EQ2341 Project Guidelines

Spring 2022

We hope you have enjoyed the four assignments. Now it is time to bring your HMM recognizer into real-life applications. You will have to choose among three topics: speech, music, and hand-drawing. For each topic, there are specific guidelines with example datasets. It is recommended that you work on the same topic as the previous assignments.

1 Speech recognition

Train and test your HMM model to classify different speech records into single commands: yes, no, up, down, left, right, on, off, stop, go, and noise. The dataset comes from a competition initiated by the Google Brain team in 2018.
https://www.kaggle.com/c/tensorflow-speech-recognition-challenge/data

An alternative task could be to detect emotions from the speech. This task might need you to extract other features than phonemes. The dataset is available here: https://www.kaggle.com/datasets/dmitrybabko/speech-emotion-recognition-en

Recording your speech (or even your pets'), should be done after your success on the well-established dataset.

2 Song recognition

For query of humming, there exists a dataset of recorded human singing to be used as 'humming' for testing:
https://zenodo.org/record/1290712.YlbkHXUzbmG

Select a few songs to train your HMM. You will need to prepare the chosen song files yourself for license reasons. It is recommended to train on short main melody pieces (<60s) instead of the entire song. One reliable song database is the Million Song Dataset. Note that instead of audio files (wav, mp3, etc.), you directly get the sample matrix in hdf5, which can be opened in python through h5py packages.
http://millionsongdataset.com/

An alternative task is to perform music genre classification. An example dataset can be the GTZAN dataset.
https://www.kaggle.com/datasets/andradaolteanu/gtzan-dataset-music-genre-classification

Potentially, the GTZAN dataset could also be used for the query of the humming training set. Recording your own humming is recommended after you have shown success in well-established datasets.

3 Character recognition

This topic is for those of you who like to create a vanilla dataset by hand. You will need to create a dataset of handwritten digits (0 to 9) plus another type of data, .e.g. simple line art or handwriting characters of a foreign/ancient language. You could draw flowers, cats, and dogs and see when your recognizer starts to understand your art vibes. For a different language, choose one different from your language family, even hieroglyphs.

4 Notes

You probably have noticed that many datasets are acquired from Kaggle. You can use any suitable dataset from there, or other public dataset, to perform the project task.

Remember, 'Optimization after implementation'. Aim for a non-random choice classifier first, then think hard on how to improve the performance. Avoid trying to achieve 99% accuracy on the first try!

Check the grading criteria on next page before your presentation.

5 Grading Criteria

EQ2341 Final Project Presentation (6p)

Element	**Point**
Talk about your application and your data. Which pattern recognition application inspired your choice of examples? Talk about your HMM design. How many states did you use, and why? What about the output distributions?	1
Briefly describe your feature extraction scheme. Are the features discrete or continuous? Scalars or vectors? Name a number of ways the data can vary, along with the innovations or techniques used by the feature extractor to be more robust against these kinds of variation.	1
Outline how you trained and tested your classifier. In particular, explain the principle of your classifier and how did you perform the cross-validation.	1
Plot or illustrate some example training sequences from one class, and compare these against random output sequences generated by the corresponding trained HMM using @HMM/rand. In what ways are they similar, and how do they differ? Discuss what aspects of the data the HMM has learned to distinguish each class.	1
Report the results of cross-validation. Show the mean and variance of error rate. Play back or show some misclassified instances to illustrate the errors. You may present a confusion matrix C, a table with elements c_{ij} showing how often examples from class i were classified as class j.	1
Present your conclusions: How did the choices you made in the design process affect your classifier? What are the strengths and weaknesses of your system? What have you learned?	1