Theoretical and practical metagenomic approaches to viral discovery

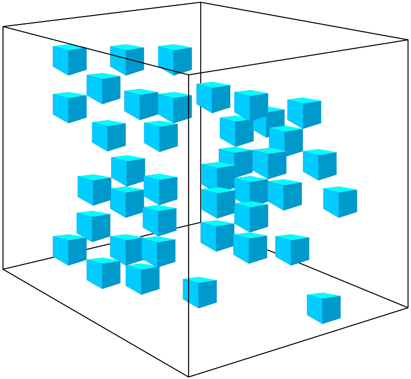
Practical Session: Dimension Reduction and Clustering

Kevin Lamkiewicz, Manja Marz

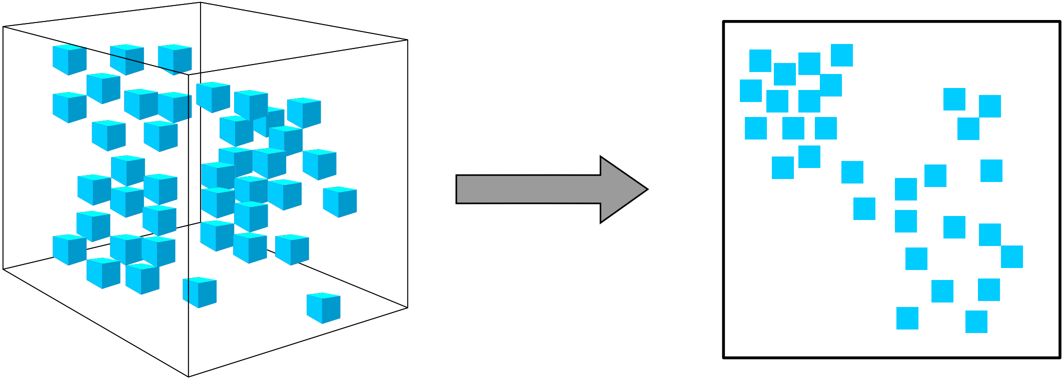
23.10.2019

European Virus Bioinformatics Center

Dimension Reduction



Dimension Reduction



PCA with scikit-learn

A QUICK EXAMPLE

```
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt


data = []
target = []
target_names = list(class2id.keys())
with open(’virus.csv’, ’r’) as inputStream:
    ...
```

A QUICK EXAMPLE

```
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt


data = []
target = []
target_names = list(class2id.keys())
with open(’virus.csv’, ’r’) as inputStream:
    ...
df = pd.DataFrame(data= np.c_[data, [id2class[x] for x in target]],
        columns= [’feature1’,’feature2’,’feature3’,’feature4’] + [’target’])
```

Normalize your data

```
features = [’feature1’,’feature2’,’feature3’,’feature4’]

# Here we seperate features from the target column
x = df.loc[:, features].values
y = df.loc[:,[’target’]].values

# Standardizing the only the features
x = StandardScaler().fit_transform(x)
```

Applying PCA

```
# create a PCA object that reduces the data to 2D
pca = PCA(n_components=2)
# fit the data
principalComponents = pca.fit_transform(x)
# bring everything a format / data structure for the PCA
principalDf = pd.DataFrame(data = principalComponents
           , columns = [’principal component 1’, ’principal component 2’])
finalDf = pd.concat([principalDf, df[[’target’]]], axis = 1)
```

Other Application of PCAs

Dimension Reduction in Machine Learning

Normally, you would like to use PCA (or any other method for dimension reduction) to **speed up** your machine learning algorithm. Instead of learning many instances with over 700 features (e.g. our handwritten letter example from earlier), we can **reduce the number of features**, by only taking the most **important combinations** of features into account.

Virus Classification with PCA

Exercise / Homework

Have a look at the `viral_data.fasta`.
There are some known viruses, but also some unknown viruses inside. Try to use k-mer frequencies as features and cluster the sequences based on that.
Can you roughly classify the unknown viruses?

You do not have to do this in Python. If you are more familiar with, for example, R, feel free to use this as well.
I am just a little bit Python addicted. :)

Coffee Break

European VIRUS BIOINFORMATICS Center