Clustering – Introduction

Machine Learning
uczeniemaszynowe@cs.uni.wroc.pl

deadline: 18.11.2025 10:00
tutorial / lab

Task 1. [2 pts]

In this task you will build a Naive Bayes classifier[1] for the dataset `Titanic Dataset.csv`[2].

1. You may reuse the data preprocessing from list 4.
2. To build the `CategoricalNB` model, convert numerical variables into categorical ones by replacing their exact values with assignments to value groups. Group values using:
   - dividing the variable range into equal-width bins,
   - dividing training observations into equal-sized bins,
   - clustering the values of each variable.
3. Build the models. Compare classifier results for different quantization methods and different numbers of groups ($k = 2, 5, 10, 25$). Collect the results in a table.

Hints:

- To find boundaries for equal-sized bins, you may use the function `np.quantile`[3].

Task 2. [2 pts]

For the (synthetic) datasets `data<index>.csv`[4] (`index` $\in 1, 2, \ldots, 5$), build k-means and Gaussian Mixture models with $k = 3$ components. Based on visualizations, evaluate which model (kMeans[5] or GMM[6]) performs better.

[1] https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.CategoricalNB.html
[2] https://www.kaggle.com/code/sakshisatre/titanic-s-missing-data-visualizing-null-values/input
[3] https://numpy.org/doc/2.3/reference/generated/numpy.quantile.html
[4] https://numpy.org/doc/2.3/reference/generated/numpy.load.html#numpy.load
[5] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html
[6] https://scikit-learn.org/stable/modules/generated/sklearn.mixture.GaussianMixture.html

Task 3. [2 pts]

Bazyli is passionate about photography. After several years he has gathered a large collection of photos. Having a sizable dataset, he decided to train a classifier on it. He began annotating the photos, but it turned out to be tedious and boring, so he stopped after only a dozen examples. Still determined, Bazyli used a pretrained model. He managed to obtain representations for all his photos, but still did not know how to assign them to the correct classes. Help Bazyli finish his task.

In the file `image_emb.npy`[7] you will find the image embeddings. In the file `image_labels.npy` you will find indices and labels assigned to selected images (a few for each class). In the files `image_emb_test.npy` and `image_labels_test.npy` you will find data for evaluating your solution.
Hints:

- Perform clustering on the embeddings.
- Translate cluster membership into class membership.
- Evaluate your solution like a standard classification model.

Task 4. [2 pts]

Build an appropriate clustering model for the `fetch_20newsgroups`[8] dataset. Choose the number of clusters based on the Silhouette coefficient. Visualize and describe how to choose the number of clusters using the *elbow method* for the average quantization error. Using the original texts, analyze which thematic groups were discovered.
Hints:

- Represent texts as vectors. You may follow the example below.

```
from sklearn.datasets import fetch_20newsgroups
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.decomposition import KernelPCA
from sklearn.pipeline import Pipeline

newsgroups_train = fetch_20newsgroups()

pipeline = Pipeline([
    ('tfidf', TfidfVectorizer()),
    ('pca', KernelPCA(n_components=100, random_state=42))
])

pca_matrix = pipeline.fit_transform(newsgroups_train.data)

print("Shape of PCA matrix:", pca_matrix.shape)
```

- To visualize clusters on a 2D plot, you may use a dimensionality reduction technique such as tSNE.

[7] ResNet50 model https://docs.pytorch.org/vision/main/models/generated/torchvision.models.resnet50.html, CIFAR10 dataset https://www.cs.toronto.edu/ kriz/cifar.html
[8] https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_20newsgroups.html

```
from sklearn.manifold import TSNE
import matplotlib.pyplot as plt

tsne = TSNE(n_components=2, random_state=42, perplexity=15)
matrix_2d = tsne.fit_transform(pca_matrix)

plt.scatter(matrix_2d[:, 0], matrix_2d[:, 1], c=cluster_labels)
```

Task 5. [2 pts]

Propose your own implementation of the k-means algorithm. Train the model on the `fetch_20newsgroups`[9] data. Visualize the training process of the model. Propose a modification of the algorithm to an *online* version, such that it updates itself with incoming data. Visualize the online fine-tuning process.
Hints:

- Use interactive visualizations or animations.
- Split the data into model pretraining data (e.g., 30%) and online fine-tuning data.
- Iterate using small batches of incoming data (not necessarily one observation at a time).

Discussion for the laboratory

- Simple methods of text vectorization (bag of words, TF-IDF)

[9] https://scikit-learn.org/stable/modules/generated/sklearn.datasets.fetch_20newsgroups.html