

CSE 6363: Machine Learning

University of Texas at Arlington

Summer 2022

Alex Dillhoff

Assignment 4

This assignment covers Hidden Markov Models.

Gesture Recognition

Gesture recognition models assign some sort of label given an input sequence. HMMs are particularly well suited for this task as they can produce a set of parameters that best encapsulate each individual gesture. In this assignment, you will implement a gesture recognition model using HMMs.

The Dataset

The dataset used in this assignment consists of 350 sequences collected by 5 volunteers, with 50 samples per gesture. All data is contained in a CSV file named `Original_Dataset.csv`. You can download it here (<https://github.com/sergiuoprea/Hand-Gesture-Recognition-Datasets>).

Preparing the Data

Since there are 7 classes, you will need to train 7 different models. We will need some subset of the data to test on, so take 10% of the data and set it aside as the test set. **Make sure that you select 10% randomly from each class.** For each of the 7 classes, there should be 40 training samples and 10 test samples.

Each sample consists of the hand centroid and finger locations for both hands over 30 frames. The models may not need all of that information and will probably perform well using only the hand centroids. When preparing the data, only take the x and y values of the hands. The final dataset size, before splitting into training and test, should be (350, 30, 4).

Normalizing the Data

The raw data contains the locations in pixel coordinates. This could lead to misclassifications depending on where the hands were in the original image. To adjust for this, normalize each sequence based on the minimum and maximum values in that sequence. The resulting locations should be normalized to range $[0, 1]$.

Building the Model

The models can be created and trained using `hmmlearn`, an HMM library for Python. The documentation can be found here (<https://hmmlearn.readthedocs.io/en/stable/index.html>). The observation model used with this dataset should follow a normal distribution.

Using `hmmlearn`, construct and train a model for each gesture using the training set. Specifically, you should have 7 models each trained on 40 training samples.

Evaluating the Test Data

Evaluate your test set on the trained models by computing the likelihood of each input test sample against all models. You can use the `score` function to evaluate the likelihood of a model given some input sequence. Given an input sample, the predicted class label corresponds to the model with the highest likelihood.

Finding the Best Configuration

Conduct a hyperparameter search by training the models using a different number of components for each model. In your report, include a table that shows the test accuracy for each configuration you tried. Highlight the best performing model by showing its results in the table in bold.

Sampling from the HMM

Use the `sample` function in `hmmlearn` to generate samples from each trained model. Visualize these samples using `matplotlib`.

In your report, show a few samples that you generated compared to a few training samples. Discuss the quality of the generated samples as compared to the real samples.

The Forwards Algorithm

Implement the forwards algorithm in Python that uses the learned parameters from your HMMs to compute the likelihood of an input sequence. Use this to evaluate your parameters on the test set. You should get the same results as before.

Submission

Create a zip file that includes all of your code as well as your report. The TA should be able to easily run the code to reproduce all plots and results. Include any additional instructions, if necessary.