CS 276A: Project 3 Report

Face Social Traits and Political Election Analysis by SVM

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.

1.1 Classification by Landmarks

The data given to us were in .mat format. The annotation file contained the trait annotations which were extracted as labels {*Old, Masculine, Baby-faced, Competent, Attractive, Energetic, Well-groomed, Intelligent, Honest, Generous, Trustworthy, Confident, Rich, Dominant*} and face landmarks which were extracted as the features {(491 x 160 matrix) containing 80x and 80y facial locations for each image.}

I read the data and loaded it using scipy and scaled the features to (0,1) range. Then I calculated the mean to set the threshold and converted the labels to +1 and -1 accordingly.

I split the features and labels in a 80:20 ratio of training and testing data.

The algorithm used for training was the Support Vector Machine which is an inbuilt function of the ScikitLearn library. The SVC function (Support Vector Classifier) was used with the following parameters:

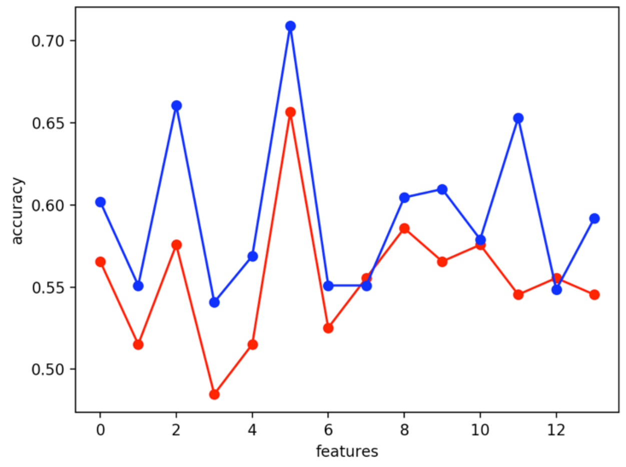
Gamma: 2^-17 to 2^5 over 10 different samples

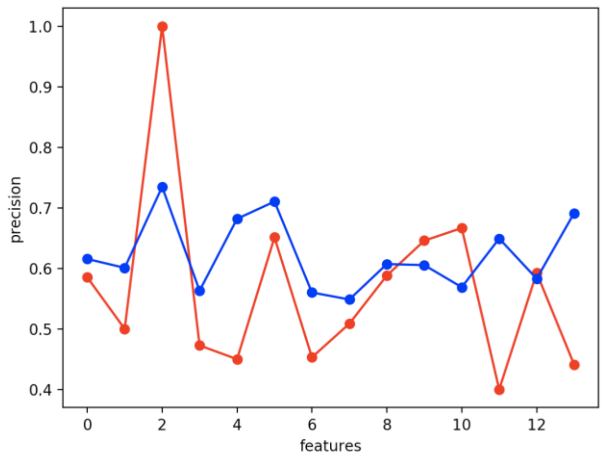
Cost: 2^-5 to 2^13 over 10 different samples

Then GridSearch was performed to find the best parameters for the classifier for each of the 14 features using the scoring function ‘accuracy’ and ‘precision’. The best parameters for each feature were as follows:

Parameters	1	2	3	4	5	6	7	8
cost	32.0	2.0	0.5	2048.0	2.0	8.0	8.0	0.5
gamma	0.0078	0.125	0.125	0.00049	0.03125	0.00782	0.125	0.5

Parameters	9	10	11	12	13	14
cost	32.0	2048.0	0.5	512.0	512.0	0.5
gamma	0.0078125	0.00049	0.125	0.000488	0.00195	0.03125





All the accuracies for each feature were reported to be better than 50% as they do better than random guess.

1.2 Classification by Rich Features

For thus part of this question, I used additional Histogram of Gradient (HOG) features to do classification. After reading each image I computed the Hog features using the hog function from the ScikitLearn Library. The orientation used was 32(sbin). After computing it, I concatenated the hog features with the original features for each image getting a vector of 30912 features for each of the 491 images.

For training I used a Support Vector Regression (SVR) algorithm which is an inbuilt function of the scikitlearn library of svm. The model was used with the following parameters:

Gamma: 2^-17 to 2^5 over 10 different samples

Cost: 2^-5 to 2^13 over 10 different samples

Epsilon: 2^-9 to 2^5 over 10 different samples

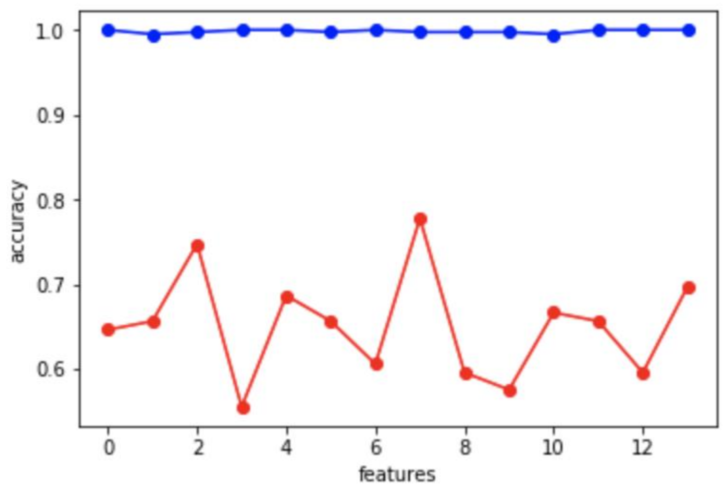
Then GridSearch was performed to find the best parameters for the classifier for each of the 14 features using the scoring function negative mean squared error (negmse). The best parameters for each feature were as follows:

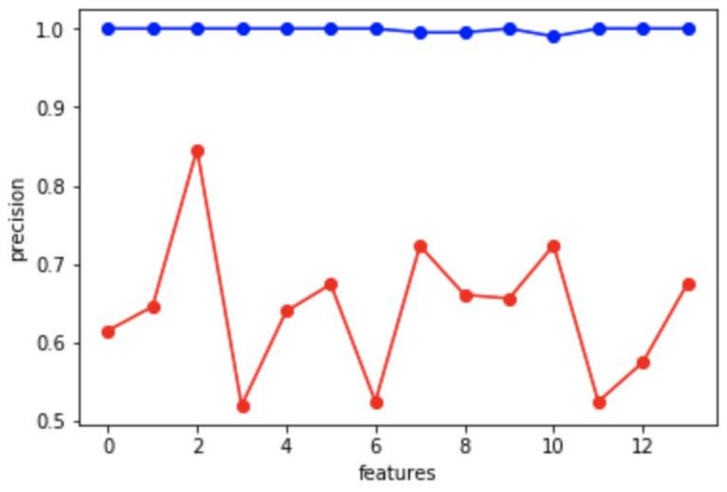
Parameters	1	2	3	4	5
Cost	910.25	8.0	8.0	2.0	8.0
Gamma	7.6293e-06	0.00012207	0.00012207	0.001953125	0.00781
Epsilon	0.001953	0.03125	0.001953	0.001953125	0.00048828

Parameters	6	7	8	9	10	11
Cost	512.0	2.0	2.0	2.0	2.0	8.0
Gamma	7.6294e-06	0.000488	0.000488	0.00195	0.00195	0.000488
Epsilon	0.001953	0.0019	0.00195	0.00195	0.00195	0.0078

Parameter	12	13	14
Cost	8.0	2.0	2.0
Gamma	0.00048828125	0.001953125	0.00048828125
Epsilon	0.125	0.001953125	0.0078125

After getting the best parameters the model was used to calculate the accuracy by predicting the training data and thresholding it with the mean to +1 and -1. Cross validation was done on 5 folds.
Then the model was used for testing the data and the testing accuracy was calculated in the same way. All the testing accuracies were greater than 50% and less than the training accuracies which were close to 100%.
The accuracies and precision for each feature was as follows:





All the accuracies for each feature were reported to be better than the ones reported in 1.1 as there are more features we train with than the first one.

2.1 Direct Prediction by Rich Features

For this part we use a rank SVM to predict the election outcome for governors and senators. The images were read like before and the features and hog features were computed in the same way as mentioned above and concatenated together.
The senator and governor data was read in pairs along with the voting differences. For half the data, the first value(loser) was subtracted from the second(winner) and a negative label of -1 was attached to it. For the other half of the data, the opposite was done, with the assignment of +1 label. Then the data was concatenated for both governors and senators separately.

After computing the full governor and senator datasets along with their corresponding labels, the feature set is scales to (0,1) range and then both the datasets are split in the 80:20 ratio with random state=20.
To implement rank SVM, the LinearSCV function from scikitlearn library is used with only one parameter of cost with values ranging from 2^-5 to 2^13 over 10 different values.

Then GridSearch was performed to find the best parameters using the scoring function accuracy and the fit_intercept parameter was set to False and hinge loss was selected for the LinearSVC function. The best parameters was:

C= 0.03125

The final accuracies were:
Governor:
training data: 0.690909090909091
testing data: 0.6666666666666666
Senator:
training data: 0.75
testing data: : 0.6086956521739

We achieve an accuracy above chance which is the aim of this project.

2.2 Prediction by Face Social Traits

For this question I loaded the classifier built in question 1.2 to get the values for each of the 14 fe atures.
The data was read and loaded the same way as above in section 2.2 and the features and hog feat ures were calculated for each image and appended together for the governor and senator separate ly. The features were scaled to the (0,1) range. The loaded model from 1.2 was used to predict th e governor and senator features and stored as traits. The difference in these traits were computed and a label was assigned to it in the same way as 2.1. Half of them were assigned +1 where the w inner traits were subtracted from the losers traits and -1 was assigned to the remaining half, in wh ich the vice versa was done. These trait differences were used as labels for classification in our ra nk SVM model.
The data was split into training and testing sets in the 80:20 ratio with random state=42.
To implement rank SVM, the LinearSCV function from scikitlearn library is used with only one parameter of cost with values ranging from 2^-5 to 2^13 over 10 different values.

Then GridSearch was performed to find the best parameters using the scoring function 'accuracy' and the fit_intercept parameter was set to False and hinge loss was selected for the LinearSVC function. The best parameters were:

C: 128.0 ~for the governor

C: 0.03125 ~for the senator

The final accuracies were:
Governor:
training data: 0.7954545454
testing data: 0.717391304347826

Senator:
training data: 7727272727272727
testing data: 0.6521739130434783

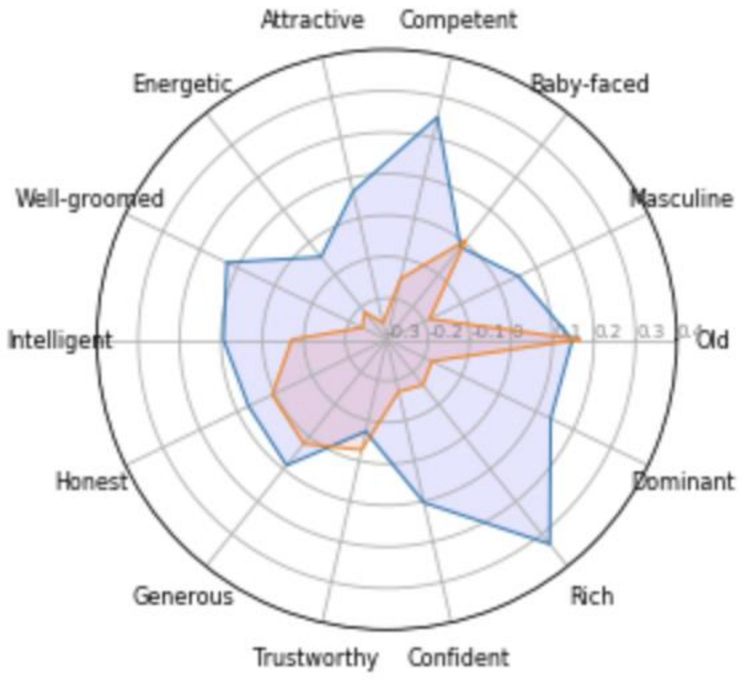
2.3 Analysis of Results

For this part the difference of the features were computed same as above, for governors and senators separately. The correlation between the difference in features and the absolute voting difference was computed using the corrcoef function of the numpy library for each of the 14 features to get an idea about which attributes in a person determined the election outcome.

The correlation values were as follows:

Features	1	2	3	4	5	6	7
Governor	0.148	0.053	-0.014	0.251	0.066	-0.045	0.1286
Senator	0.1651	-0.186	0.006	-0.149	-0.259	-0.215	-0.2342
Features	8	9	10	11	12	13	14
Governor	0.094	0.071	0.088	-0.07	0.104	0.3310	0.1392
Senator	-0.071	0.007	0.0213	-0.027	-0.171	-0.160	-0.1793

The correlation values were plotted in a radar plot and was as follows:


Attractive
Competent
Baby-faced
Energetic
Masculine
Well-groomed
Intelligent
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4
Old
Honest
Dominant
Generous
Rich
Trustworthy
Confident