Machine learning for visualization

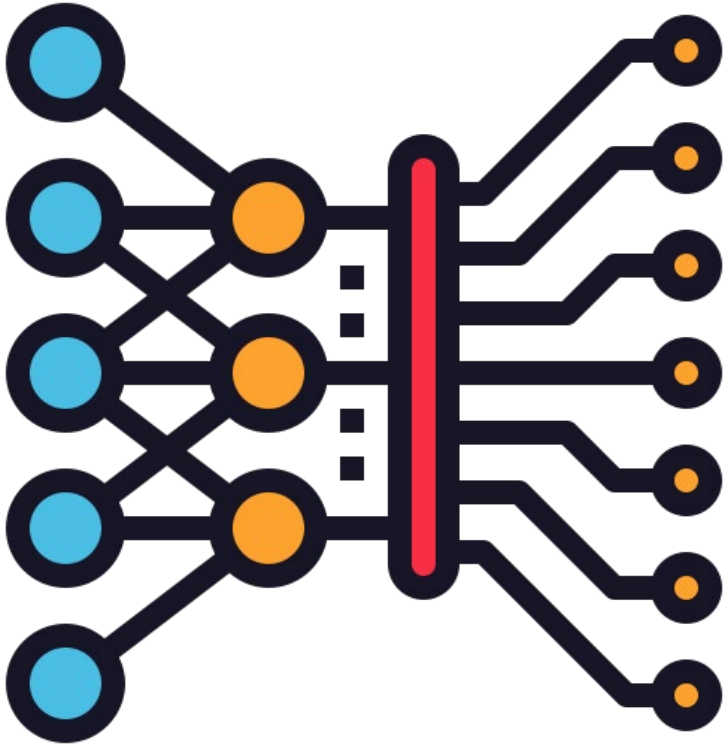
CS424: Visualization & Visual Analytics

Fabio Miranda

https://fmiranda.me

Slides based on Claudio Silva ml+vis course

Machine learning is pervasive.



Machine learning in many domains

Banks are using tabular data to identify fraud or calculate loan risk.

Machine translation has improved remarkably over the past few years.

Healthcare providers look to interpretable models to guide decisions.

Understanding the gap

- Model understanding has yet to catch up to model performance improvements.
- Visualization is key to improve model understanding to a wide range of audiences.
- There are many rich domains to apply VisML techniques, including industry and research.

Where does visualization fit in?

Many tasks in machine learning inevitably involve some form of visualization.

In other cases, visualization is a byproduct of the process.

We have also seen an increase in the number of interactive visualization systems, designed with machine learning as a critical component.

Visual Analytics in Deep Learning | Interrogative Survey Overview

§4 WHY

Why would one want to use visualization in deep learning?

Interpretability & Explainability
Debugging & Improving Models
Comparing & Selecting Models
Teaching Deep Learning Concepts

§6 WHAT

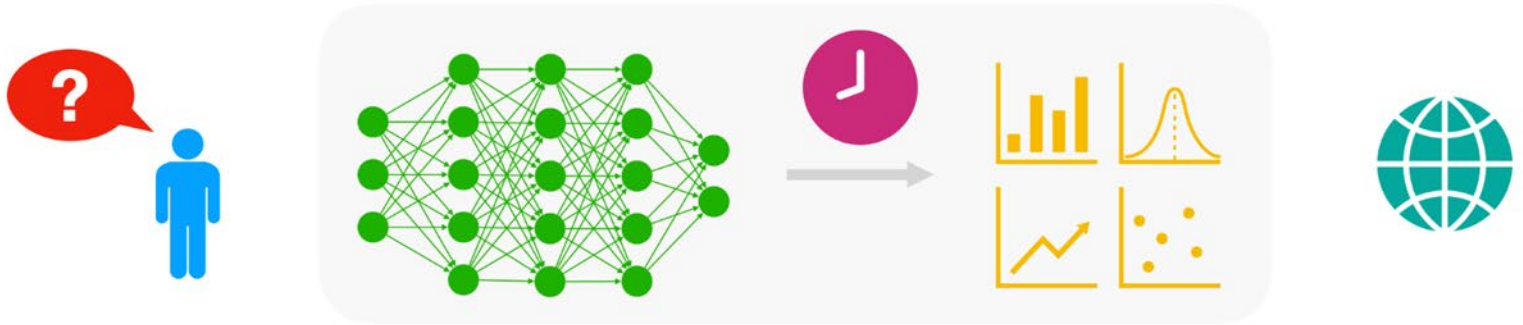
What data, features, and relationships in deep learning can be visualized?

Computational Graph & Network Architecture
Learned Model Parameters
Individual Computational Units
Neurons In High-dimensional Space
Aggregated Information

§8 WHEN

When in the deep learning process is visualization used?

During Training
After Training



§5 WHO

Who would use and benefit from visualizing deep learning?

Model Developers & Builders
Model Users
Non-experts

§7 HOW

How can we visualize deep learning data, features, and relationships?

Node-link Diagrams for Network Architecture
Dimensionality Reduction & Scatter Plots
Line Charts for Temporal Metrics
Instance-based Analysis & Exploration
Interactive Experimentation
Algorithms for Attribution & Feature Visualization

§9 WHERE

Where has deep learning visualization been used?

Application Domains & Models
A Vibrant Research Community

[Hohman et al., 2018]

Dimensionality reduction

- Input data may have thousands or millions of dimensions
  - E.g., text, images, videos, …
- **Dimensionality reduction** aims to represent data with fewer dimensions.
  - Easier learning: fewer parameters.
  - Discover “intrinsic dimensionality” of data.
    - High dimensional data that is truly lower dimensional.
    - Noise reduction.
  - Visualization: show high-dimensional data in a visual space (2D or 3D).
    - How are the points spread?
    - Are there well defined groups of similar instances?

Technique	Taxonomy												
	Data Types					Linearity	Supervision	Multi-level	Locality	Steerability	Stability	OOC	Comp.
	Di	Or	Ca	Ne	Ct								
PCA [65]			✔			✔					•	✔	$O(p^3)$
LDA [50]			✔			✔	✔					✔	$O(np^2+p^3)$
Classical MDS [155]	✔		✔	•		✔	•			•	•		$O(n^3)$
Kruskal [79]	✔	✔	✔	•			•			•		•	$O(in^2)$
NLM [132]	✔		✔	•			•			•		•	$O(in^2)$
MCA [17]			•		✔								$O(n^3)$
Smacof [42]	✔	✔	✔	•			•					•	$O(in^2)$
SOM [126]			✔									✔	$O(l^2np+l^2)$
FastMap [48]	✔		✔	•			•					•	$O(n)$
Chalmers [32]	✔		✔	•			•	•		•	○	✔	$O(in)$
GTM [22]			✔									•	$O((lnp)^3+m^3)$
Pekalska [120]	✔		✔	•		✔	•						$O(r^3+rn)$
CCA [43]	✔		✔	•			•					✔	$O(l^2)$
LLE [129]	✔		✔	•			•		✔			•	$O(n^3)$
Isomap [153]	✔		✔	✔			•					•	$O(n^3)$
Lapl. Eigenmaps [15]	✔		✔	✔			•		✔			•	$O(n^3)$
Force-Directed [152]	✔		✔				•	•				✔	$O(in^2)$
LTSA [180]			✔			•			✔				$O(n^3)$
MVU [169]	✔		✔	•			•		✔			•	$O(n^3)$
LSP [117]	✔		✔	✔			•	•	✔	✔	○		$O(n^3)$
SNE [64]	✔		✔	•			•	•		○		•	$O(in^2)$
PLMP [118]			✔			•	•	•	✔	✔	○		$O(n^3)$
LAMP [73]			✔				•	•	✔	✔	✔	✔	$O(pn)$
RBF-MP [2]	✔		✔	•			•			✔	○	•	$O(r^3+n)$
LoCH [47]	✔		✔	•			•		✔	✔		✔	$O(n\sqrt{(n)})$
ClassiMap [90]	✔		✔	•			✔					•	$O(in^2)$
Kelp [13]	✔		✔	•		•	•		✔	✔	○	•	$O(r^3)$

Technique	Taxonomy												
	Data Types					Linearity	Supervision	Multi-level	Locality	Steerability	Stability	OOC	Comp.
	Di	Or	Ca	Ne	Ct								
PCA [65]						✔					•	✔	$O(p^3)$
LDA [50]						✔	✔					✔	$O(np^2+p^3)$
Classical MDS [155]	✔		✔	•		✔	•			•	•		$O(n^3)$
Kruskal [79]	✔	✔	✔	•			•			•		•	$O(in^2)$
NLM [132]	✔		✔	•			•			•		•	$O(in^2)$
MCA [17]			•		✔								$O(n^3)$
Smacof [42]	✔	✔	✔	•			•					•	$O(in^2)$
SOM [126]			✔									✔	$O(l^2np+l^2)$
FastMap [48]	✔		✔	•			•					•	$O(n)$
Chalmers [32]	✔		✔	•			•	•		•	○	✔	$O(in)$
GTM [22]			✔									•	$O((lnp)^3+m^3)$
Pekalska [120]	✔		✔	•		✔	•						$O(r^3+rn)$
CCA [43]	✔		✔	•			•					✔	$O(l^2)$
LLE [129]	✔		✔	•			•		✔			•	$O(n^3)$
Isomap [153]	✔		✔	✔			•					•	$O(n^3)$
Lapl. Eigenmaps [15]	✔		✔	✔			•		✔			•	$O(n^3)$
Force-Directed [152]	✔		✔				•	•				✔	$O(in^2)$
LTSA [180]			✔			•			✔				$O(n^3)$
MVU [169]	✔		✔	•			•		✔			•	$O(n^3)$
LSP [117]	✔		✔	✔			•	•	✔	✔	○		$O(n^3)$
SNE [64]	✔		✔	•			•	•		○		•	$O(in^2)$
PLMP [118]			✔			•	•	•	✔	✔	○		$O(n^3)$
LAMP [73]			✔				•	•	✔	✔	✔	✔	$O(pn)$
RBF-MP [2]	✔		✔	•			•			✔	○	•	$O(r^3+n)$
LoCH [47]	✔		✔	•			•		✔	✔		✔	$O(n\sqrt{(n)})$
ClassiMap [90]	✔		✔	•			✔					•	$O(in^2)$
Kelp [13]	✔		✔	•		•	•		✔	✔	○	•	$O(r^3)$

Core Techniques

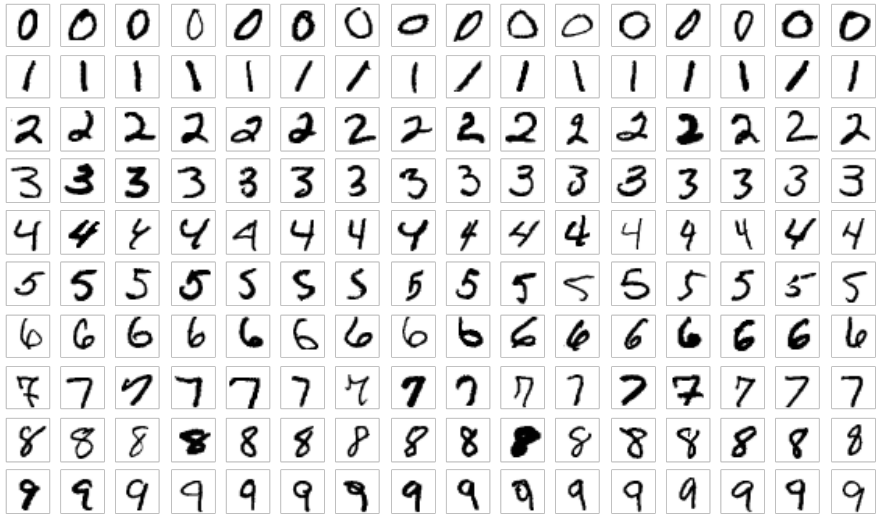
Technique	Taxonomy												
	Data Types					Linearity	Supervision	Multi-level	Locality	Steerability	Stability	OOC	Comp.
	Di	Or	Ca	Ne	Ct								
PCA [65]						✔					•	✔	$O(p^3)$
LDA [50]						✔	✔					✔	$O(np^2+p^3)$
Classical MDS [155]	✔		✔	•		✔	•			•	•		$O(n^3)$
Kruskal [79]	✔	✔	✔	•			•			•		•	$O(in^2)$
NLM [132]	✔		✔	•			•			•		•	$O(in^2)$
MCA [17]			•		✔								$O(n^3)$
Smacof [42]	✔	✔	✔	•			•					•	$O(in^2)$
SOM [126]			✔									✔	$O(l^2np+l^2)$
FastMap [48]	✔		✔	•			•					•	$O(n)$
Chalmers [32]	✔		✔	•									$O(in)$
GTM [22]			✔										$O(lnp)^3+m^3)$
Pekalska [120]	✔		✔	•									$O(r^3+rn)$
CCA [43]	✔		✔	•								✔	$O(l^2)$
LLE [129]	✔		✔	•								•	$O(n^3)$
Isomap [153]	✔		✔	✔								•	$O(n^3)$
Lapl. Eigenmaps [15]	✔		✔	✔								•	$O(n^3)$
Force-Directed [152]	✔		✔									✔	$O(in^2)$
LTSA [180]			✔										$O(n^3)$
MVU [169]	✔		✔	•								•	$O(n^3)$
LSP [117]	✔		✔	✔						✔	○		$O(n^3)$
SNE [64]	✔		✔	•						○		•	$O(in^2)$
PLMP [118]			✔							✔	○		$O(n^3)$
LAMP [73]			✔							✔	✔	✔	$O(pn)$
RBF-MP [2]	✔		✔	•			•			✔	○	•	$O(r^3+n)$
LoCH [47]	✔		✔	•			•		✔	✔		✔	$O(n\sqrt{(n)})$
ClassiMap [90]	✔		✔	•			✔					•	$O(in^2)$
Kelp [13]	✔		✔	•		•	•		✔	✔	○	•	$O(r^3)$

Core Techniques

Core Method	Variants
PCA [65]	see [181] for variants; iPCA [71], PCP [182]
LDA [50]	see [61] for variants
Classical MDS [155]	L-MDS [144], Pivot-MDS [24], CFMDS [113]
MCA [17]	see [154] for variants
SOM [126]	see [150] for variants
Chalmers [32]	Glimmer [70]
LLE [129]	SLLE [178]
Isomap [153]	S-Isomap [53]
LTSA [180]	LLTSA [179]
MVU [169]	MUHSIC [147]
LSP [117]	PLP [115], E-LSP [33], Hipp [116]
SNE [64]	NeRV [162], t-SNE [158], DS t-SNE [77], Q-SNE [69] A-tSNE [122], H-SNE [121], BH-tSNE [157]

Many Variants

Dimensionality reduction



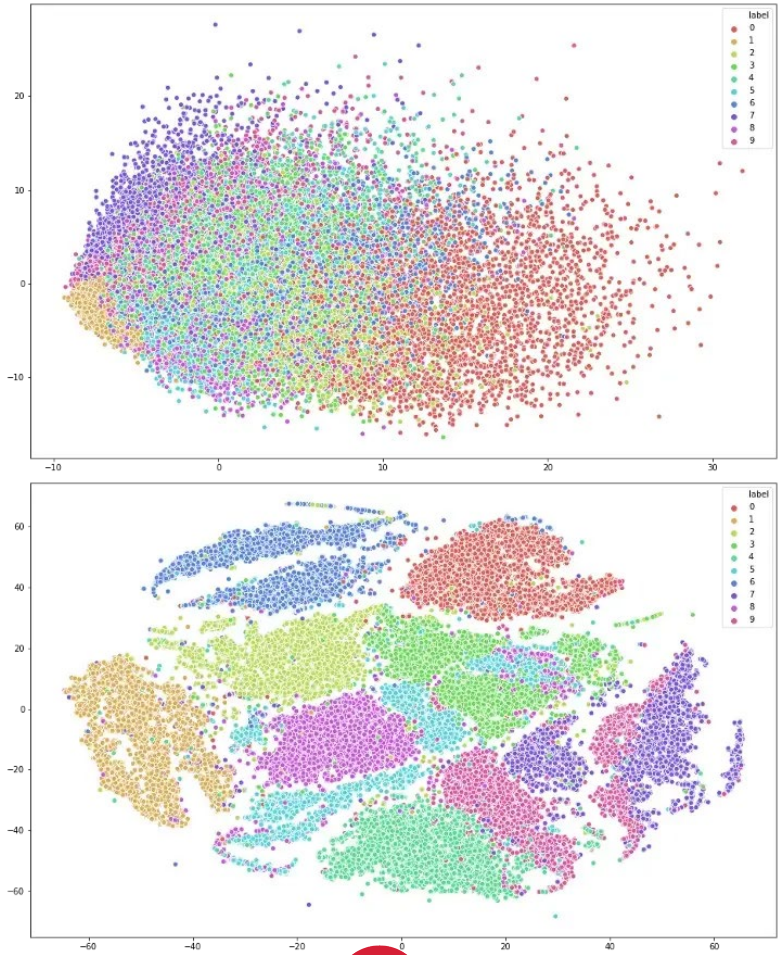
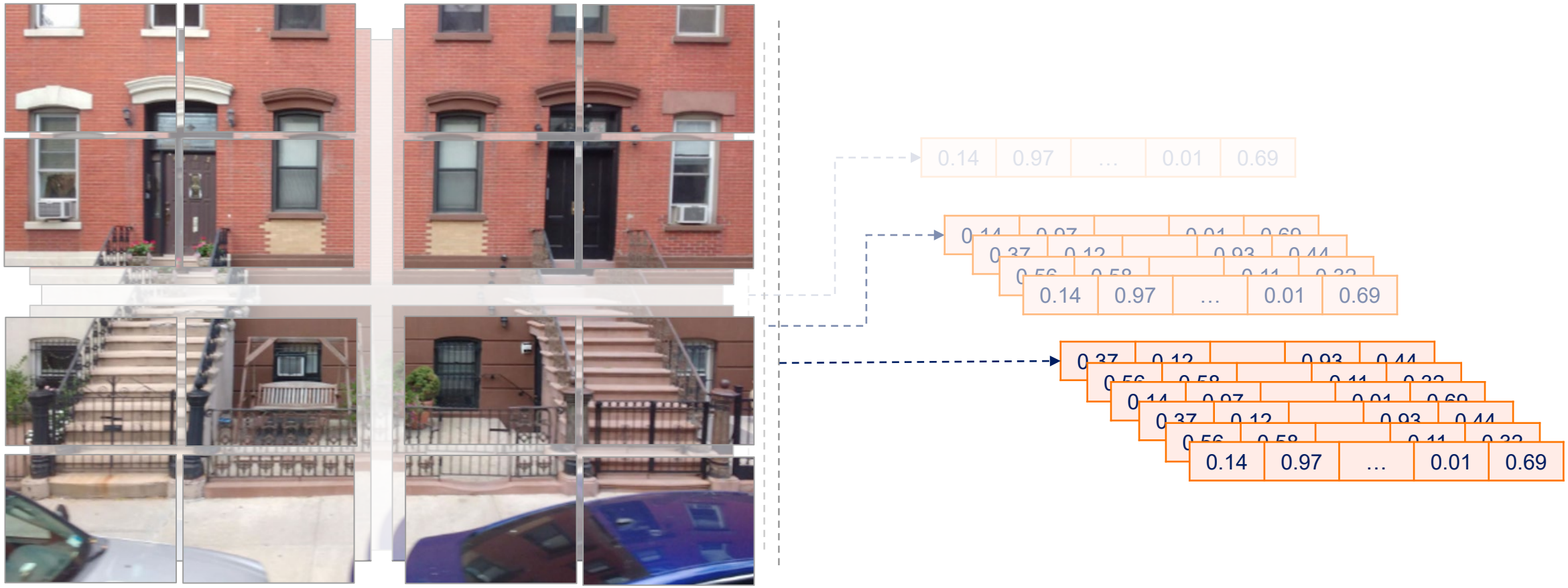
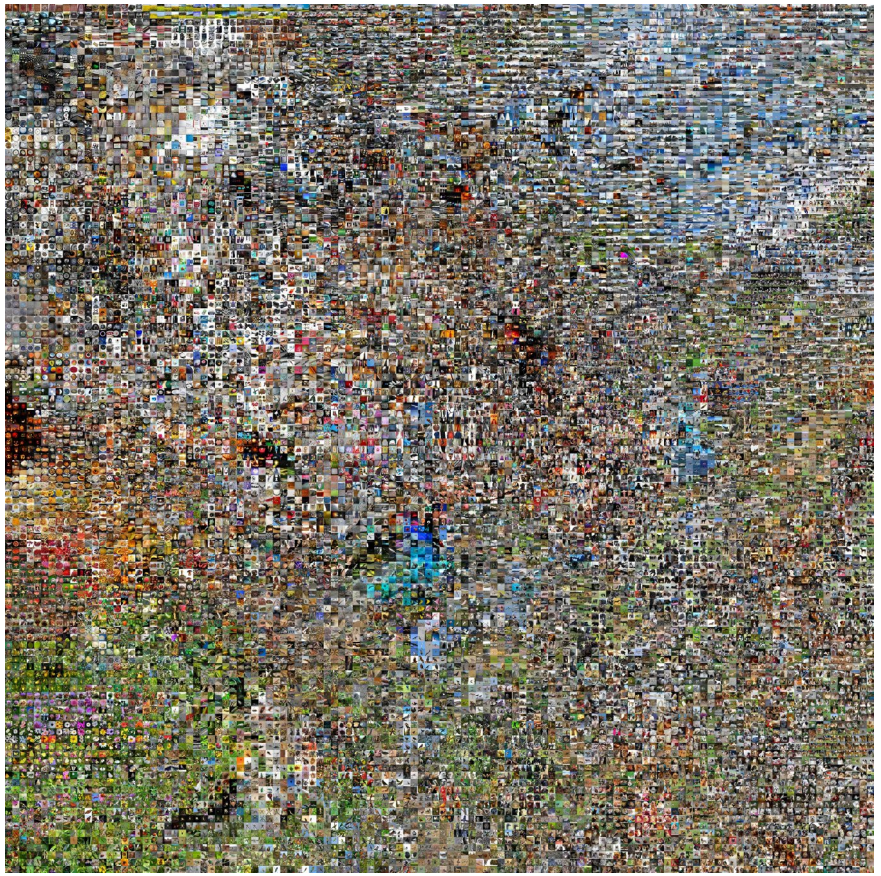


Image embeddings



Dimensionality reduction

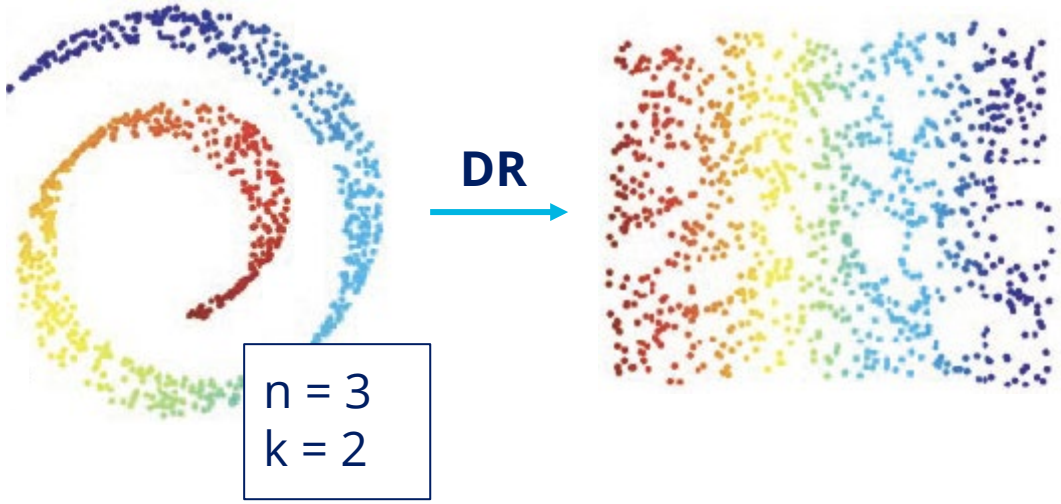


Dimensionality reduction

Dimensionality reduction

- High-dimensional data typically has an intrinsic dimension smaller than the space in which it is embedded.

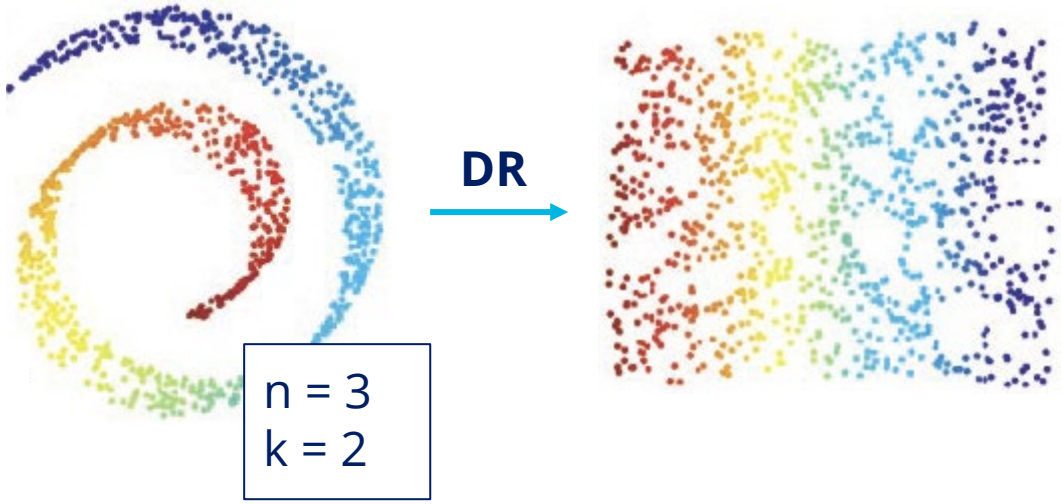
DR comprises a category of unsupervised methods that aim to capture, to a certain extent, the intrinsic dimension of the data.



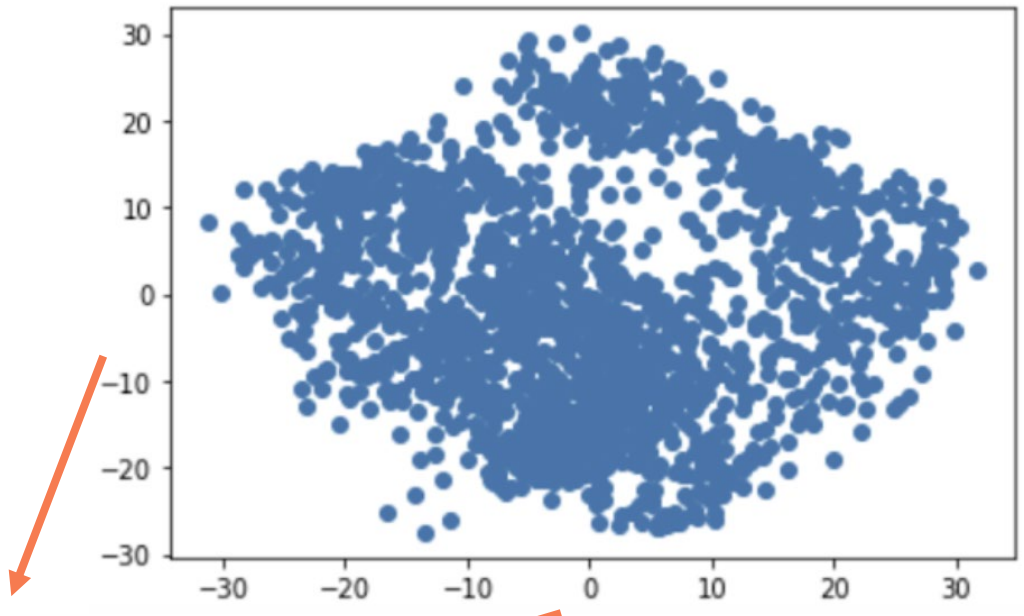
Dimensionality reduction

- High-dimensional data typically has an intrinsic dimension smaller than the space in which it is embedded.

The goal is to perform the dimensionality reduction preserving some similarity information among data instances.



Dimensionality reduction



It is not possible to interpret the meaning of the axes.

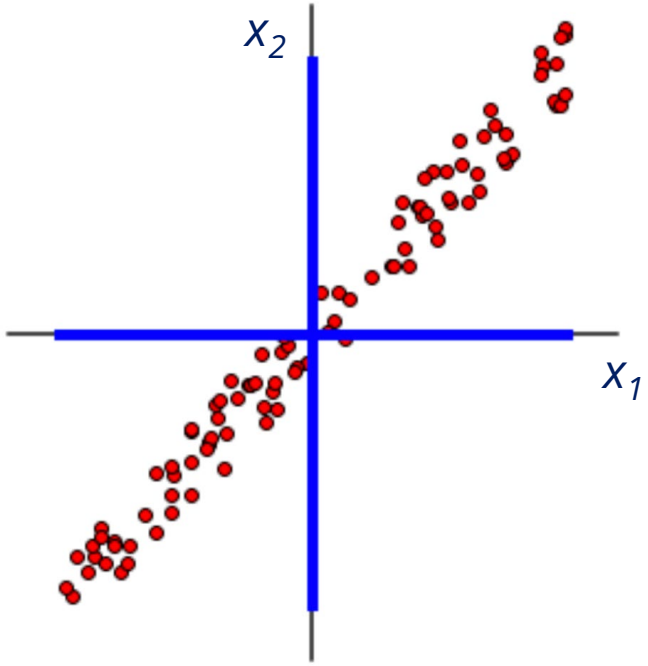
Dimensionality reduction techniques embed data in a latent space where the point coordinates (axes) do not have a semantic meaning.

$\Omega \rightarrow \mathbb{R}^p$

- Dimensionality reduction is a mapping between two spaces.
- For visualization purposes, usually $p = 2$ or $p = 3$.
- Principal Component Analysis (PCA).
- Multidimensional Scaling (MDS).
- t-distributed Stochastic Neighbor Embedding (t-SNE).
- Uniform Manifold Approximation and Projection (UMAP).
- Local Affine Multidimensional Projection (LAMP).

Principal Component Analysis (PCA)

- Given data matrix $\mathbf{X}$, PCA finds a new basis to represent the data so that the coordinates of the points in the new basis are uncorrelated.



Principal Component Analysis (PCA)

Given a dataset of n points $a_1, \ldots, a_n \in \mathbb{R}^d$:

- $d = 1$
  - Mean
    - $\mu = \frac{1}{n}\sum_{i=1}^{n} a_i \in \mathbb{R}$
  - Variance
    - $\sigma^2 = \frac{1}{n}\sum_{i=1}^{n}(a_i - \mu) \in \mathbb{R}$

Principal Component Analysis (PCA)

Given a dataset of n points $a_1, \ldots, a_n \in \mathbb{R}^d$:

- $d = 1$
  - Mean
$$\mu = \frac{1}{n}\sum_{i=1}^{n} a_i \in \mathbb{R}$$
  - Variance
$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n} (a_i - \mu) \in \mathbb{R}$$

- $d \geq 2$
  - Mean
$$\mu = \frac{1}{n}\sum_{i=1}^{n} a_i \in \mathbb{R}^d$$
  - Covariance matrix
$$\mathbf{S} = \frac{1}{n}\sum_{i=1}^{n} (a_i - \mu)\,(a_i - \mu)^T \in \mathbb{R}^{dxd}$$

Principal Component Analysis (PCA)

Given a dataset of n points $a_1, \ldots, a_n \in \mathbb{R}^d$:

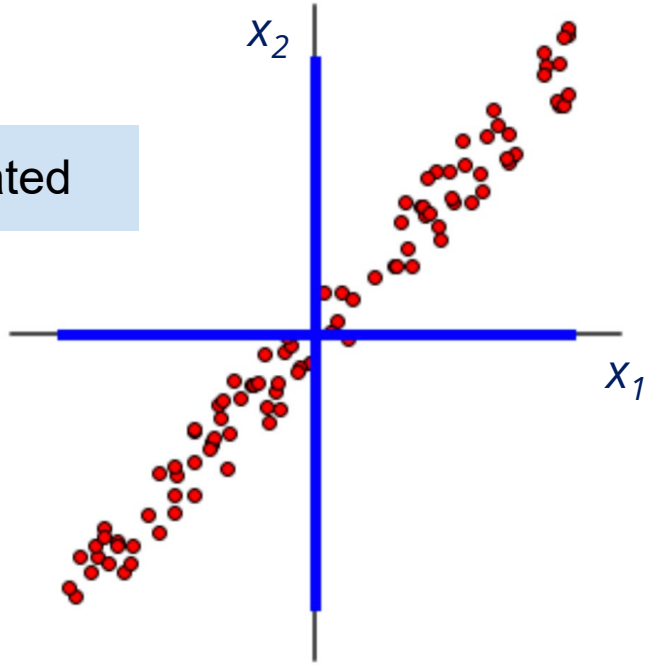
- $d = 1$
  - Mean
$$\mu = \frac{1}{n}\sum_{i=1}^{n} a_i \in \mathbb{R}$$
  - Variance
$$\sigma^2 = \frac{1}{n}\sum_{i=1}^{n} (a_i - \mu) \in \mathbb{R}$$

- $d \geq 2$
  - Mean
$$\mu = \frac{1}{n}\sum_{i=1}^{n} a_i \in \mathbb{R}^d$$
  - Covariance matrix
$$\mathbf{S} = \frac{1}{n}\sum_{i=1}^{n} (a_i - \mu)\,(a_i - \mu)^T \in \mathbb{R}^{dxd}$$
$$= \frac{1}{n}\sum_{i=1}^{n} a_i a_i^T \quad if \quad \mu = 0$$

Principal Component Analysis (PCA)

- Given data matrix $\mathbf{X}$, PCA finds a new basis to represent the data so that the coordinates of the points in the new basis are uncorrelated.

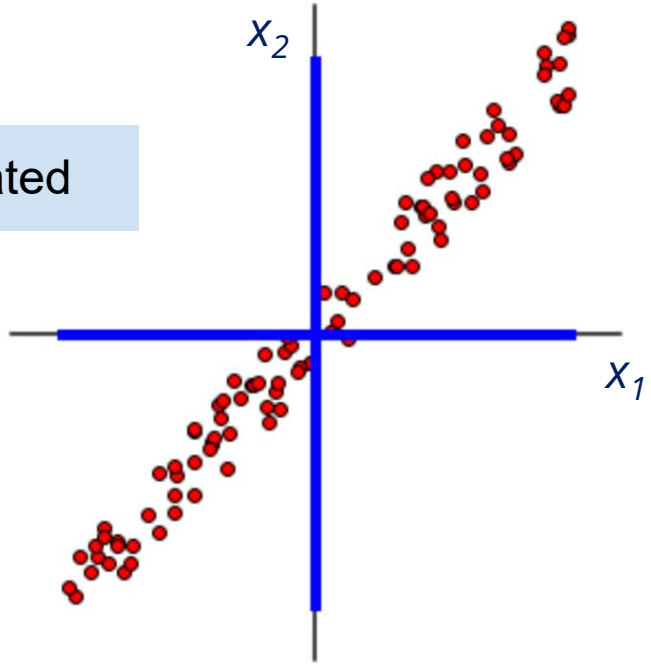
Highly correlated

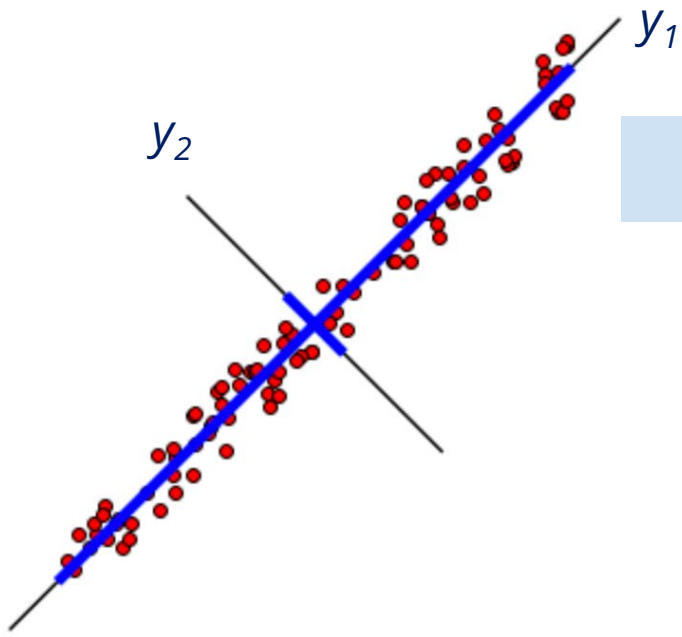


Principal Component Analysis (PCA)

- Given data matrix $\mathbf{X}$, PCA finds a new basis to represent the data so that the coordinates of the points in the new basis are uncorrelated.

Highly correlated





Uncorrelated

Principal Component Analysis (PCA)

- Given a dataset of n points $a_1, \ldots, a_n \in \mathbb{R}^d$, where d is large.
- Goal: represent this dataset in lower dimension,
  i.e., find $a_1, \ldots, a_n \in \mathbb{R}^k$ where $k \ll d$.
- Assume that the dataset is centered:

Principal Component Analysis (PCA)

- Given a dataset of n points $a_1, \ldots, a_n \in \mathbb{R}^d$, where d is large.
- Goal: represent this dataset in lower dimension, i.e., find $a_1, \ldots, a_n \in \mathbb{R}^k$ where $k \ll d$.
- Assume that the dataset is centered:

$$\sum_{i=1}^{n} a_i = 0$$

Principal Component Analysis (PCA)

- Given a dataset of n points $a_1, \ldots, a_n \in \mathbb{R}^d$, where d is large.
- Goal: represent this dataset in lower dimension, i.e., find $a_1, \ldots, a_n \in \mathbb{R}^k$ where $k \ll d$.
- Assume that the dataset is centered:

$$\sum_{i=1}^{n} a_i = 0$$

$$\mathbf{S} = \sum_{i=1}^{n} a_i a_i^T = \mathbf{A}^{\mathrm{T}}\mathbf{A} = \mathbf{A}\mathbf{A}^{\mathrm{T}}$$

Principal Component Analysis (PCA)

- Given data matrix $\mathbf{X}$, PCA finds a new basis to represent the data so that the coordinates of the points in the new basis are uncorrelated.
- Therefore, the goal is to find a change of basis matrix $\mathbf{P}$ such that:

$$\mathbf{Y} = \mathbf{PX} \Rightarrow \mathbf{YY}^{\mathbf{T}} = \mathbf{D}$$

Coordinate system in which the variables are uncorrelated.

Principal Component Analysis (PCA)

- Given data matrix $\mathbf{X}$, PCA finds a new basis to represent the data so that the coordinates of the points in the new basis are uncorrelated.
- Therefore, the goal is to find a change of basis matrix $\mathbf{P}$ such that:

$$\mathbf{Y} = \mathbf{PX} \Rightarrow \mathbf{YY}^{\mathbf{T}} = \mathbf{D}$$

Covariance matrix (new coordinates)

Principal Component Analysis (PCA)

- Given data matrix $\mathbf{X}$, PCA finds a new basis to represent the data so that the coordinates of the points in the new basis are uncorrelated.
- Therefore, the goal is to find a change of basis matrix $\mathbf{P}$ such that:

$$\mathbf{Y} = \mathbf{PX} \Rightarrow \mathbf{YY}^{\mathbf{T}} = \mathbf{D}$$

$$\mathbf{YY}^{\mathbf{T}} = (\mathbf{PX})(\mathbf{PX})^{\mathbf{T}} = \mathbf{PXX}^{\mathbf{T}}\mathbf{P}^{\mathbf{T}}$$

Covariance matrix (old coordinates)

Principal Component Analysis (PCA)

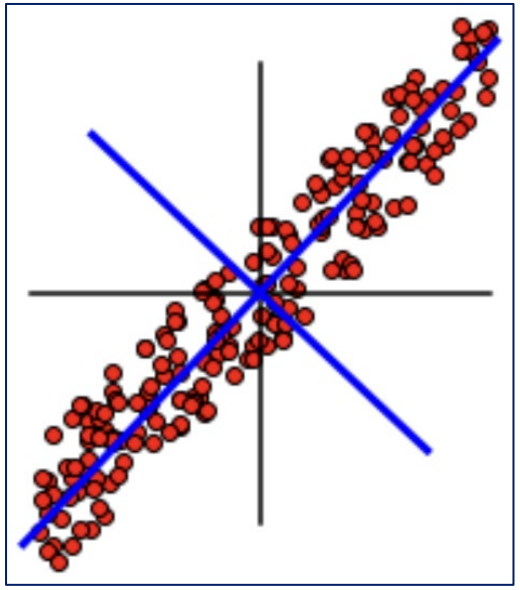
- Given data matrix $\mathbf{X}$, PCA finds a new basis to represent the data so that the coordinates of the points in the new basis are uncorrelated.
- Therefore, the goal is to find a change of basis matrix $\mathbf{P}$ such that:

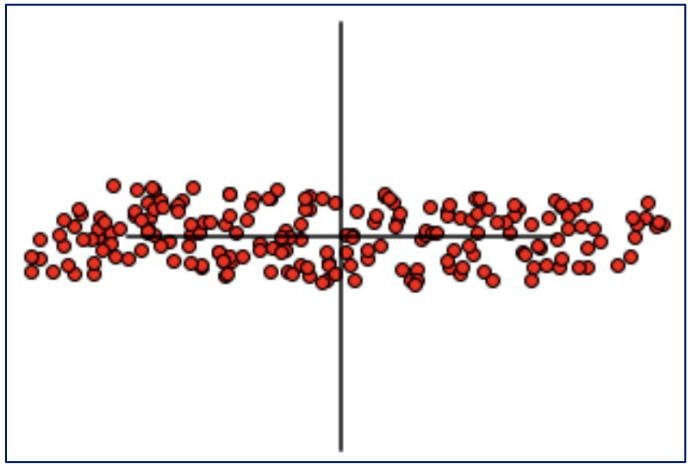
$$\mathbf{Y} = \mathbf{PX} \Rightarrow \mathbf{YY}^{\mathbf{T}} = \mathbf{D}$$
$$\mathbf{YY}^{\mathbf{T}} = (\mathbf{PX})(\mathbf{PX})^{\mathbf{T}} = \mathbf{PXX}^{\mathbf{T}}\mathbf{P}^{\mathbf{T}}$$

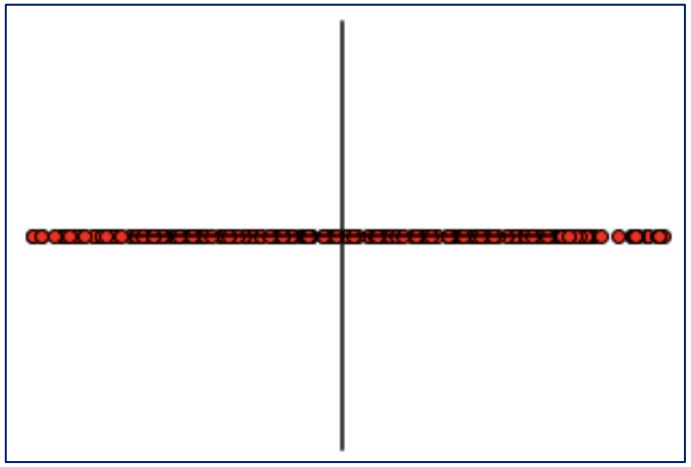
From the Spectral Theorem we conclude that rows of $\mathbf{P}$ must be the eigenvectors of $\mathbf{XX}^{\mathbf{T}}$. Moreover, the diagonal elements in $\mathbf{D}$ are the eigenvalues of $\mathbf{XX}^{\mathbf{T}}$ .In other words, the eigenvalues of $\mathbf{XX}^{\mathbf{T}}$ are the variances of each coordinate.

Principal Component Analysis (PCA)

- Typically, the variance associated to certain principal components are close to zero, meaning that the coordinate associated to those directions corresponds to noise and can be disregarded.







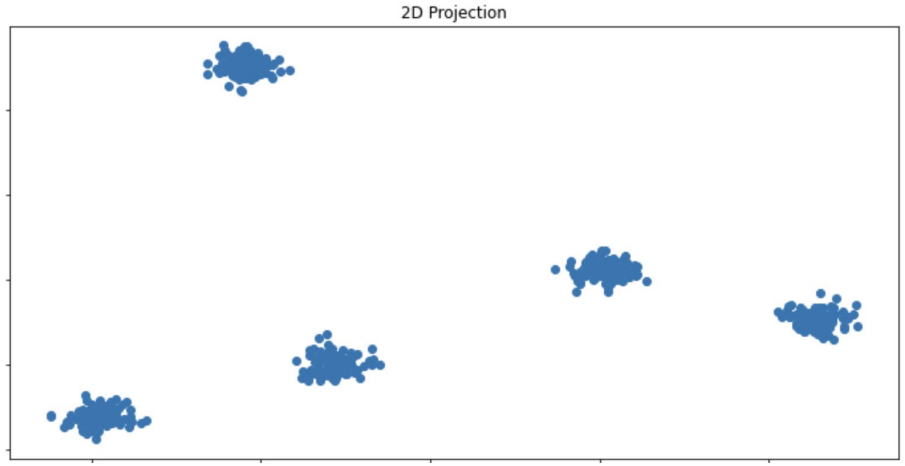
Principal Component Analysis (PCA)

```
from sklearn.decomposition import PCA

data_pca=PCA(svd_solver='full')    # svd_solver='full' computes all the principal directions
data_transformed = data_pca.fit(X) # Computes the PC basis

var = data_transformed.explained_variance_
expl_var = data_transformed.explained_variance_ratio_

Y = np.dot(data_transformed.components_[:2,:],X.T) # projects original data onto the 2 main PC
```

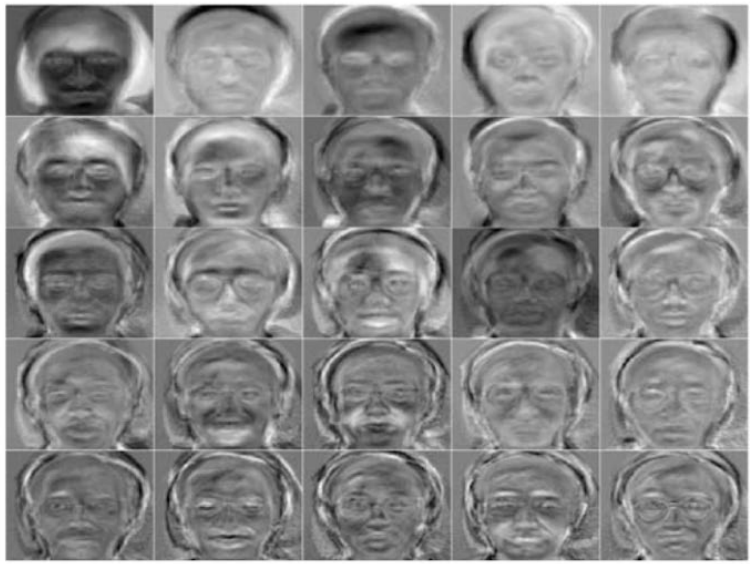


Principal Component Analysis (PCA)

Eigenfaces [Turk, Pentland '91]

- Input images:

- Principal components:



Multidimensional Scaling (MDS)

- Given the pairwise distance between data points

$$\begin{bmatrix} d_{12} & d_{13} & d_{14} & \cdots & d_{1n} \\ & d_{23} & d_{24} & \cdots & d_{2n} \\ & & & \ddots & \\ & & & & d_{(n-1)n} \end{bmatrix}$$

- Find an Euclidian embedding with total minimal distortion:

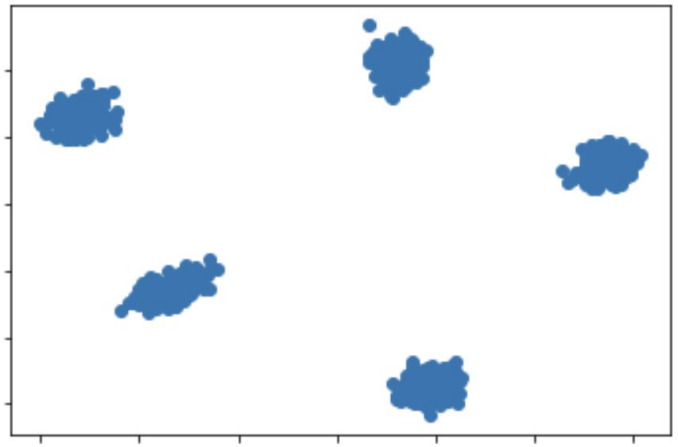
$$\min_{Y_i \in \mathbb{R}^k} \sum_{i,j} (\|Y_i - Y_j\|^2 - d_{ij}^2)^2$$

Multidimensional Scaling (MDS)

```
from sklearn.manifold import MDS
from sklearn.metrics.pairwise import euclidean_distances

D = euclidean_distances(X)  # pairwise distance matrix

embedding = MDS(n_components=2,dissimilarity='precomputed') # 2D embedding from D
Y = embedding.fit_transform(D) # computes the embedding
```

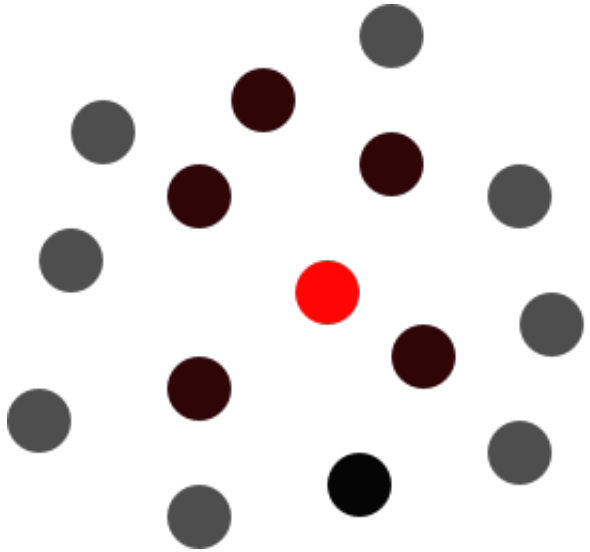


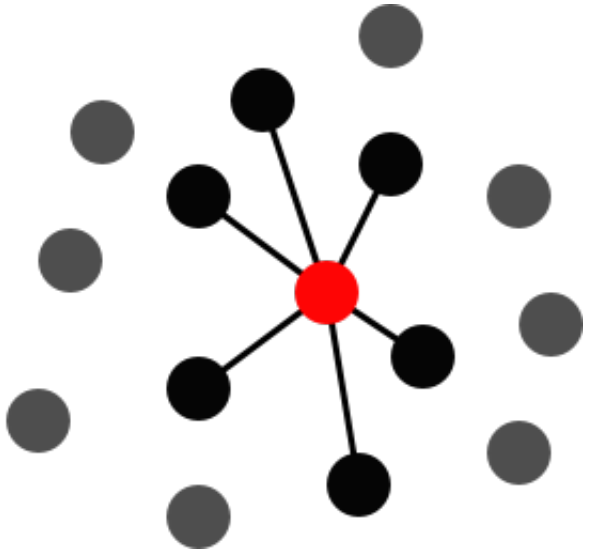
tSNE

- tSNE: tool for dimensionality reduction of high-dimensional data.
- Main idea:
  - Similarity between data points in the high-dimensional space treated as distribution P.
  - Similarity between data points in the low-dimensional space Q.
  - Achieve a representation (embedding) in the low dimension where Q faithfully represents P.
- Steps:
  1. Computation of similarities ($O(n^2)$).
  2. Minimization of cost function (divergence between P and Q).

tSNE

- The core idea of Stochastic Neighbor Embedding (SNE) is to convert the pairwise Euclidean distances between data points into conditional probabilities that represent similarities:





tSNE

- The core idea of Stochastic Neighbor Embedding (SNE) is to convert the pairwise Euclidean distances between data points into conditional probabilities that represent similarities:

$$p_{j|i} = \frac{\exp\left(-\|x_i - x_j\|^2 / 2\sigma_i^2\right)}{\sum_{k \neq i} \exp\left(-\|x_i - x_k\|^2 / 2\sigma_i^2\right)}$$

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

tSNE

- The core idea of Stochastic Neighbor Embedding (SNE) is to convert the pairwise Euclidean distances between data points into conditional probabilities that represent similarities:

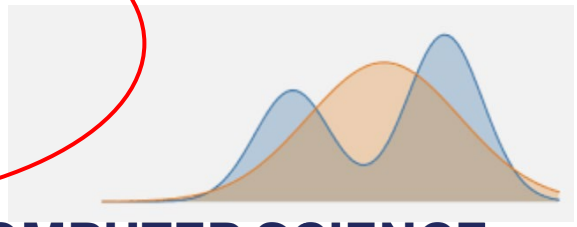
$$p_{j|i} = \frac{\exp\left(-\|x_i - x_j\|^2 / 2\sigma_i^2\right)}{\sum_{k \neq i} \exp\left(-\|x_i - x_k\|^2 / 2\sigma_i^2\right)}$$

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2n}$$

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

KL-divergence

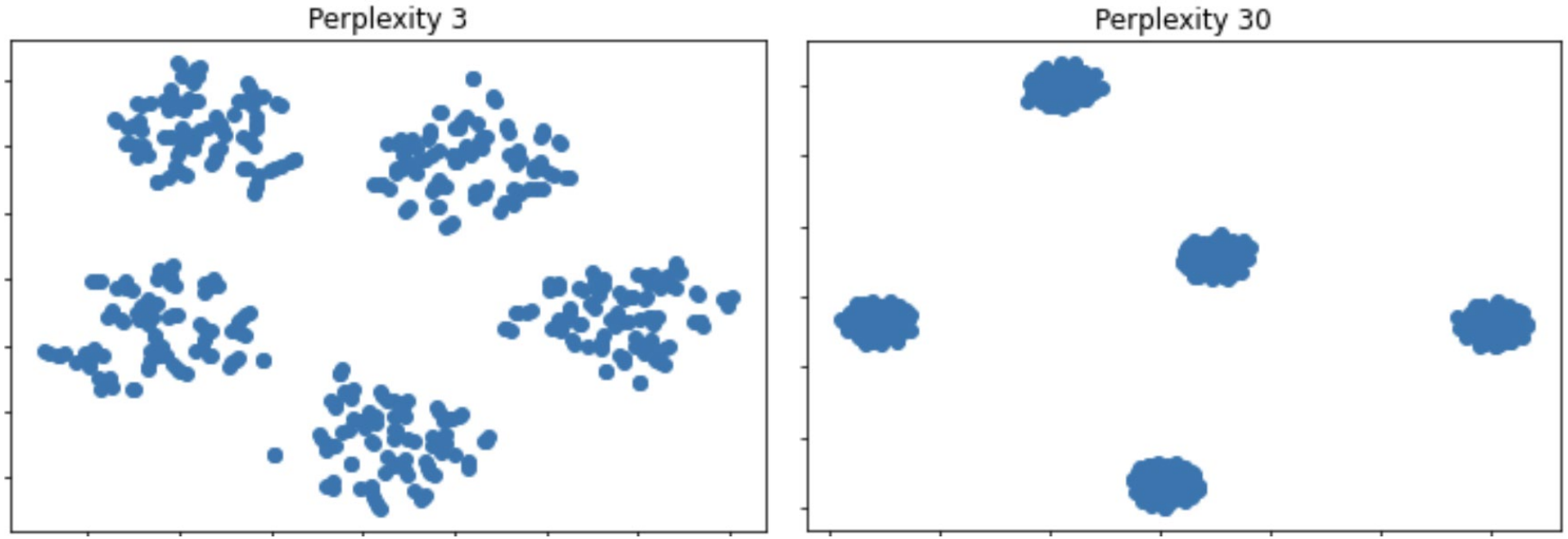
$$\arg\min_{y_i \in \mathbb{R}^k} \sum_{i,j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$



tSNE

```
from sklearn.manifold import TSNE

t_sne = TSNE(n_components=2, perplexity=3)
Y = t_sne.fit_transform(X)
```

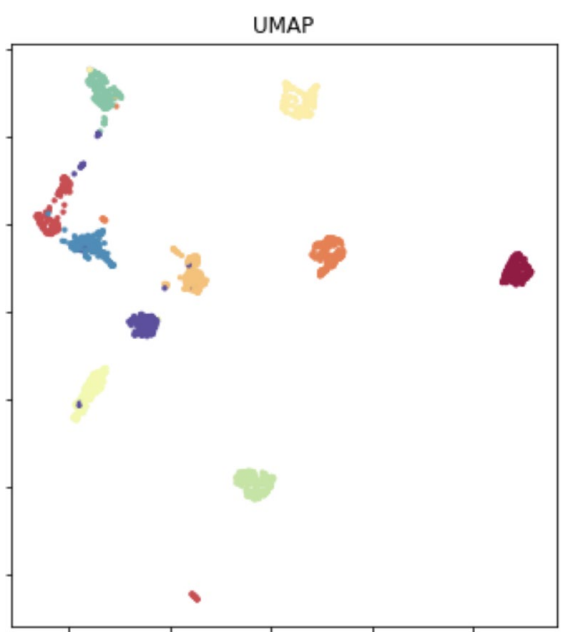


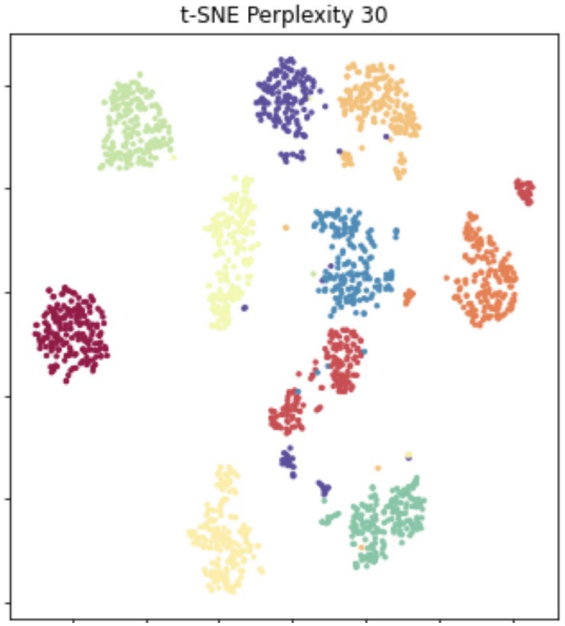
UMAP

```
conda install -c conda-forge umap-learn
```

```
import umap

umap_r = umap.UMAP()
Y = umap_r.fit_transform(X)
```





What about interactivity?

- Dimensionality reduction methods discussed previously are unsupervised, so users can only tune hyperparameters.
- **What if users want to have some control where certain points must be in the visual space while preserving their neighborhood?**

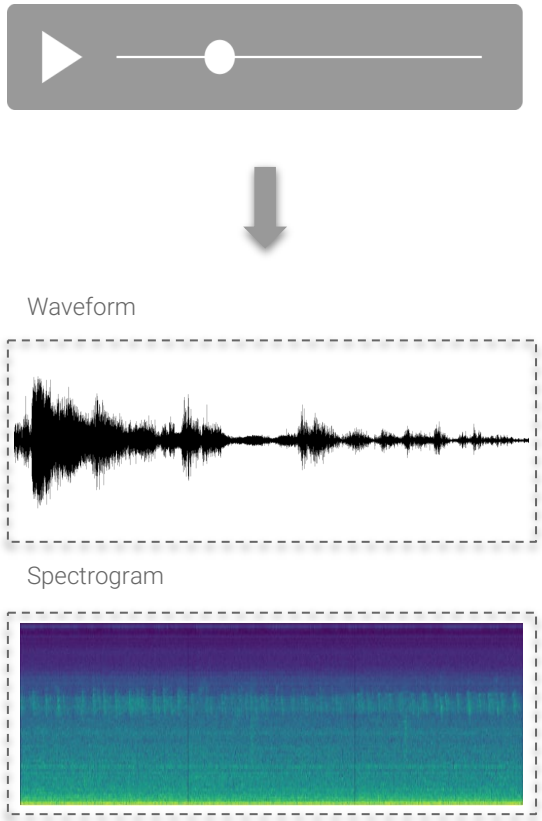
What about interactivity?

- Dimensionality reduction methods discussed previously are unsupervised, so users can only tune hyperparameters.
- **What if users want to have some control where certain points must be in the visual space while preserving their neighborhood?**
- Interactive dimensionality reduction methods are designed to enable users with some control over the mapping.

Urban audio data exploration

Audio data

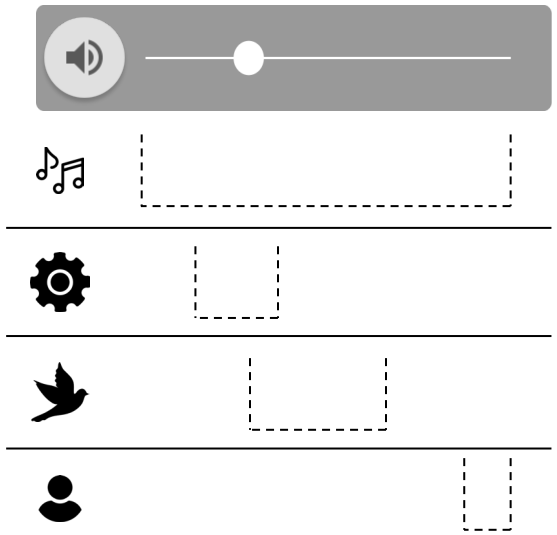
- Audio recordings are consumed in a **serial** way by us.
- To understand events happening in a 10-second audio snippet, users **must listen through the entire recording.**
- Although the **visualization of specific frequencies or loudness** can help identify interest periods of the recording, it is still difficult to build a **semantic understanding** of the recording.



Urban audio data exploration

Audio data

- Unlike images, where visual objects are opaque, sound objects are conceptually ***transparent***, meaning that multiple objects (sound sources) can have energy at the same frequency.
- At any given instant in time, a sound recording might have a **mixture of background** (birds, dog barks) and **foreground sounds** (party, sirens).

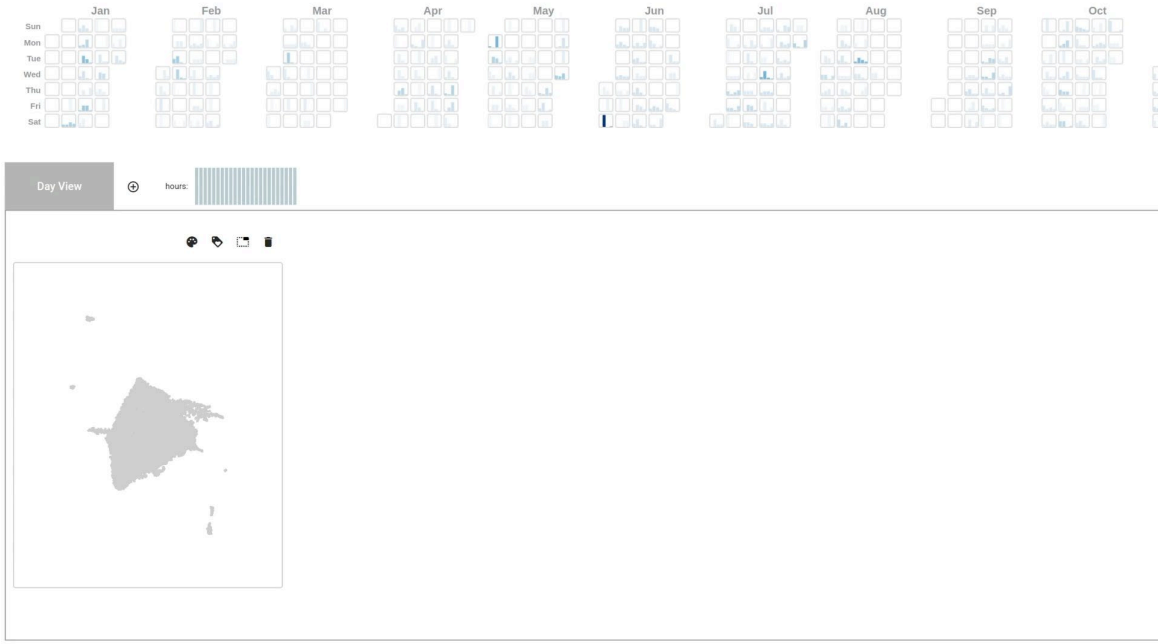


Urban Rhapsody: Creating prototypes

Binary Classification Model

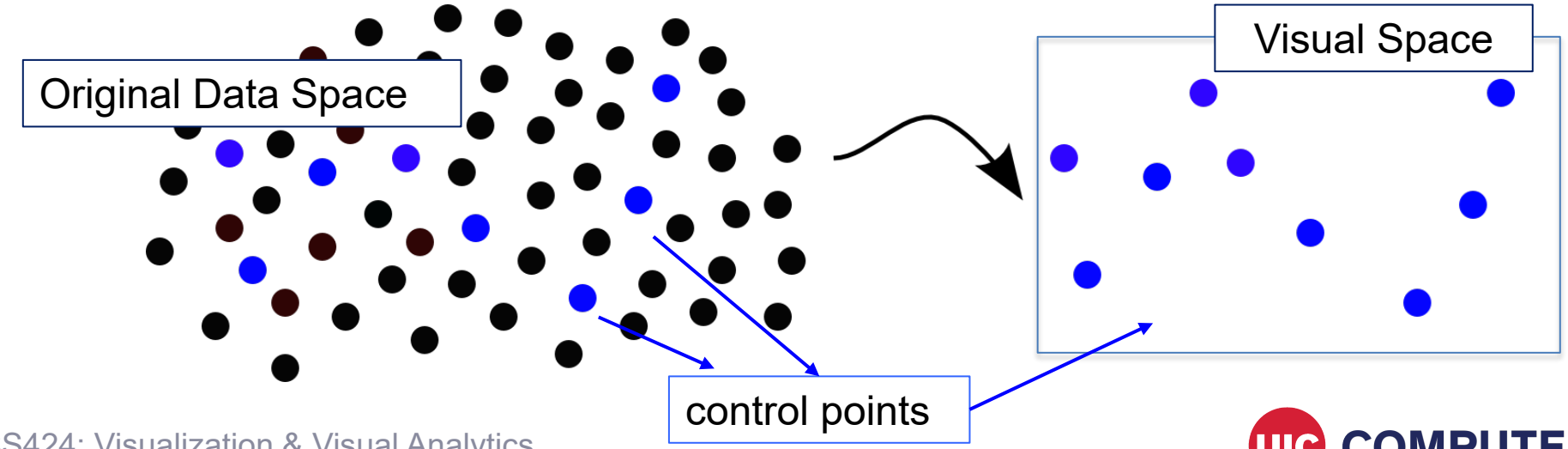
+

Representatives



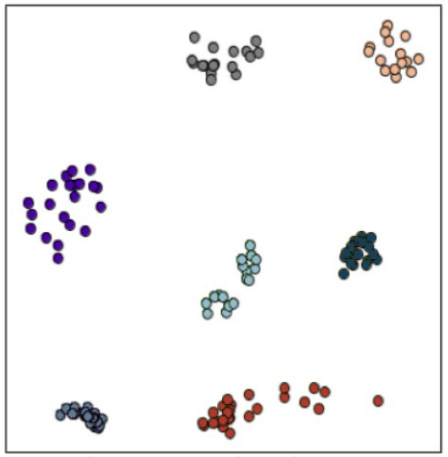
Interactive dimensionality reduction

- There are several interactive dimensionality reduction methods:
  - Least Squares Projection (LSP).
  - Piecewise Laplacian Projection (PLP)
  - Local Affine Multidimensional Projection (LAMP)



LAMP

Control points



Tuning the hyperparameter σ_i