homework 4: Unsupervised Learning and Expectation Maximization

CSE 142: Machine Learning

University of California, Santa Cruz

Please do this assignment on your own. Discussing concepts with your classmates is fine, but please do not share any code.

For this assignment, we're doing some more programming. We'll implement both K-Means Clustering and Gaussian Mixture Models clustering with expectation-maximization.

For your code, you can either build your implementation in a Jupyter notebook, or standalone Python code.

The deliverables for this assignment will be your code, as well as a written report of your findings from your programming. One third of your grade will depend on the quality of the report, which you ought to submit as a PDF. For the full experience, you might like to typeset your scientific writing using LATEX. Some free tools that might help: Overleaf (online), TexLive (cross-platform), MacTex (Mac), and TexStudio (Windows).

Datasets and problems to solve

For this homework, you'll be working with a dataset that's included in the homework zip file, `hw4_dataset.csv`.

Sparing no expense, we asked some UC Santa Cruz biology majors to go out into the world and measure the weights (in kilograms) and lengths (in centimeters) of some animals that they found, from a small finite number of different species. Unfortunately, we forgot to ask them to write down which which animal was from which species, how many different species they measured, or what the species were.

So it's going to be your job to load up the collected dataset and do some clustering, with your own bespoke implementation of the clustering algorithms we discussed in class.

You can load the CSV data into a Pandas dataframe like so:

```
df = pd.read_csv("hw4_dataset.csv")
```

Programming: implementing K-Means and Gaussian Mixture Models clustering

For this homework, you'll be implementing K-Means and Gaussian Mixture model clustering algorithms.

What does it mean to have implemented these clustering algorithms? You ought to implement code (perhaps two distinct Python classes) so that...

- You can initialize the clusterer to have a specific number of clusters and a maximum number of iterations to run (these are hyperparameters).
- You have a `fit` method that determines the parameters of the clustering model, based on the dataset, which is passed in as a matrix `X`. This code will set the parameters of your model so they can be used later, for clustering the data in...
- The `predict` function (or method), which given an input matrix `X` (each row is an instance from a dataset, just like in `fit`), assigns each instance to a cluster ID ranging from 0 to the number of clusters minus 1.

What are the parameters of your models that you'll need? For both clusters, you'll need to find the means (which will be points in 2 dimensional space, for this dataset), and for the GMM model, you'll need to estimate a covariance matrix, since these are multivariate Gaussian distributions.

Experiments

Once you have a working implementation of K-Means and GMM clustering, try to find good clusterings for the given dataset!

How does your implementation compare with the versions in scikit-learn?

Report

Your writeup for this homework can be fairly short, but it ought to include:

- A quick prose description of how your code works.
- How many clusters does it seem like there ought to be here? Why would you say this?
- A visualization of your clusterings.
- Do you have a sense about which animals these might be, that our biologists went out and measured?

Deliverables

You should turn in your code along with a README, explaining how how to run each of the variants you developed, as well as your report as a PDF.

Submission Instructions

Submit a zip file (hw4.zip) on Canvas, containing the following:

- Code: You code should be implemented in Python 3, and needs to be runnable. Submit your code together with a neatly written README file explaining how to run your code with different settings. Follow good software engineering practice here – code is meant to be read!
- Report: As noted above, your writeup should be in PDF; please put your name at the top.

References

- *Pattern Recognition and Machine Learning* by C. Bishop, Chapter 9 – contains really clear explanations of the algorithms to implement and math you'll need to know to do this! (links from the course Canvas if you haven't already got it)