UNIVERSITÄT
LEIPZIG

ICDS: Group 5

Dimensionality Reduction

Leipzig, 09.06.2023

Harpreet Int-Veen, Karoline Schuster, Moritz Huhle,
Paula Hagen, Paul Kühnel, Sepideh Kavianpoor Esfahani

Structure

- Dimensionality reduction - overview
- Methods
  - PCA (Principal Component Analysis)
  - Exploratory Factor Analysis
  - Auto-Encoder
  - LDA (Linear Discriminant Analysis)
  - LLE (Locally Linear Embedding)
  - TSNE (T-distributed Stochastic Neighbor Embedding)
- Python libraries
- Data Set
- Project Goal
- Road Map
- GitHub usage

Dimensionality reduction - Overview

What is Dimensionality Reduction

- Unsupervised machine learning techniques
- Reduces the dimension of data while preserving relevant information [1]
  - Reduction to a smaller set of original or new variables

Why do we need it?

- Curse of dimensionality: high dimensional, sparse with large number of observations [2]
- Decreases complexity of the calculation, risk of overfitting [1]
- New variables are linear combinations or nonlinear functions of the original variables

Applications

- To counter overfitting $\rightarrow$ noise reduction, feature extraction
- Image compression
- Saves computational resources when training models
- Reduces the training time of models

PCA (Principal Component Analysis)

Goals of the PCA:

- Transform high-dimensional data into a low-dimensional representation while retaining the most important information

How does it work?

- Identify the principal components to explain the maximum amount of variation present in the data

Eigenvectors and eigenvalues:

- The eigenvectors represent the directions or axes in the original feature space
- The eigenvalues indicate the amount of variance explained by each eigenvector
- Obtain the principal components by selecting a subset of the eigenvectors with the highest eigenvalues

What are the results?

- Project the original data onto the selected principal components
- The new representation only requires a smaller number of principal components instead of the original variables

PCA (Principal Component Analysis)

How many principal components do we need?

- Explained Variance: The cumulative explained variance of the principal components represents the proportion of total variance in the data
- Scree Plot: A scree plot helps identify an elbow point where the eigenvalues sharply drop off. We choose the number of components before the elbow point [3]

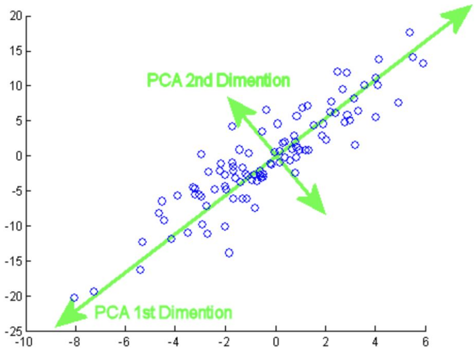


Image source: https://programmathically.com/principal-components-analysis-explained-for-dummies/

PCA (Principal Component Analysis)

Advantages

- Efficient for dim. reduction and maximizing the variance
- Easy to interpret and visualize

Disadvantages

- Assumes linear relationships between variables
- Does not consider the underlying structure / latent factors

When do we use this technique?

- When reducing the dimensionality of the dataset while preserving most of the variation in the data

→ Data visualization, noise reduction [4]

Exploratory Factor Analysis

- Similar to PCA, but aims to find **latent variables** (factors) → Not directly measured in a single variable
- Assumption, that there exist latent variables in our data [3]
- Common in psychology to uncover latent dimensions (e.g. personality traits)

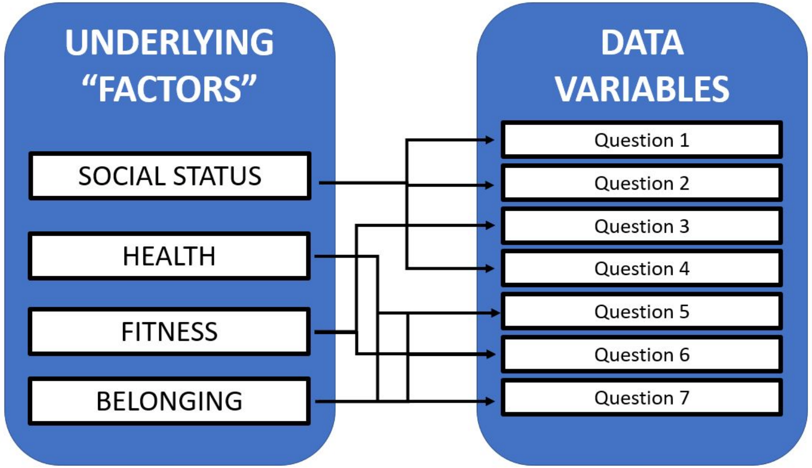


Image source: https://golden.com/wiki/Factor_analysis-RWGG9

Auto-Encoder

- Encoder network compresses the data into a lower-dimensional representation
- Decoder network reconstructs the original data from the compressed representation
- Tries to minimize the reconstruction error during training to capture the most important features of the data
- Used for high-dimensional data: Anomaly detection, image generation [5]

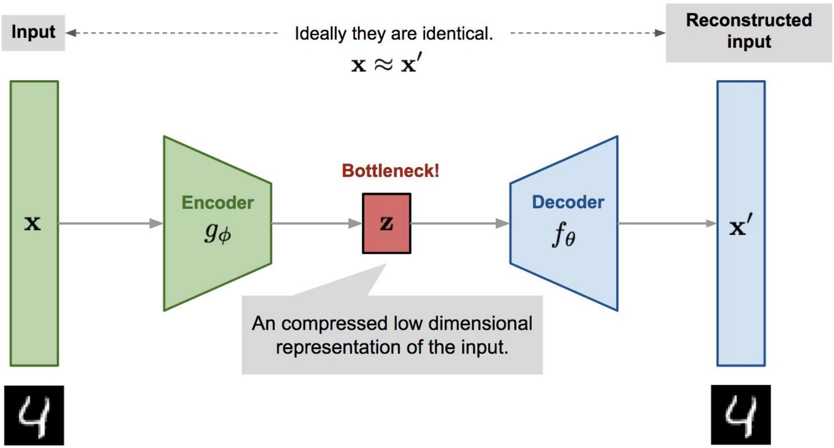


Image source: https://lilianweng.github.io/posts/2018-08-12-vae/

LDA (Linear Discriminant Analysis)

- In LDA, we use the information from class labels to transform the feature space into a lower-dimensional space that maximizes the distance between classes and minimizes the distances within classes [6]

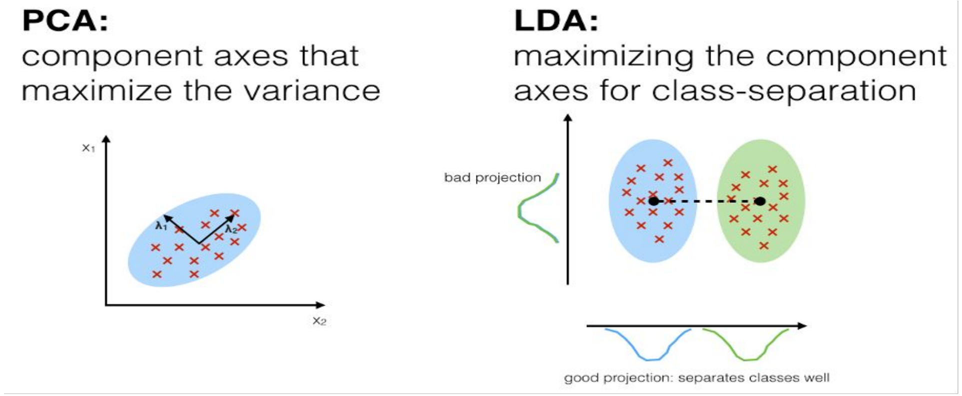


Image source: https://sebastianraschka.com/Articles/2014_python_lda.html

LDA (Linear Discriminant Analysis)

Advantages

- Supervised projection method
- Maintains class-discriminatory information

Disadvantages

- Assumes classes to be evenly distributed and the data of each class is normally distributed
- If these assumptions are not true, LDA's performance may be adversely affected

When do we use this technique?

- Takes class information into account, making it useful for classification tasks

LLE (Locally Linear Embedding)

- LLE preserves the geometric features of the original non-linear feature structure

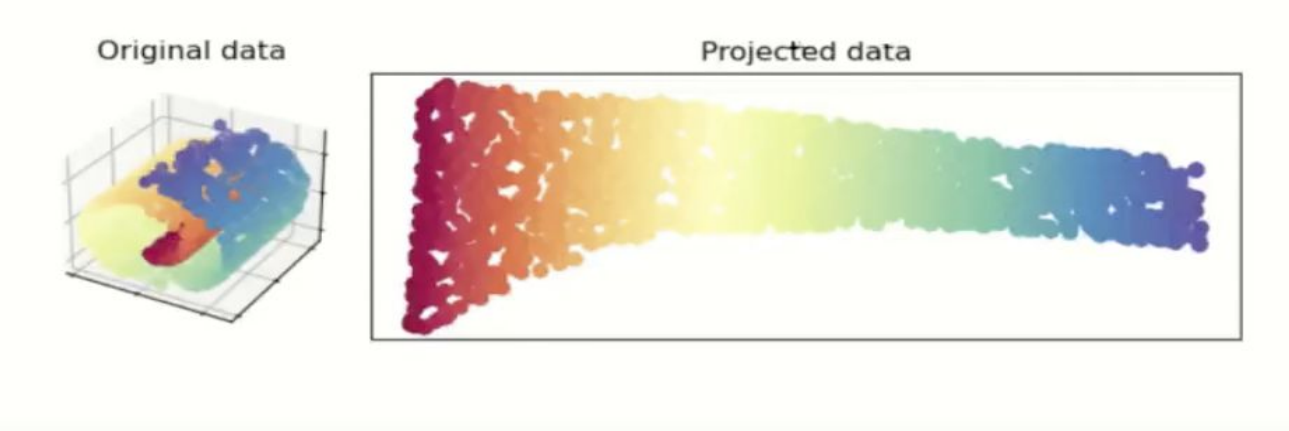


Image source: https://cs.nyu.edu/~roweis/lle/swissroll.html

LLE (Locally Linear Embedding)

Advantages

- It attempts to preserve the local relationships between points in the data, making it particularly useful for certain types of data.

Disadvantages

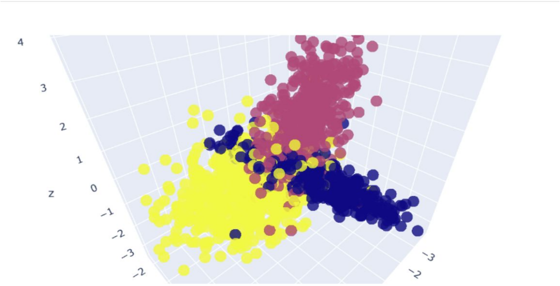
- Can be relatively slow for large datasets.

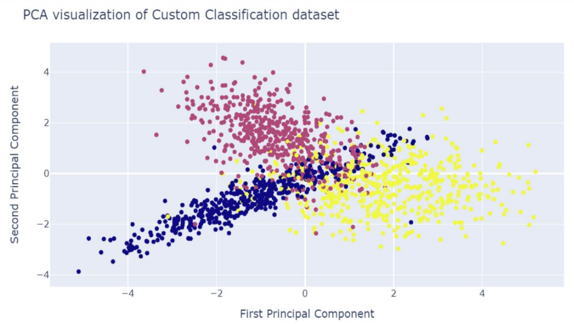
When do we use this technique?

- It's particularly effective when dealing with data that resides on a lower-dimensional manifold within a higher-dimensional space.[7]

TSNE (T-distributed Stochastic Neighbor Embedding)

- Unsupervised, non-linear technique
- Able to separate data that cannot be separated by a straight line





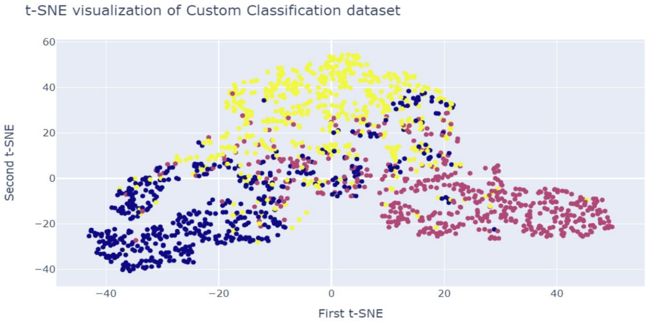


Image source: https://www.datacamp.com/tutorial/introduction-t-sne

TSNE (T-distributed Stochastic Neighbor Embedding)

Advantages

- Works particularly well on visualization of higher dimensional data

Disadvantages

- Results are not deterministic

When do we use this technique?

- Dimensionality reduction and visualization of high-dimensional datasets. It is particularly effective when visualizing data that consists of several distinct groups or clusters.[8]

Python libraries

Basic libraries useful with all methods:

- **numpy/scipy**, **pandas** and **matplotlib**

PCA:

- pca (`from pca import pca`)
- statsmodels (`from statsmodels.multivariate.pca import PCA`)
- sklearn (`from sklearn.decomposition import PCA`)

Comparison: [9]

- **pca**: Is based on sklearn but has additional features
- **statsmodels:** Closer to R, focus is statistics & economics
- **sklearn**: Most widespread, good documentation, focus is machine learning

Exploratory Factor Analysis:

- statsmodels (`from statsmodels.multivariate.factor import Factor`)
- factor-analyzer (`from factor_analyzer import FactorAnalyzer`)
- sklearn (`from sklearn.decomposition import FactorAnalysis`)

Auto-Encoder:

- keras (`from keras import Model, layers`)
- autoencoder (`from autoencoder import model_parts`)
- PyTorch

LDA:

- sklearn (`from sklearn.discriminant_analysis import LinearDiscriminantAnalysis`)

Graph-Methods (t-SNE, LLE):

- sklearn (`from sklearn.manifold import TSNE`)

```
import torch
import torch.nn as nn

# Define the autoencoder model
class Autoencoder(nn.Module):
    def __init__(self):
        super(Autoencoder, self).__init__()
        self.encoder = nn.Sequential(
            nn.Linear(784, 32),
            nn.ReLU(True))
        self.decoder = nn.Sequential(
            nn.Linear(32, 784),
            nn.Sigmoid())

    def forward(self, x):
        x = self.encoder(x)
        x = self.decoder(x)
        return x
```

Image source: https://medium.com/nerd-for-tech/deep-learning-keras-vs-pytorch-cnn-rnn-gan-autoencoder-88179564c0b3

Dataset

ICMR Dataset (Indian Council of Medical Research published on Kaggle [10])

- Contains gene samples from people associated with different cancer types
- Which genes or combination of genes are responsible for each cancer type
- Usability: classification and clustering

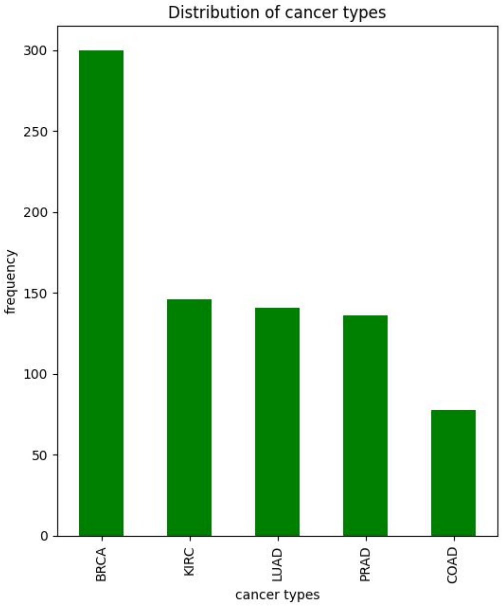
Descriptive Details

- Input dataset contains 801 samples for the corresponding 801 people with cancer
- Who have been detected with different types of cancer
- Each sample contains expression values of more than 20K genes
- Samples have one of the types of tumours: BRCA, KIRC, COAD, LUAD and PRAD

Dimensionality Reduction:

- Input: complete dataset including all genes (20.531)
- Output: the principal components → genes that are most responsible

Dataset - Distributions of classes and genes



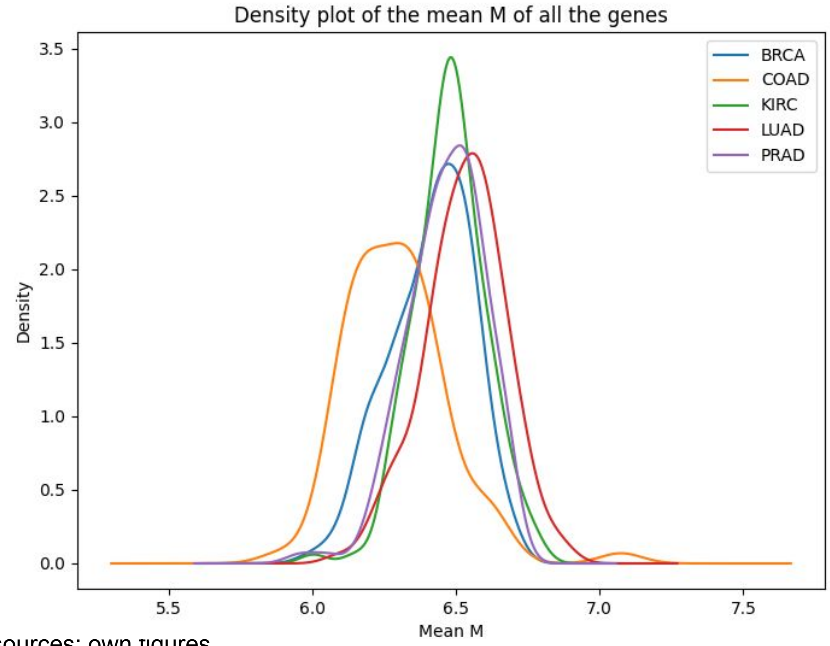


Image sources: own figures

Project goal

Performing **Dimensionality Reduction** on **ICMR Dataset** because

- $p > n$, making it impossible to use standard classification models
- Reducing dimensionality would lead to better runtime for later classification
- Cancer research shows there is no one gene responsible for developing cancer [11, 12]

We will use **PCA** because

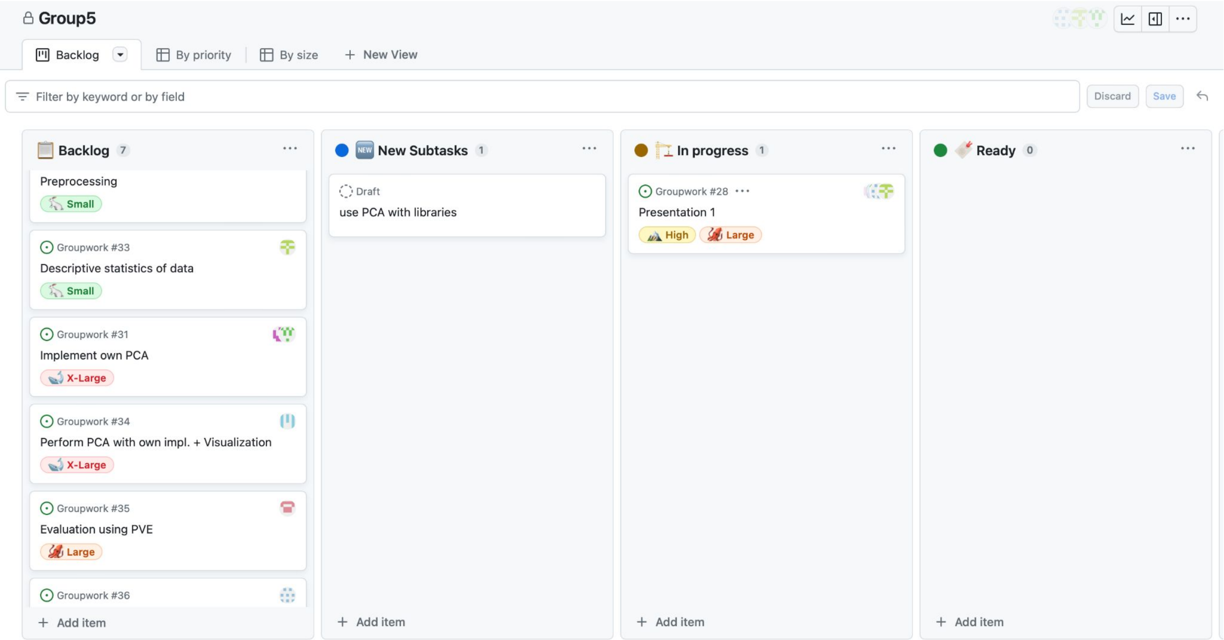
- Very popular method
- Faster than other dimension reduction techniques [13]
- Performs well, if features are correlated [13]
- Visualizes data well
- PCA results are deterministic and applicable to new data

Road Map

1. Implement PCA (including standardization)
2. Load ICMR data sets and merge raw data with label data
3. Investigate data set for missings
4. Descriptive statistics (distribution of cancer types, age…)
5. Perform PCA on genetic data using own implementation
6. Evaluate proportion of variance explained and choose number of components
7. Compare results & run time with PCA implementation
8. (Optional) use dimensions to classify subjects into cancer type

How we will use git/GitHub in our project

- dev & main branch
- Kanban in GitHub with user stories & issues
- Documentation: **.md** files in a docs folder



References

[1] Van Der Maaten, Laurens, Eric Postma, and Jaap Van den Herik. "Dimensionality Reduction: A Comparative Review." *J Mach Learn Res* 10.66-71 (2009): 13

[2] Altman, Naomi, and Martin Krzywinski. "The curse(s) of dimensionality." *Nat Methods* 15.6 (2018): 399-400

[3] Wolf, C., Best, H. “Handbuch der sozialwissenschaftlichen Datenanalyse”. (2010)

[4] Brownlee, Jason. “Advantages and disadvantages of PCA.” https://machinelearningmastery.com/discover-feature-engineering-how-to-engineer-features-and-how-to-get-good-at-it/

[5] Xu, Jinchao, et al. “Autoencoder: A Review of Theories and Applications.” (2020)

[6] Raschka, Sebastian. “Linear Discriminant Analysis.” (2014)

[7] Raj, Judy T. ”A beginner’s guide to dimensionality reduction in Machine Learning.” (2019) https://towardsdatascience.com/dimensionality-reduction-for-machine-learning-80a46c2ebb7e (last access: 05.06.2023)

[8] Awan, Abid Ali. “t-distributed Stochastic Neighbor Embedding.” (2023) https://www.datacamp.com/tutorial/introduction-t-sne (last access: 05.06.2023)

References

[9] Persson, Isaiah, and Jam Khojasteh. "Python packages for exploratory factor analysis." *Structural Equation Modeling: A Multidisciplinary Journal* 28.6 (2021): 983-988.

[10] ICMR Dataset. https://www.kaggle.com/datasets/shibumohapatra/icmr-data (last access: 05.06.2023)

[11] Breast Cancer Consortium. “Breast Cancer Risk Genes — Association Analysis in More than 113,000 Women.” *N Engl J Med* 384 (2021), 428-439.

[12] Mendiratta, G., et al. “Cancer gene mutation frequencies for the U.S. population.” *Nature Communications* 12, 5961 (2021)

[13] Ellis, Christina. “When to use principal component analysis.” *Crunching the data*. https://crunchingthedata.com/when-to-use-principal-component-analysis/ (last access: 05.06.2023)

Thank You!

Introductory Computing for Data Science - Dimensionality reduction

Paula Hagen, Moritz Huhle, Sepideh Kaviapoor Esfahani, Paul Kühnel,
Karoline Schuster, Harpreet Int-Veen