# DAT565/DIT407
# Introduction to Data Science and AI

Assignment 5

**Deadline:** 2024-10-10 08:00

The file `seeds.tsv`, available on Canvas, contains geometric measurement data on wheat kernels of different species, obtained by x-raying the kernels [1]. Each observation has seven features plus a numerical class label that identifies the species of wheat in question. The features are listed in Table 1.

Your task will be to apply clustering methods to the dataset and report your findings.

## Problem 1: Preprocessing the dataset

Read the dataset into Pandas. As the features have different units and ranges of values, you should normalize the dataset appropriately. Describe what you did in your report.

## Problem 2: Determining the appropriate number of clusters

For now, let us set the species label aside; we shall *not* make use of it in this problem, so you should not use it as part of any code you are writing at this stage.

Iterate over different values $k = 1, 2, \ldots$, and apply $k$-means clustering on the data. For each value of $k$, compute the *inertia* or the sum of squared residual error, that is, the squared distance from each point to its nearest cluster center:

$$\sum_{\ell=1}^{k} \sum_{i \in C_\ell} ||x_i - \bar{c}_\ell||^2 \,,$$

where $C_\ell$ is the $\ell$th cluster, and $\bar{c}_\ell = \frac{1}{|C_\ell|} \sum_{i \in C_\ell} x_i$ is the associated cluster center.

Plot the inertia as a function of $k$. Eyeballing the plot, what would be an appropriate number of clusters? Justify your answer.

Table 1: Columns of the dataset.

| Column | Feature |
|---|---|
| 1 | Area $A$ |
| 2 | Perimeter $P$ |
| 3 | Compactness $C = \frac{4\pi A}{P^2}$ |
| 4 | Length of kernel |
| 5 | Width of kernel |
| 6 | Asymmetry coefficient |
| 7 | Length of the kernel groove |
| 8 | Numerical class label |

## Problem 3: Visualizing the classes

**(a)** Visualizing multidimensional data like this is a challenge. Visualizations seldom work in more than two dimensions. One way to systematically visualize the data is by projecting it to the constituent dimensions.

Plot a scatter plot between each pair of features, coloring the points by the class label. Can we find structure? Can some pair of features tell all point reliably tell all or some classes apart? Include *one* interesting plot in your report and explain why you think it is interesting.

**(b)** Another way to obtain more feasible visualizations is through dimensionality reduction. Apply *Gaussian random projection* to project the data to two dimensions. Plot a scatter plot and include the plot in your report. What does the data look like?

Note that due to randomness, the results may vary, so you may want to try to do this a couple of times.

**(c)** Another principled way to obtain an interesting projection to fewer dimensions is by applying UMAP, or *Uniform Manifold Approximation and Projection for Dimensional Reduction* [2] which projects the data on a (hypothesized) Riemannian manifold. Use the `umap-learn` package to project the data into two dimensions, plot a scatter plot, and include the plot in your report.

**(d)** Based on your observations in (a)–(c), does the data *look* linearly separable? Why / why not? What implications does this have for clustering?

## Problem 4: Evaluating clustering

Apply $k$-means clustering to the data, but choose $k$ to match the number of underlying species of wheat. It would seem like a reasonable assumption that there should be an equal number of clusters to the number of interesting classes in the data.

Compute the *Rand index* of the clustering you obtain. The Rand index looks at each pair of points, and reports the fraction of points that are either in the same cluster in both clusterings, or in different clusters in both clusterings. That is, suppose $c_i$ is the cluster label of point $i$ in one clustering and $c'_i$ in another. The Iverson bracket notation is defined as follows:

$$[\![P]\!] = \begin{cases} 1 & \text{if } P \text{ is true, and} \\ 0 & \text{otherwise.} \end{cases}$$

Using the Iverson bracket notation, the Rand index $R$ could be thus defined as

$$R = \frac{1}{\binom{n}{2}} \left( \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} [\![c_i = c_j \wedge c'_i = c'_j]\!] + [\![c_i \neq c_j \wedge c'_i \neq c'_j]\!] \right) .$$

Compute the Rand index between the clustering you obtained using $k$-means and the assumed correct clustering based on class labels.

Furthermore, compute the *accuracy* of your clustering: the fraction of points that have been assigned to the correct class. As clustering is unsupervised, we do not have cluster identities as such; you may thus find the correct *permutation* of cluster labels, such that it best matches the class labels.

**Explanation.** Suppose we have $k$ classes and $k$ clusters. Let us denote $[k] = \{1, \dots, k\}$, and suppose $c_1, c_2, \dots, c_n \in [k]$ are the cluster labels (output of the clustering algorithm) and $y_1, y_2, \dots, y_k \in [k]$ are the correct class labels. We are looking for a permutation $\pi : [k] \to [k]$ that reorders the cluster labels such that the reordered cluster labels best match the correct class labels; that is, we define accuracy to be

$$\text{Accuracy} = \max_{\pi \in S_k} \frac{1}{n} \sum_{i=1}^{n} [\![ \pi(c_i) = y_i ]\!] \,,$$

where $S_k$ is the *symmetric group* of degree $k$, the set of all permutations of $k$ objects.

## Problem 5: Agglomerative clustering

Use agglomerative clustering to compute a hierarchical clustering for the data. Try different linkage options. Which works best and which worst, in terms of accuracy at the assumed correct number of clusters? Why do you think this is the case?

Furthermore, plot a dendrogram of the hierarchical clustering with the best linkage option. Use the correct class labels as the *labels* for the leaf nodes (or singleton clusters) such that it is possible to see how the observations of different species fall into the cluster tree.

## Hints

- Use `pd.read_table` to read the data.[1]
- Scikit-Learn comes with many useful transformations directly, meaning, for example, doing Z-score normalization is very simple: just apply `fit_transform` of `StandardScaler`.[2]
- Scikit-Learn can also do Gaussian random projections very easily with the `GaussianRandomProjection` class.[3]
- The `umap-learn` library has a very good tutorial on its use.[4]
- If the definition Rand index seems complicated, have a look at the Wikipedia page for alternative descriptions.

[1] https://pandas.pydata.org/docs/reference/api/pandas.read_table.html

[2] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html

[3] https://scikit-learn.org/stable/modules/generated/sklearn.random_projection.GaussianRandomProjection.html

[4] https://umap-learn.readthedocs.io/en/latest/basic_usage.html

- Accuracy is simply the fraction of correctly classified observations; however, since the clustering has no notion of class labels, you need to consider every possible permutation of mappings between the class labels and cluster labels (if the clustering is any good, then the correct permutation should stand out).
- Agglomerative clustering is availabe through both Scikit-Learn[5] and SciPy.[6]
- You should use `scipy.cluster.hierarchy.dendrogram` for plotting the dendrogram; to do this, you are going to need the linkage matrix which you can obtain either with `scipy.cluster.hierarchy.linkage` or by following the Scikit-Learn example.[7]

## Returning your report

Write a report, typeset in LaTeX, that answers *all* questions above. Include all your Python code in your report as an appendix, preferably using the `listings` package. Your report should be legible even without having a look at your code.

If you refer to outside sources, remember to add an appropriate literature reference (including websites) in references by `\cite`ing the references. It is recommended that you use the package `biblatex` to manage citations.

Place your figures in numbered `figure` environments, with descriptive captions and `\ref` to the figures in your discussion. Likewise, place your tables in numbered `table` environments with desciprive captions and `\ref` to the tables in your discussion.

After grading, you will be given another attempt to revise your report according to TA comments if it is not considered acceptable.

The deadline is *hard*. Late submissions will not be read at all and are considered failed. This means you will not get any feedback for the first round and the submission is considered a revision; there will be no third attempt, so if a late submission is failed, you will need to participate in a later iteration of the course for a re-attempt.

## References

[1] Magorzata Charytanowicz et al. *seeds*. UCI Machine Learning Repository. 2012. DOI: `https://doi.org/10.24432/C5H30K`. URL: `https://archive.ics.uci.edu/dataset/236/seeds`.

[2] Leland McInnes, John Healy, and James Melville. *UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction*. 2018. arXiv: `1802.03426 [stat.ML]`.

[5] `https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html`

[6] `https://docs.scipy.org/doc/scipy/reference/cluster.hierarchy.html`

[7] `https://scikit-learn.org/stable/auto_examples/cluster/plot_agglomerative_dendrogram.html`