CSCE 624 Sketch Recognition
Fall 2019
Assignment 1

Due: 9/17/2019 2pm CT

Overview

Your first assignment will involve coding a variety of features to build an arrow recognizer. You will be provided with a training data set to help build the best recognizer possible.

For this assignment, we will be using JavaScript, a flexible language which forms the basis of many webapps on both their client & server sides (via NodeJS). The NodeJS starter code will call a feature extraction function, and you are responsible for writing the logic which will compute the features for the provided stroke in order to return an array of feature values.

The final step will be using the resulting feature set to train Weka classifiers. You will then write a report about your results and other relevant insights.

Setup/Usage Instructions

1. Download HW files from eCampus. Unzip the folder which contains the code files and data files.
2. Install NodeJS LTS (v10) from https://nodejs.org
3. Set the number of sketches to use near the top of features.js (a value of 0 means use all sketches)
4. Open a command line/terminal in the directory of the files and run the command

   ```
   node main.js
   ```

   This will run the feature extraction (which you will implement) on the number of sketches specified in features.js
5. Install Weka from https://www.cs.waikato.ac.nz/ml/weka/
6. Use the CSV file generated by Node with Weka to build an arrow/other classifier
7. Save the Weka model for submission (https://machinelearningmastery.com/save-machine-learning-model-make-predictions-weka/)

Assignment Instructions

1. Implementing feature extraction

   The first step is to implement feature extraction for the data. You should implement the 13 Rubine features, 11 unique features from Long (12-22 in his paper), and at least 6 of your own features. The feature values should be returned from the extractFeatures method in features.js as an array of length 30.

   a. Rubine features 1-13 (indices 0-12)
   b. Long features 12-22 in the Long paper (indices 13-23)
   c. Your own features (indices 24-29+)

   In features.js, there is a function called extractFeatures (exports.extractFeatures). This function will be called by main.js and is given a stroke each time it is called. For simplicity, every sketch in this dataset contains only a single stroke. The stroke is passed to extractFeatures in its most basic form: an array of point objects. The stroke parameter is simply an array of Objects which contain keys (or properties) x, y, and time. These are the properties of each point in the stroke. The x and y values are in pixels. The time is an epoch timestamp in milliseconds.

2. Weka classifiers

   Use the CSV file of the features you generated to build a classifier in Weka. The classifier should be classifying the label of “arrow” or “other” and you should leave 10-fold cross validation enabled.

   Try out several different models (linear regression, logistic regression, multilayer perceptron, J48, Random Forest, Random Tree, Random Table are good ones to try). Try subset selection (Select attributes tab) to minimize the number of features and find out which seem important. Try changing the parameters on the classifiers to see how they affect the results.

   After trying several of these options, save your best model by right clicking the model in the Results lists and hitting “save model”.

3. Report

   Your report should be written in LaTeX (see www.overleaf.com for an easy to use, online LaTeX editor) and contain the following sections

a. Feature explanation - Provide an explanation for all features including the Rubine and Long features. The explanation should include a brief explanation and an equation if it helps the description.
b. Feature motivation - For the new features that you came up with, provide a brief explanation of your motivation behind the feature (e.g. I saw this in paper XYZ, I use a similar feature in other fields, I thought arrows would have a distinct feature value because of XYZ).
c. Weka classifiers - Provide an overview of the Weka models you tried and report the f-measure for each. A table would be a good choice. Make sure to mention if the results are with the full feature set or just a subset. Why do you think a certain model performed better than others?
d. Weka subset selection - Provide the method used in Weka for subset selection and list which features were selected. Make sure the classifier section of the report contains an entry with the best classifier (when using all features) using just the features from subset selection. Why do you think the features chosen were chosen?

4. Submission

You should submit a zip file of the following files. The zip should be named lastName_firstName_UIN_HW1.zip where you use your info. For example, John Smith with a UIN 123001234 would name their zip Smith_John_123001234_HW1.zip

a. features.js
b. lastName_firstName_UIN_weka.model
c. lastName_firstName_UIN_report.pdf

Your submission should be made to eCampus by the due date at the top of these instructions.