Data Science & Artificial Intelligence

Summer Term 2022

IFM Institut für Fahrerassistenz und vernetzte Mobilität

L04 Dimension Reduction

J. Haselberger, D. Schneider, B. Stühr

L04.1 Motivation

Why do we need to reduce the Dimensions?

Curse of Dimensionality
real world problems could have more than 1000 features

Application for:

- data compression
- visualization (otherwise max 3D/4D)
- noise reduction
- reduce redundancy

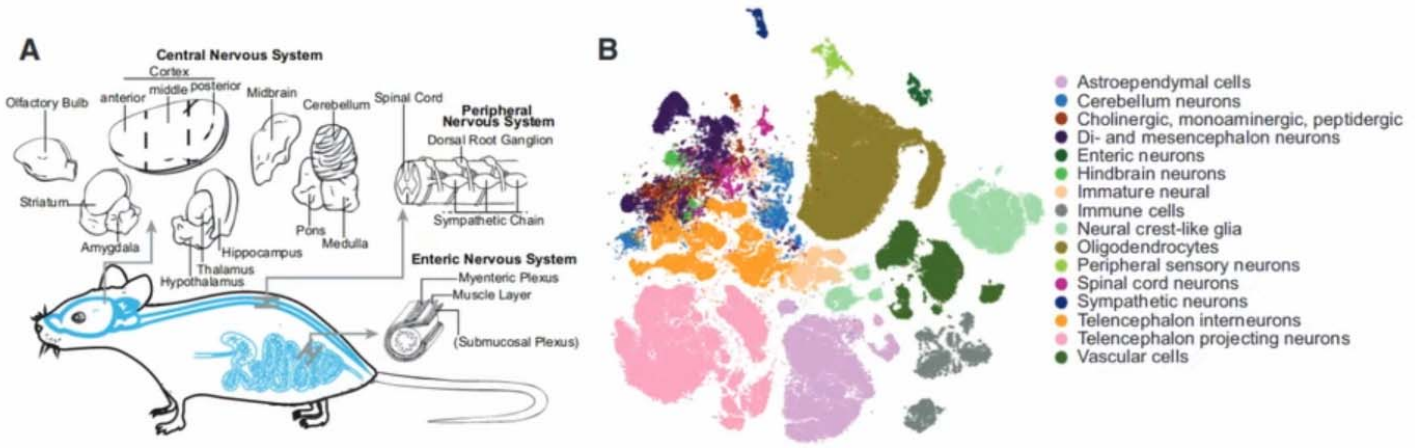
	PassengerId	Survived	Pclass	Name	Sex	Age	SibSp	Parch	Ticket	Fare	Cabin	Embarked
0	1	0	3	Braund, Mr. Owen Harris	male	22.0	1	0	A/5 21171	7.2500	NaN	S
1	2	1	1	Cumings, Mrs. John Bradley (Florence Briggs Th...	female	38.0	1	0	PC 17599	71.2833	C85	C
2	3	1	3	Heikkinen, Miss. Laina	female	26.0	0	0	STON/O2. 3101282	7.9250	NaN	S
3	4	1	1	Futrelle, Mrs. Jacques Heath (Lily May Peel)	female	35.0	1	0	113803	53.1000	C123	S
4	5	0	3	Allen, Mr. William Henry	male	35.0	0	0	373450	8.0500	NaN	S
5	6	0	3	Moran, Mr. James	male	NaN	0	0	330877	8.4583	NaN	Q
6	7	0	1	McCarthy, Mr. Timothy J	male	54.0	0	0	17463	51.8625	E46	S
7	8	0	3	Palsson, Master. Gosta Leonard	male	2.0	3	1	349909	21.0750	NaN	S
8	9	1	3	Johnson, Mrs. Oscar W (Elisabeth Vilhelmina Berg)	female	27.0	0	2	347742	11.1333	NaN	S

sample data from the titanic dataset [Jr.2017]

L04.1 Motivation

Why do we need to reduce the Dimensions?

Single-cell transcriptomics (single-cell RNA sequencing): samples are cells, features are genes.

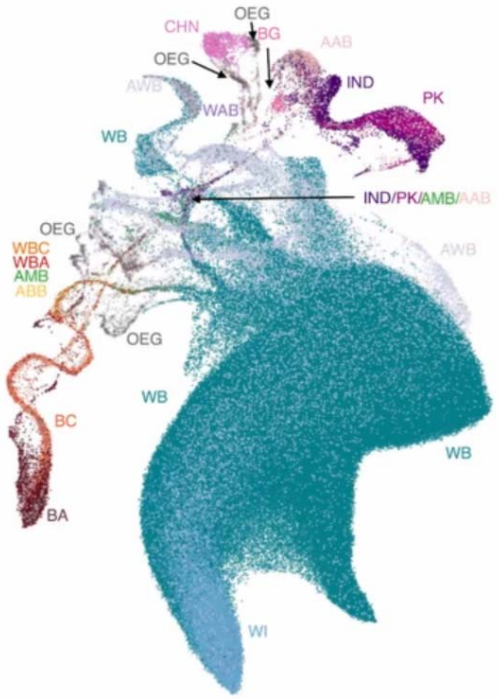


Zeisel et al. (2018)
$n \approx 500{,}000$

L04.1 Motivation

Why do we need to reduce the Dimensions?

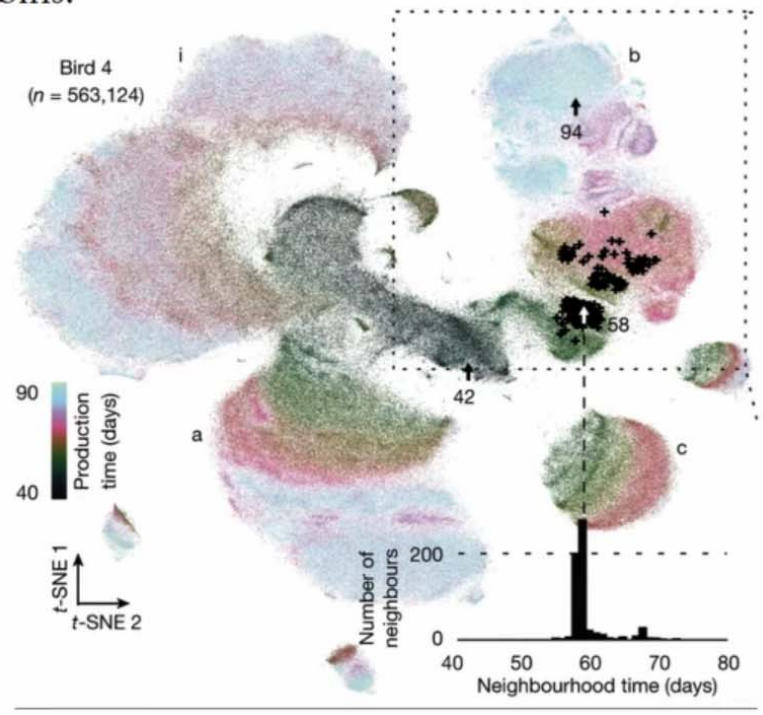
Population genomics: samples are people, features are single-nucleotide polymorphims.



L04.1 Motivation

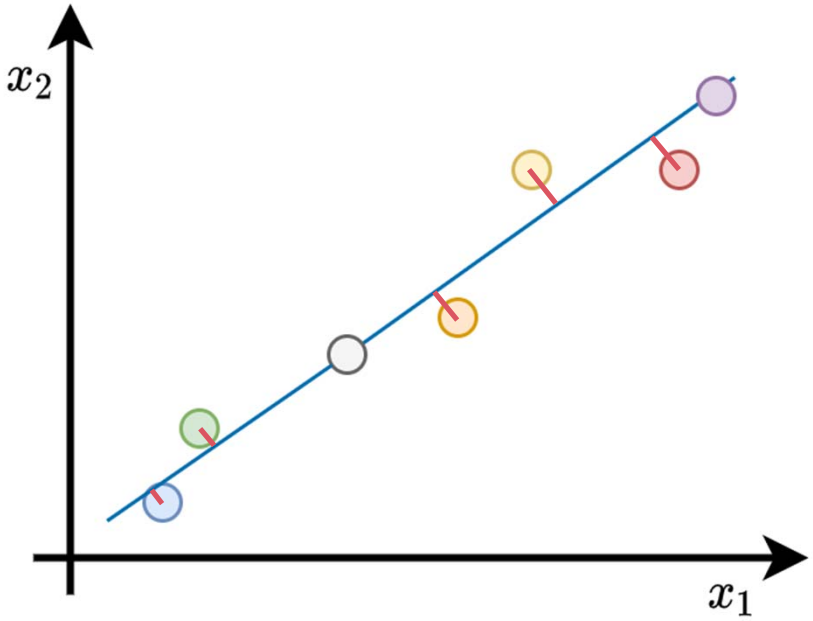
Why do we need to reduce the Dimensions?

Behavioural physiology: samples are syllable renditions, features are spectrogram bins.



Kollmorgen et al. (2020)
$n \approx 600{,}000$

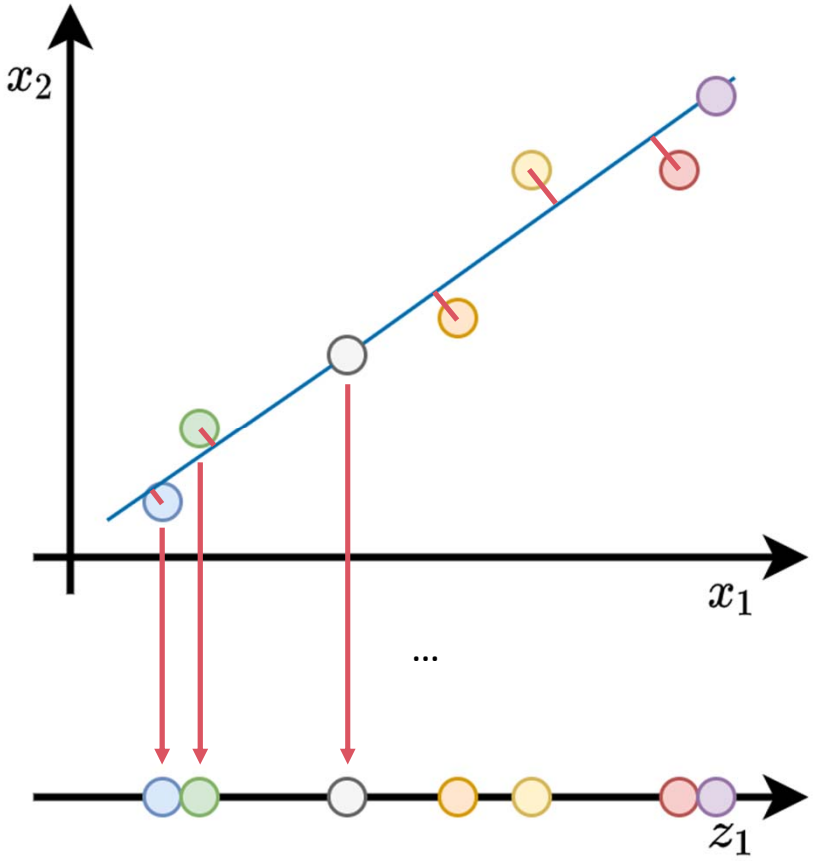
L04.1 Intuition



Data reduction from 2D to 1D

- find a direction to project the data to in order to minimize the projection error
- $X_i \in \mathbb{R}^2 \rightarrow Z_i \in \mathbb{R}^1$

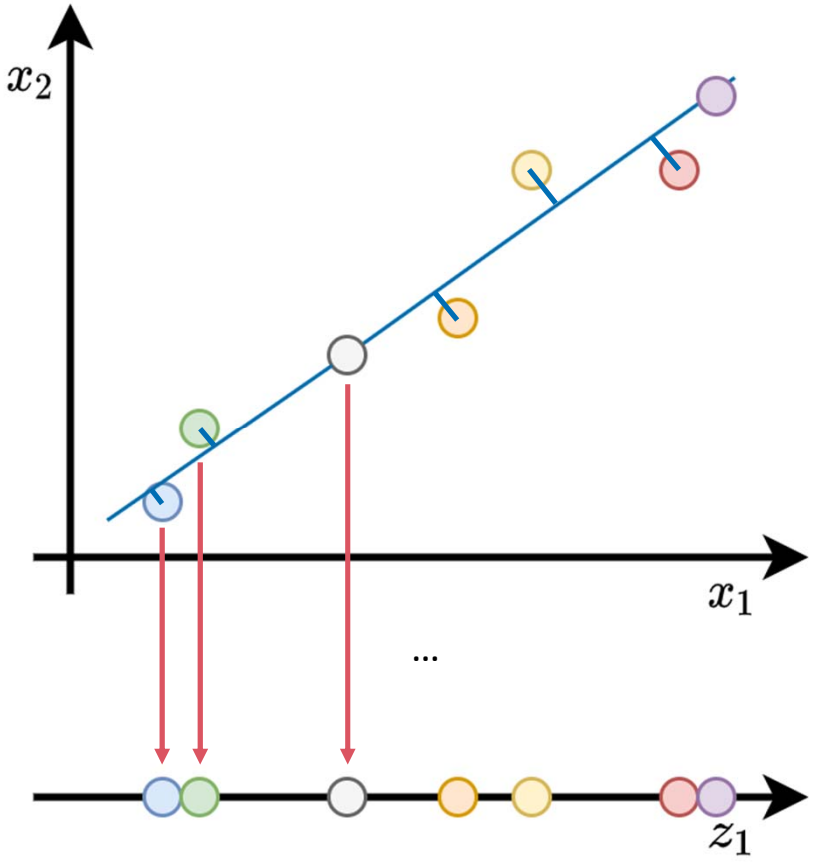
L04.1 Intuition



Data reduction from 2D to 1D

- find a direction to project the data to in order to minimize the projection error
- $X_i \in \mathbb{R}^2 \rightarrow Z_i \in \mathbb{R}^1$

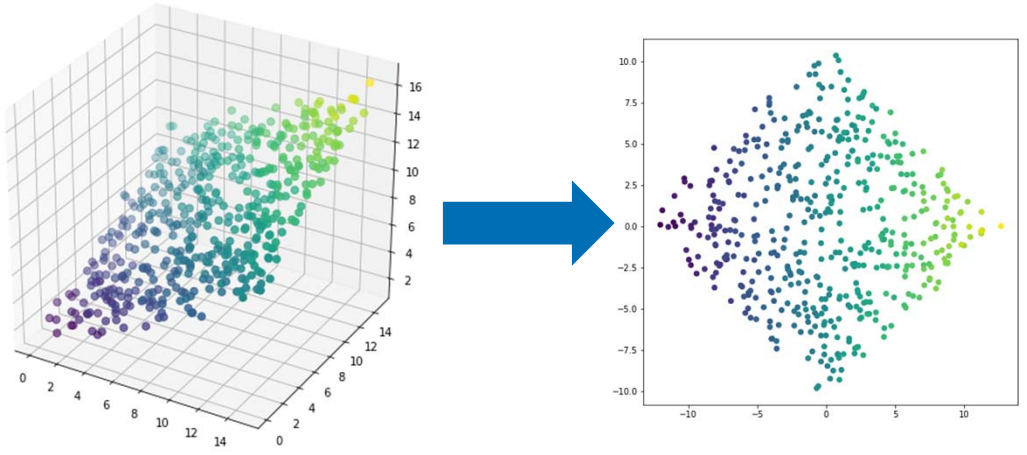
L04.1 Intuition



Data reduction from 2D to 1D

- find a direction to project the data to in order to minimize the projection error
- $X_i \in \mathbb{R}^2 \rightarrow Z_i \in \mathbb{R}^1$

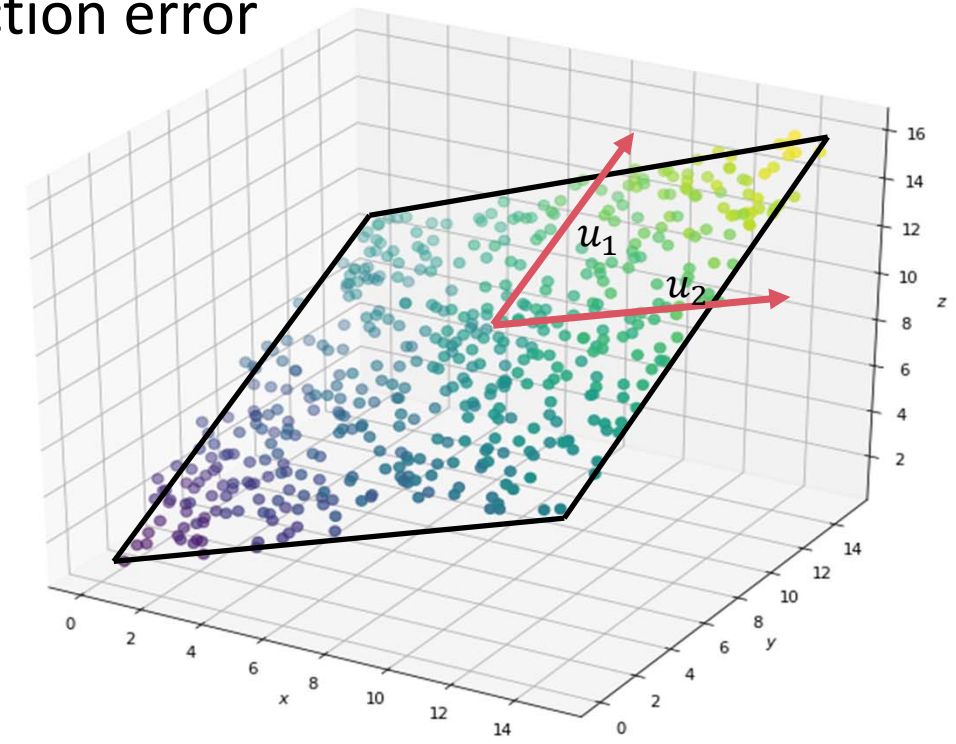
But also applicable for multiple dimensions



L04.2 Principal Component Analysis (PCA)

From n to k Dimensions: Problem Formulation

- search for k vectors $(u_1, u_2, \ldots, u_k)$ to project the data onto in order to minimize the projection error



L04.2 Principal Component Analysis (PCA)

Basic Algorithm

Step One: compute the covariance matrix

Covariance Matrix:

$$\Sigma = \frac{1}{m} \sum_{i=1}^{n} x_i x_i^T$$

```
import numpy as np

# calculate the covariance matrix of X
sigma = np.cov(X)

# calculate the eigenvectors of sigma
u, s, v = np.linalg.svd(sigma)
```

Step Two: compute the eigenvectors of the covariance matrix

- process called „Singular Value Decomposition“ (svd)

$$USV = svd(\Sigma)$$

L04.2 Principal Component Analysis (PCA)

Basic Algorithm

- U contains the u_1, u_2 … vectors that we need to project the data

$$U = \underbrace{\begin{bmatrix} | & | & | & & | \\ u_1 & u_2 & u_3 & \dots & u_m \\ | & | & | & & | \end{bmatrix}}_{k} \in \mathbb{R}^{n \times n}$$

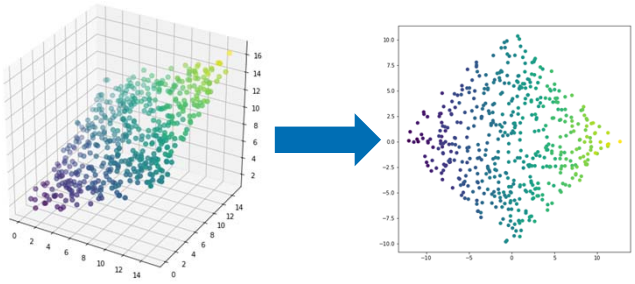
Step Three: select first k coloums of U

- k is the target dimensionality

$$U_{reduced} = \begin{bmatrix} | & | & & | \\ u_1 & u_2 & \dots & u_k \\ | & | & & | \end{bmatrix} \in \mathbb{R}^{n \times k}$$

Step Four: calculate the projection

$$Z = U_{reduced}^T X$$



$X \in \mathbb{R}^3$ $Z \in \mathbb{R}^2$

L04.2 Principal Component Analysis (PCA)

Data Preperation

- PCA is affected by the scale
- features of the data must be standardized before PCA is applied
- remove mean ($\mu = 0$) and scale to unit variance ($\sigma = 1$)

StandardScaler:

$$\acute{X} = \frac{(X - \bar{X})}{\sigma}$$

```python
from sklearn.preprocessing import StandardScaler

# apply the standard scaling
X_scaled = StandardScaler().fit_transform(X)
```

L04.2 Principal Component Analysis (PCA)

Choosing the number of principal components

- if we use dimensionality reduction for visualization, the number of components is defined by the visualization type, e.g. $k = 3$ for 3D plot

- if we use reduction for compression, we are interesed in the amount of information we loose due to the reduction to k dimensions

- The **Explained Variance** (ratio between projection error and data variation) indicates how much information (variance) can be attributed to each of the principal components

Explained Variance:

$$1 - \frac{\overbrace{\frac{1}{m}\sum_{i=1}^{m} ||x_i - \hat{x}_i||^2}^{\text{average squared projection error}}}{\underbrace{\frac{1}{m}\sum_{i=1}^{m} ||x_i||^2}_{\text{total data variation}}}$$

L04.2 Principal Component Analysis (PCA)

Choosing the number of principal components

- Rule of thumb: set k so that an Explained Variance of $\geq 99\%$ is reached

$$S = \begin{bmatrix} \sigma_{11} & 0 & 0 & 0 \\ 0 & \sigma_{22} & 0 & 0 \\ & & \ddots & \\ 0 & 0 & 0 & \sigma_{nn} \end{bmatrix}$$

But how to find k?

- Option 1: Run PCA several times with different values of k and calculate explained variance
  - highly inefficient!

- Option 2: use linear algebra, the already computed S matrix
  - almost no additional computing time

Explained Variance:

$$\frac{\sum_{i=1}^{k} \sigma_{ii}}{\sum_{i=1}^{n} \sigma_{ii}}$$

L04.2 Principal Component Analysis (PCA)

Impact of k on training duration and accuracy

- in the following example, the MNIST dataset is used to train a Logistic Regression model
- the dataset has 784 features
- PCA is applied successively to reduce the dimensions

Variance Retained	Number of Components	Time (seconds)	Accuracy
1.00	784	48.94	0.9158
0.99	541	34.69	0.9169
0.95	330	13.89	0.9200
0.90	236	10.56	0.9168
0.85	184	8.85	0.9156

Example taken from [Galarnyk2017]

Reducing the dimensions:

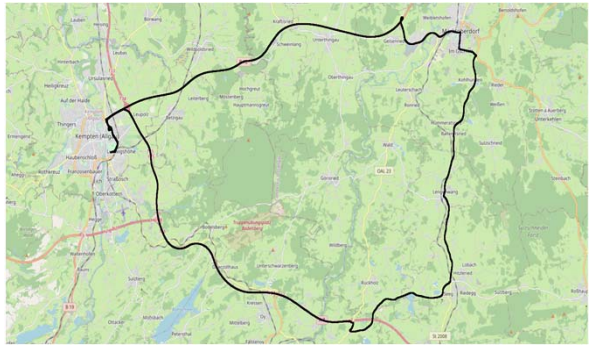
- training time is strongly reduced
- accuracy can be maintained, if not improved

Prevent Overfitting
By reducing the data dimensionality, the risk of overfitting can be reduced.

L04.2 Principal Component Analysis (PCA)

Use Case Example

Self-Perception versus Objective Driving Behavior
Subject Study of Lateral Vehicle Guidance



#	Item Content
2	Purposely tailgate other drivers
5	Drive through traffic lights that have just turned red
9	When in a traffic jam and the lane next to me starts to move. I try to move into that lane as soon as possible
16	In a traffic jam. I think about ways to get through the traffic faster
17	When a traffic light turns green and the car in front of me doesn't get going immediately. I try to urge the driver to move on
32	Get impatient during rush hours
6	Enjoy the sensation of driving on the limit
20	Fix my hair/ makeup while driving
22	Like to take risks while driving
24	Like the thrill of flirting with death or disaster
44	Enjoy the excitement of dangerous driving
15	Lost in thoughts or distracted. I fail to notice someone at the pedestrian crossings
27	Forget that my lights are on full beam until flashed by another motorist
30	Misjudge the speed of an oncoming vehicle when passing
34	Intend to switch on the windscreen wipers. But switch on the lights instead
35	Attempt to drive away from traffic lights in third gear (or on the neutral mode in automatic cars)
36	Plan my route badly. So that i hit traffic that i could have avoided
39	Nearly hit something due to misjudging my gap in a parking lot
11	I daydream to pass the time while driving
10	Driving makes me feel frustrated
31	Feel nervous while driving
33	Feel distressed while driving
26	Meditate while driving
12	Swear at other drivers
28	When someone does something on the road that annoys me. I flash them with the high beam
43	Honk my horn at others
3	Blow my horn or "flash" the car in front as a way of expressing frustrations
13	When a traffic light turns green and the car in front of me doesn't get going. I just wait for a while until it moves
18	At an intersection where I have to give right-of-way to oncoming traffic. I wait patiently for cross-traffic to pass
23	Base my behavior on the motto "better safe than sorry"
14	Drive cautiously
41	Always ready to react to unexpected maneuvers by other drivers
42	Tend to drive carefully

L04.2 Principal Component Analysis (PCA)

Use Case Example

- The result of PCA with multiple components can be used to group the different questionnaire questions into unique clusters
- Calculation of a score for the different factors

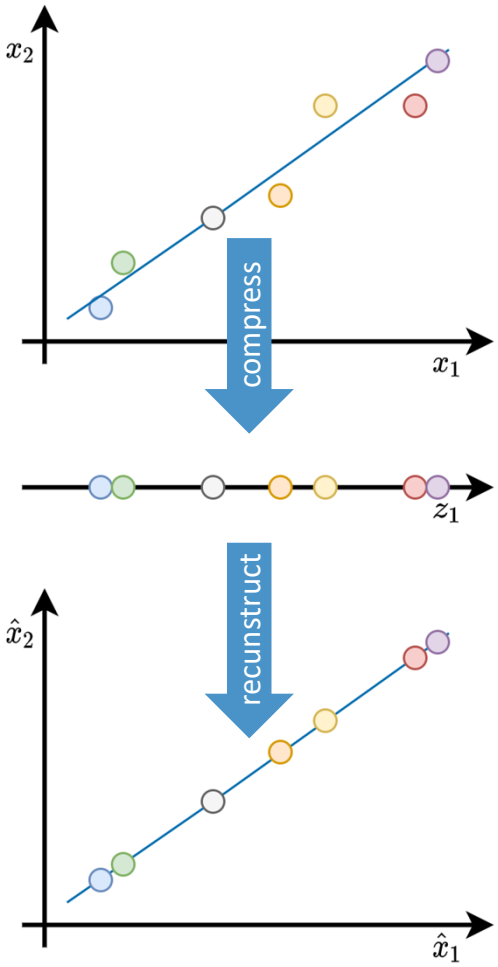
Table 1: Results of two factor analysis (Loadings are from the second analysis)

	Factor 1 | Angry Driving	**Loadings**
43	Honk my horn at others	.803
3	Blow my horn or "flash" the car in front as a way of expressing frustrations	.771
28	When someone does something on the road that annoys me, I flash them with high beam	.733
12	Swear at other drivers	.657
17	*When a traffic light turns green and the car in front of me doesn't get going immediately, I try to urge the driver to move on*	*.692*
13	*When a traffic light turns green and the car in front of me doesn't get going, I wait for a while until it moves [-]*	*.503*
2	*Purposely tailgate other drivers*	*.413*
	Factor 2 | Risky Driving	
44	Enjoy the excitement of dangerous driving	.832
22	Like to take risks while driving	.737
24	Like the thrill of flirting with death or disaster	.701
6	Enjoy the sensation of driving on the limit	.657
29	*Get a thrill out of breaking the law*	*.732*
	Factor 3 | Anxious Driving	
31	Feel nervous while driving	.799
10	Driving makes me feel frustrated	.744
40	Feel comfortable while driving [-]	.720
4	Feel I have control over driving [-]	.534
25	It worries me when driving in bad weather	.499
33	Feel distressed while driving	.445
41	*Always ready to react to unexpected manoeuvres by other drivers [-]*	*.429*
8	*While driving, I try to relax myself [-]*	*.331*
	Factor 4 | Dissociative Driving	
35	Attempt to drive away from traffic lights in third gear (or on the neutral mode in automatic cars)	.648
27	Forget that my lights are on full beam until flashed by another motorist	.597
39	Nearly hit something due to misjudging my gap in a parking lot	.530
36	Plan my route badly, so that I hit traffic that I could have avoided	.465
34	Intend to switch on the windscreen wipers, but switch on the lights instead	.453
20	*Fix my hair / makeup while driving*	*.508*
21	*Distracted or preoccupied, and suddenly realize the vehicle ahead has slowed down, and have to slam on the breaks to avoid a collision*	*.496*
5	*Drive through traffic lights that have just turned red*	*.447*
19	*When someone tries to skirt in front of me on the road, I drive in an assertive way in order to prevent it*	

L04.2 Principal Component Analysis (PCA)

What about the other way round?

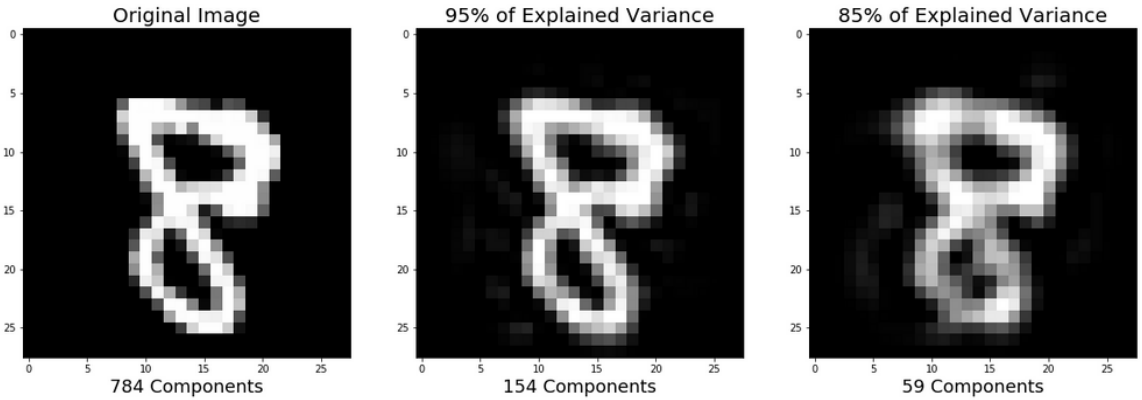
- PCA can also return the compressed representation of the data (lower dimensions) to an approximation of the original high-dimensional data
- due to the compression there is a reconstruction error
- k has also an influence on the recunstruction quality



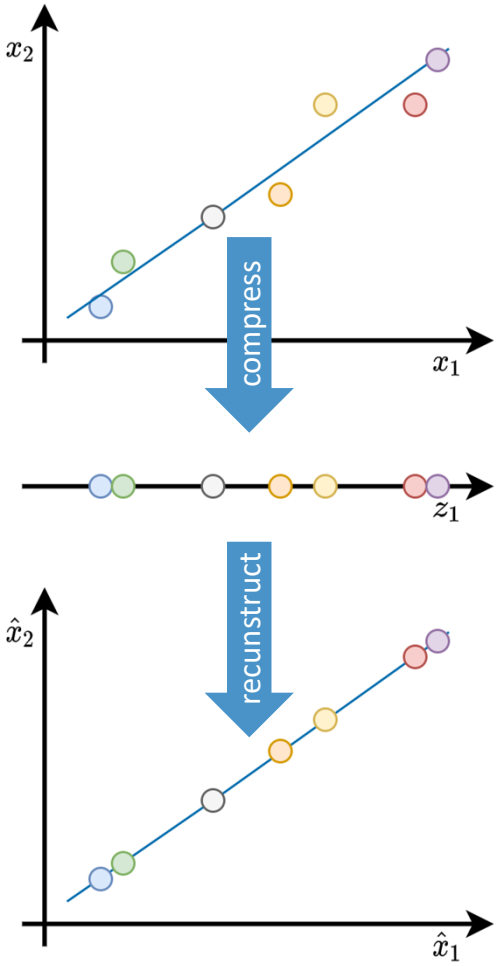
L04.2 Principal Component Analysis (PCA)

What about the other way round?

- PCA can also return the compressed representation of the data (lower dimensions) to an approximation of the original high-dimensional data
- due to the compression there is a reconstruction error
- k has also an influence on the recunstruction quality



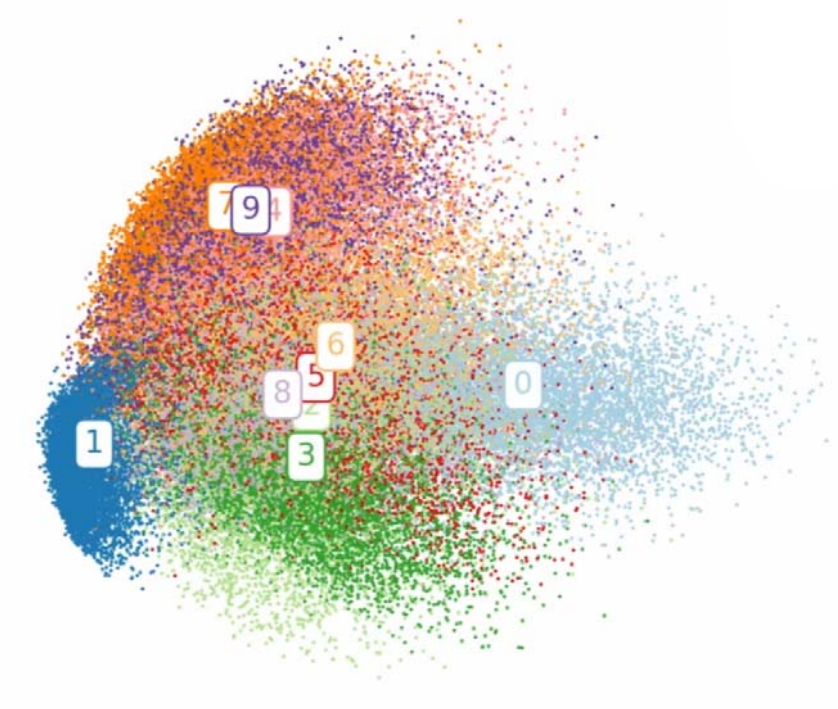
Example taken from [Galarnyk2017]



L04.2 Principal Component Analysis (PCA)

Disadvantages and Limitations of PCA

- Low interpretability of principal components
  - Principal components are only linear combinations of the original data
  - no physical meaning derivable
- Trade-off between information loss and dimensionality reduction
- assumes a correlation between features
- **assumes a linear relationship between features**
- not robust against outliers
- not optimised for 2-dimensions



PCA on MNSIT

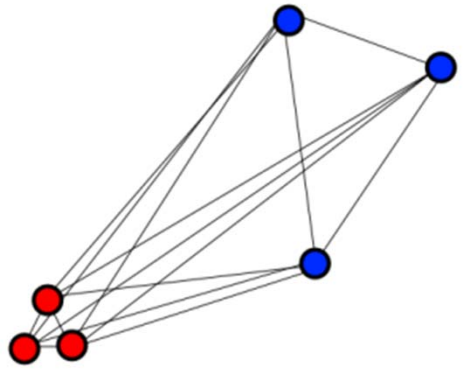
L04.3 T-Distributed Stochastic Neighbour Embedding

T-Distributed Stochastic Neighbour Embedding (tSNE)

- Iterative, non-linear dimensionality reduction

Working principle

- embeds points from a higher dimension into a lower dimension, trying to preserve the neighborhood of these points
- all-vs-all table of pairwise data to data distances
- Shuffles data based on point distances until convergence



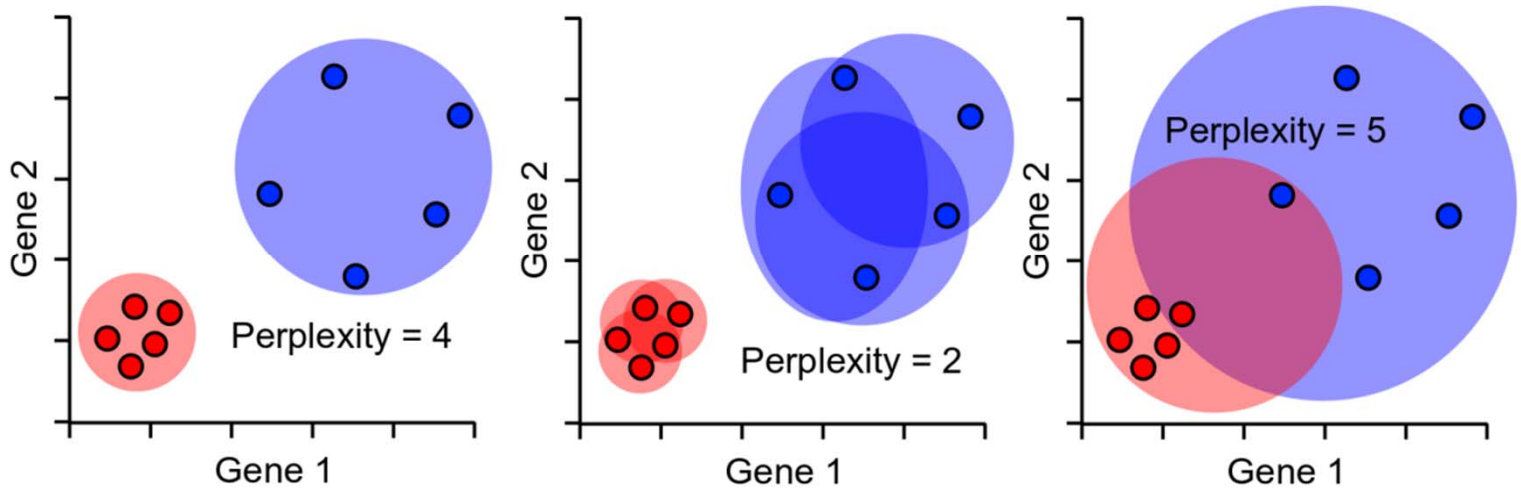
Drawbacks:

- takes a lot of time and space to compute
- has a quadratic time and space complexity in the number of data points

L04.3 T-Distributed Stochastic Neighbour Embedding

Hyperparameter Perplexity

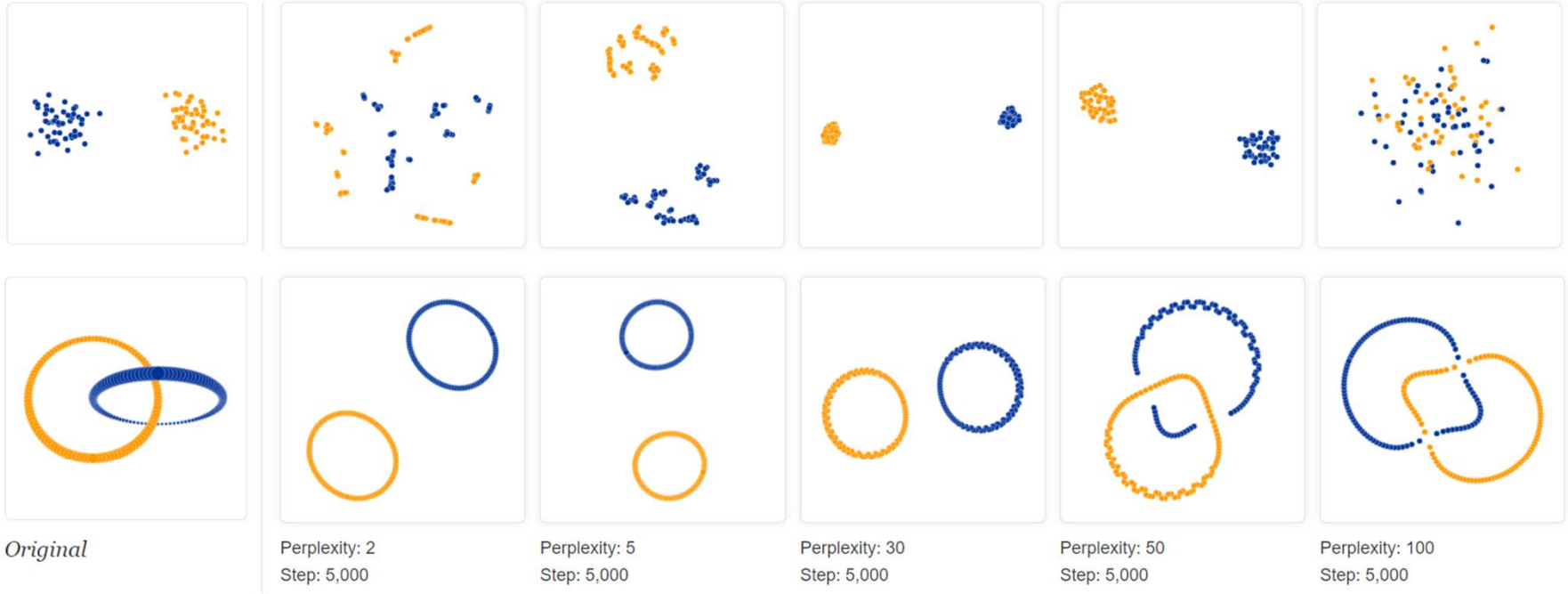
- expected number of neighbors within a cluster
- distances are scaled relative to the perplexity neighbors
- Range: 5 - 50 (suggested by van der Maaten & Hinton)



L04.3 T-Distributed Stochastic Neighbour Embedding

Hyperparameter Perplexity

- Example: tSNE on 100 points (50 per class)



high perplexity values show a kind of global connectivity

L04.3 T-Distributed Stochastic Neighbour Embedding

Iterative Process

- tSNE requires several runs until stability or a maximum number of iterations has been reached
- If strange "squashed" shapes appear, the process was probably stopped too early
- there is no fixed number of steps that leads to a stable result

L04.3 T-Distributed Stochastic Neighbour Embedding

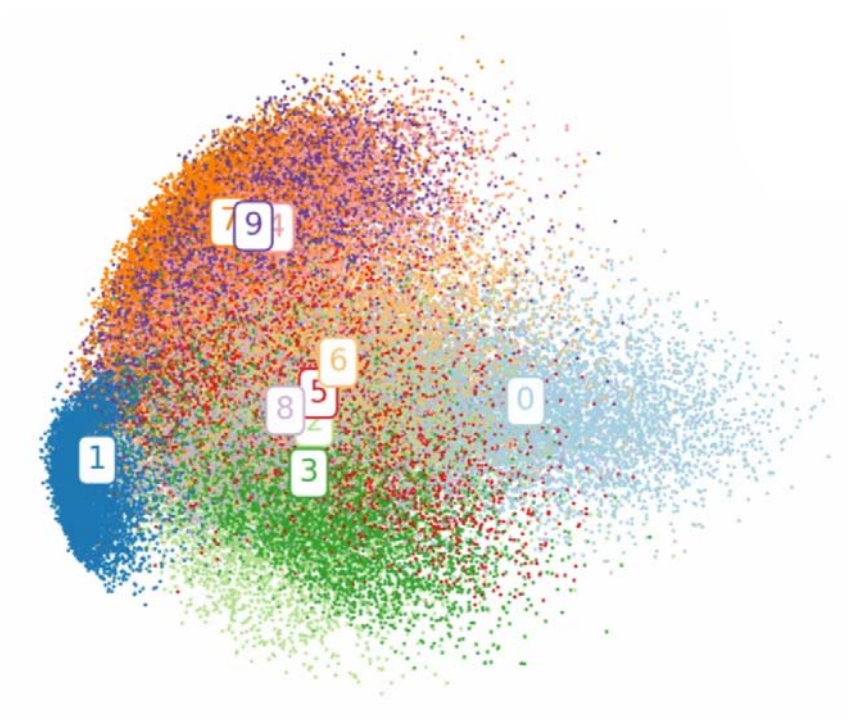
tSNE Projection

- x and y axis have no specific meaning
- point distances have no meaning
- close proximity is very informative (large distances not)

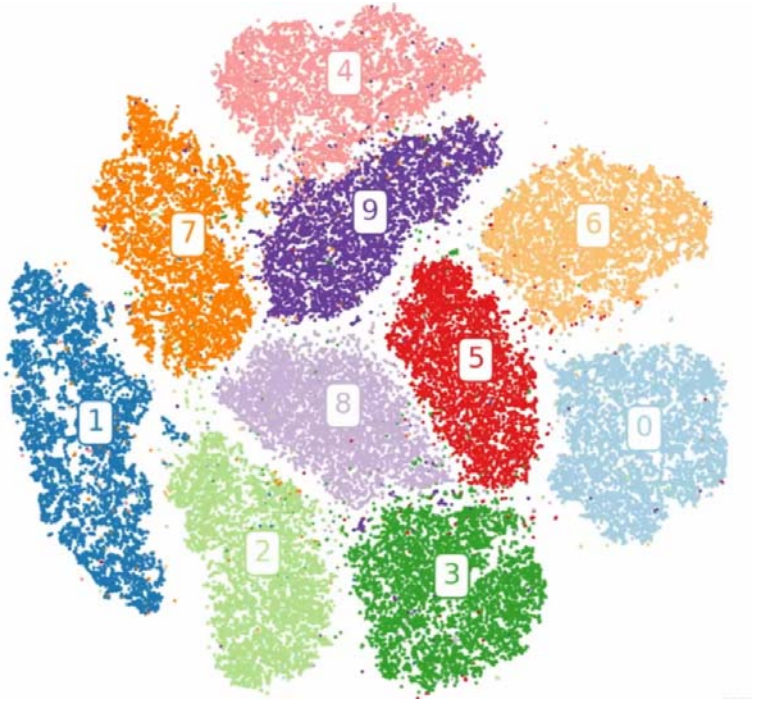
MNIST convergence animation of tSNE [Rossant2015]

L04.3 T-Distributed Stochastic Neighbour Embedding

PCA vs tSNE



PCA on MNSIT



tSNE on MNIST

[Kobak2021]

L04.3 T-Distributed Stochastic Neighbour Embedding

PCA vs. tSNE

PCA	tSNE
linear reduction	non-linear reduction
tries to preserve global structure of the data	tries to preserve local structure (cluster) of the data
no hyperparameters	multiple hyperparameters
affected by outliers	can handle outliers
determinisitc	non-deterministic (randomized)
amount of explained variance can be controled	explained variance canot be influenced

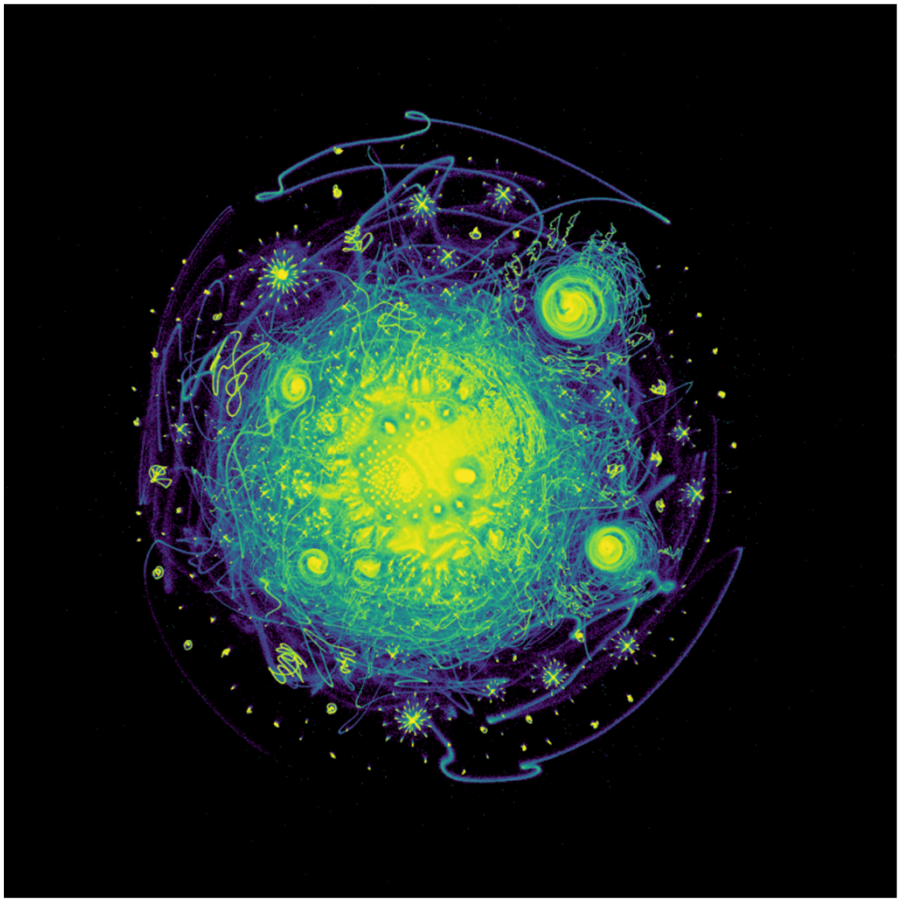
! **No applicability to new/unknown data**
tSNE is non-parametric: the embedding is learned directly by moving the data across the low dimensional space

! **Clustering on tSNE output**
Clustering of the tSNE output can be misinterpreted as it can only be the artifacts of tSNE

L04.4 UMAP

Uniform Manifold Approximation and Projection (UMAP) [McInnes2018]

- **Goal:** create a low-dimensional graph that preserves the high-dimensional clusters and their relationship to each other
- **Iterative process:** low-dimensional points are initialized and reordered pointwise until the same relationships as in high-dimensional space are established
- relationships are measured using Similarity Scores

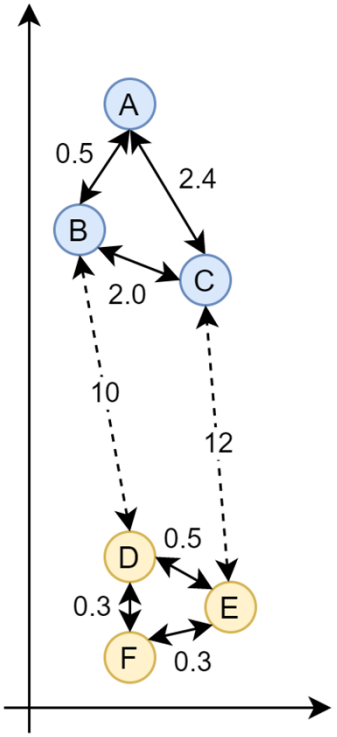


Visualization of 30,000,000 integers as represented by binary vectors of prime divisibility, colored by density of points [McInnes2018]

L04.4 UMAP

Working principle

- To preserve the high-dimensional relationships, Similarity Scores in the high-dimensional space are calculated using
  - the distance metric d
  - the number of neighbors k



HD Similarity Scores:

$$w((x_i, x_{i_j})) = \exp\left(\frac{-\max(0, d(x_i, x_{i_j})) - p_i}{\sigma_i}\right)$$

- $w((x_i, x_{i_j}))$ describes the similarity score between point i and j
- p_i is the distance d to the nearest neighbor

L04.4 UMAP

Working principle

- per definition Similarity Scores of the k neighbors sum up to $\log_2(k)$

HD Similarity Scores:

$$\sum_{j=1}^{k} \exp\left(\frac{-\max(0, d(x_i, x_{i_j})) - p_i}{\sigma_i}\right) = \log_2(k)$$

- To fulfill this condition for each point σ is adjusted
- The resulting Similarity Scores (e.g. A→B and B→A) are not symmetrical and are therefore scaled using the fuzzy union operation

Fuzzy Union Operation:

$$B = A + A^T - A \circ A^T$$

Working principle

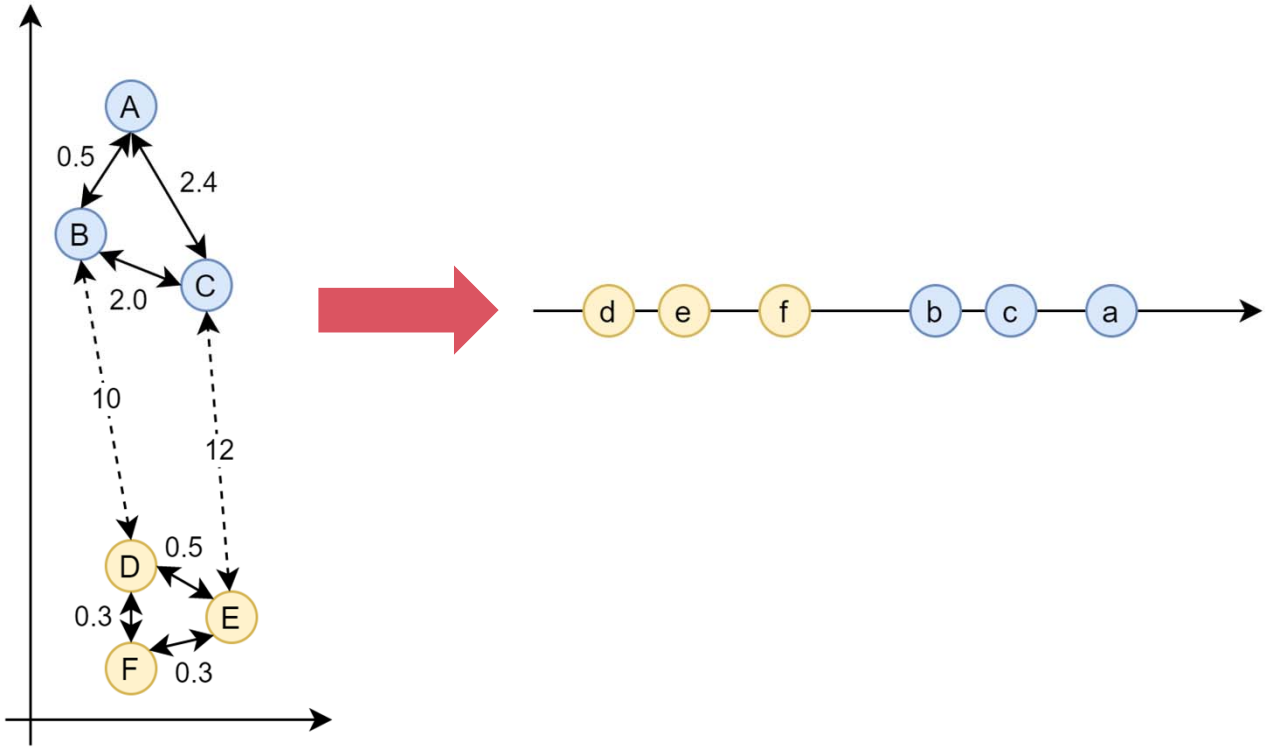
- per definition Simil
- To fulfill this condit
- The resulting Simila and are therefore s

L04.4 UMAP

Working principle

Initialization

- the low-dimensional space gets initialized using Spectral Embedding
- for comparison: tSNE initializes randomly

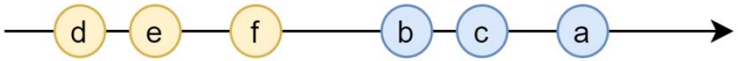


L04.4 UMAP

Working principle

Attraction

- UMAP tries to cluster points together with high Similarity Scores in the high-dimensional space
  - random selection of pair of points proportionally to their high-dimensional Similarity Scores
  - random selection which point gets moved to the other one



Separation

- UMAP tries to separate clusters with low Similarity Scores in the high-dimensional space
  - random picks a point with very low Similarity Score in the high-dimensional space

L04.4 UMAP

Working principle

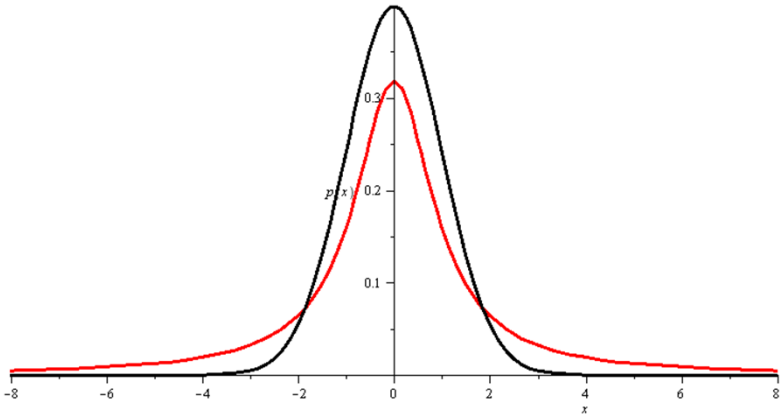
Move the points in the low-dimensional space

- calculation of Similarity Scores in the low-dimensional space
- UMAP uses a fixed t-distribution:

LD Similarity Scores:

$$w((z_i, z_{i_j})) = \frac{1}{1+\alpha d^{2\beta}}$$

- d defines the distance metric in the low-dimensional space



gaussian vs **t-distribution**

- Fun Fact: for $\alpha = 1$ and $\beta = 1$ the Similarity Scores are the same as with tSNE

L04.4 UMAP

Working principle

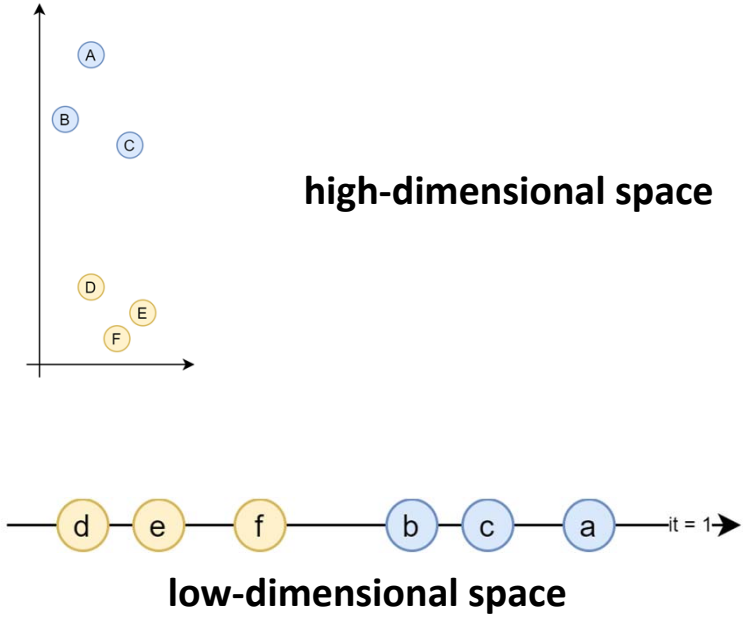
Move the points in the low-dimensional space

- Cost function:

UMAP Cost Function:

$$C = \log\left(\frac{1}{w_{z,\text{neighbor}}}\right) + \log\left(\frac{1}{1-w_{z,\text{notneighbor}}}\right)$$

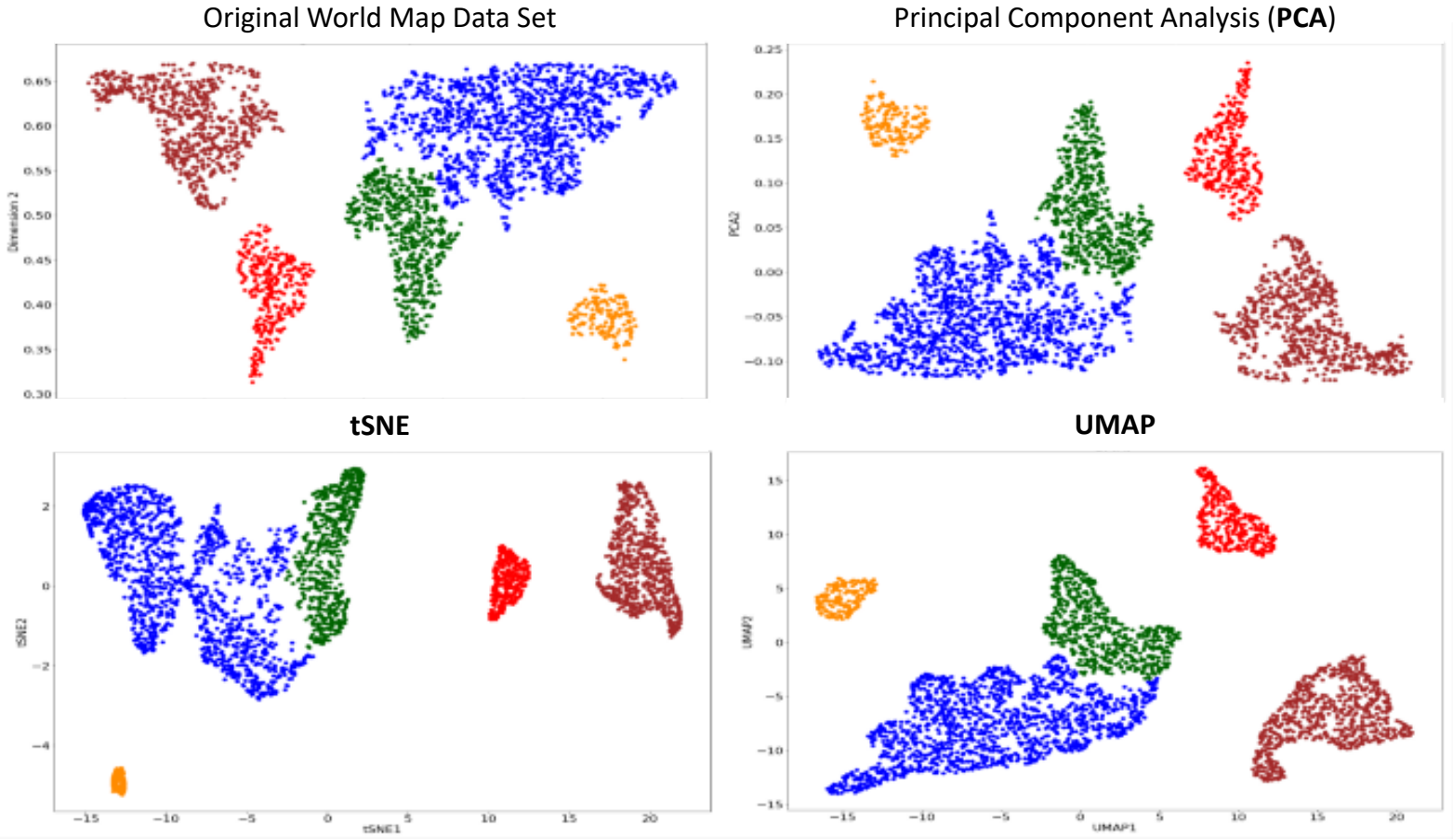
- UMAP solves the cost function using Stochastic Gradient Descent
- only one point is moved per iteration (in contrast, with tSNE every point is moved per iteration)
- per iteration, the point are moved just a little bit



L04.4 UMAP

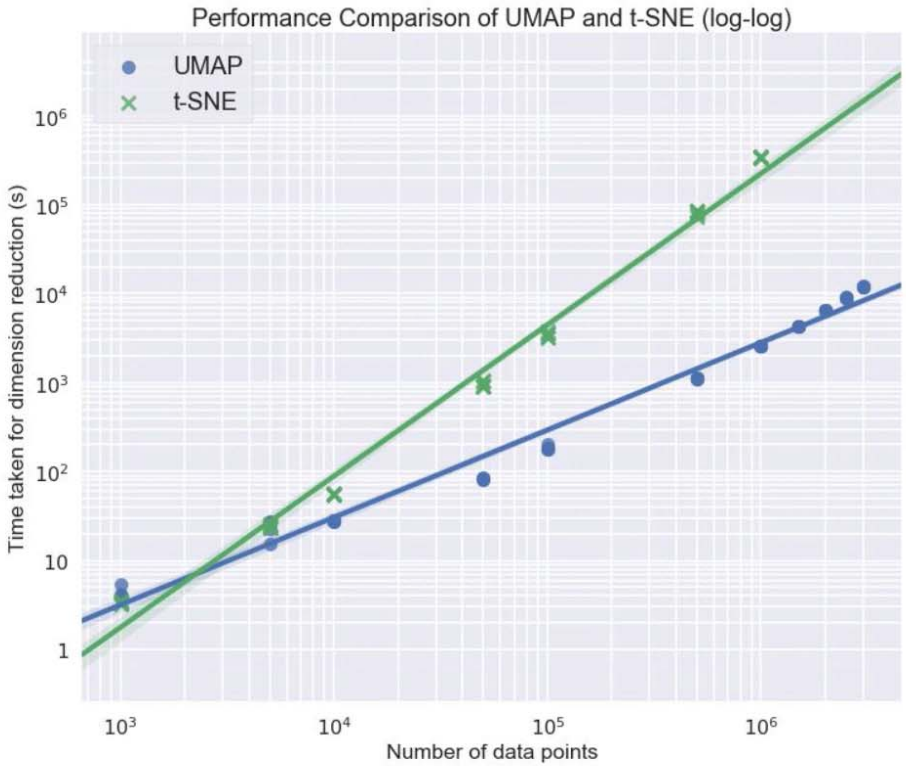
UMAP

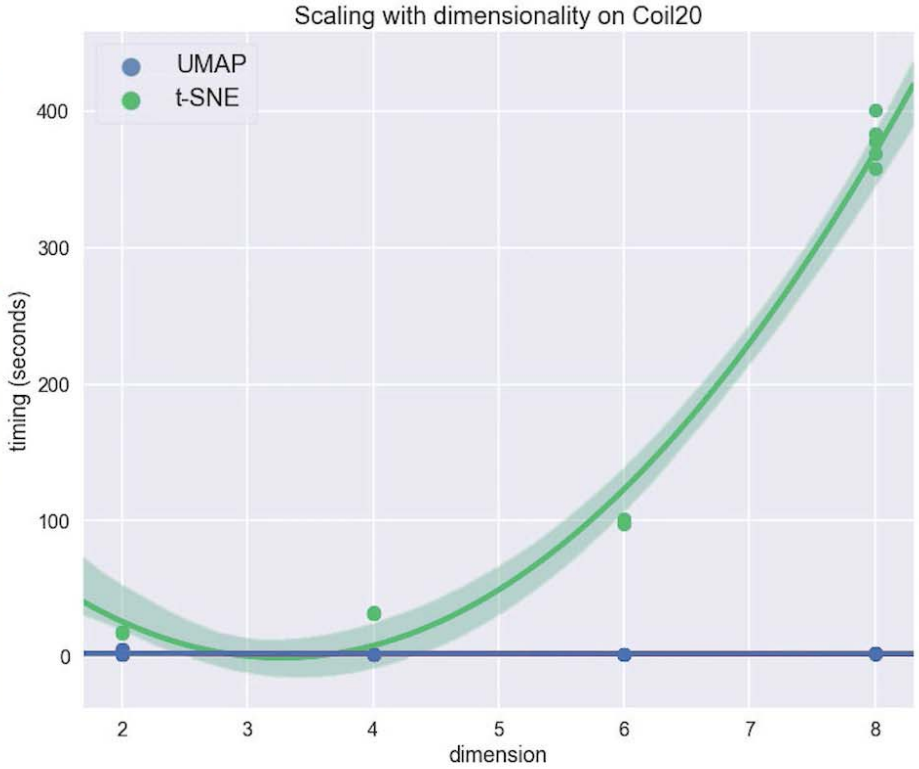
- UMAP is a replacement for tSNE
- Conceptually very similar to tSNE but ways faster (min vs h)
- **Preserve more of the global structure**
- can run on raw data without PCA preprocessing
- **allow new data to be added to an existing projection**



Example taken from [Oskolkov2020]

L04.4 UMAP


Performance Comparison of UMAP and t-SNE (log-log)
UMAP
t-SNE
Time taken for dimension reduction (s)
Number of data points


Scaling with dimensionality on Coil20
UMAP
t-SNE
timing (seconds)
dimension

L04.4 UMAP

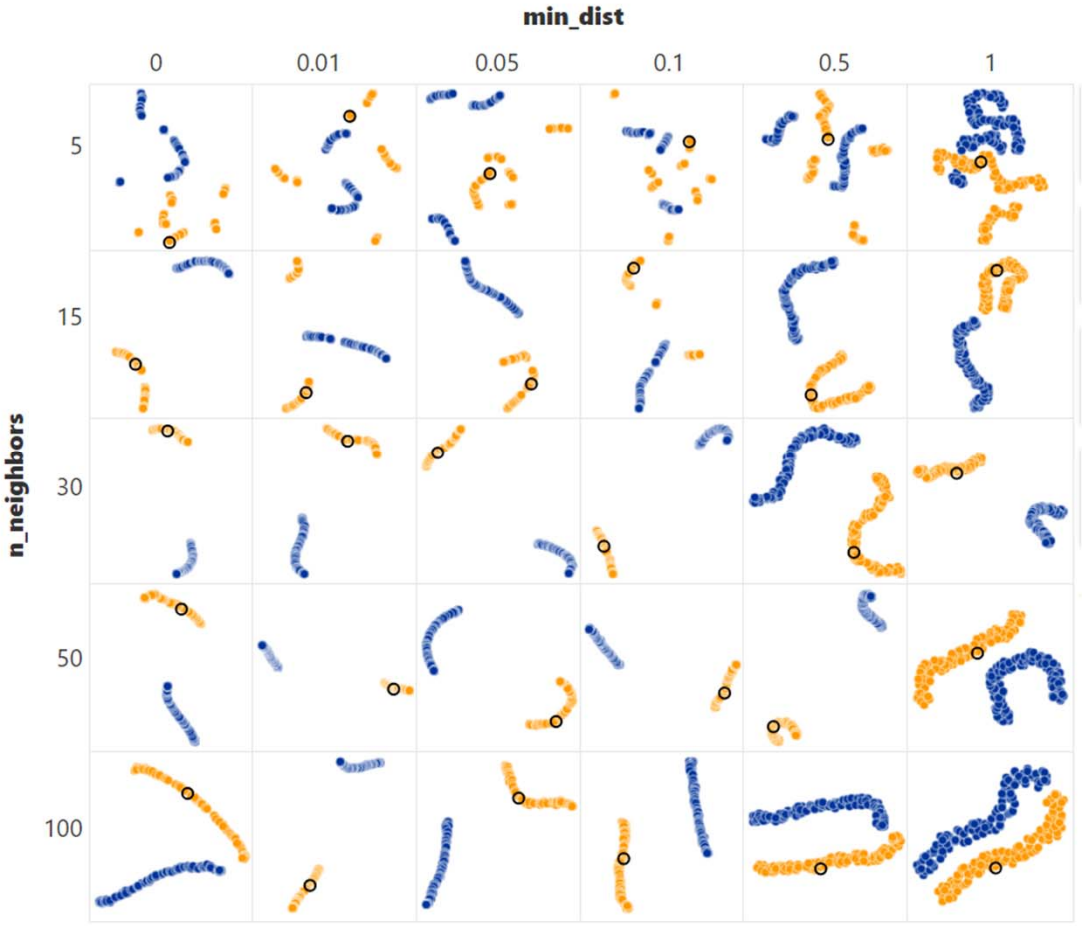
Hyperparameter

Nearest Neighbours

- number of expected nearest neighbours
- basically the same concept as perplexity
- affect the influence given to global vs local information

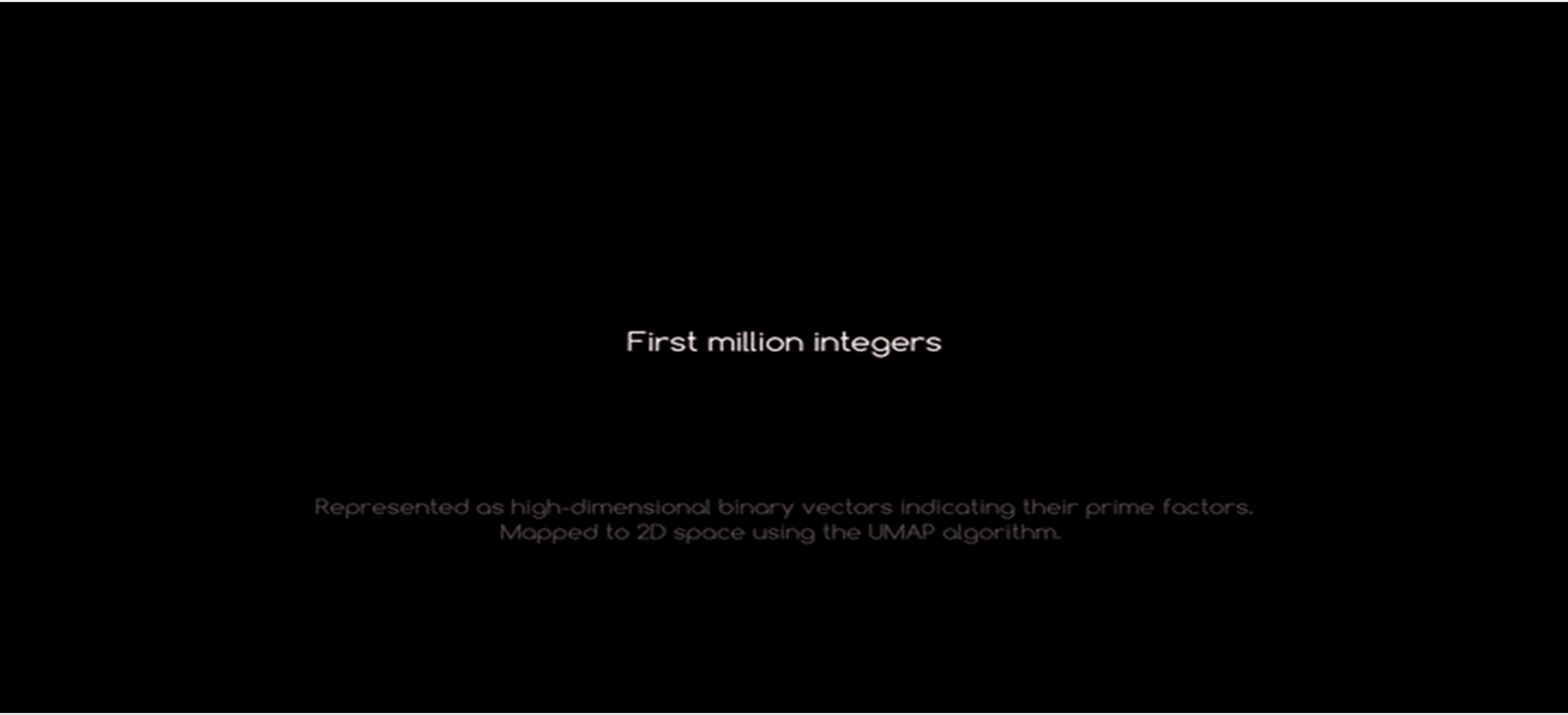
Minimum Distance

- how tightly UMAP packs points which are close together
- affect how compactly packed the local parts of the plot are



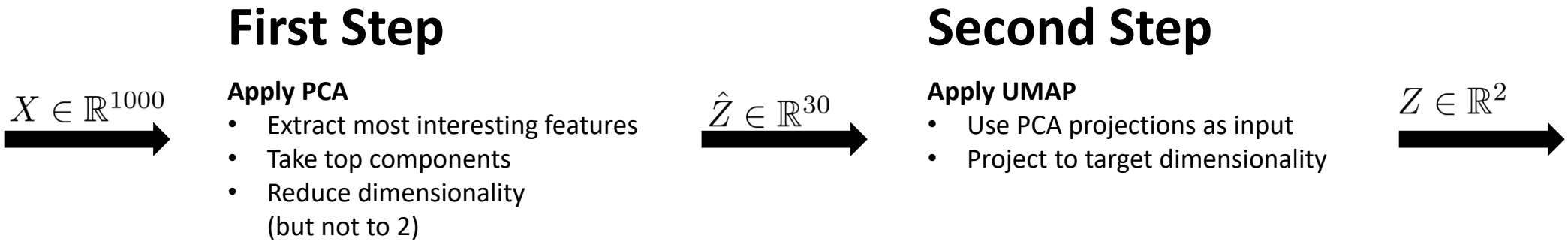
L04.4 UMAP

L04.4 UMAP


First million integers
Represented as high-dimensional binary vectors indicating their prime factors.
Mapped to 2D space using the UMAP algorithm.

L04.5 Practical approaches

PCA preprocessing



PCA Preprocessing
PCA preprocessing can also help tSNE to preserve the global structure!

L04.5 Practical approaches

Real world example: from 134400D to 2D

$X \in \mathbb{R}^{134400}$

Encoder

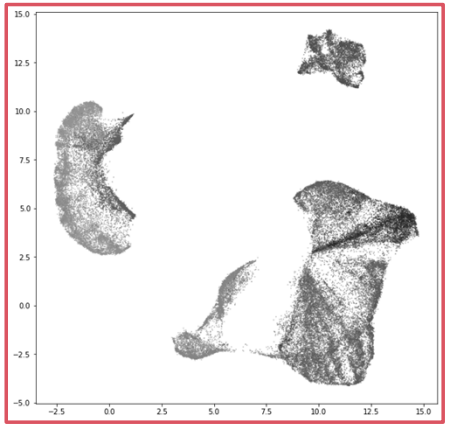
$\hat{Z} \in \mathbb{R}^{2074}$

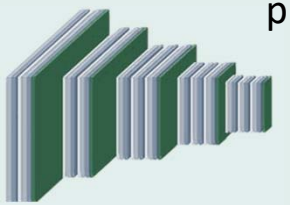
PCA

$\tilde{Z} \in \mathbb{R}^{100}$

UMAP

$Z \in \mathbb{R}^{2}$





CNN based image encoder
pretrained on other task
(e.g. classification)

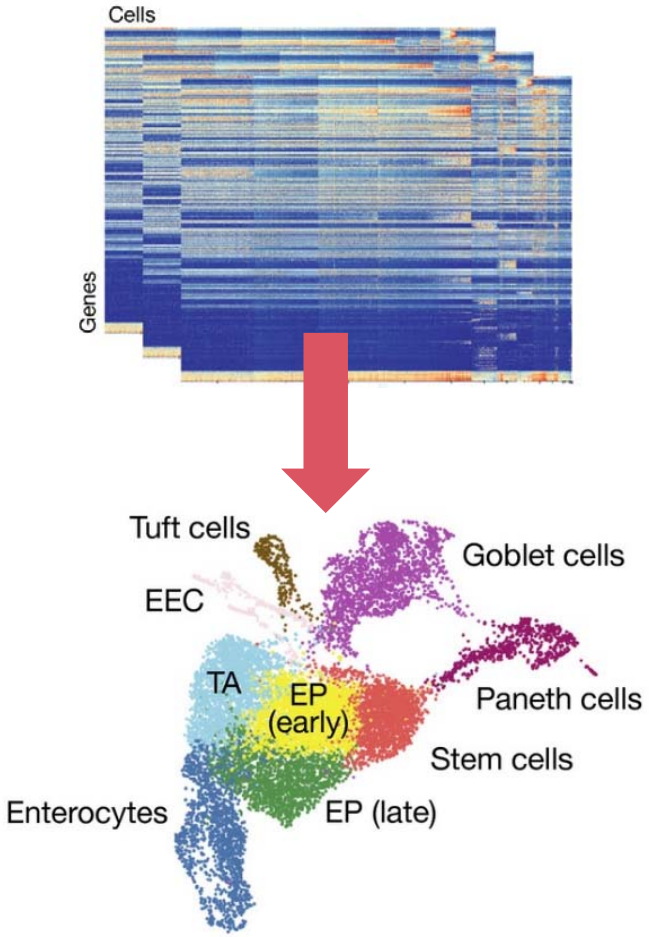
[Lab2022]

L04.6 Honorable Mentions

Single-Cell RNA-Seq Analysis

- very high-dimensional space: specialized solutions required
- however, the methods can also be transferred to other use cases

- **IVIS** [Szubert2019]
  - utilizes a **siamese neural network architecture**
  - minimising a **triplet-loss function**
  - can handle presence of random noise
  - preserves global data structures in a low-dimensional space
  - adds new data points to existing embeddings using a parametric mapping function



L04.6 Honorable Mentions

Trimap [Amid2019]

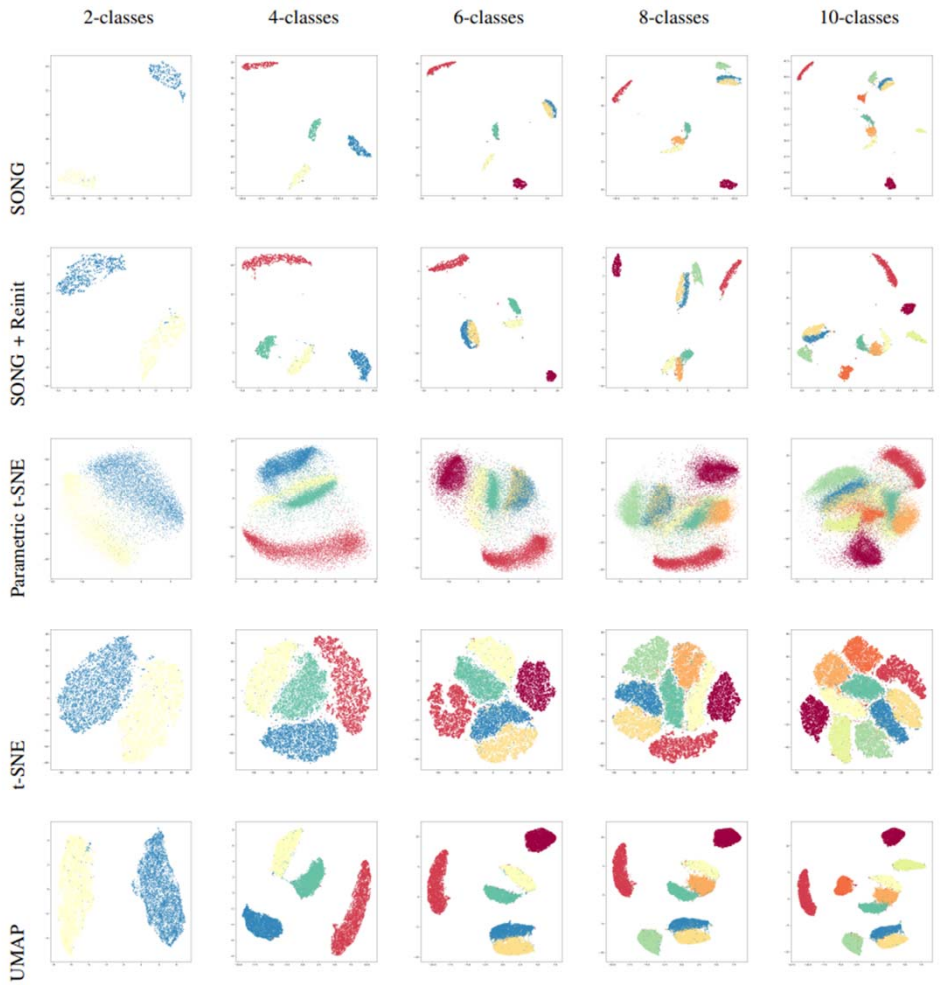
- uses triplet constraints to form a low-dimensional embedding
- The triplet constraints are of the form "point i is closer to point j than point k“
- provides a significantly better global view of the data than the other dimensionality reduction methods such t-SNE, LargeVis, and UMAP
- t-SNE can provide additional insight about the local neighborhood of individual points
- appears to be ineffective when the data is highly non-linear or contains a large amount of outliers

L04.6 Honorable Mentions

Self Organizing Nebulous Growths for Robust and Incremental Data Visualization (SONG) [Senanayake2021]

- parametric nonlinear dimensionality reduction
- provide topology-preserving visualizations of high-dimensional data
- **allowing new data to be mapped into existing visualizations without complete reinitialization**
- robust to noisy and highly mixed clusters
- can handle considerable heterogeneity

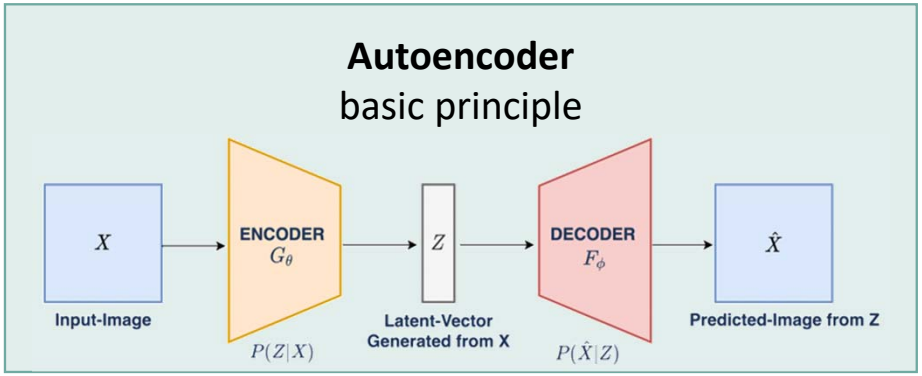
- has a higher computational complexity than UMAP because the high-dimensional parametric graph in the input space needs to be recalculated several times

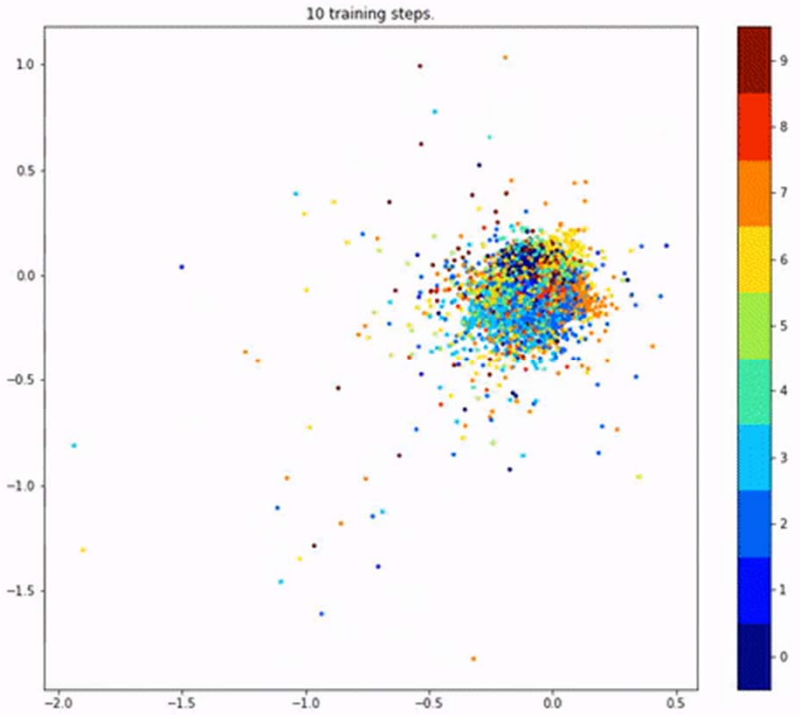


L04.6 Honorable Mentions

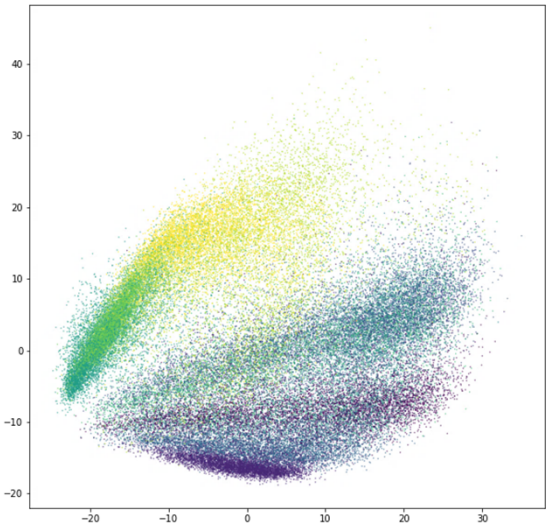
CompressionVAE [Frenzel2020]

- based on the Variational Autoencoder (VAE) [Kingma2014]
- provides a reversible and deterministic function, mapping from data space to embedding space
- fast and scales much better to large datasets, and high dimensional input and latent spaces

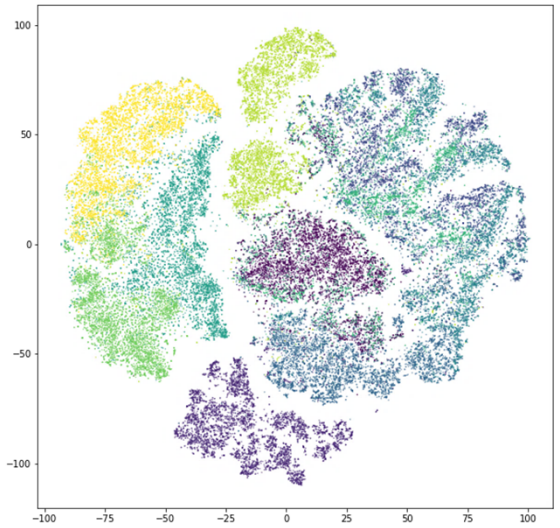




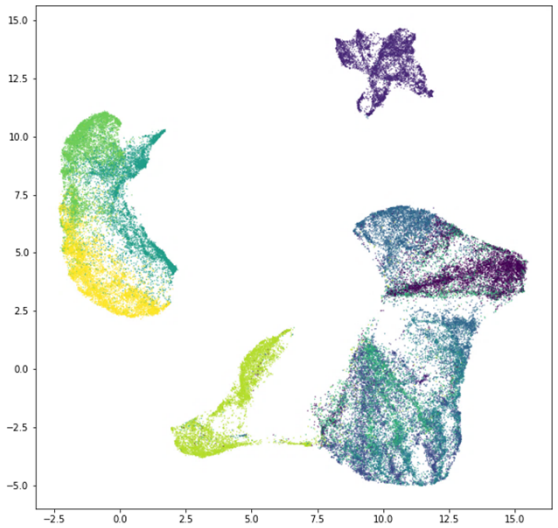
L04.7 Fit on FashionMNIST



PCA

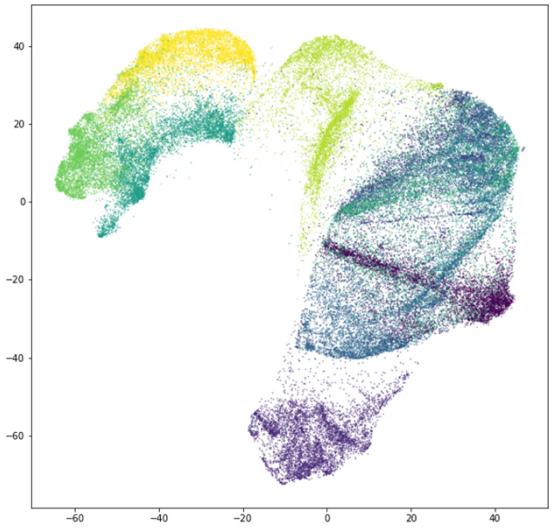


tSNE

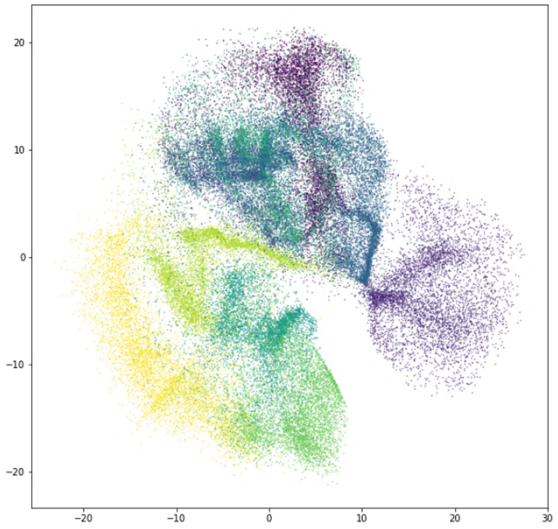


UMAP

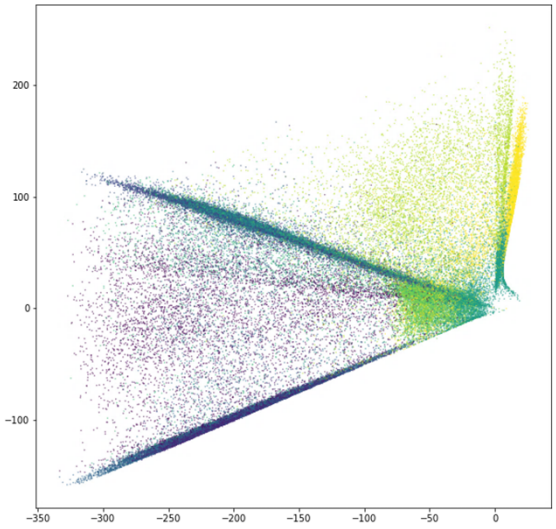
L04.7 Fit on FashionMNIST



TRIMAP



IVIS



CompressionVAE


End of Presentation
ANY QUESTIONS?
memecreator.org

www.hs-kempten.de/ifm