SVM Drawing Classification – Proposal

Evan A Bonsignori

10/24/2017

1. **SVM Classifier for Labelled Drawings**

Users will be provided with a GUI canvas for drawing a picture and a textbox to enter the label of the picture drawn. After drawing and adding a label for the picture, users can draw additional pictures and label them with the existing label. For example, if a user draws a flower on the canvas and types “flower” into the new label box, then the data points used to draw the flower will be taken and labelled as a flower. If the user draws another flower, they can select the existing label, “flower” and add it to the training examples.

Upon entering a few training examples (number needed for accuracy to be determined in project), the user can select to have a new drawing classified. The classifier will then learn using the training examples provided by the user and output the label of the drawing.

1. **Data Sources**

Since the classifier is trained using data provided by the user at runtime (the points that make up the drawings), the application does not require external data sources. In this sense, the application is self-contained, however, there is room for additional features that do require premade data sources. The application could be modified to learn a series of ‘x’ drawings provided before runtime and label any of the ‘x’ drawings when a user attempts to draw them.

In the case that we train the classifier before runtime with drawings provided by myself, the classification turns into an experiment of classifier accuracy when changing domains. If we take me (Evan) as the first domain, I provide the classifier with training examples and label them accordingly. If we take any user who is not me as another domain, then the drawing they provide will be classified using a classifier that was initially trained from my drawings.

Another modification to the application could be implementing unsupervised learning. If instead of allowing users to label their drawings, they were only asked to submit as series of them, and most are off the same things (same labels), then unsupervised analysis would take place to say if a new drawing is like a series of old drawings. This modification would require clustering and the data to be provided at runtime, thus not requiring external data sources.

1. **The SVM Classification Process**

The input for training the classifier will be a set of data points (that form a drawing) and the corresponding label (or title) of each set of data points. To simplify terminology we will simply use, “drawing” to signify the set of data points that make up the picture, and “label” to indicate the title applied to the drawing. The SVM will take each of the points and try to separate them with a hyperplane. Since the points are linear, but with a lot of crossover, so the Kernel will calculate the hyperplane in a higher dimension. Thus, the hyperplane won’t be linear.

1. **API Library**

Currently, the Scikit-learn module (library) for Python seems to be the best approach for prepackaged SVM calculations. However, after successfully using Scikit’s methods for classification, I’d like to attempt an implementation of my own SVM kernel, but use helper methods from other (to be determined) modules for LaGrange multiplier calculations.