Computer-Linguistische Anwendungen

CLA | B.Sc. | LMU

Visualization

Visualization

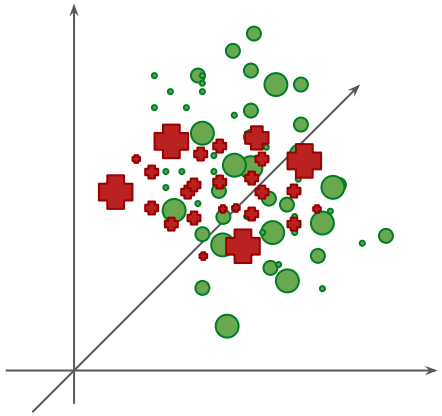
Wie kann man Embeddings verstehen / interpretieren?

- Oft verwendet: zweidimensionale Projektion
  - PCA (Principal Component Analysis)
  - t-SNE (t-distributed stochastic neighbor embedding)
  - Gensim library

- Wichtig: Wenn ein hochdimensionaler Raum auf 2 Dimensionen projiziert wird, sind die Dimensionen nicht mehr interpretierbar

Visualization: PCA

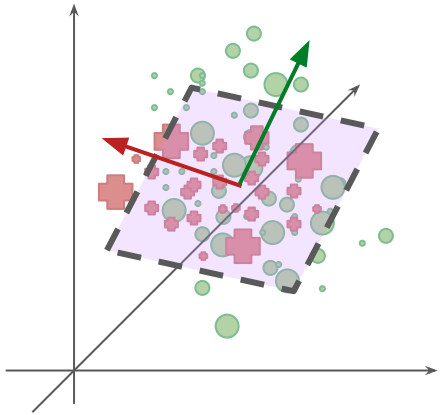
Principal Component Analysis

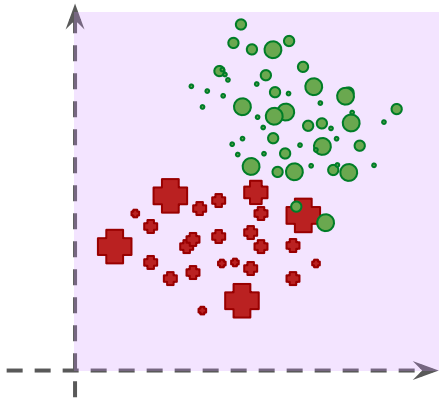


3 Dimensions (or more)

Visualization: PCA

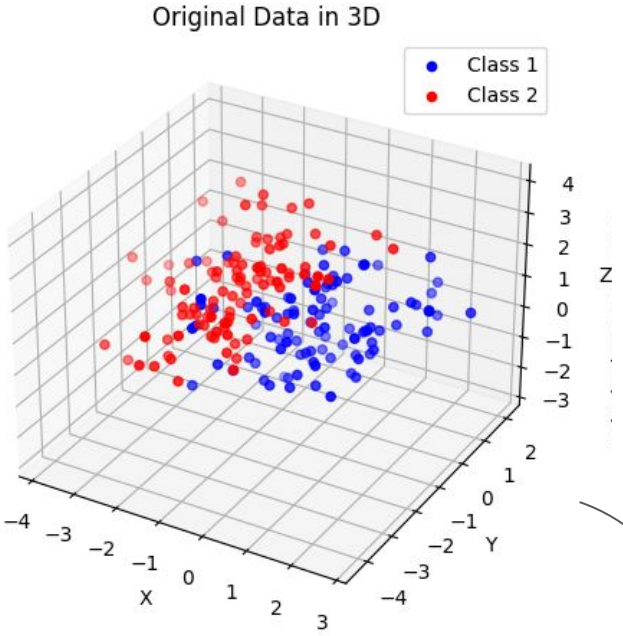
Principal Component Analysis

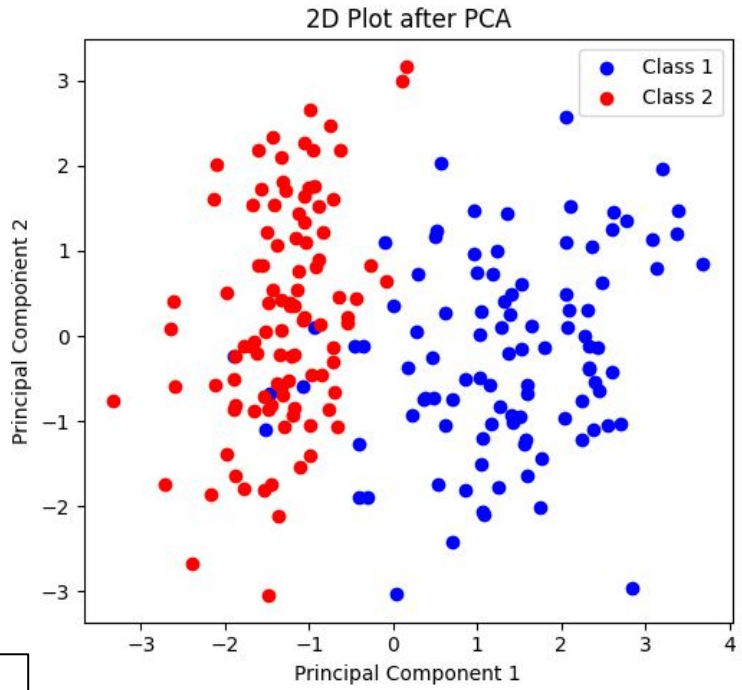




PCA findet neue Achsen, die die **größte Varianz** der Daten erklären, und **projiziert die Daten** auf diese Achsen, um sie in einem neuen Koordinatensystem darzustellen. Dies ermöglicht eine einfachere Interpretation und Visualisierung der Daten.

Visualization: PCA






```
from sklearn.decomposition import PCA
pca = PCA(n_components=2)
X_pca = pca.fit_transform(X)
```

Visualization: t-SNE

- t-SNE (t-Distributed Stochastic Neighbor Embedding)
- Bekannte Technik für Dimensionsreduktion, oft verwendet in Machine Learning und Daten-Visualization.
- Idee:
  - Miss die Ähnlichkeit zwischen Datenpunkten im höher-dimensionalen Raum und übertrage sie in den niederdimensionalen Raum auf eine Art und Weise, **welche die Ähnlichkeiten möglichst beibehält.**
  - Dies wird durch Wahrscheinlichkeiten, welche die Datenpunkte des Models abbilden, ermöglicht.

Visualization: t-SNE

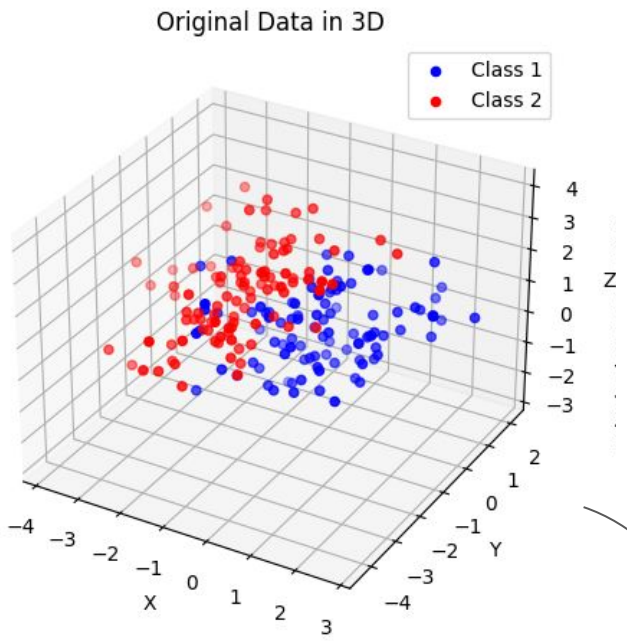
1. Start by calculating **pairwise similarities between all data points** in the high-dimensional space. Similarity is usually measured using a Gaussian distribution, where nearby points have a higher similarity than distant points.

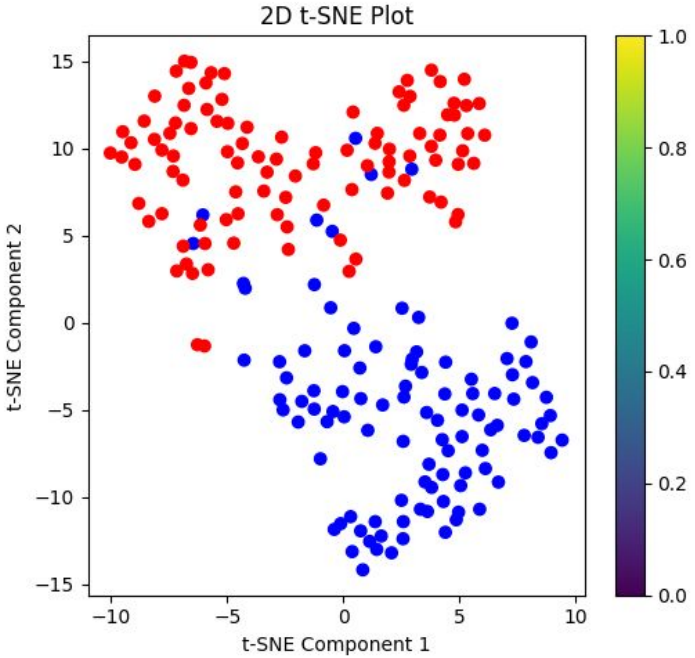
2. **Randomly initialize** the positions of the data points **in the lower-dimensional space**.

3. Iteratively update the positions of the points in the lower-dimensional space based on two objectives:
   a. **Minimize the mismatch** between the pairwise similarities of the data points in the high-dimensional space and the lower-dimensional space. This is done by defining a similarity measure between points in the lower-dimensional space using a Student's t-distribution.
   b. **Maintain the distances** between the points as best as possible. Points that are far apart in the high-dimensional space should remain far apart in the lower-dimensional space.

4. Repeat the iterations until the algorithm converges or reaches a specified number of iterations.

Visualization: t-SNE






```
from sklearn.manifold import TSNE
tsne = TSNE(n_components=2, random_state=0)
X_tsne = tsne.fit_transform(X)
```

Visualization

http://projector.tensorflow.org/