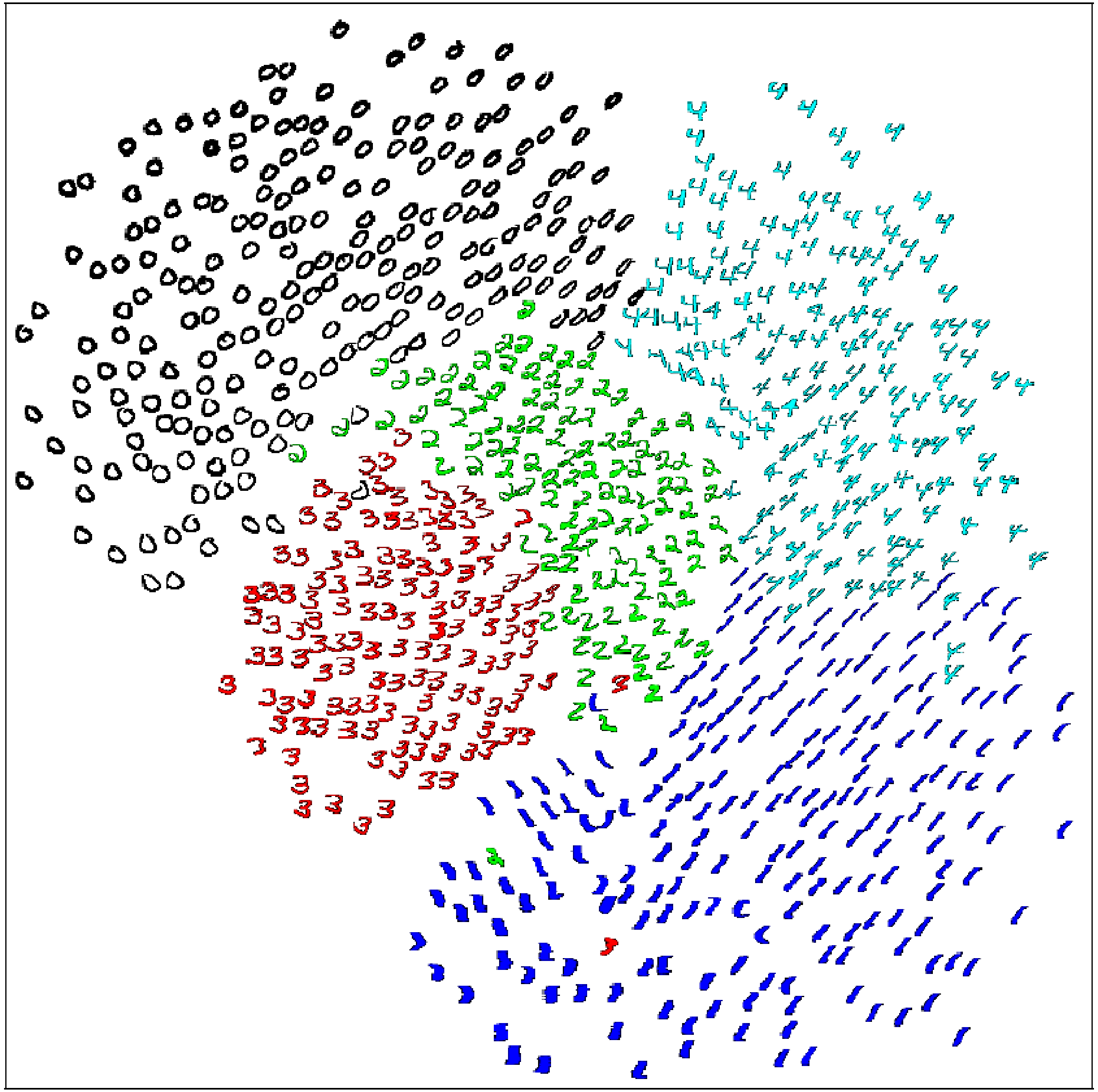


Graphic: Sivan Leviyang, Georgetown University

Example 2: SNE

- Hand-written digits:
  1. Data: $\boldsymbol{x}_1, \dots, \boldsymbol{x}_{3000} \in \mathbb{R}^{256}$ (grayscale images)
  2. Lower dimension: $\mathbb{R}^2$

- Problems:
  - Difficult to optimize the cost function
  - “Crowding” of points



Graphic: Hinton and Roweis 2002

Stochastic neighbor embedding (t-SNE)

- Big picture:
  - **Same idea as SNE, but easier to optimize – and "works better."**

- Key differences from SNE:
  - New cost function
    - Symmetric probabilities
    - Simpler gradient

    ➡ Faster to optimize
  - Instead of Gaussian, uses Student-t distribution (df = 1; same as Cauchy distribution) to compute similarity of points in low-D space

    ➡ Fixes "crowding" by separating clusters

t-SNE algorithm (part 1) (changes to SNE are in yellow)

1. Define a distance metric between points *i* and *j* in original (high-dimensional) space ($\boldsymbol{x}_i, \boldsymbol{x}_j \in \mathbb{R}^N$):

$$d_{ij}^2 := \frac{\left|\left|\boldsymbol{x}_i - \boldsymbol{x}_j\right|\right|^2}{2\sigma_i^2}, \qquad \boldsymbol{x}_i, \boldsymbol{x}_j \in \mathbb{R}^N, \sigma_i \in \mathbb{R}$$

2. For each object *i* and each potential neighbor *j*, compute probability p_{ij} that *i* picks *j* as its neighbor:

$$p_{ij} := \frac{p_{j|i} + p_{i|j}}{2n}, \qquad p_{ii} = 0$$

Note:
$p_{ij} = p_{ji}$

t-SNE algorithm (part 2) (changes to SNE are in yellow)

3. Now in low-dimensions ($\mathbb{R}^m, m \ll N$): imagine low-dimensional images $\boldsymbol{y}_i \in \mathbb{R}^m$ of each point *i*.

Then the induced probability q_{ij} that *i* picks *j* as its neighbor is:

$$q_{ij} = \frac{\left(1 + ||\boldsymbol{y}_i - \boldsymbol{y}_j||^2\right)^{-1}}{\sum_{k \neq l} \exp\left(1 + ||\boldsymbol{y}_k - \boldsymbol{y}_l||^2\right)^{-1}}$$

Note:
$q_{ij} = q_{ji}$

Fixed normalizing constant across all points

4. **Goal**: choose mappings $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_n\}$ so that $p_{j|i}$ = $q_{j|i}$

For each object *i*, minimize cost function: sum of Kullback-Leibler divergences between p_{ij} and q_{ij}:

$$C = \sum_{i=1}^{N} \sum_{j=1}^{N} p_{ij} \log\left(\frac{p_{ij}}{q_{ij}}\right) = KL(P||Q)$$

No longer a sum

t-SNE algorithm (condensed)

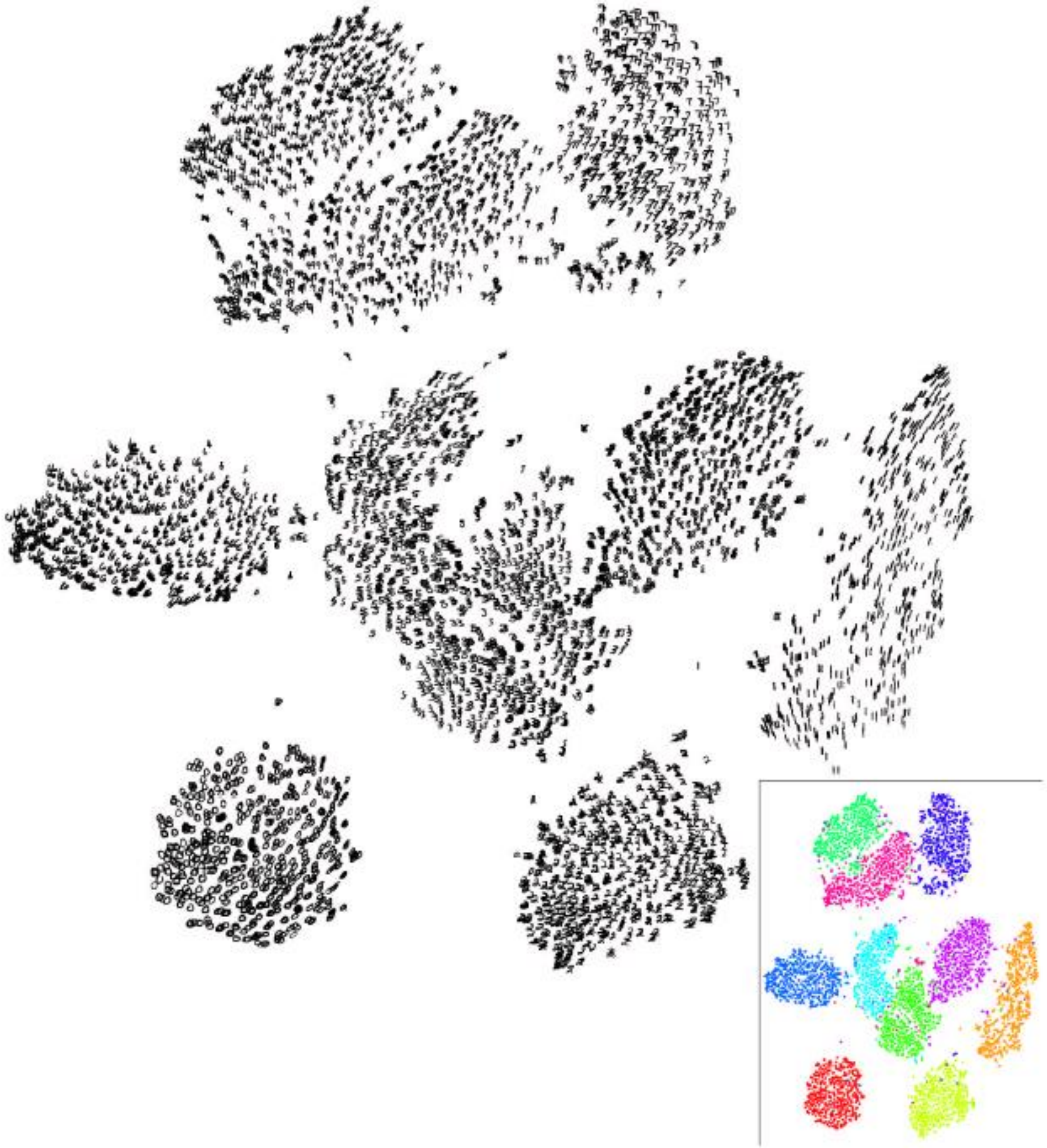
Algorithm 1: Simple version of t-Distributed Stochastic Neighbor Embedding.

Data: data set $X = \{x_1, x_2, ..., x_n\}$,
cost function parameters: perplexity $Perp$,
optimization parameters: number of iterations T, learning rate η, momentum $\alpha(t)$.
Result: low-dimensional data representation $\mathcal{Y}^{(T)} = \{y_1, y_2, ..., y_n\}$.
begin
  compute pairwise affinities $p_{j|i}$ with perplexity $Perp$ (using Equation 1)
  set $p_{ij} = \frac{p_{j|i}+p_{i|j}}{2n}$
  sample initial solution $\mathcal{Y}^{(0)} = \{y_1, y_2, ..., y_n\}$ from $\mathcal{N}(0, 10^{-4}I)$
  for t=*1* **to** T **do**
    compute low-dimensional affinities q_{ij} (using Equation 4)
    compute gradient $\frac{\delta C}{\delta \mathcal{Y}}$ (using Equation 5)
    set $\mathcal{Y}^{(t)} = \mathcal{Y}^{(t-1)} + \eta \frac{\delta C}{\delta \mathcal{Y}} + \alpha(t) \left(\mathcal{Y}^{(t-1)} - \mathcal{Y}^{(t-2)}\right)$
  end
end

Van der Maaten and Hinton, 2008

Example 2: t-SNE

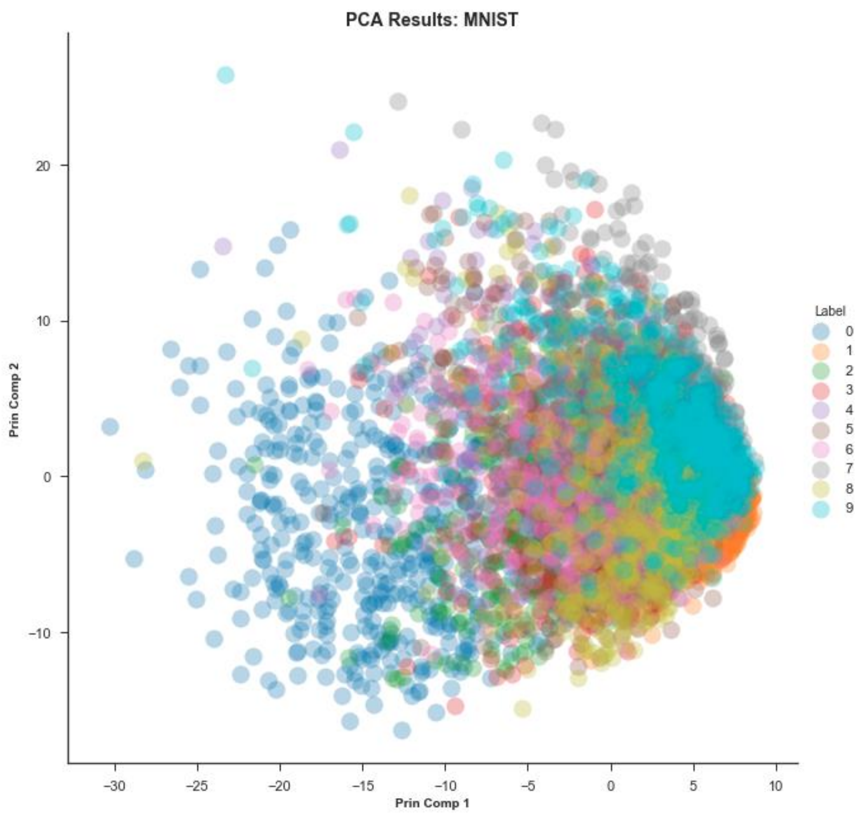
- Hand-written digits (MNIST):
  1. Data: $\boldsymbol{x}_1, \dots, \boldsymbol{x}_{6000} \in \mathbb{R}^{784}$ (grayscale images)
  2. Lower dimension: $\mathbb{R}^2$
  3. Then used k=20 nearest neighbors (color version)

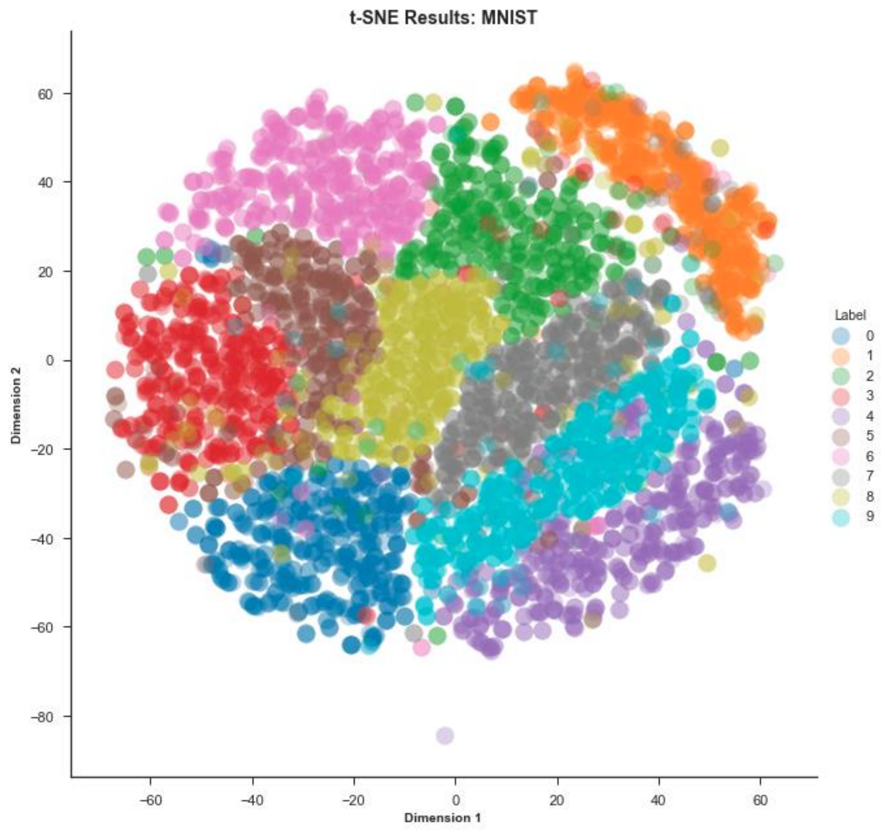


Graphic: Hinton and Roweis 2002

Example 3: PCA vs. t-SNE

- Hand-written digits (MNIST):
  1. Data: $\boldsymbol{x}_1, \dots, \boldsymbol{x}_{6000} \in \mathbb{R}^{784}$ (grayscale images)
  2. Lower dimension: $\mathbb{R}^2$





Graphics: Andre Violante 2009

Example 4: t-SNE

- Facial Expression Recognition (FER)
  Japanese Female Facial Expression (JAFFE) by Jizheng Yi et al, 2013

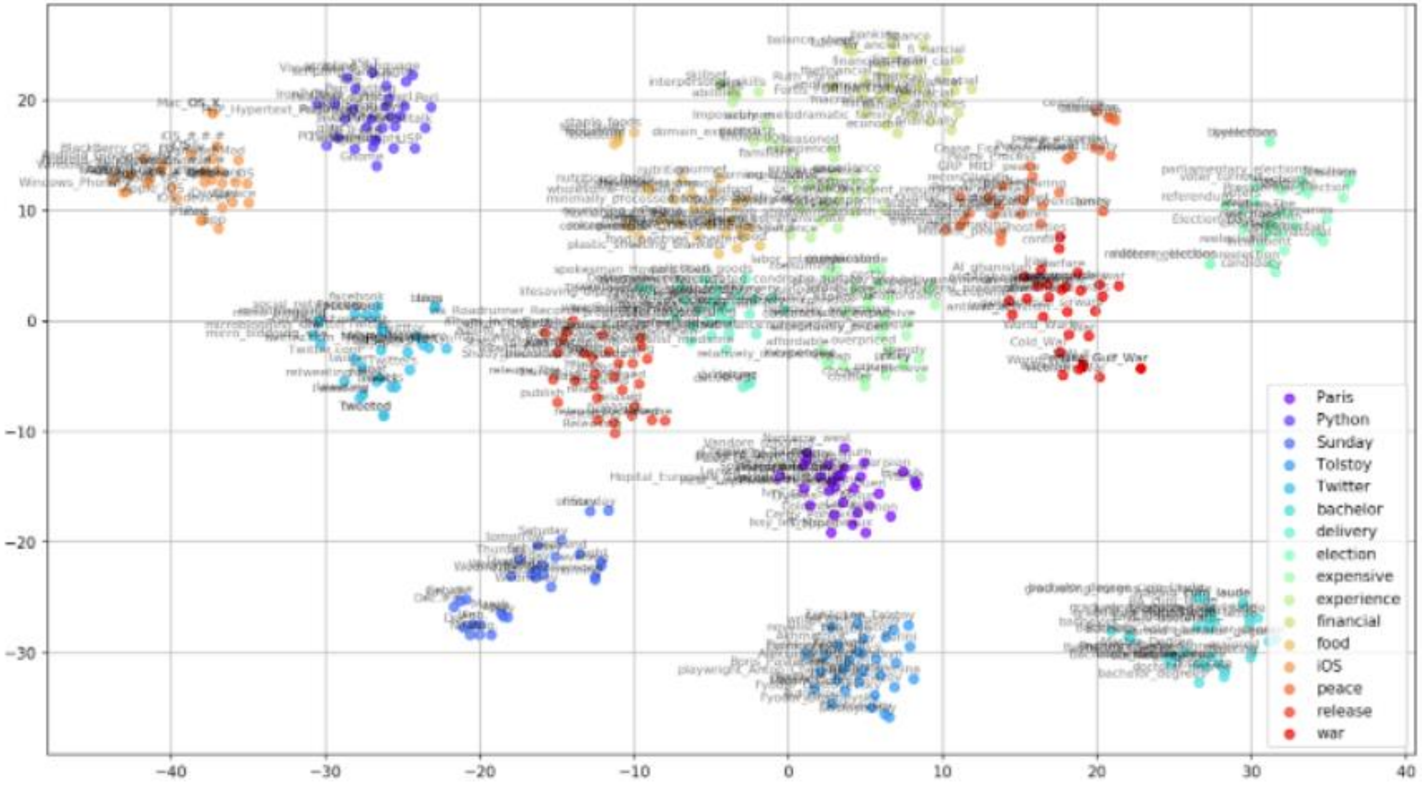
TABLE I. THE FER RATES BETWEEN DIFFERENT ALGORITHMS

	PCA	LDA	LLE	SNE	t-SNE
SVM	73.5%	74.3%	84.7%	89.6%	90.3%
AdaboostM2	75.4%	75.9%	87.7%	90.6%	94.5%

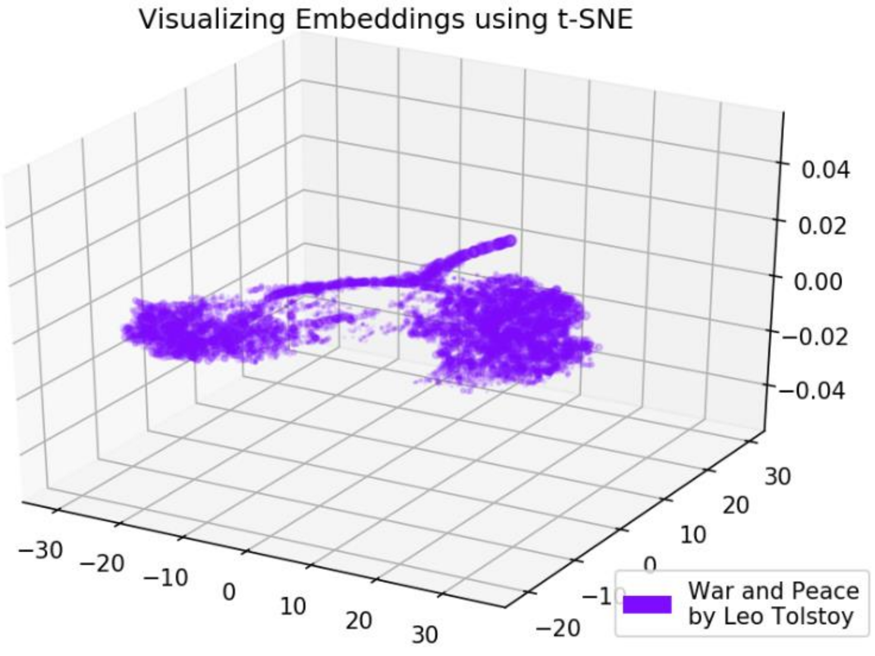
Graphic: Jizheng Yi et al, 2013

Example 5: t-SNE

- Word embeddings
  - Words in Google News (100 billion words, $\mathbb{R}^{200}$)
  - *War and Peace* by Tolstoy



Clusters of similar words from Google News (preplexity=15)



Visualization of the Word2Vec model trained on War and Peace

Graphic: Sergey Smetanin, 2018

Example 6: t-SNE

- Interactive viz & tutorial: https://distill.pub/2016/misread-tsne/

Questions?