Face Social Traits and Political Election Analysis by SVM

1 Objective

In this project, we follow the paper by Jungseock et al. in ICCV 2015. The rationale is that election outcomes can be predicted solely based on geometric and appearance facial features. Further, these features can be mapped to high-level concepts of perception such as attractiveness or trustworthiness.

We exploit such low-level facial features and high-level perceptual to analyze election outcomes 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.

Project 4 is an exercise for studying the social attributes of human faces using Support Vector Machines (SVM). The goal is to:

(1) train classifiers that can infer the perceived face social traits from low-level features,
(2) apply the model to analyze the outcomes of real-world political elections.

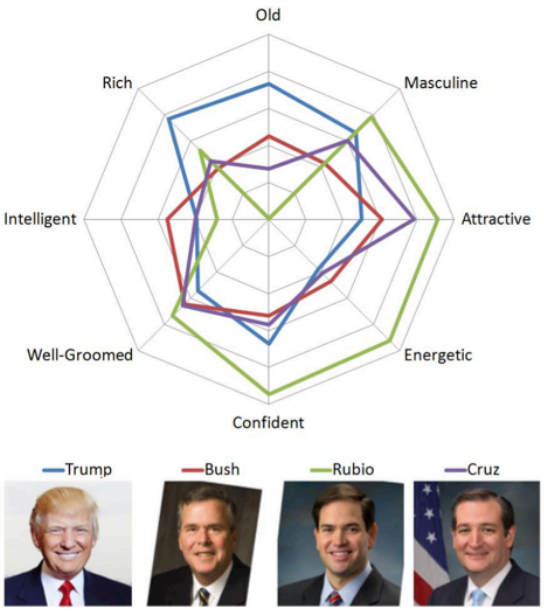

Figure 1. The inferred social dimensions of faces of a few Republican politicians who may run for 2016 presidency, predicted by our learned model.

[NOTE: 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 Tools

Images

.
/img # This is the set of images to train and test trait classifiers.
/M*.jpg ## 491 images

/img-elec # This folder contains two sub-folders (senator and governor).
/governor
/G*.jpg ## 112 images of governors
/senator
/S*.jpg ## 116 images of senators

Annotations

./train-anno.mat # This contains the perceived trait annotations (491 x 14 matrix). The 14 variables correspond to {*Old, Masculine, Baby-faced, Competent, Attractive, Energetic, Well-groomed, Intelligent, Honest, Generous, Trustworthy, Confident, Rich, Dominant*}

In addition, we provide you with pre-computed facial landmark coordinates (491 x 160 matrix). Each row represents 80 keypoint locations [x1, x2, …, x80, y1, y2, … , y80]

[NOTE: The trait annotations are obtained from ranking, thus you see ***real-valued*** scores instead of binary classes. Consider these options:
a) Set an arbitrary decision threshold and divide the whole set into a positive and a negative set, or
b) Use regression (e.g. SVR or RankingSVM).]

./stat-sen.mat
./stat-gov.mat # These two files contain the pre-computed facial landmarks for the Part II images and the actual voting share differences between the candidate pairs.

Tools

./libsvm_3.21/ # This is a library for Support Vector Machines. Refer to libsvm_doc.pdf for more details
./demo.m # An example script of using HoGfeatures.cc file
./HoGfeatures.cc # A mex implementation of HOG extraction.

3 Tasks

Part 1: Face Social Traits Classification (or Regression)

The goal of this task is to train binary SVMs (or SVRs) to predict the perceived traits (social attributes) from facial photographs. You can use the pre-computed facial keypoint locations and extract HoG (histogram of oriented gradient) features using the enclosed MATLAB function. You can further try your own favorite features.

[NOTE: We do not explicitly divide the image set into train/test sets. Therefore, you need to perform k-fold cross-validation and report the obtained accuracy.]

1.1 Classification by Landmarks

The first step of your assignment is to train 14 SVMs or SVRs only using the provided facial landmarks as features. Write a script which reads the annotation file and the landmark file. Then train 14 models - 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 number of positive vs negative examples), you should report the average precisions. Perform k-fold cross-validation to choose the LIBSVM parameters.

[Report: (1) Average accuracies and precisions on training and testing data for each of the 14 models. (2) The LIBSVM parameters of the 14 models.]

[NOTE: 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.]

1.2 Classification by Rich 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. Then repeat the earlier step to train and test your models, but using augmented features: [landmark] and [new appearance feature]. You can concatenate two types of feature vectors into one. Compare the performance with the previous one.

[Report: (1) Average accuracies and precisions on training and testing data for each of the 14 models. (2) The LIBSVM parameters of the 14 models. (3) The names of the features you have used. (4) Comparison to classification by landmarks (1.1).]

Part 2: Election C

2.1 Direct Prediction by Rich Features

Using the same features that you developed in the section 1.2, train a classifier to classify the election outcome.

[Report: (1) Average accuracies on training and testing data. (2) The chosen model parameters.]

[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 / precisions. The point is to achieve an accuracy higher than chance.]

2.2 Prediction by Face Social Traits

We finally consider a two-layer-model in which we first project each facial image in a 14-dimensional attribute space and second perform binary classification of the election outcome in the obtained feature space. Specifically, you need to apply the classifiers that you trained in the section 1.2 to each politician's image and collect all the outputs of 14 classifiers (use real-valued confidence instead of label). Treat these outputs in 14 categories as a new feature vector that represents the image.

Since each race comprises two candidates, a simple trick is to define a pair of politicians as one data point by subtracting a trait feature vector A from another vector B, and train a binary classifier: $\boldsymbol{F_{AB} = F_A - F_B}$. Do not include a bias term. Then you can again train SVM classifiers using these new feature vectors. Compare the result with direct prediction in 2.1.

[Report: (1) Average accuracies on training and testing data. (2) The chosen model parameters. (3) Comparison to direct prediction by rich features (2.1).]

2.3 Analysis 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?

[Report: Correlation coefficients of each of the facial attributes with the election outcomes.]