Machine Learning Project Proposal

William Clark
will.clark@chicagobooth.edu
University of Chicago Booth School of Business

Matthew DeLio
mdelio@chicagobooth.edu
University of Chicago Booth School of Business

I. Summary

For our final project we will enter a Kaggle contest called Facial Keypoints Detection. The goal of the contest is to predict the (x, y) coordinates of the following facial keypoints for a given set of portrait pictures:

- center of each eye (2),
- inner and outer corner of each eye (4),
- end of each eyebrow (4),
- tip of nose (1),
- corners of mouth (2), and
- center of each lip (2),

for a total of 15 features per face. Our submission will be scored by the average root mean square error (RMSE) across all keypoints and images.

II. Data Set

Our data are a set of 8,832 96-by-96 grayscale images of tightly cropped portraits (sample images are shown in Figure 1). The images have been vectorized and each pixel is represented by its grayscale intensity on a $(0, 255)$ scale. The (x, y) coordinates of all facial keypoints are given, so there are 30 variables to predict for each image. Our models/algorithms will be tested on a testing set of 1,783 similarly formatted images.

III. Project Description

This project can be thought of in two discrete parts: (1) image pre-processing to reduce the dimensionality of our data set and make our modeling efforts more robust, and (2) a predictive modeling exercise to identify facial keypoints.

A. Image Processing

A common practice in other computer vision studies is to use principal components analysis to reduce the dimensionality of the images. In most cases, nearly all the variation across images can be explained with a fraction of the original data set. Rather than defining each image as a vector of length $p = 96 \cdot 96$, we can express it as a linear combination of the first x principal components where $x \ll p$. This process should reduce the noise in each vectorized image and increase the accuracy of the predictive algorithms we need to train and tune.

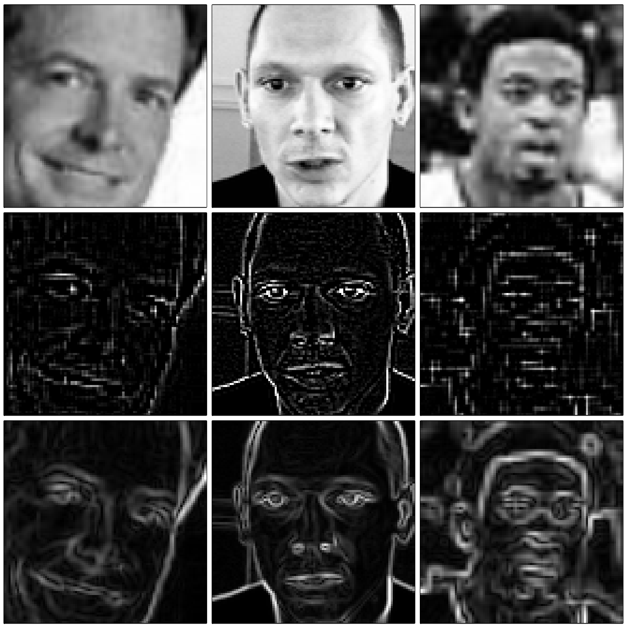

Fig. 1. Sample Faces in Training Data Set

In addition to PCA, we will also employ edge-detection to help us better identify the key components needed to perform the facial keypoint detection. Some of the filtering kernels we have already employed on the data-set can be seen in Figure 1 (see the second and third lines). We may end up combining this with neural network to create a multi-stage convolutional neural network. Unfortunately, since h2o does not support this, we may need to research some other languages (like python) with appropriate libraries/packages. We may explore the techniques used in the LeNet-5 by Yann LeCun as he has a tutorial on how to use Python to do something similar to what we're trying to do.

B. Predictive Modeling

This project requires training and tuning 30 different predictive algorithms/models, one for each x and y coordinate of all 15 facial features. The following list of models/algorithms is neither exclusive nor exhaustive, but represents our best current estimate of what the final paper might include:

- linear regression with L1/L2 penalization,
- nearest neighbor searches,
- regression trees and bagged regression trees,
- random forests,
- boosting trees, and
- neural networks.

With this variety of forecasting models, we could make our prediction based on the best performing model or best performing set of models (such that our final algorithm is an ensemble of some combination of listed models).