Strasbourg University / French-Azerbaijani University

Artificial Intelligence

Computer Science track - Year 3

Lab: Iris dataset clustering with K-Means

The objective of this lab is to implement the K-means algorithm in order to cluster Irises. A code template is available to you on Moodle.
Specific objectives:

- Observe the data, understand their nature and how to adapt them (if needed) so you can use them in a Decision Tree model.
- Understand how K-Means clustering works so as to implement this model in a computer program.
- Evaluate the results and put them into perspective with what we know about the data.

1 The Iris dataset

See lab sheet "Iris dataset classification using a Decision Tree" for a presentation of the data.

★ *Is k-means clustering used for supervised or unsupervised classification? Explain your answer.*

2 A code template to help you

A code template (written in Java) is avaiable on Moodle. You do not *have* to use it, but if you do, then you start with a working data model (and you can begin the interesting part –implement the core of the k-means algorithm– right away).

★ *Get this code, have a look at it and localize the parts you need to implement the procedures to build a Decision Tree.*

3 Implementing k-means clustering

The main idea is to classify data that are not necessary labelled. K-means will partion the dataset in k groups (or clusters).

★ *Knowing that we do have a labelled dataset, how many clusters are you going to set in your program?*

If you do not have labelled data, you still have to propose a value for k: such models are referred to as **semi-supervised** models.

Obviously, different values for k will lead to different results. Without labels, you don't know a priori which value to use (k can be approximated using dedicated methods (*e.g.* `https://link.springer.com/article/10.1007\%2FBF02294245`).

K-means algorithm Once you have set k, the procedure to create the clusters is the following:

1. *Cluster initialisation:* you can randomly pick k datapoints in the dataset and set them as the center of each cluster.
2. *Assigning datapoint to clusters:* for each datapoint, compute the distance to the center of each cluster. You must then assign the datapoint to the closest cluster.
3. *Updating clusters center:* compute the average of each cluster. For each cluster, the resulting average is the new center of the cluster.
4. *Repeat* steps 2 and 3 while cluster still change (or set a maximal number of iteration so you compensate for possible oscillations).

★ *Implement the procedure. After running it, export the clusters in a* `csv` *file in which one column is used to indicate to which cluster a given instance belongs. Plot the clusters using GnuPlot, LibreOffice, or whatever plotting software you choose.*

★ *Run you program with different values for k and different cluster initialization (random, or with the method presented during the lectures).*

Visualizing k-means clustering with different initialisations : `http://shabal.in/visuals/kmeans/1.html`