Dimension Reduction

Accelerating t-SNE using Tree-Based Algorithms

Journal of Machine Learning Research 15(Oct):3221-3245, 2014

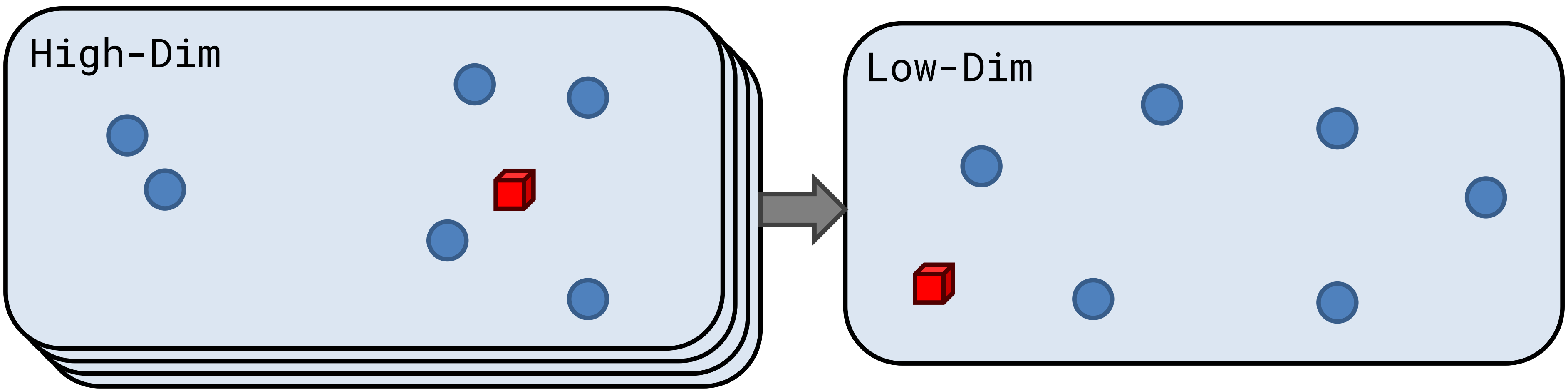
Laurens van der Maaten
Tilburg University

Geoffrey Hinton
University of Toronto

2015.10.21
Group Meeting Report
NCTU
指導教授 ｜ 林志青
0356624 ｜ 葉美伶

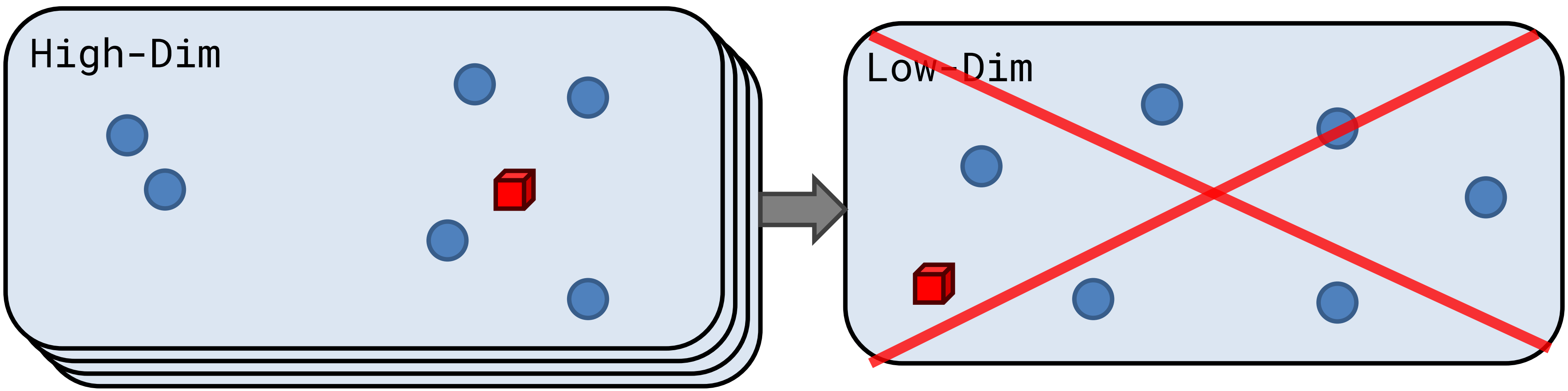
Introduction

Purpose: build ***map*** in which distances between points reflect similarities in the data



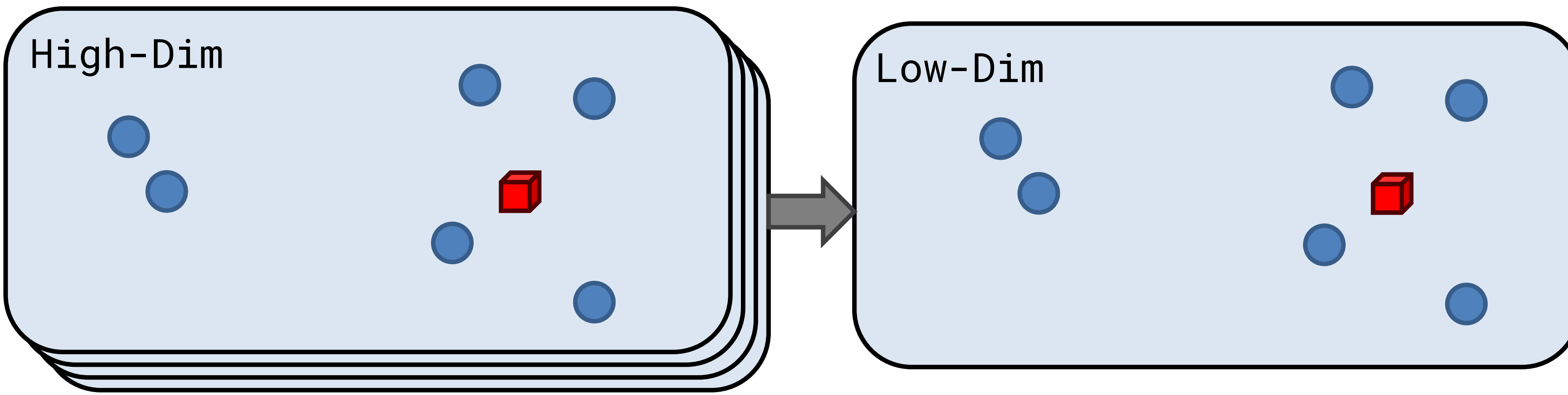
Introduction

Purpose: build ***map*** in which distances between points reflect similarities in the data



Introduction

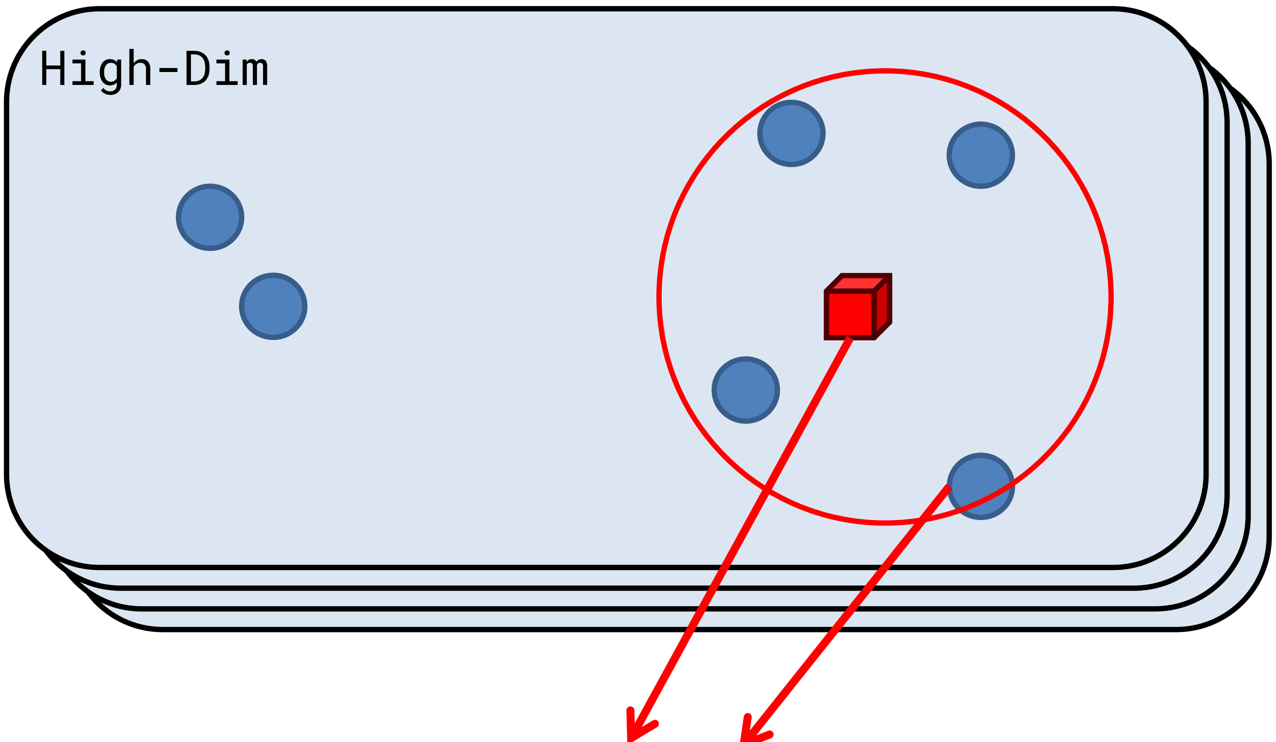
Purpose: build ***map*** in which distances between points reflect similarities in the data



To do this we need to minimize some objective function that measures the ***discrepancy*** between similarities in the data and similarities in the map

t-Distributed Stochastic Neighbor Embedding

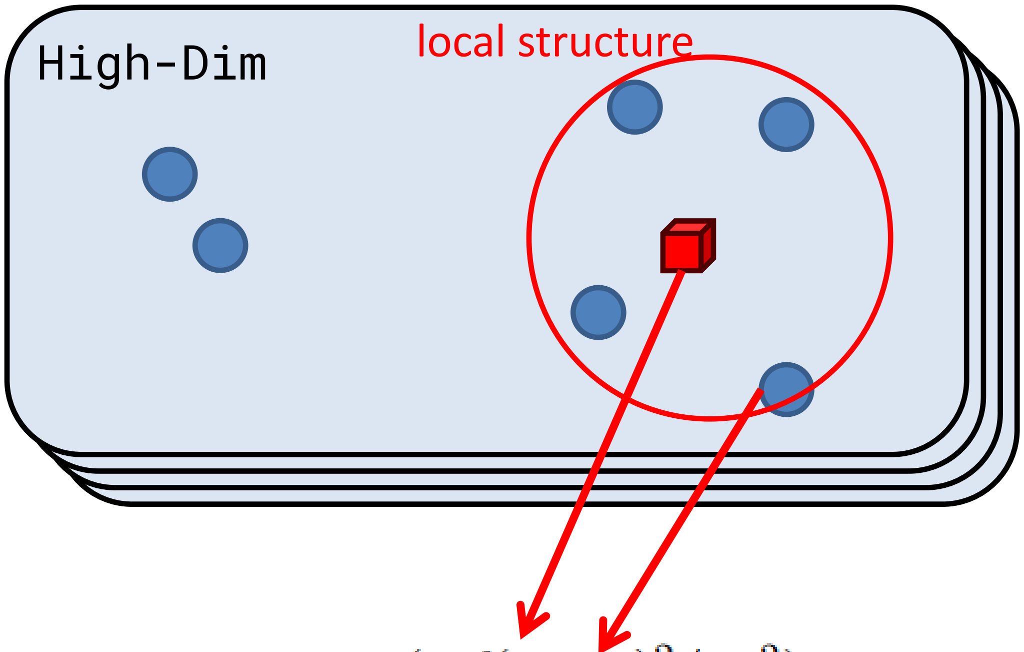
Measure pairwise similarities between high-dimension points



$$p_{ij} = \frac{\exp\left(-\|x_i - x_j\|^2 / 2\sigma^2\right)}{\sum_{k \neq l} \exp\left(-\|x_k - x_l\|^2 / 2\sigma^2\right)}$$

t-Distributed Stochastic Neighbor Embedding

Measure pairwise similarities between high-dimension points

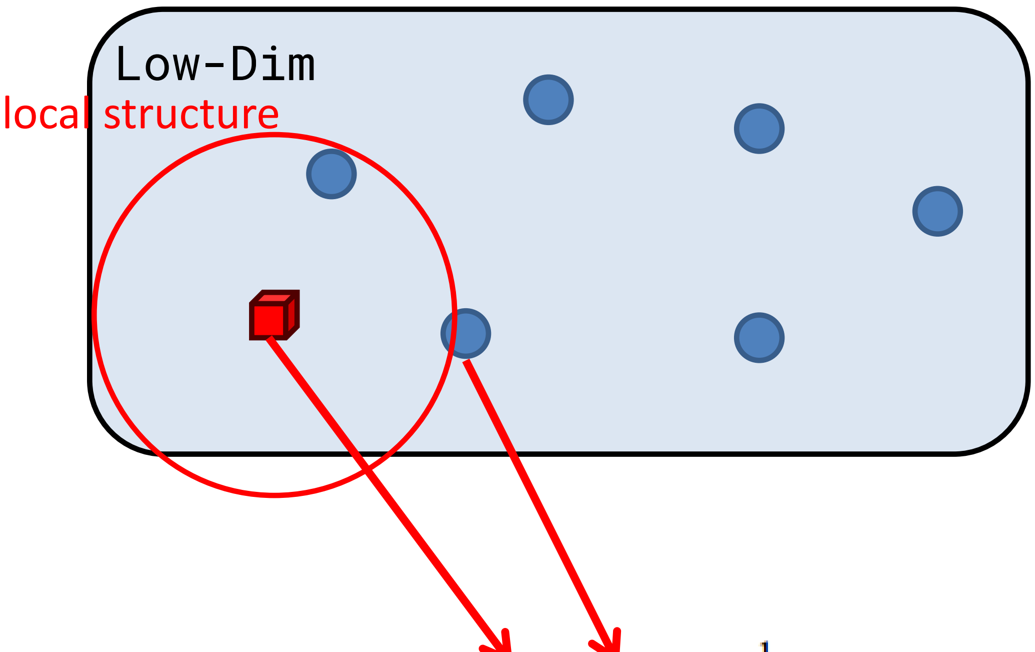


- The bandwidth, σ(i) is scaled by predefined **perplexity** (default=30)

$$p_{j|i} = \frac{\exp(-d(\mathbf{x}_i, \mathbf{x}_j)^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-d(\mathbf{x}_i, \mathbf{x}_k)^2 / 2\sigma_i^2)} \qquad p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N}$$

t-Distributed Stochastic Neighbor Embedding

Measure pairwise similarities between low-dimension map points

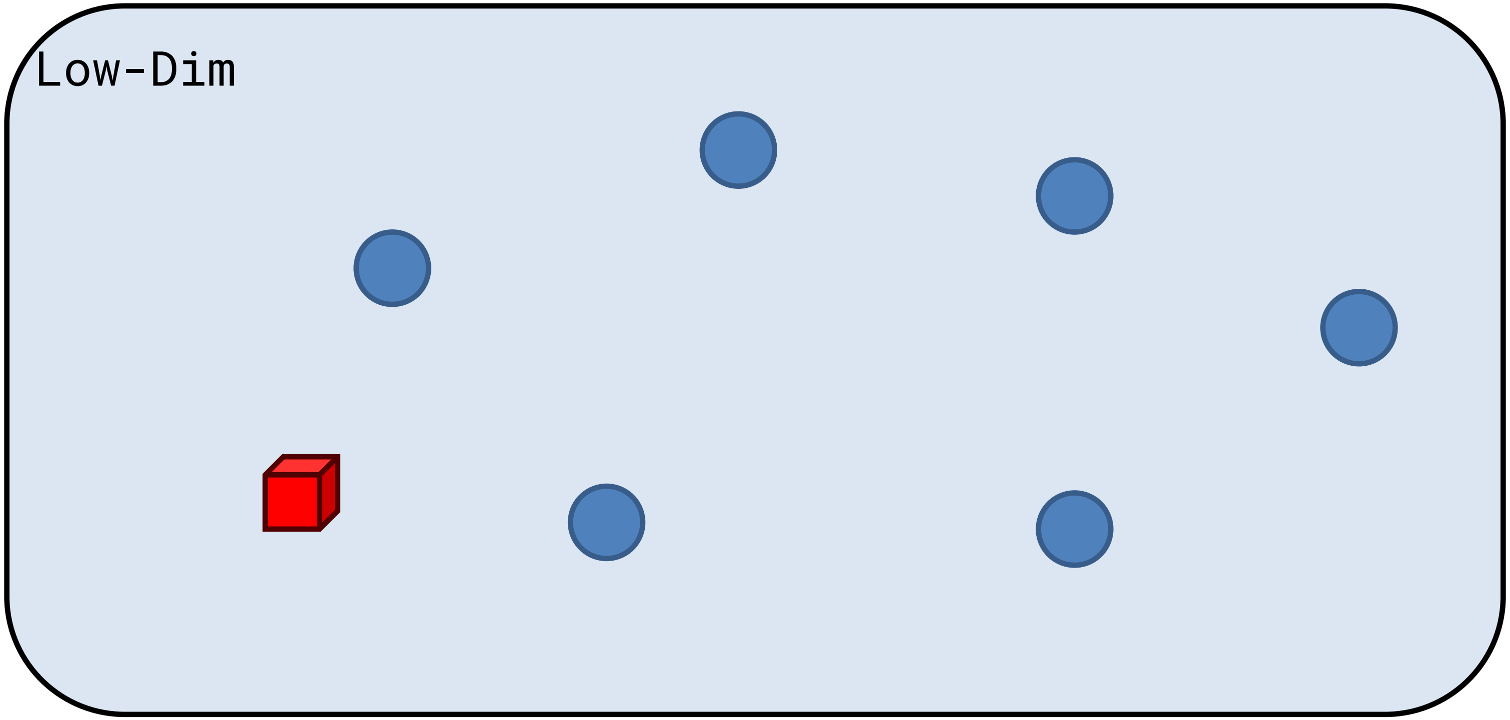


- Points of low-dim local structure are from high-dim local structure

$$q_{ij} = \frac{\left(1 + \|y_i - y_j\|^2\right)^{-1}}{\sum_{k \neq l} \left(1 + \|y_k - y_l\|^2\right)^{-1}}$$

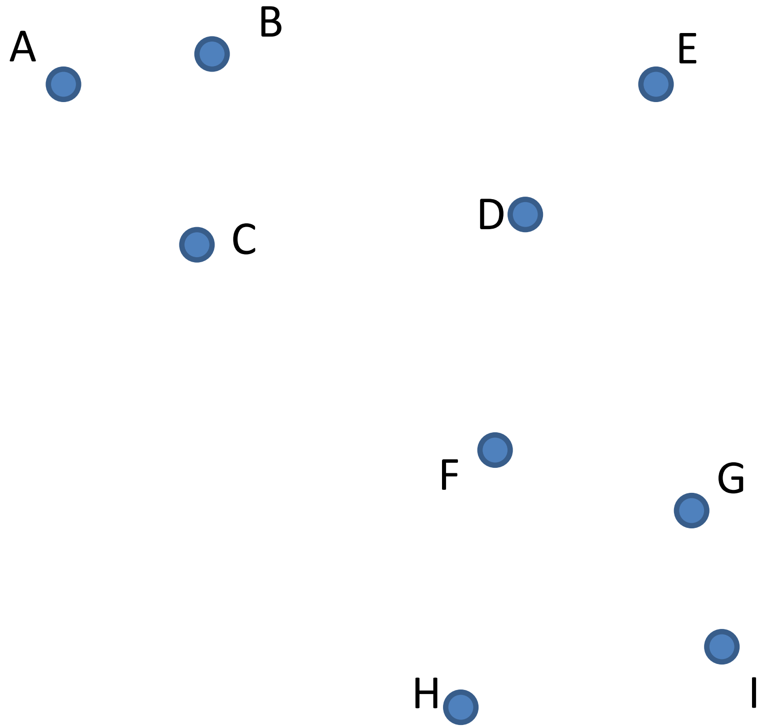
t-Distributed Stochastic Neighbor Embedding

Move points around to minimize: $C = KL(P||Q) = \sum_i \sum_j p_{ij} \log \frac{p_{ij}}{q_{ij}}$:



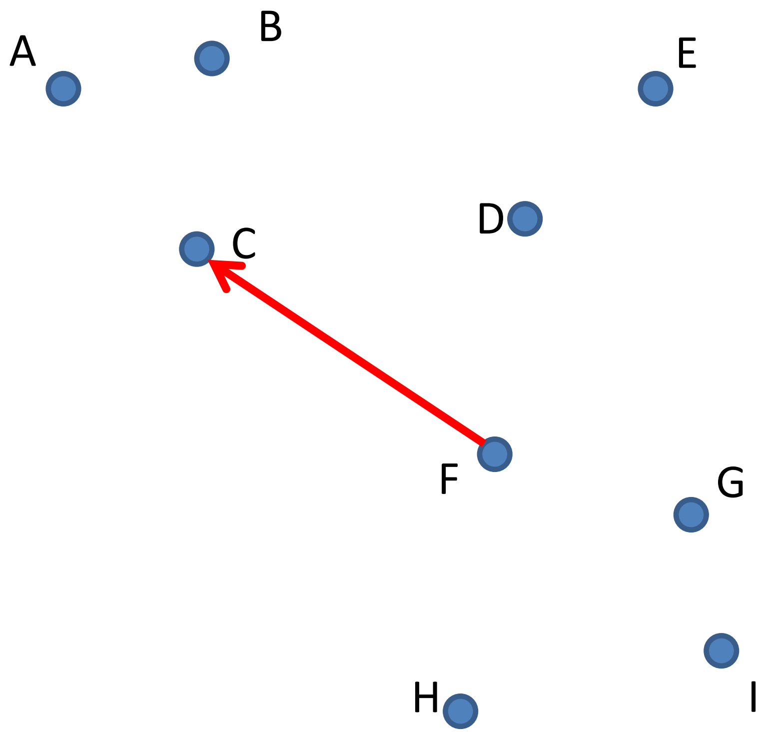
Gradient interpretation

$$\frac{\delta C}{\delta y_i} = 4\sum_j (p_{ij} - q_{ij})(y_i - y_j)\left(1 + \|y_i - y_j\|^2\right)^{-1}$$



Gradient interpretation

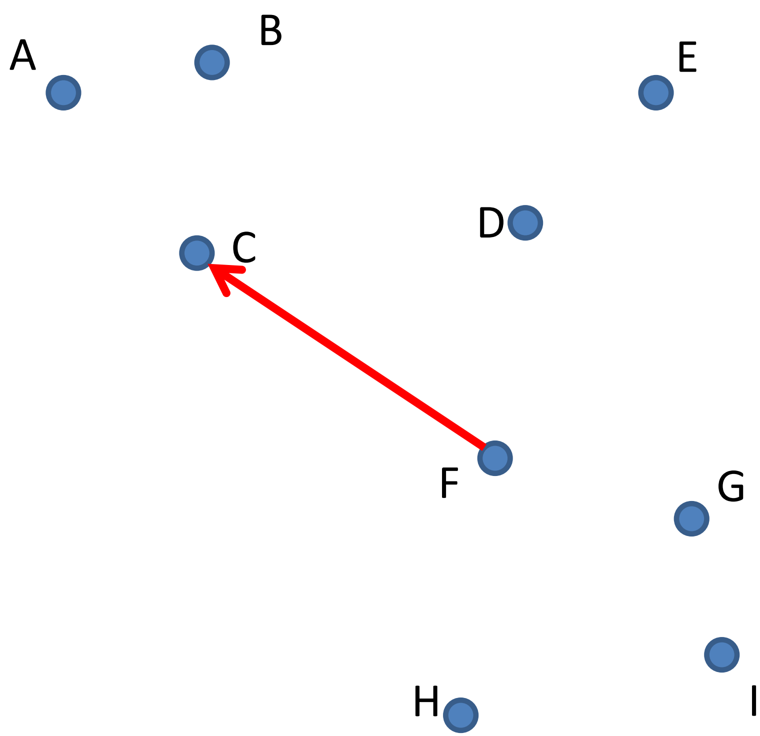
$$\frac{\delta C}{\delta y_i} = 4\sum_j (p_{ij} - q_{ij})\overset{\textbf{spring}}{\boxed{(y_i - y_j)}}\left(1 + \|y_i - y_j\|^2\right)^{-1}$$



Gradient interpretation

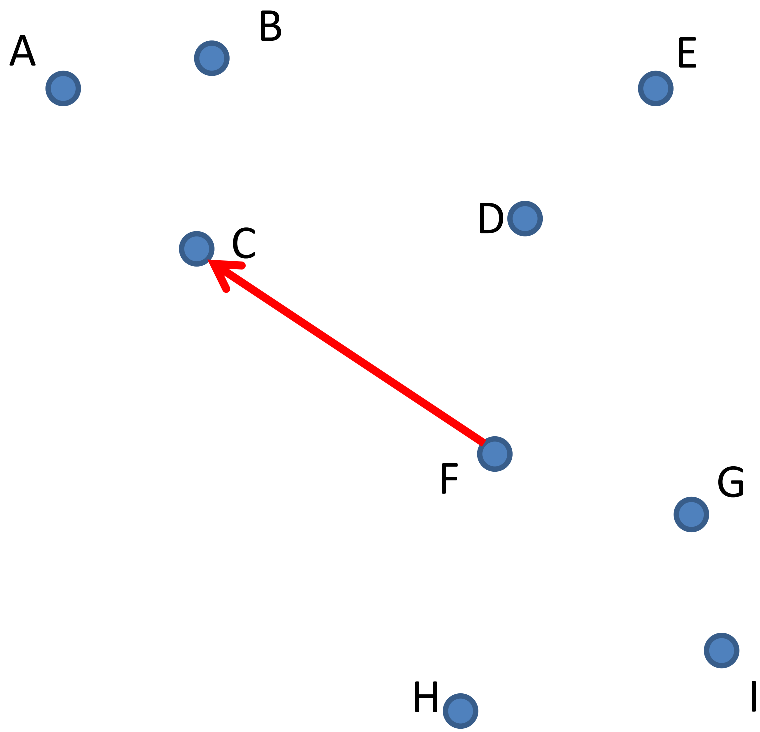
Normalization term

$$\frac{\delta C}{\delta y_i} = 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j) \boxed{\left(1 + \|y_i - y_j\|^2\right)^{-1}}$$



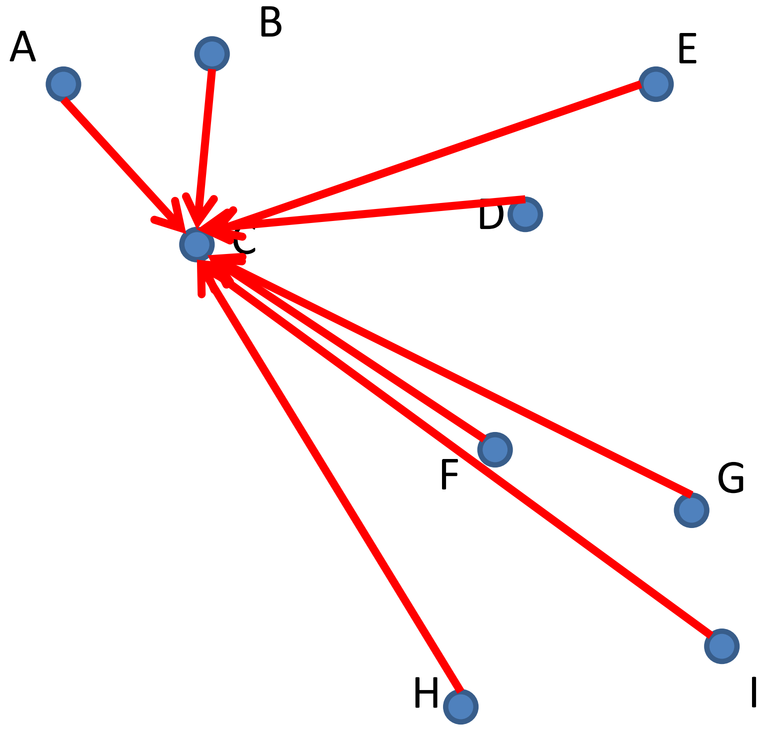
Gradient interpretation

$$\frac{\delta C}{\delta y_i} = 4 \sum_j \overset{\textbf{scale}}{\boxed{(p_{ij} - q_{ij})}} (y_i - y_j) \left(1 + \|y_i - y_j\|^2\right)^{-1}$$



Gradient interpretation

$$\frac{\delta C}{\delta y_i} = 4 \boxed{\sum_j} (p_{ij} - q_{ij})(y_i - y_j)\left(1 + \|y_i - y_j\|^2\right)^{-1}$$



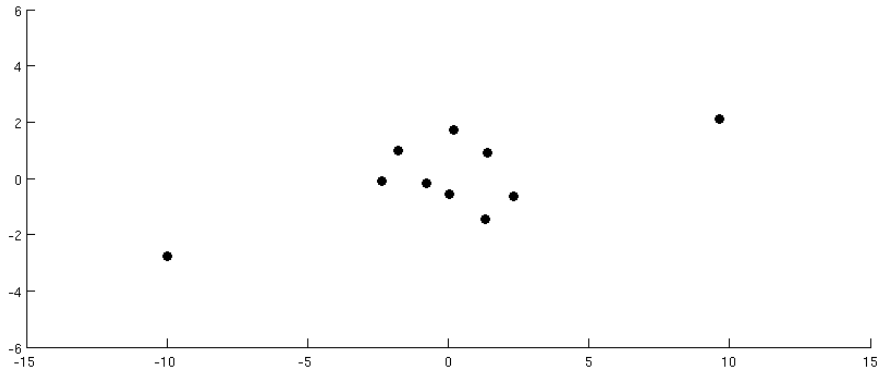
Simple version of t-Distributed Stochastic Neighbor Embedding

Data: data set $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$,
cost function parameters: perplexity $Perp$,
optimization parameters: number of iterations T, learning rate η, momentum $\alpha(t)$.
Result: low-dimensional data representation $\mathcal{Y}^{(T)} = \{y_1, y_2, \ldots, y_n\}$.
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 $\mathcal{Y}^{(0)} = \{y_1, y_2, \ldots, y_n\}$ 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

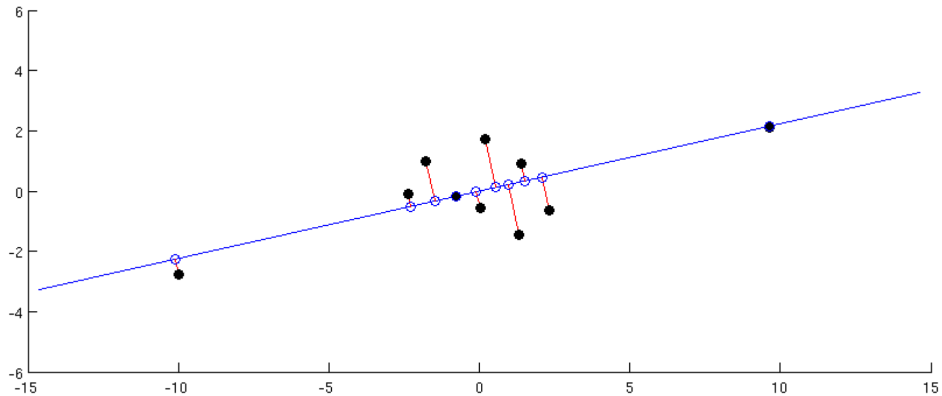
Simple version of t-Distributed Stochastic Neighbor Embedding

Data: data set $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$,
cost function parameters: perplexity $Perp$,
optimization parameters: number of iterations T, learning rate η, momentum $\alpha(t)$.
Result: low-dimensional data representation $\mathcal{Y}^{(T)} = \{y_1, y_2, \ldots, y_n\}$.
begin
O(n^2) 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 $\mathcal{Y}^{(0)} = \{y_1, y_2, \ldots, y_n\}$ from $\mathcal{N}(0, 10^{-4}I)$
for t=*1* **to** T **do**
O(n^2) compute low-dimensional affinities q_{ij} (using Equation 4)
O(n^2) 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

Use PCA to generate initial solution

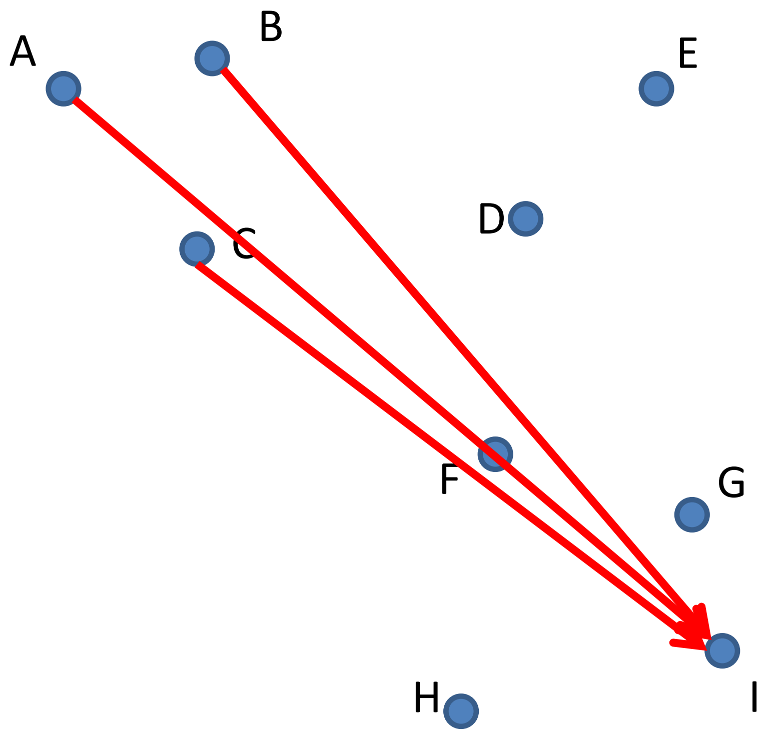


Use PCA to preserve global structure

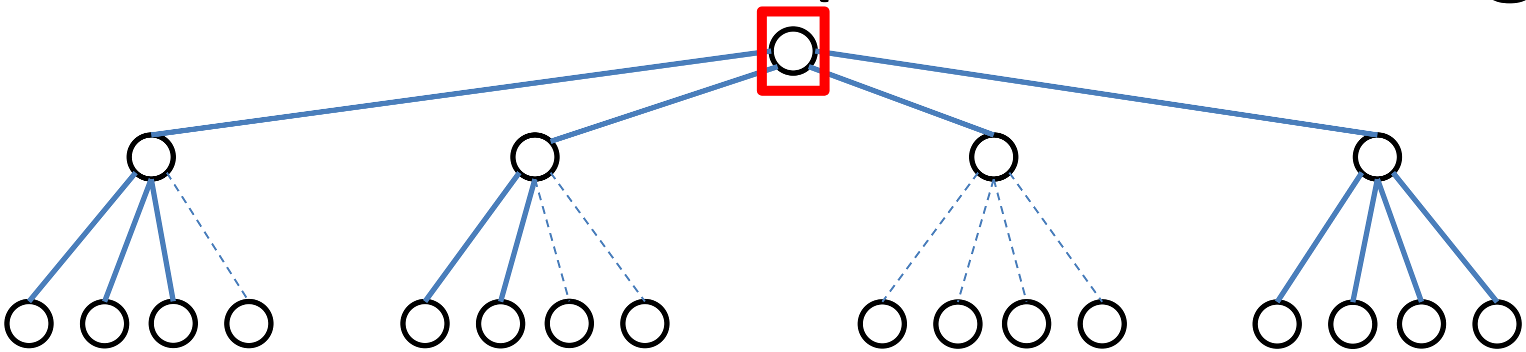


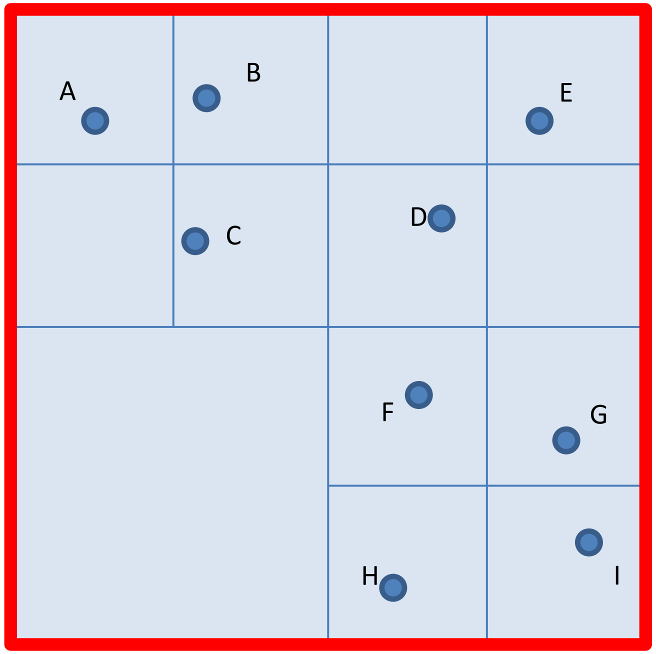
Barnes-Hut-SNE (Tree-based Alg)

$$\frac{\delta C}{\delta y_i} = 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j)\left(1 + \|y_i - y_j\|^2\right)^{-1}$$

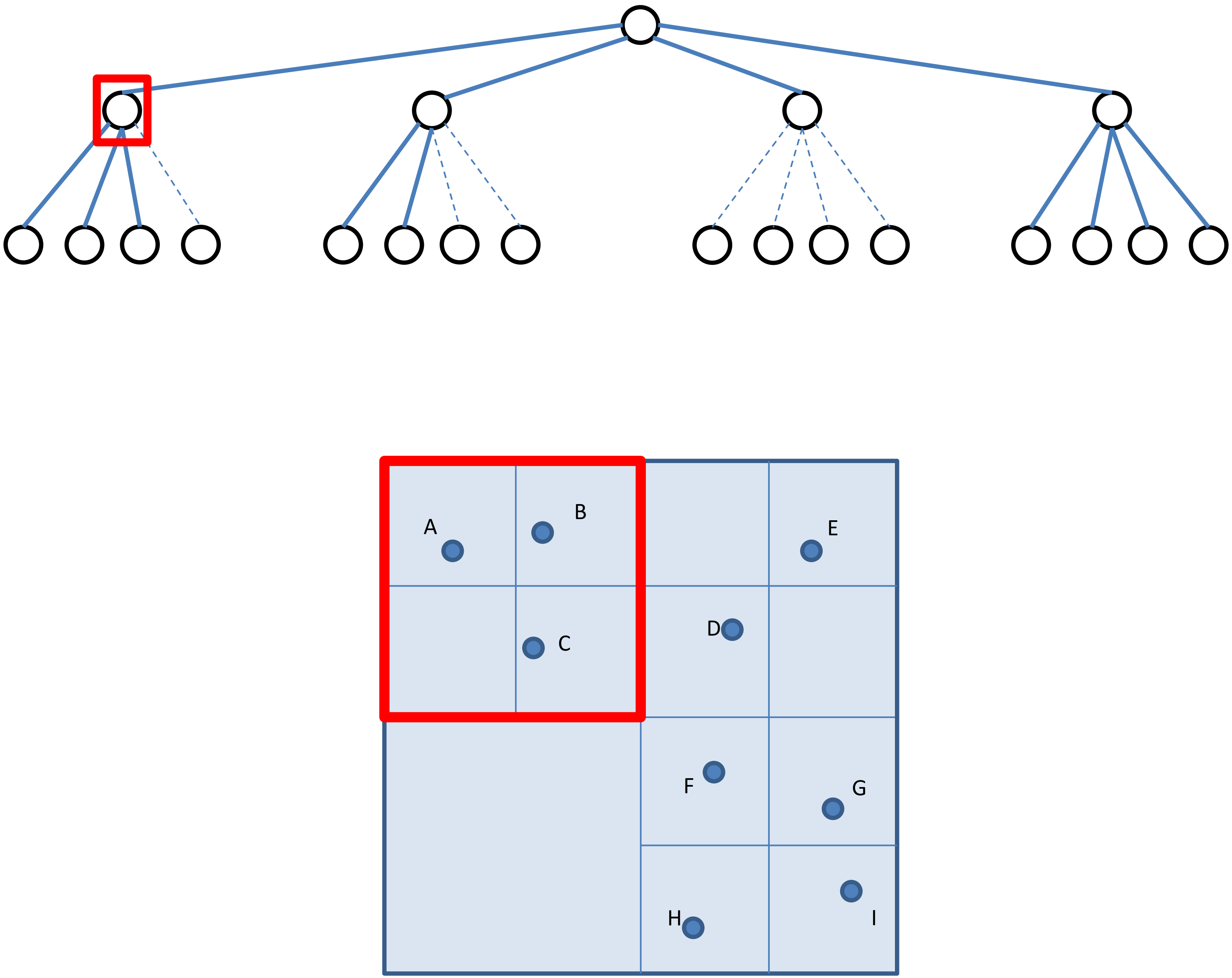


Barnes-Hut-SNE (Tree-based Alg)

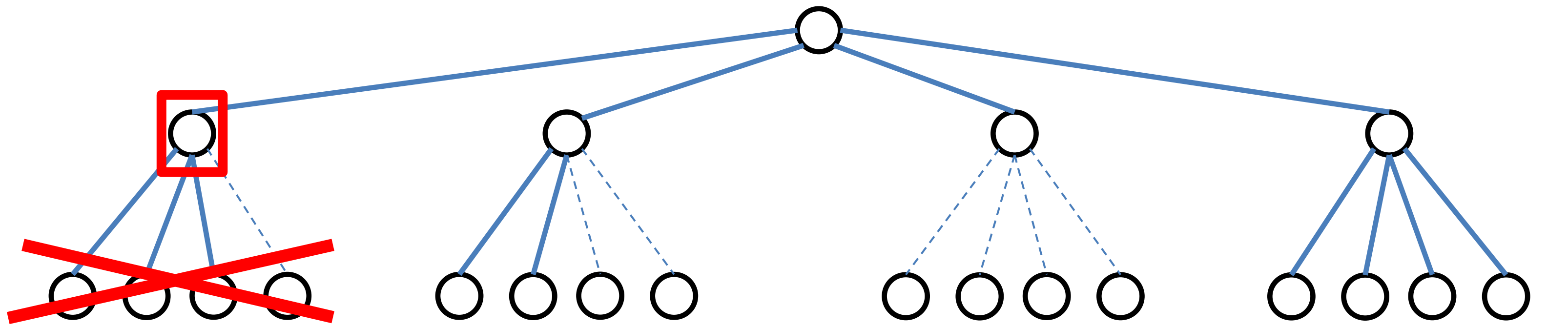


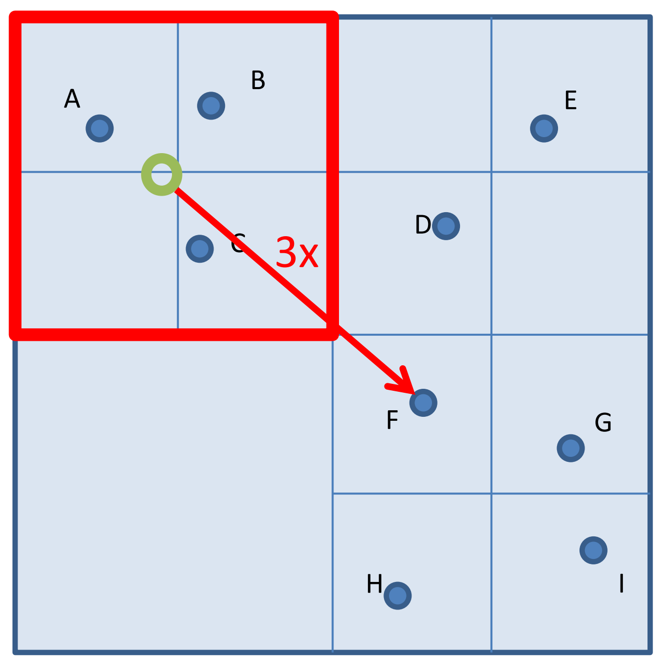


Barnes-Hut-SNE (Tree-based Alg)

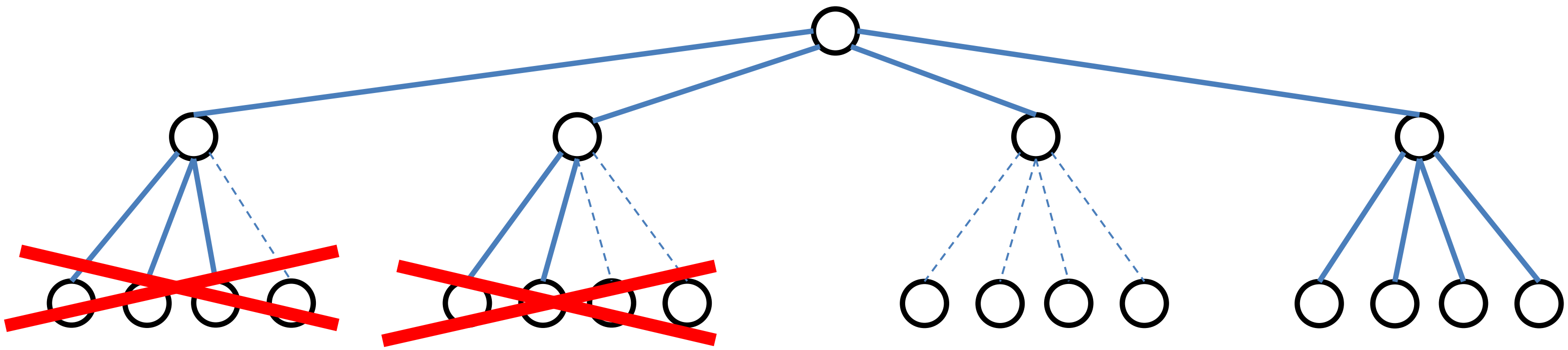


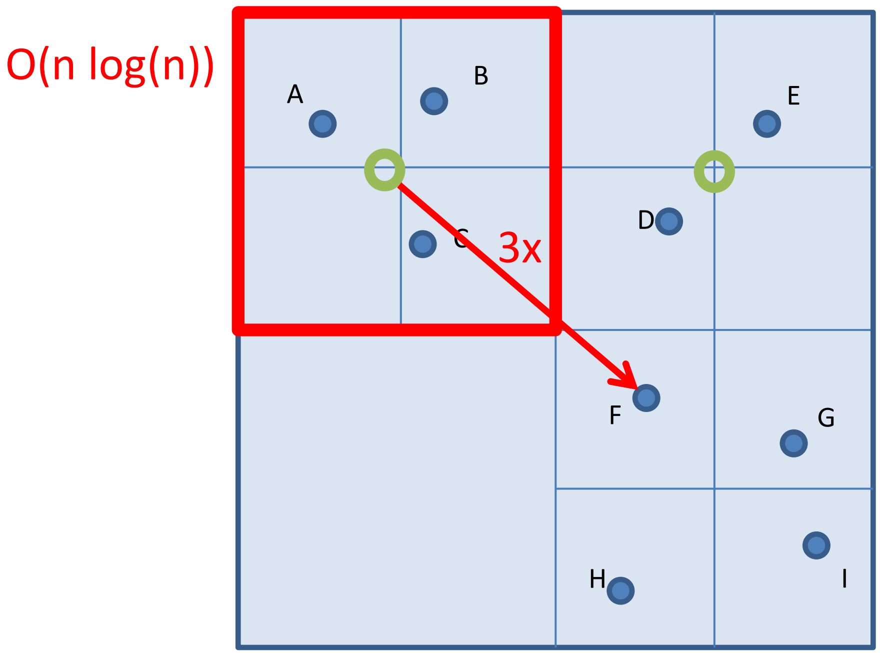
Barnes-Hut-SNE (Tree-based Alg)



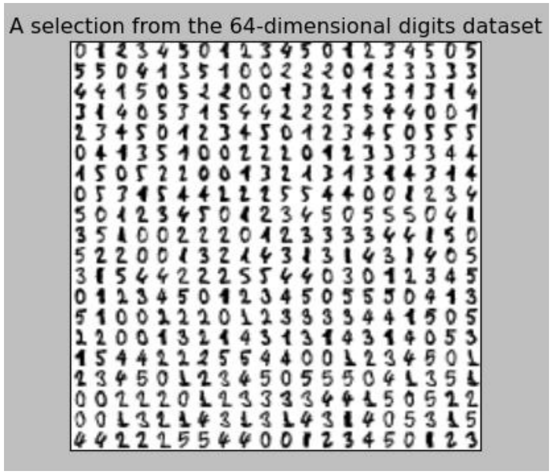


Barnes-Hut-SNE (Tree-based Alg)



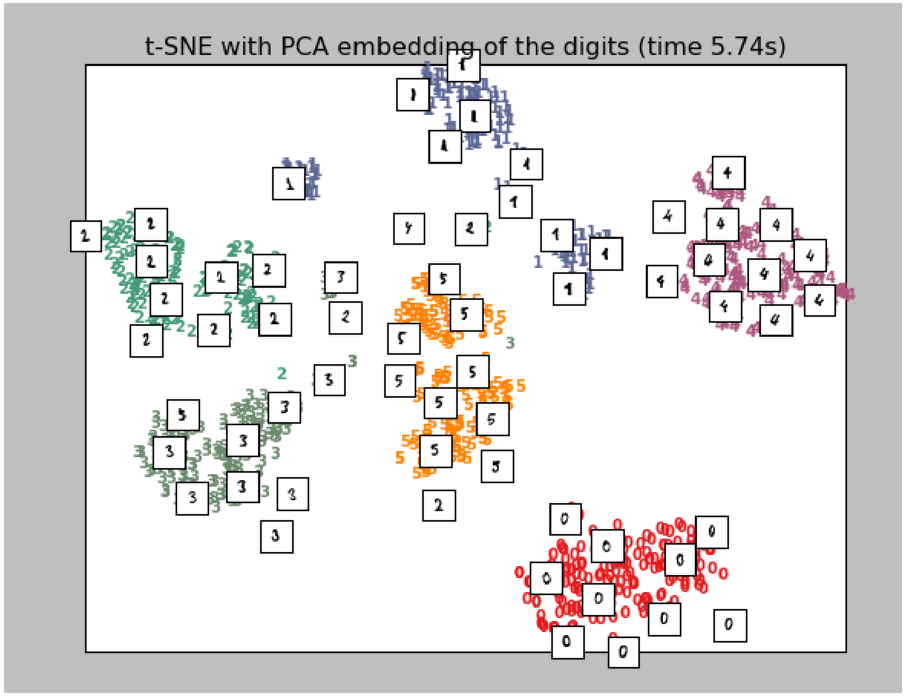


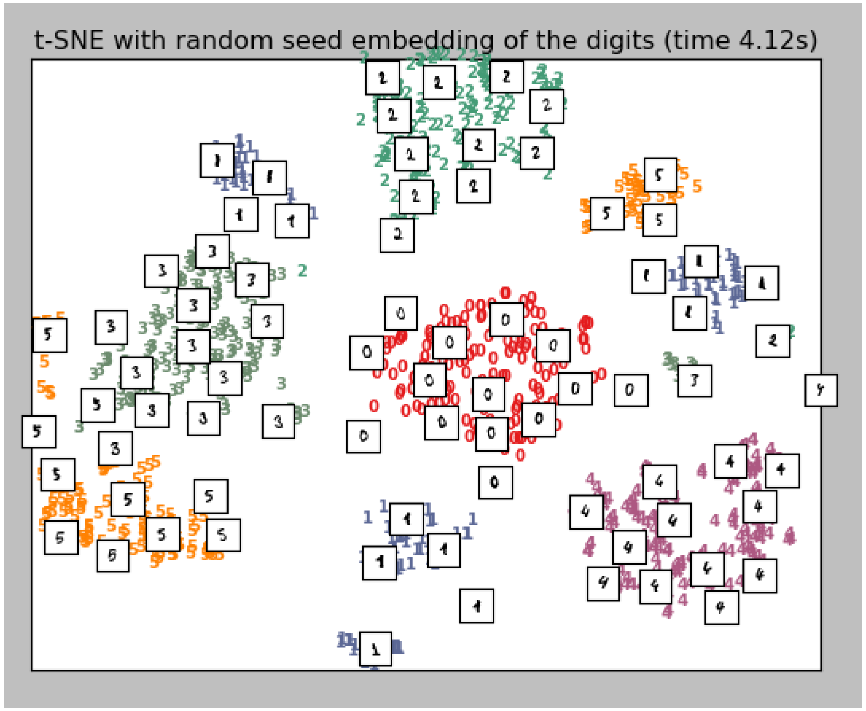
Experiments



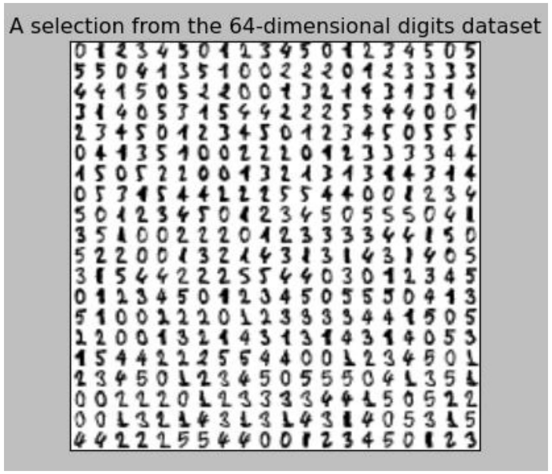
Each datapoint is a 8x8 image of a digit.

Classes	10
Samples per class	~180
Samples total	1797
Dimensionality	64
Features	integers 0-16



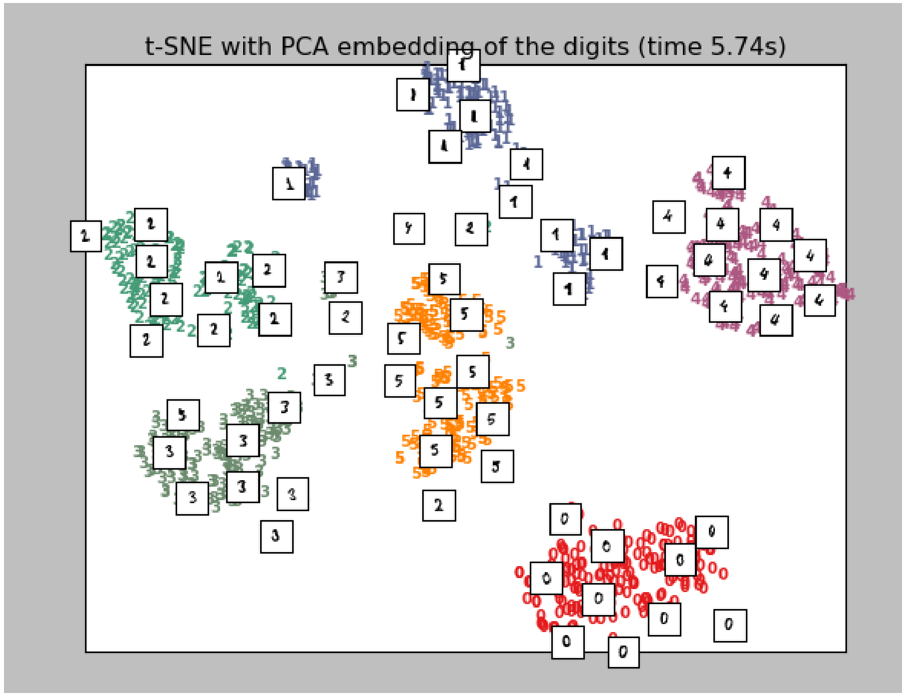


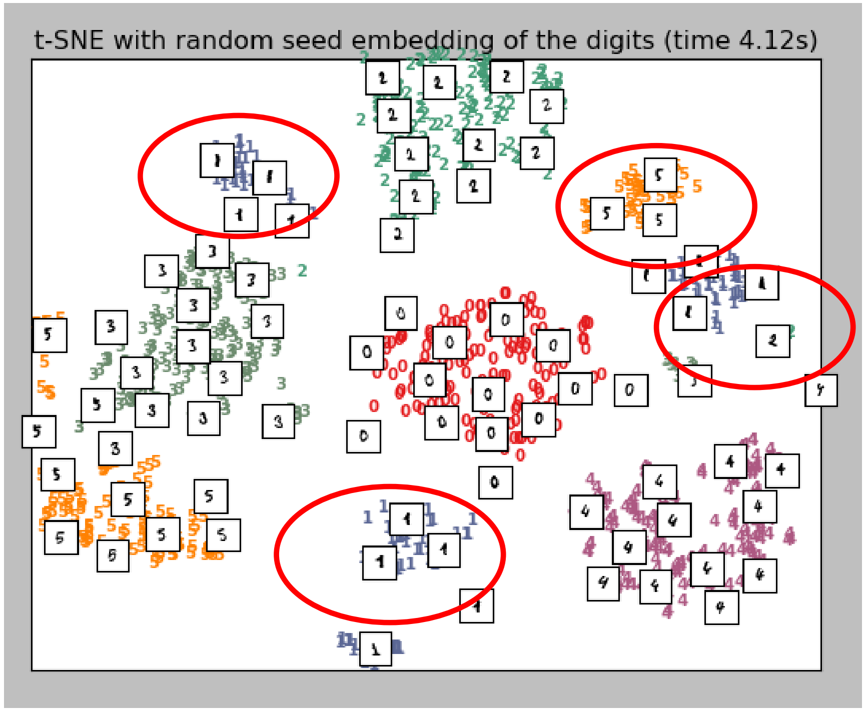
Experiments



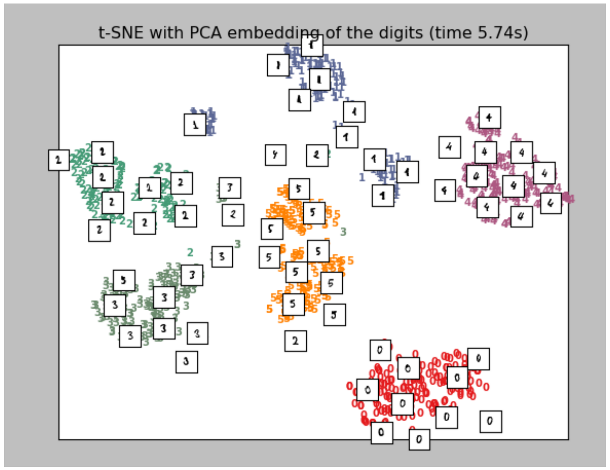
Each datapoint is a 8x8 image of a digit.

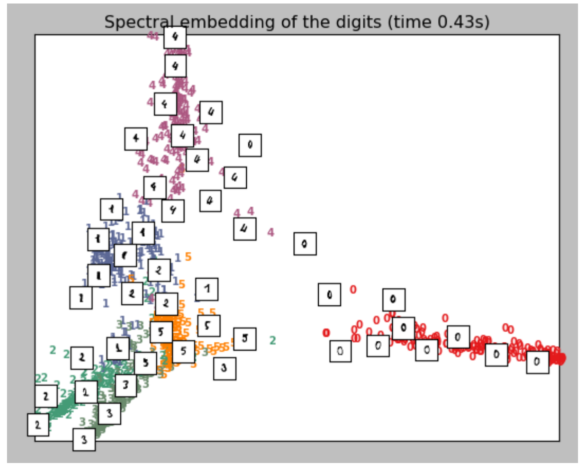
Classes	10
Samples per class	~180
Samples total	1797
Dimensionality	64
Features	integers 0-16

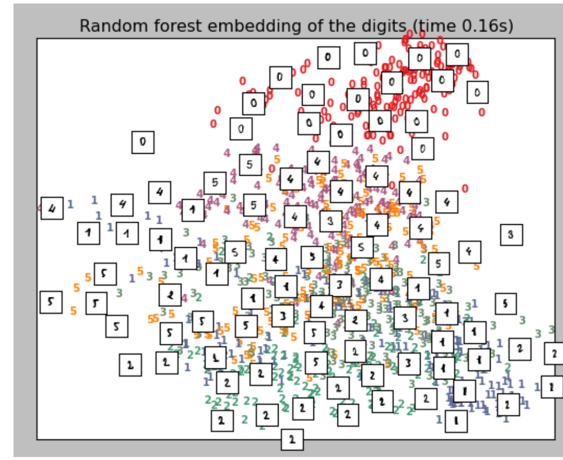


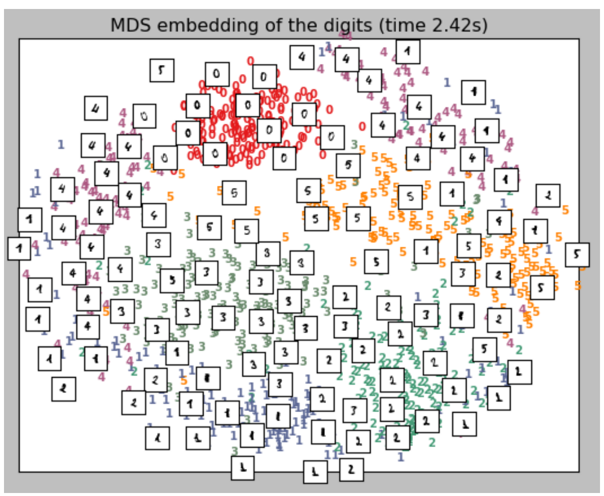


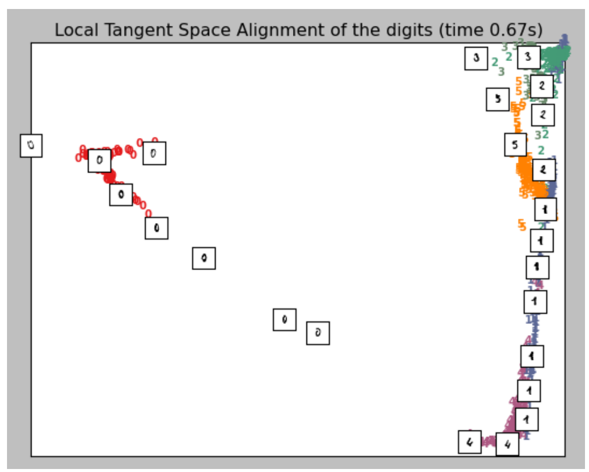
Experiments

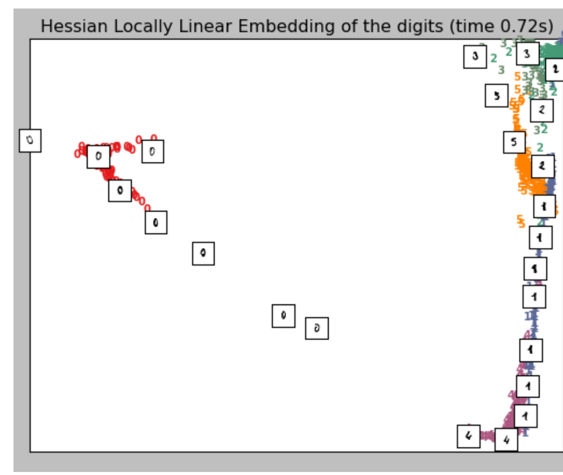




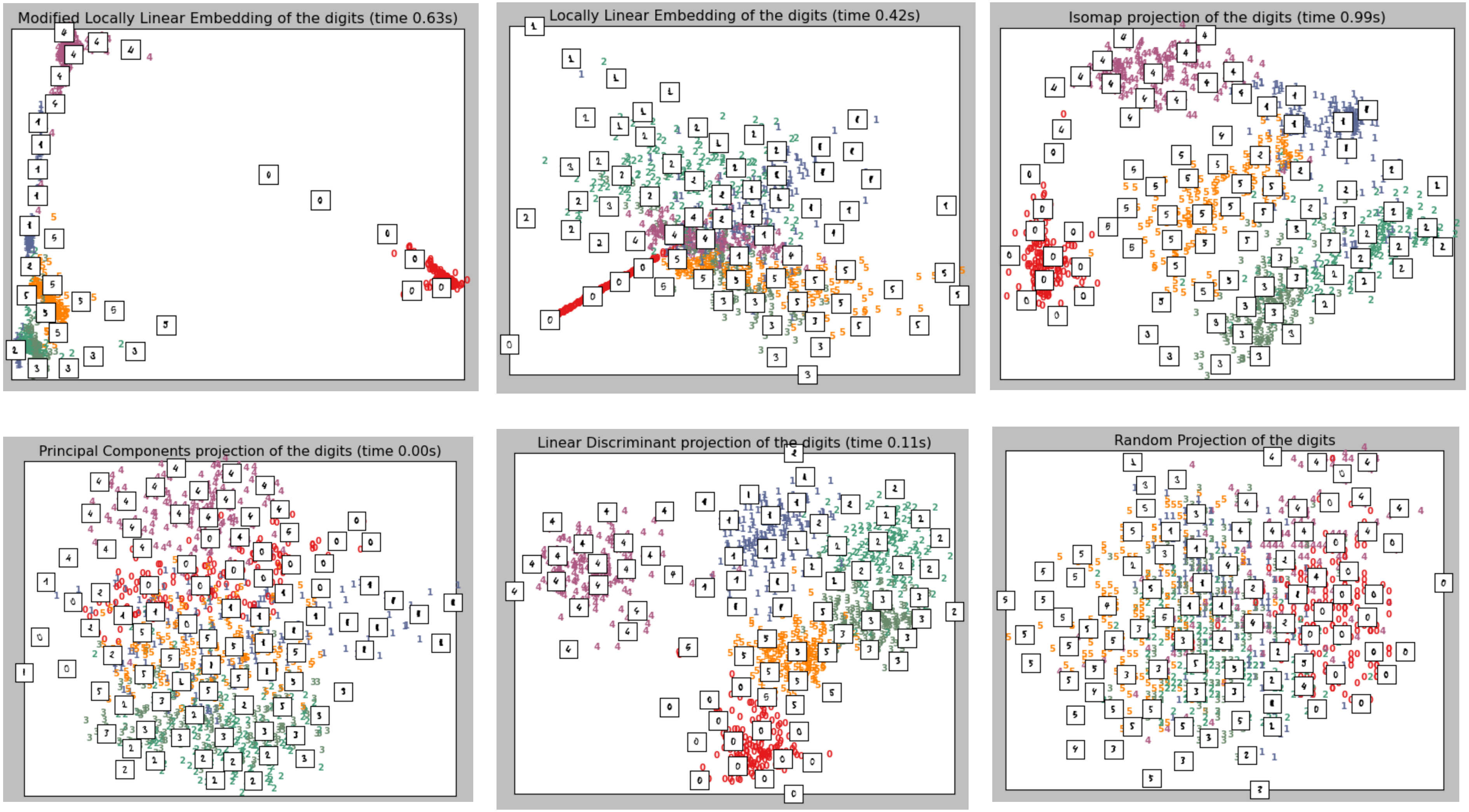




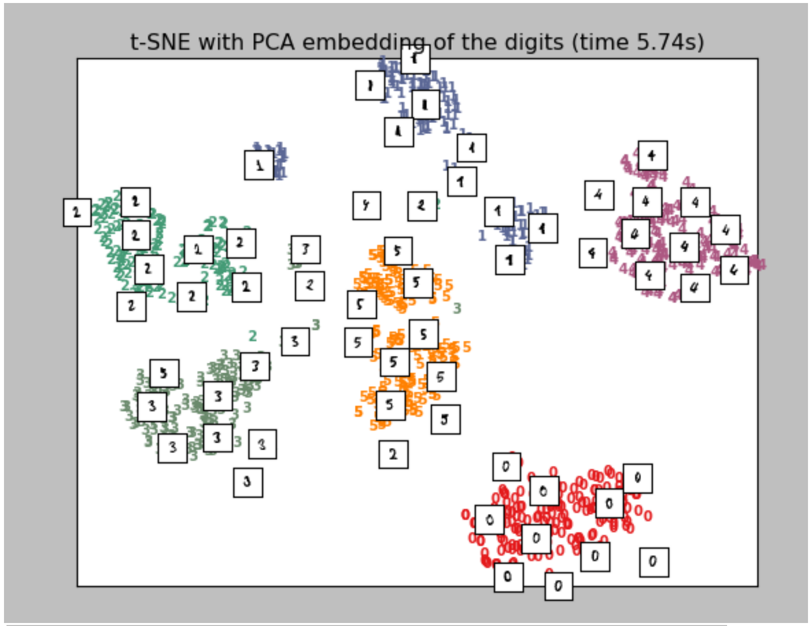


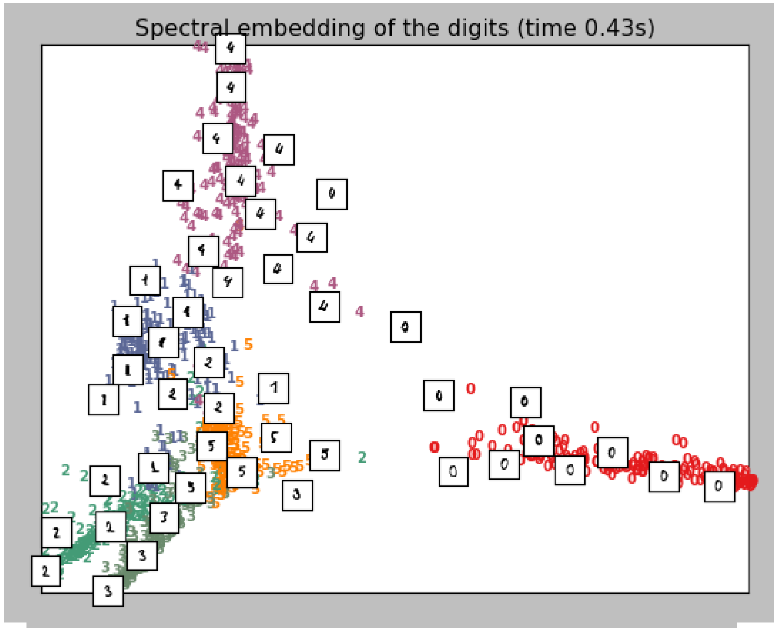


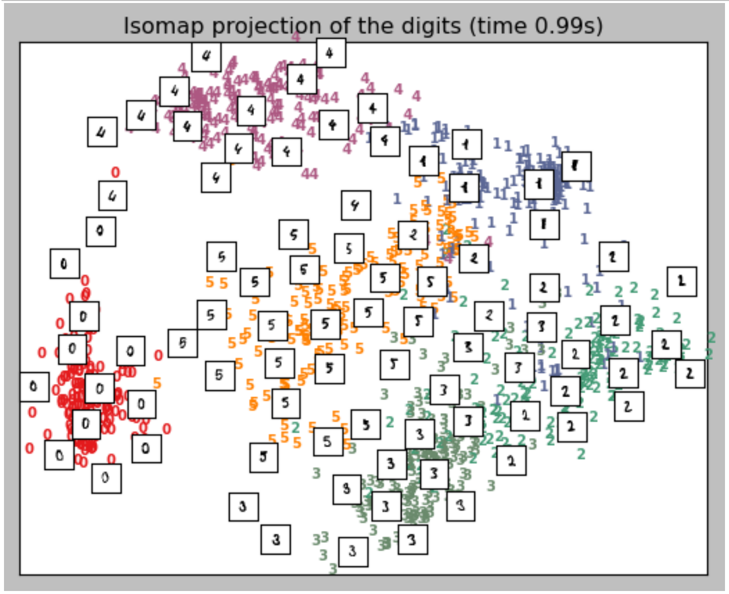
Experiments

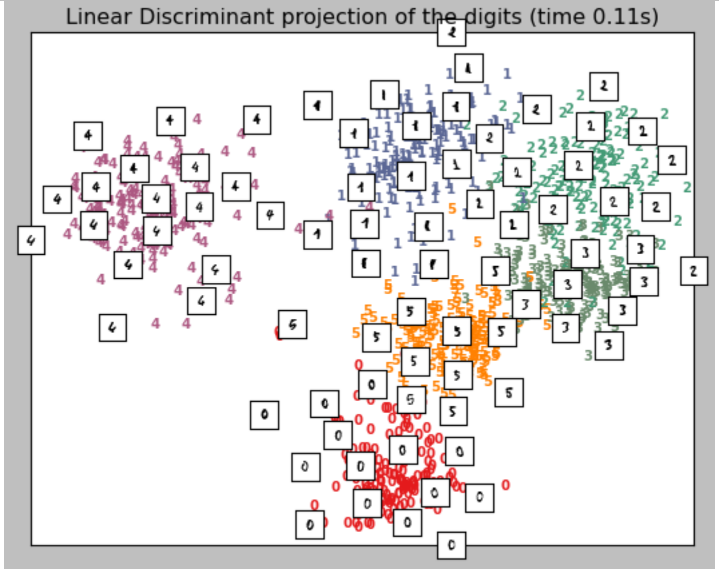


Experiments



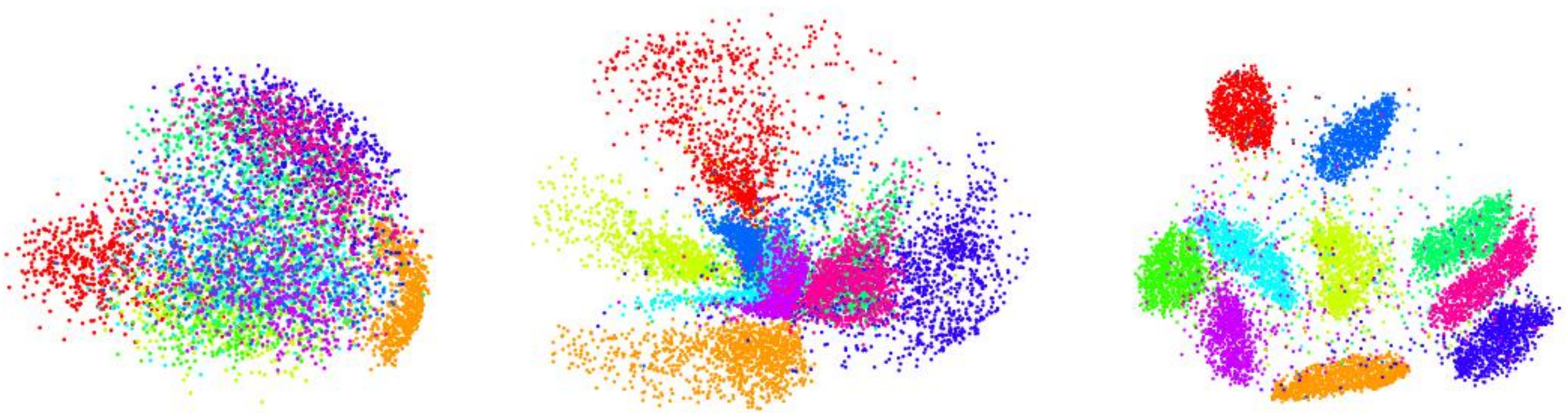






Experiments

handwritten digits	
sample points	10,000 / 70,000
dimensions	784
classes	10 (0~9)



(a) Visualization by PCA. (b) Visualization by an autoencoder. (c) Visualization by parametric t-SNE.

Figure 2: Visualizations of 10,000 digits from the MNIST dataset by parametric dimensionality reduction techniques.

Experiments

	MNIST			**Characters**			**20 Newsgroups**		
	2D	10D	30D	2D	10D	30D	2D	10D	30D
PCA	78.16%	43.03%	10.78%	86.72%	60.73%	20.50%	35.99%	27.05%	28.82%
NCA	56.84%	8.84%	7.32%	72.90%	24.68%	17.95%	**30.76%**	26.65%	26.09%
Autoencoder	66.84%	6.33%	**2.70%**	82.93%	17.91%	**11.11%**	37.60%	29.15%	27.62%
Par. t-SNE, $\alpha = 1$	**9.90%**	5.38%	5.41%	**43.90%**	26.01%	23.98%	34.30%	**24.40%**	24.88%
Par. t-SNE, $\alpha = d - 1$	**9.90%**	**4.58%**	2.76%	**43.90%**	**17.13%**	13.55%	35.10%	25.28%	**23.75%**
Par. t-SNE, learned α	12.68%	4.85%	**2.70%**	44.78%	17.30%	14.31%	33.82%	27.21%	24.72%

Table 1: Generalization errors of 1-nearest neighbor classifiers on low-dimensional representations of the MNIST dataset, the characters dataset, and the 20 newsgroups dataset.

Summary

- The disadvantages to using t-SNE are roughly:
  - t-SNE is **computationally expensive**, and can take several hours on million-sample datasets where PCA will finish in seconds or minutes
  - The Barnes-Hut t-SNE method is **limited to two or three** dimensional embeddings.
  - The algorithm is stochastic and multiple restarts with different seeds can yield different embeddings. However, it is perfectly legitimate to pick the the embedding with the least error.
  - **Global structure is not explicitly preserved**. This is problem is mitigated by initializing points with PCA (using init='pca').

References

1. L.J.P. van der Maaten. **Accelerating t-SNE using Tree-Based Algorithms**. *Journal of Machine Learning Research* 15(Oct):3221-3245, 2014.
2. L.J.P. van der Maaten and G.E. Hinton. **Visualizing Non-Metric Similarities in Multiple Maps**. *Machine Learning* 87(1):33-55, 2012.
3. L.J.P. van der Maaten. **Learning a Parametric Embedding by Preserving Local Structure**. In *Proceedings of the Twelfth International Conference on Artificial Intelligence & Statistics (AI-STATS), JMLR W&CP* 5:384-391, 2009.
4. L.J.P. van der Maaten and G.E. Hinton. **Visualizing High-Dimensional Data Using t-SNE**. *Journal of Machine Learning Research* 9(Nov):2579-2605, 2008.
5. G.E. Hinton and S.T. Roweis. Stochastic Neighbor Embedding. In Advances in Neural Information
6. Processing Systems, volume 15, pages 833–840, Cambridge, MA, USA, 2002. The MIT Press.
7. https://github.com/scikit-learn/scikit-learn
8. http://scikit-learn.org/stable/index.html