STAT 231 / CS 276A - Project 3

Face Social Trait and Political Election Analysis by SVM

Due December 2

1. Objectives

In this project, we apply Support Vector Machines (SVM) to scoring the social attributes/traits of faces. We consider a wide range of facial attributes including demographic (gender, race, age), geometric and appearance facial features, expressions, and even some high-level perceptual dimensions (i.e. attractiveness, trustworthiness etc.). Furthermore, we exploit such facial attributes to analyze election outcomes (winner or loser) and party affiliations (GOP vs DEM) of politicians. This study is motivated by prior behavior studies in psychology which suggest that people judge others by facial appearance. Some evidence was also found in election and jury sentencing.

This project is based on a paper by Jungseock Joo et al in ICCV 2015. All the materials and project theme are confidential, and provided only to the students in this class solely for educational purpose. Therefore, you must not distribute, use or present these materials for any purposes other than submission for this course project.

2. Data and code

Data and codes are downloadable in one zip file.

NOTE: Do not change the folder structure. Put all of your codes in this folder, work there, and submit only the codes in the folder. Do not use any absolute path in your implementation. Your code must run on any other machines as long as the same data is located under the working folder. Name your scripts as p1.m, p2.m, p3.m, ..., pk.m so that they can be tested sequentially. Do not submit any data or temporary files.

Provided Codes:
./demo.m - an example script
./HoGfeatures.cc - a mex implementation of extracting the so called HOG features.
./HoGfeatures.mexw64 - a pre-compiled mex file
./ libsvm_matlab/ - is the Matlab lib for svm

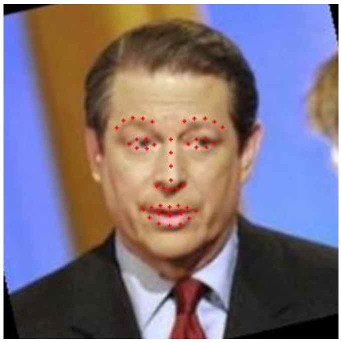
Data - LFW:
./LFW_meta.mat - attribute annotation, landmark, image file names, is_train
./ LFW_image/*.jpg - images

Data - face images of politicians
./Politic_meta.mat - election outcome (1: win, -1: loss), political party (1: DEM, -1: GOP), landmark,
image file names
./Politic_image/*.jpg - images

NOTE: The LFW dataset is a public benchmark dataset of facial photographs (http://vis-www.cs.umass.edu/lfw/). For your convenience, a subset of images (frontal faces) have been selected and facial landmarks have been detected from those images.

3. PART I : Facial Attribute Classification

The goal of the part I is to train binary SVMs to classify a series of facial attributes from a facial photograph. You use the LFW dataset for training and test. Each facial photograph is labeled with 73 binary facial attributes as well as facial landmarks, as shown below:



Male	1	Sunglasses	-1	Sideburns	0	No Beard	0	Heavy Makeup	-1
Asian	-1	Mustache	-1	Fully Visible Fo	1	Goatee	0	Rosy Cheeks	-1
White	1	Smiling	-1	Partially Visible	-1	Round Jaw	0	Shiny Skin	-1
Black	-1	Frowning	1	Obstructed For	-1	Double Chin	-1	Pale Skin	1
Baby	-1	Chubby	-1	Bushy Eyebrow	-1	Wearing Hat	-1	5 o' Clock Shad	0
Child	-1	Blurry	-1	Arched Eyebro	-1	Oval Face	1	Strong Nose-M	-1
Youth	-1	Harsh Lighting	-1	Narrow Eyes	1	Square Face	-1	Wearing Lipstic	-1
Middle Aged	0	Flash	-1	Eyes Open	1	Round Face	-1	Flushed Face	-1
Senior	-1	Soft Lighting	1	Big Nose	0	Color Photo	1	High Cheekbon	-1
Black Hair	-1	Outdoor	-1	Pointy Nose	1	Posed Photo	1	Brown Eyes	1
Blond Hair	-1	Curly Hair	1	Big Lips	-1	Attractive Man	-1	Wearing Earrin	-1
Brown Hair	1	Wavy Hair	-1	Mouth Closed	0	Attractive Wor	-1	Wearing Neckt	1
Bald	-1	Straight Hair	0	Mouth Slightly	0	Indian	-1	Wearing Neckl	-1
No Eyewear	1	Receding Hairli	1	Mouth Wide O	-1	Gray Hair	-1		
Eyeglasses	0	Bangs	-1	Teeth Not Visit	-1	Bags Under Eye	1		

3-A. Attribute Classification by Landmark. The first step of your assignment is to train SVM classifiers only using the provided facial landmarks as features (no your own feature extraction step). Write a script which reads the annotation file and the landmark file and trains 73 binary SVM classifiers -- one for each attribute dimension -- using the training examples. After training is done, you should apply the learned classifiers on the test examples and measure performance (classification accuracy) of the classifiers. Since the labels are imbalanced (different # of positive vs negative examples), you should report the average precisions.

NOTE : Through all experiments in this project, you need to measure both training error and testing error and report both in your report. When training SVM classifiers with LIBSVM or other libraries, you can specify a parameter ("C") to control the trade-off between classification accuracy and regularization. You should tune this parameter if you believe your classifiers are over-fitting.

3-B. Attribute Classification by Richer Features. The next step is to extract richer visual features (appearance) from the images. Here, you should include the HoG (histogram of oriented gradient) features and can additionally choose whatever feature you want to try such as LBP (local binary pattern) or color histogram. Then repeat the earlier step to train and test SVM classifiers, but using augmented features: [landmark] and [new appearance feature]. You can simply concatenate two types of feature vectors into one. Compare the performance with the previous one.

NOTE: You can further implement a joint feature representation such that you first locate a facial landmark and extract the appearance feature around the location. A similar treatment to this is the "warping" strategy that you used in the project 1.

4. PART II : Election Outcome Prediction.

Now, we use the learned classifiers to analyze the outcomes of real-world elections. You will use the photographs of US politicians for the experiment in this section. Each image in this dataset is labeled with i) election outcome (win or loss) and ii) party affiliation (DEM or GOP). Your task is to train SVMs to automatically classify these two variables given a photograph. (The election outcome is better defined when one considers a "pair" of politicians who run for the same race, but we will disregard the actual parings and simply divide all the examples into two classes: winners and losers to simplify the problem.)

4-A. Direct Classification. Using the same feature that you developed in the section 3-B, train two classifiers to classify the election outcome and the party affiliation. Therefore, you simply repeat the section 3-B with a different dataset and different output labels, but using the same feature representation.

NOTE: We do not divide the second image set into a train and a test set. Perform k-fold or leave-one-out cross-validation and report the average accuracy. The point is to achieve an accuracy higher than chance. Since the classes are imbalanced, you need to show that your accuracy or AP (average precision) is above 136/235 = 0.58 (which is obtained by majority classification, i.e. all winners).

4-B. Classification by Facial Attribute. We finally consider a two-layer-model in which we first project each facial image in the 73-dimensional attribute space and classify the election outcome and the party affiliation using the attribute-based representation. Specifically, you need to apply the classifiers that you trained in the section 3-B to each politician's image and collect all the outputs of 73 classifiers (use real-valued confidence instead of label). Then you treat these outputs in 73 categories as a new feature vector which represents the image. Therefore, you can again train SVM classifiers using these new feature vectors. Compare the result with 4-A.

4-C. Analyses of results. At a minimum, show the correlations between the facial attributes and the election outcomes. What are the facial attributes that lead to the electoral success?

NOTE: In order to verify the true effects of facial appearance to election outcomes, one need to control for many outside factors (covariates) such as clothing or image background. In this class project, we do not consider these variables for simplicity.

Submit your report and code in one zip file in CCLE.