UNIVERSITY OF BERGEN

INFO 284 – Machine Learning
Dimension Reduction

Bjørnar Tessem

UNIVERSITY OF BERGEN

Learning in AI

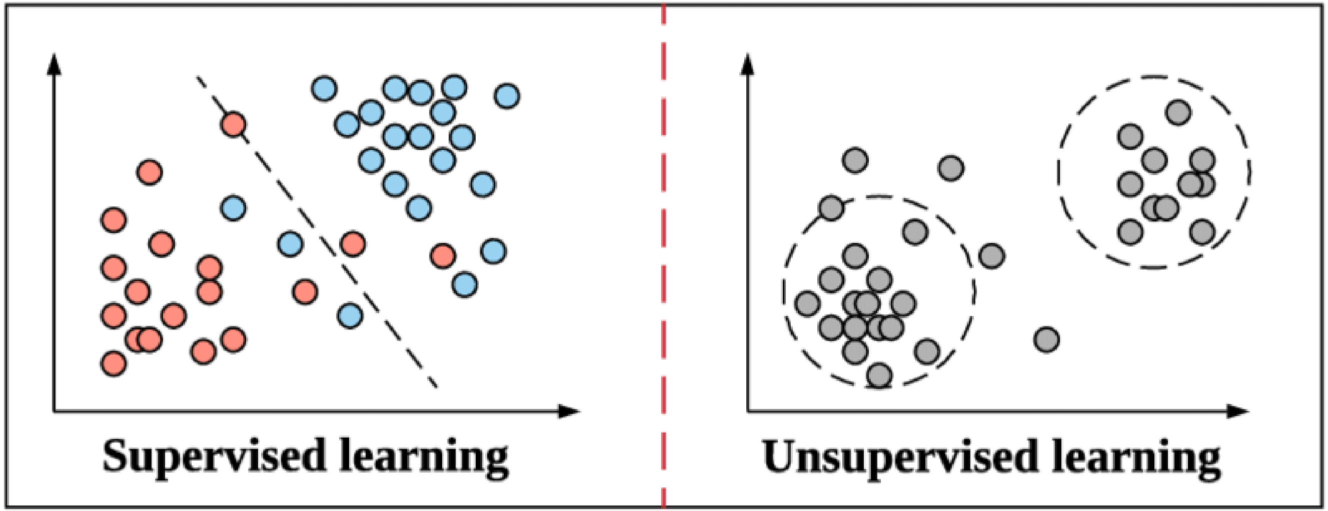
Any component of an intelligent system can be improved by learning from data. Learning depends on four factors:

- which component is to be improved
- what prior knowledge the agent has
- **what representation is used for the data and the component**
- **what feedback is available to learn from**

Available feedback

- Supervised learning - correct/wrong
- Unsupervised learning - no feedback
- Self-supervised learning
  - Generate labels from original data
  - Predict some parts of the data from other parts
- Reinforcement learning - reward/punishment

Unsupervised learning



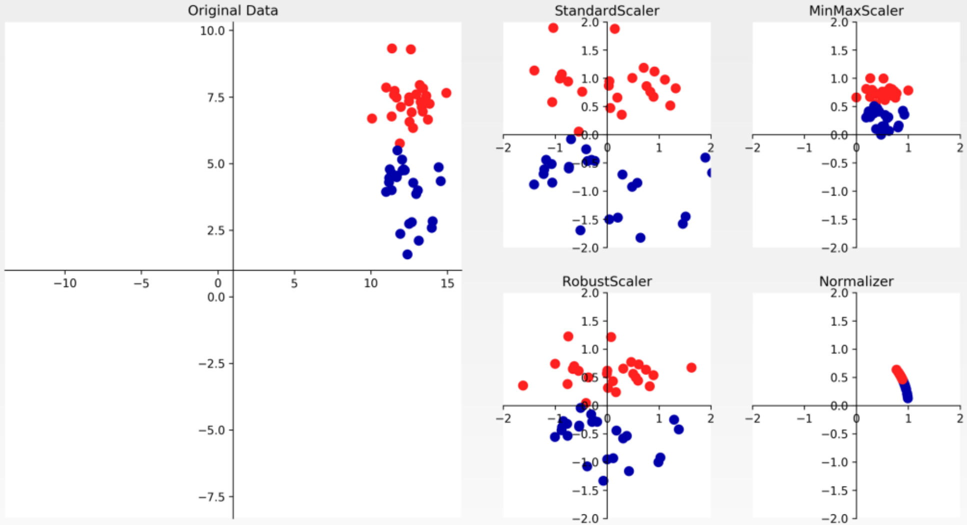
Types of unsupervised learning

- Clustering
  - Find inherent groupings in the data
    - grouping customers by purchasing behaviour
    - news by topic
- Unsupervised transformations
  - Create new representations of the data
    - easier to visualise
    - easier to learn from
  - Dimensionality reduction

Dimensionality reduction

- Scaling
  - Ensuring all features are on same/similar scale
- Principal Component Analysis
  - Extract numerical features to represent data
- Non-negative Matrix Factorisation
  - Extract **positive valued** numerical features
- t-distributed Stochastic Neighbor Embedding
  - Two- or three-dimensional representations for visualisation

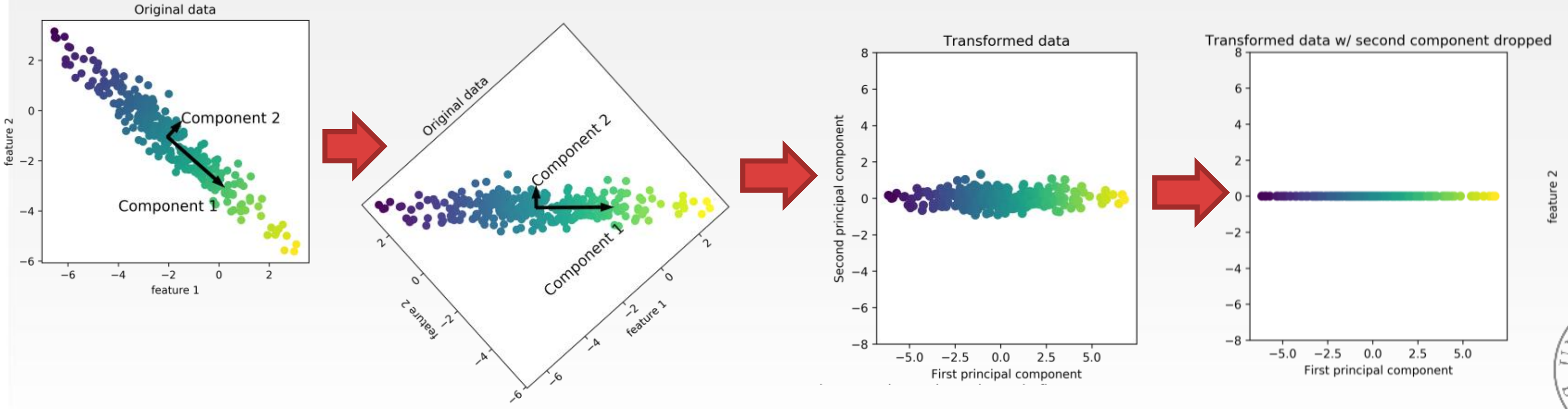
Scaling



- Standard scaler
  - transform to standard Gaussian
- MinMax scaler
  - transform to numbers in [0,1]
- Robust scaler
  - transform based on quartiles
- Normalizer
  - transform so that data vector has length (norm) 1

Principal Component Analysis

- Compute alternative feature set
- Rotate dataset so that the new features are statistically uncorrelated



Mathematics

- How much to rotate in each dimension?
- Data: $\begin{array}{cc} \begin{bmatrix} \quad X \quad \end{bmatrix} & n \\ m & \end{array}$ n datapoints, m features
- Covariance matrix: $X^TX \rightarrow \begin{array}{cc} \begin{bmatrix} \quad C \quad \end{bmatrix} & m \\ m & \end{array}$
- Eigenvectors: Find w-vectors such that $\lambda w = Cw$
  - λ-s are called eigenvalues
  - w-vectors are called eigenvectors

Use

- Place all eigenvectors in own matrix $W = [w_1, w_2, \ldots, w_m]$
  - rank by λ (eigenvalue)
- Transform all datapoints x to

$$t = xW$$

  - To reduce dimension use only the n first eigenvectors of W
  - The chosen eigenvectors are called principal components
  - t becomes an n-dimensional vector called scores
  - W columns are called loadings

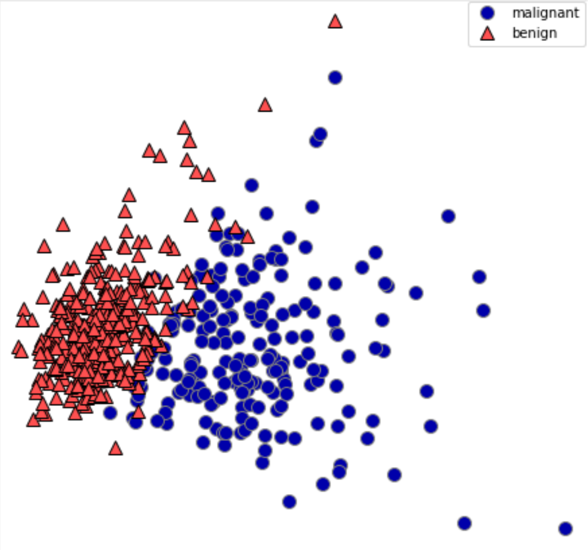
Visualisation

- Pair plots

```
from sklearn.decomposition import PCA
# keep the first two principal components of the data
pca = PCA(n_components=2)
# fit PCA model to breast cancer data
pca.fit(X_scaled)

# transform data onto the first two principal components
X_pca = pca.transform(X_scaled)

Original shape: (569, 30)
Reduced shape: (569, 2)
```



- The two axes in the plot are often not easy to interpret
  - uncorrelated combinations of the original features

PCA for feature extraction

- Find a representation of data that is better suited to analysis than the raw representation.

PCA in scikit-learn for feature extraction

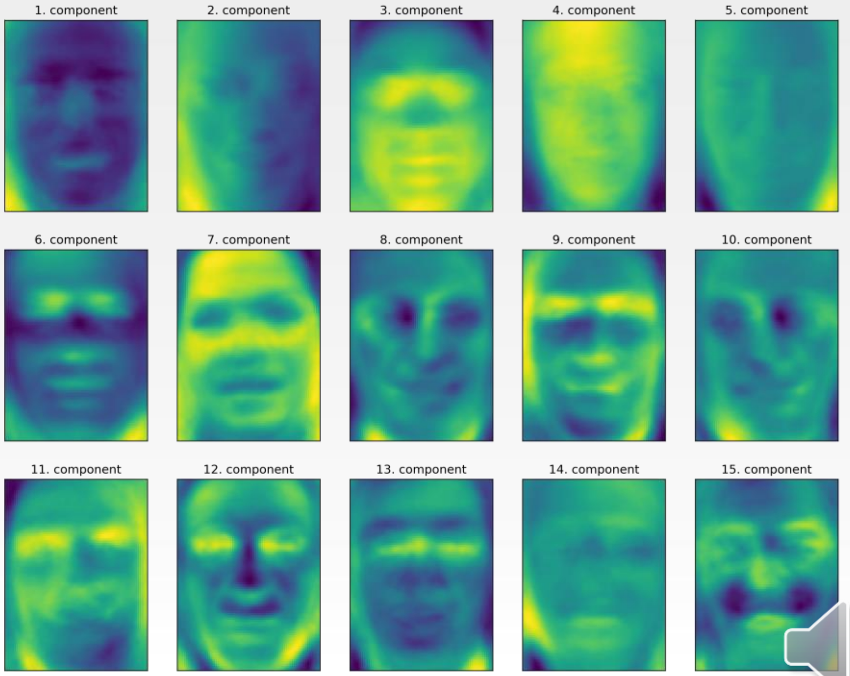
- Problem: Does a previously unseen face belong to a known person from the database?
- Approach 1: build a classifier where each person is a separate class
  - Problem:
    - many different people
    - not that many pictures of each => hard to train a classifier
- Approach 2: Classifier that **can** be trained is kNN.
  - Computing distances between pixels is not very informative
  - Using PCA of images can help

PCA for feature extraction

```
pca = PCA(n_components=100, whiten=True, random_state=0).fit(X_train)
X_train_pca = pca.transform(X_train)
X_test_pca = pca.transform(X_test)


knn = KNeighborsClassifier(n_neighbors=1)
knn.fit(X_train_pca, y_train)
```

Scaling principal components



Schematic view of PCA

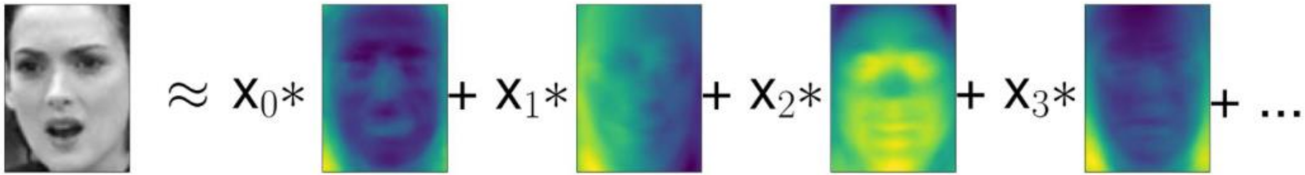
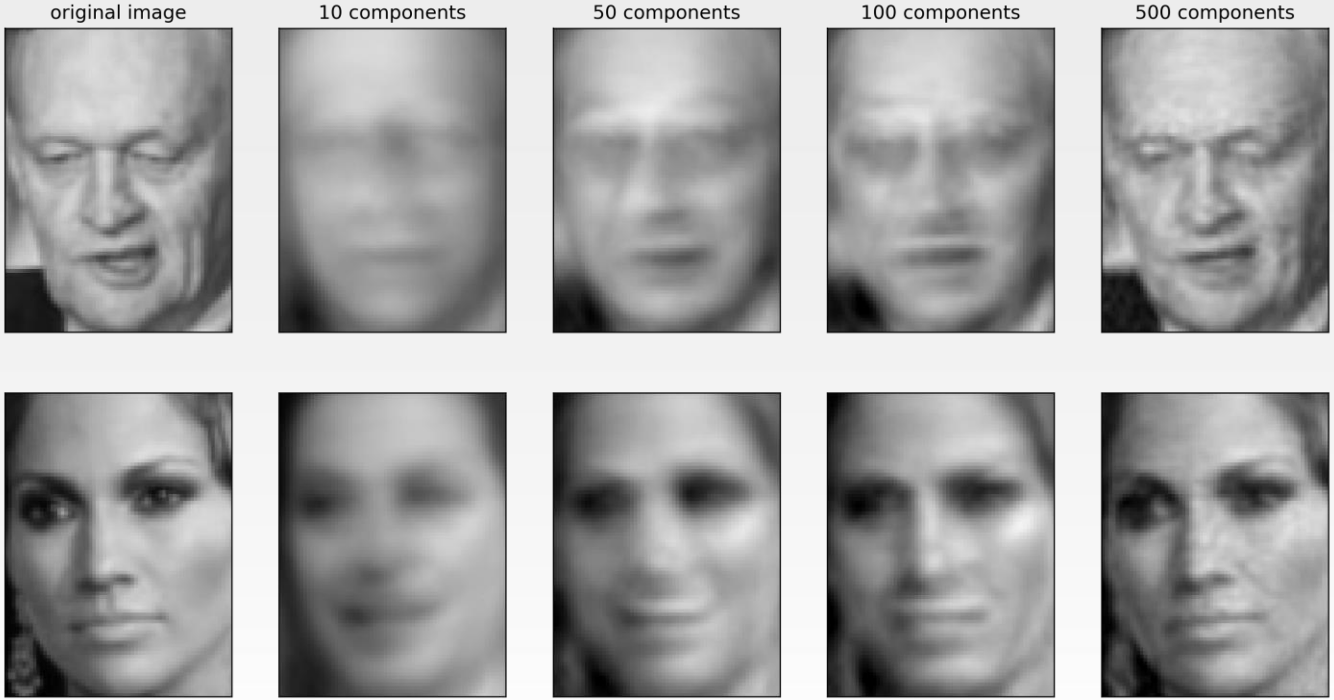
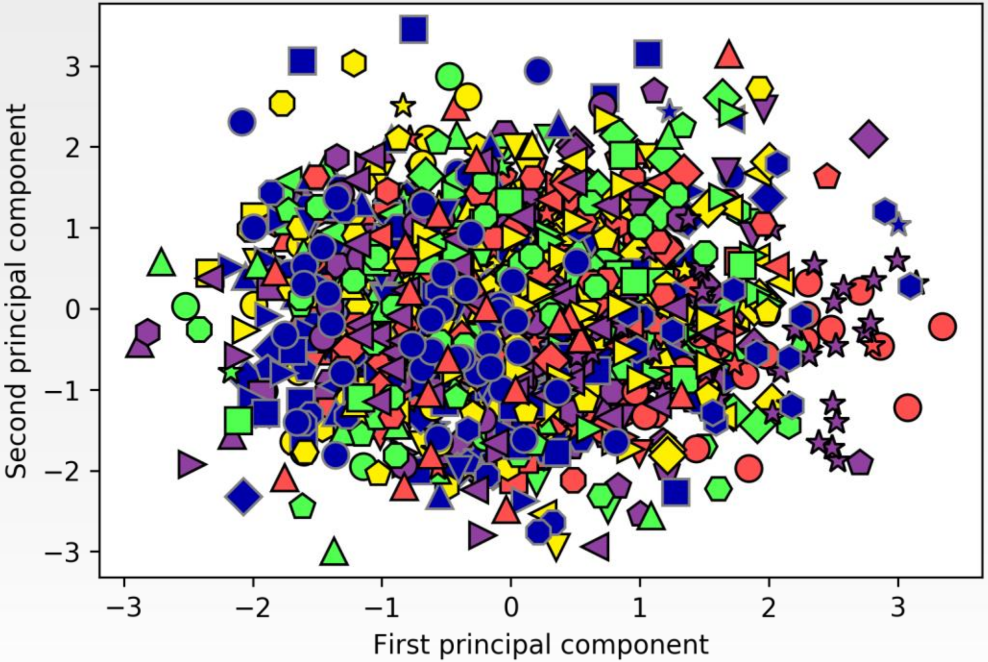


Figure 3-10. Schematic view of PCA as decomposing an image into a weighted sum of components

Reconstructing images



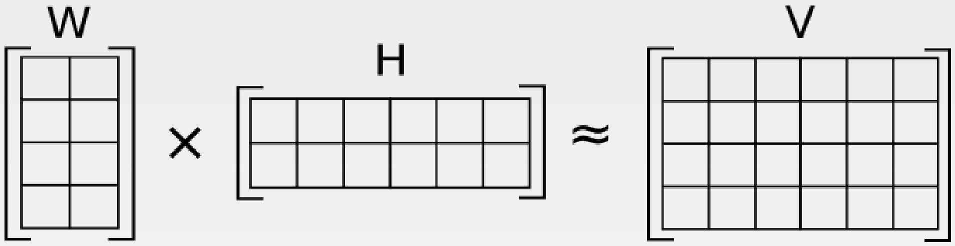
Plot of first two principle components



Non-negative Matrix Factorization

- Establish new set of features
- Similar to Principal Component Analysis
  - PCA: ortogonal components
- NMF: positive components and coefficients
  - Not ortogonal
  - $t = x \cdot W$
  - x and W only positive entries

Non-negative Matrix Factorisation



The matrix **V** (the original dataset) is represented by the two smaller matrices **W** and **H.** **W** x **H** approximately reconstruct **V**.

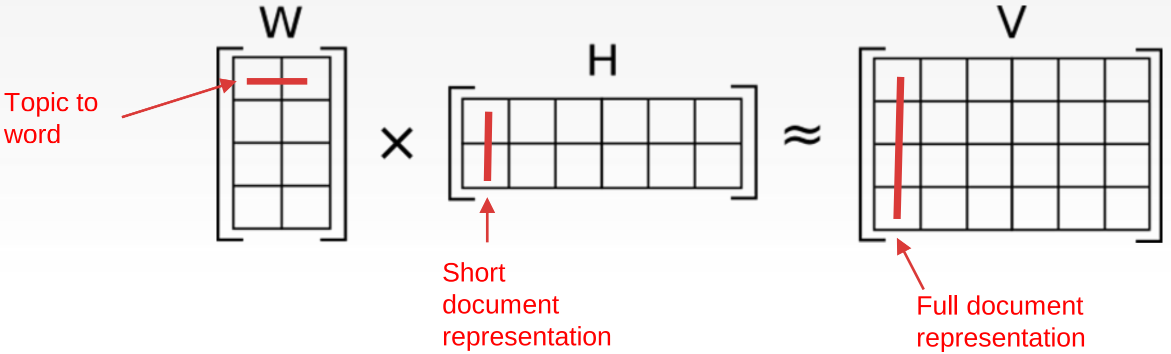
We want to transfer **V** to **H** to reduce dimensionality.

Each row in **H** represent the scores on a particular «hidden» feature. Each column is a datapoint.

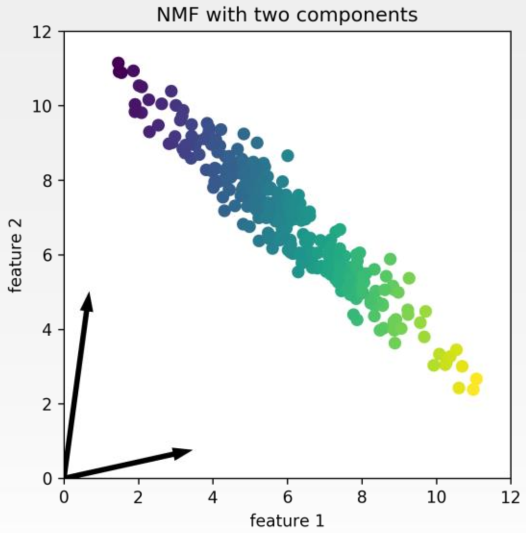
Each column in **W** tells how we transform a «hidden» feature to the observed data found in **V**

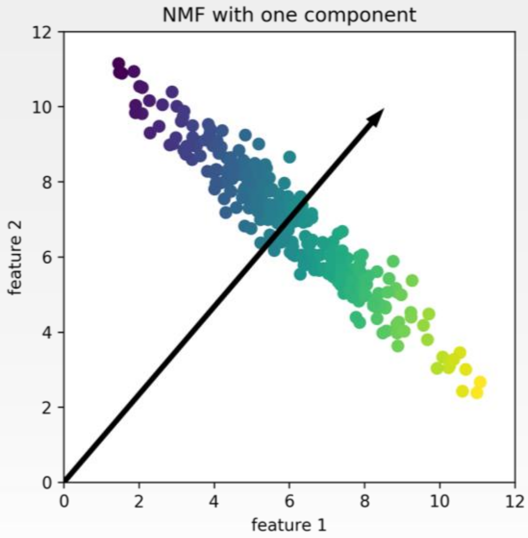
Topic modeling

- Popular approach in document analysis
- Uses NMF technique
- Represent documents as one-hot-coded document representations (or TF-IDF)
- Convert to a combination of a few topics.

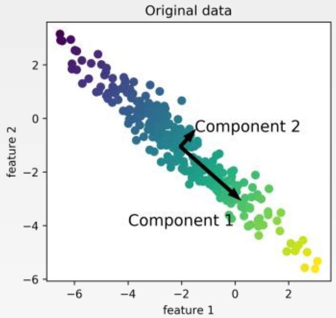


Non-negative Matrix Factorisation





PCA



Faces in MNF components

High scorers on component 3

High scorers on component 7

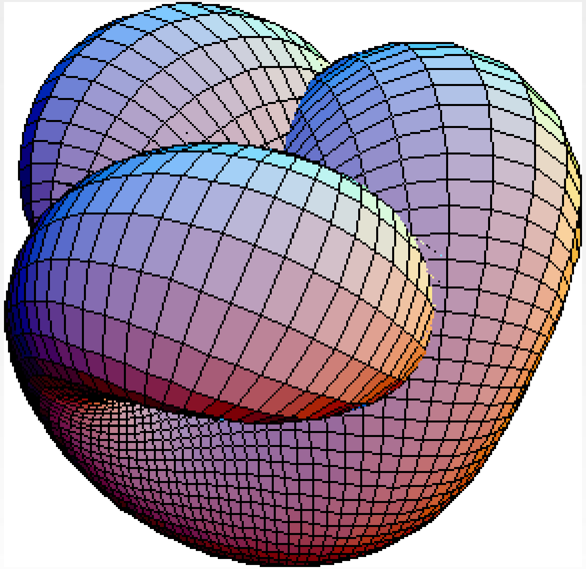
t-distributed Stochastic Neighbor Embedding

- t-SNE
  - tool for visualisation of many-dimensional data sets
- PCA maintains variance in data
  - Pairwise distances in data are preserved
- t-SNE
  - Preserve only short pairwise distances
  - Close data points in full feature space features will be close in 2(3) dimensions
  - «Trained» by gradient descent
  - New data → new training

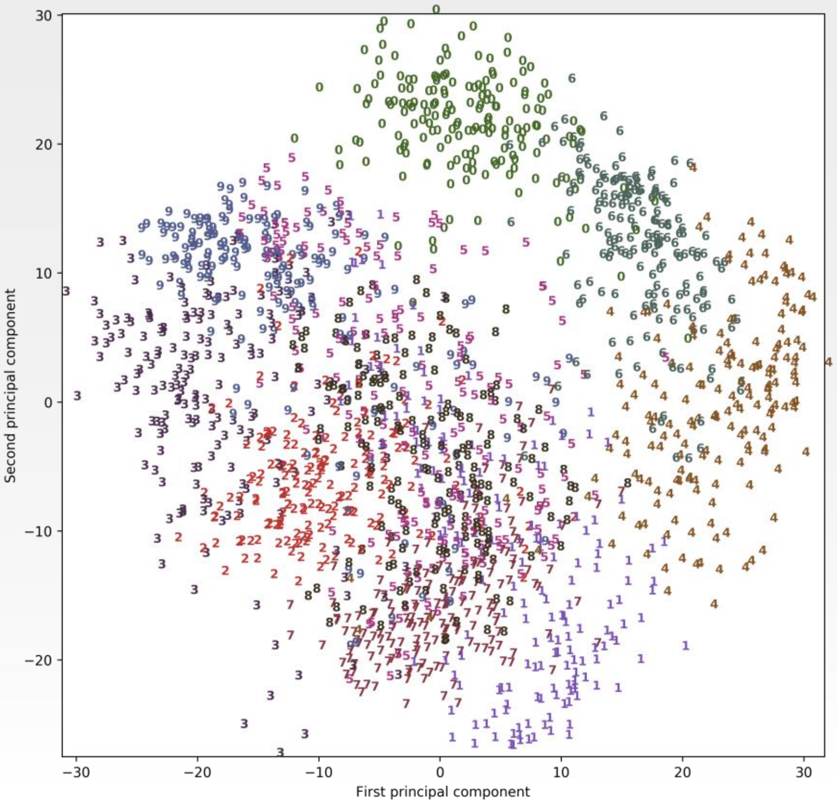
Manifolds

Data is often placed within particular continuous subsets (surfaces) of the full R^n feature space.

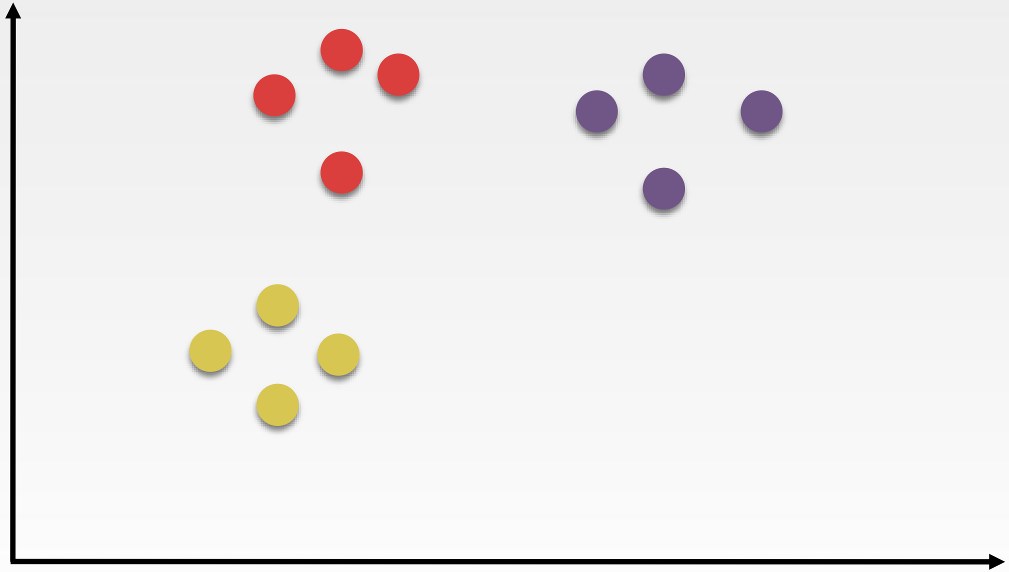
Visualise by reducing this data MANIFOLD to two-dimensions



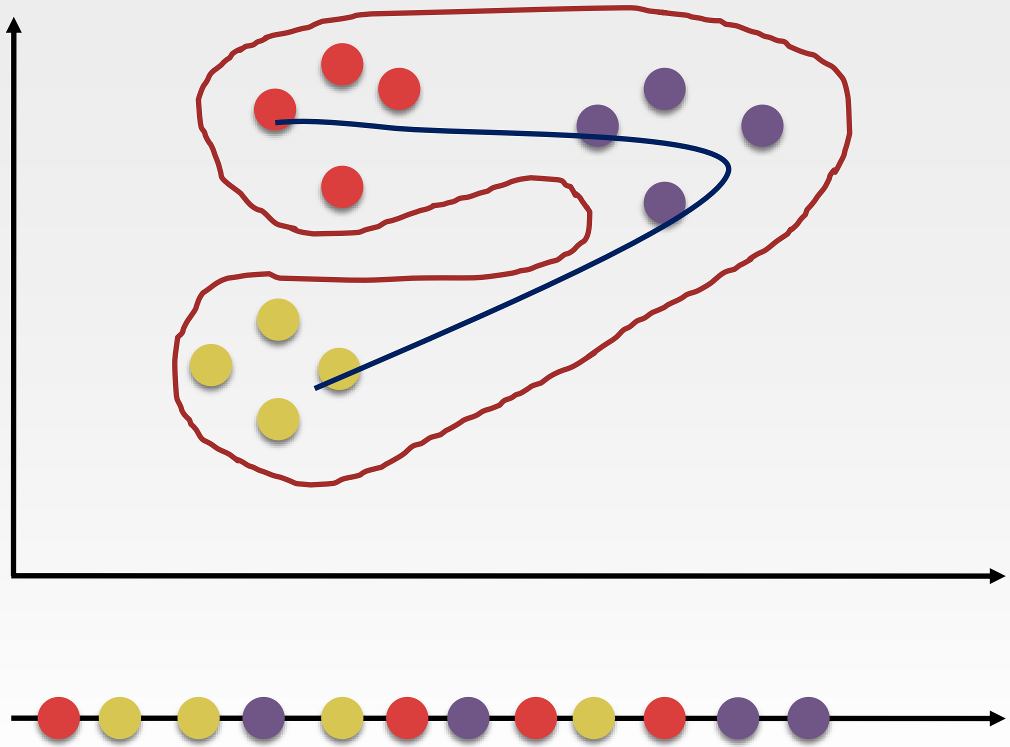
Plot of only the first two principle components MNIST



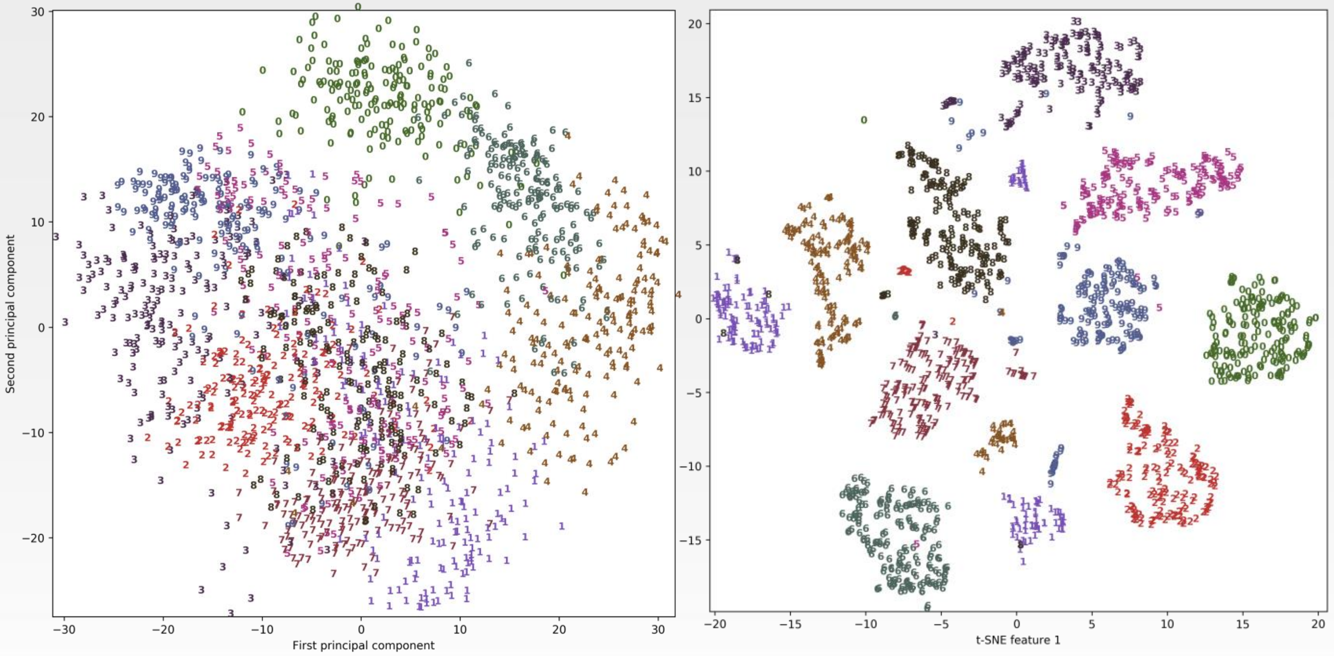
Basic ideas of t-SNE



Basic ideas of t-SNE



Plot MNIST using t-SNE



https://www.youtube.com/watch?v=RJVL80Gg3lA

UNIVERSITAS BERGENSIS

uib.no