STATS 231 – PATTERN RECOGNITION AND MACHINE LEARNING

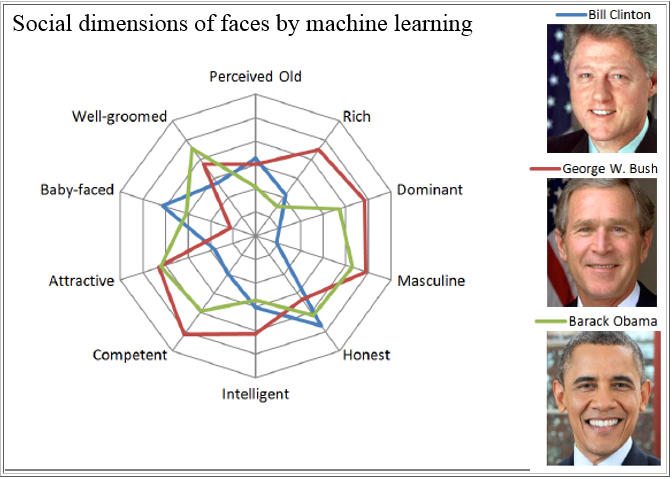
Project 2, Fall 2015

By Luis Sosa

004435372

University of California – Los Angeles

CREATING SUPPORT VECTOR MACHINES TO DETERMINE ELECTION RESULTS IN GOVERNER AND SENATORIAL CANDIDATE RACES USING SOCIAL ATTRIBUTES



**Abstract:**

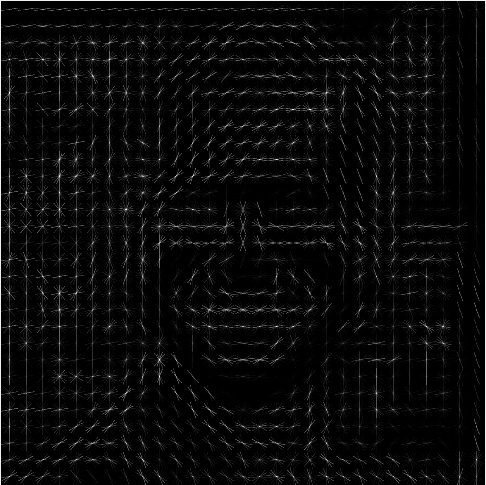
In this project, we are asked to create classifiers which study the social attributes of human faces using Support Vector Machines (SVM’s). We were given 491 facial images of political candidates, and two sets of data containing facial images of senators and governors. We were asked to use support vector machines to determine the social attributes of the faces of the politicians in this dataset—the social attributes being {Old, Masculine, Baby-Faced, Competent, Attractive, Energetic, Well-Groomed, Intelligent, Honest, Generous, Trustworthy, Confident, Rich, and Dominant}. The key idea here is to determine which of these social traits can inherently benefit politicians and help them win candidate races (by gaining more electoral votes). Here is an example of some of the training images we have which were used to create out Support Vector Machines:



Along with the images in our data, we were also given 80 landmarks per image (points throughout the face) which are used to determine the geometric aspect of political faces. We convert the 80 landmark points in one image to a vector of 160 x 1 (x1,x2,…..,x80,y1,y2,…,y80), and use this vector to train the support vectors which will help classify between the different traits.



The landmarks represent the geometric aspect, whereas the appearance aspect of the image is represented using Histogram over Gradient Values (HoG). The image of a test image being converted to Histogram over Gradient is shown below:

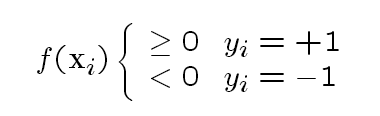


What this shows is a matrix of29x29x32 values (the images used in our dataset actually have HoG dimensions of 61x61x32 values) which describe the gradient orientations in localized regions of the image. The gradient orientations can be useful in determining different features of a face, which can be used to describe the social attributes—such as dominant face, or intelligent face. The 61x61x32 matrix can be reshaped into a vector of 119,072 x 1, which can then be used as a training attribute to our support vectors.

**Methods:**

For the first part of this project, we needed to design classifiers using the training images to predict the perceived trait scores. There are 14 traits for each of the 491 political candidates, and each trait corresponds to a ranking value between -1 and +1. Therefore a candidate who has a baby-face value of -0.9 will have a very “low” baby-face score, and a political candidate with a value of +0.9 will have a “high” baby-face score.

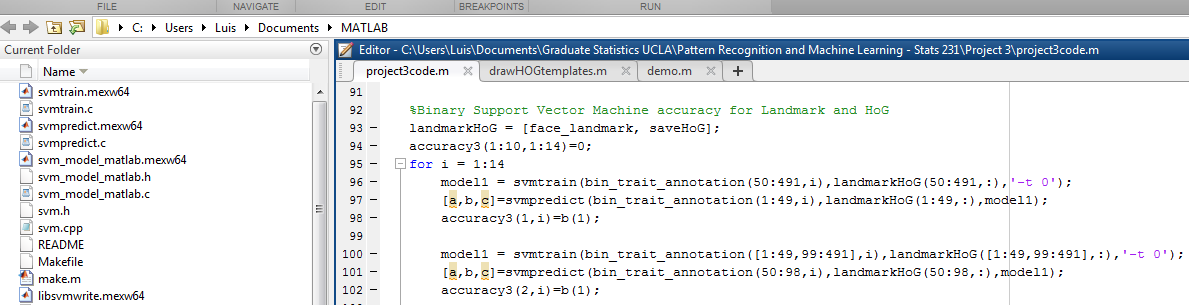
In this portion of the project, I decided that the best approach in training the support vectors is by using a Binary Support Vector Machine, where:



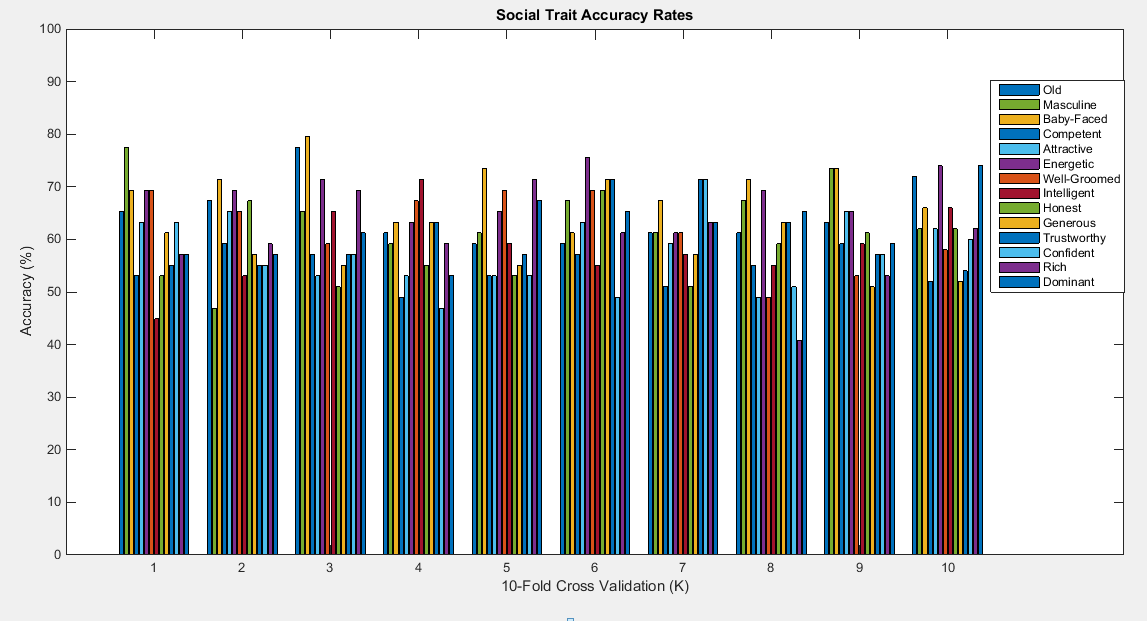
Meaning— if we make the threshold 0 among all the traits and separate the outcomes into a binary class (negative trait and positive trait), we can train binary classifiers to predict if the person has a “positive” social attribute, or a “negative” social attribute. In essence, we would be able to tell for example if, by observing the landmarks and the Histogram over Gradient values, whether candidate A is rich (positive), not intelligent (negative), not dominant (negative), masculine (positive), etc.

I use the “libsvm” library in Matlab to work with the support vector machine training functions, and perform a 10-fold cross-validation to compute the overall accuracy of these training images. I also concatenate the landmark vector (160x1) with the HoG vector (119,072x1) in order to create a new vector of 119,232x1 (which will be used as the training feature for our overall classifier).

Below is the code I used to implement the binary support vectors for the 15 social attributes. I first converted the social attributes to +1 or -1 (based on the sign of the trait) and used those values to classify the 119,232 features containing landmarks and Histogram over Gradient Values. The svmtrain model I uses a linear kernel **u’\*v**.



Below are the accuracy rates for the Binary support vector machines, for all 14 social attributes. Using a 10-fold Cross Validation, I have obtained many accuracy rates among predicting my social attrbitues. The lowest accuracy rate reported was at **40.8%** and the highest accuracy reported was at **79.6%**. Obviously, this is a big range so I used the cross-validation technique and computed the average accuracy reported, among all 14 social attributes.



The accuracy rates using the 10-fold Cross Validation among the social attributes were:

Old **= 64.75%**

Masculine **= 64.16%**

Baby-Faced **= 69.66%**

Competent **= 54.59%**

Attractive **= 58.65%**

Energetic **= 68.42%**

Well-Groomed **= 62.13%**

Intelligent **= 58.64%**

Honest **= 58.24%**

Generous **= 58.67%**

Trustworthy **= 60.50%**

Confident **= 56.41%**

Rich **= 59.67%**

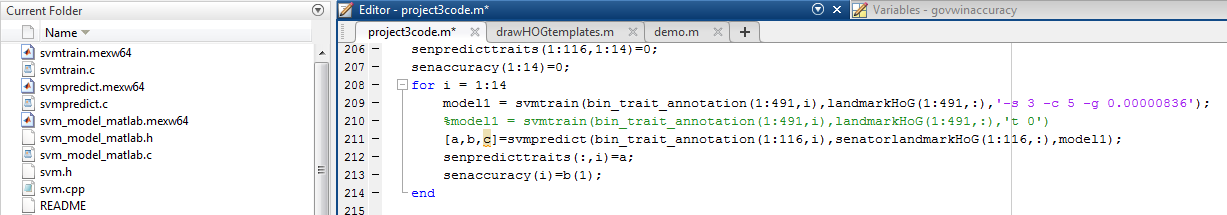
Dominant **= 62.30%**

Overall, the top three social attributes that we can predict using the political candidate’s landmarks and Histogram over Gradient features are Baby-Faced, Energetic, and Old traits. It also helps to know that the accuracy rates are all above 50%, meaning that my classifiers have a better chance at predicting traits to the correct category than flipping a coin. Still however, the accuracies are not extremely high (none of them pass 70% after 10-fold cross validation).

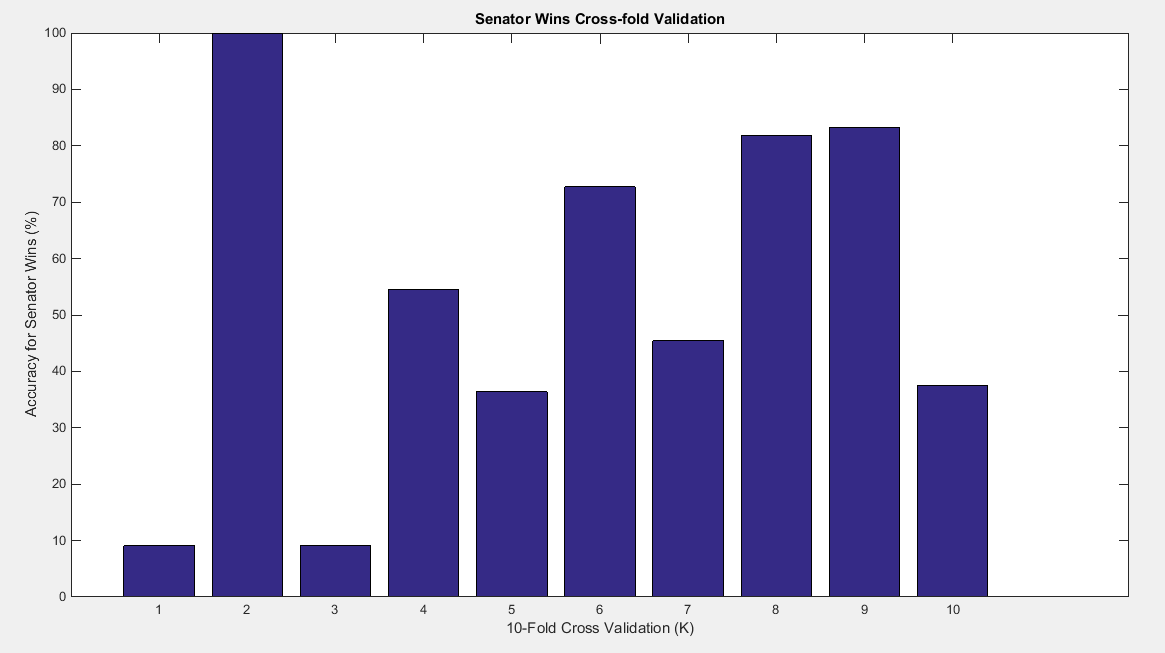
Now for the second part of this project—I use the classifiers which were trained in part one and use them to predict the traits of the senator and governor images. Then once I predict those trait values, I use those traits as features which will train an “election winner” classifier. The data provided for us in the class is a vector of “vote differences” between two candidates A and B. My method is using another binary Support Vector Machine (where there are two classes: +1 defining win and -1 defining loss). I can use my classifiers from part 1 to predict the senator and governor social traits using the concatenation of both landmarks and histogram over gradient values.

One modification I perform for this part is changing the binary support vector models from part one, and using a support vector regression (SVR) to predict the social traits instead. With these regression values (not classified values +1, -1), I can have better leverage on classifying between win +1 and loss -1. I do this by finding the difference in regression trait values for candidate A and B, then use that value has a feature which will determine who wins and who loses.

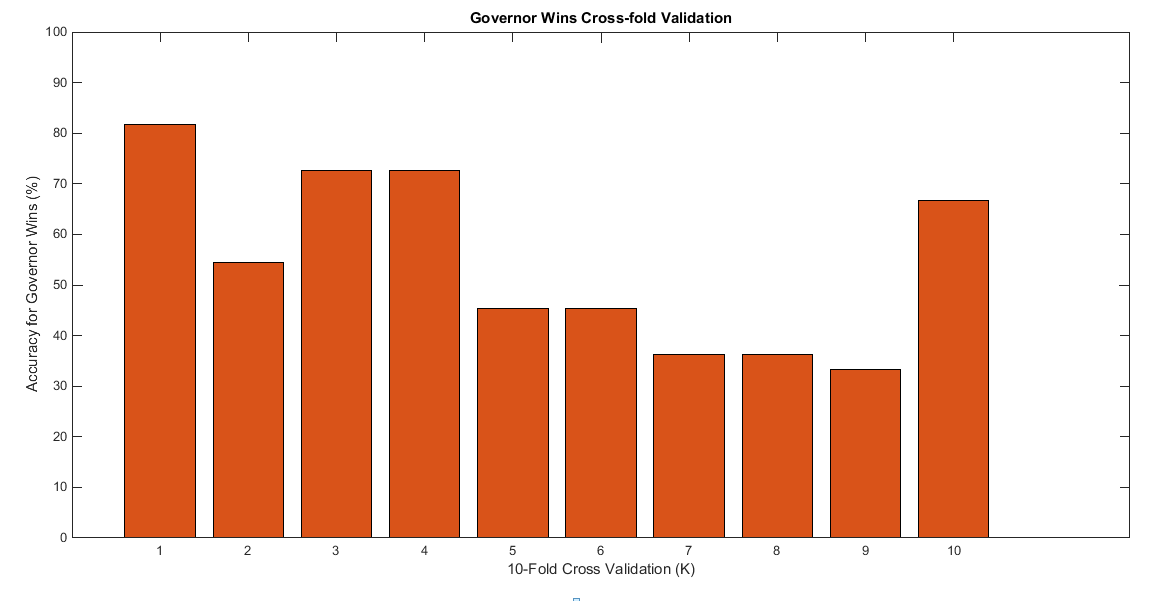
The code below displays the parameters I chose for the Support Vector Regression. I simply converted the vote difference vector into binary values of +1 and -1 in order to categorize between a win and loss.



The plot below shows the 10 Fold Cross-Validations for the Senator Candidate Races, and the Governor Candidate Races using the predicted support vector regressions:



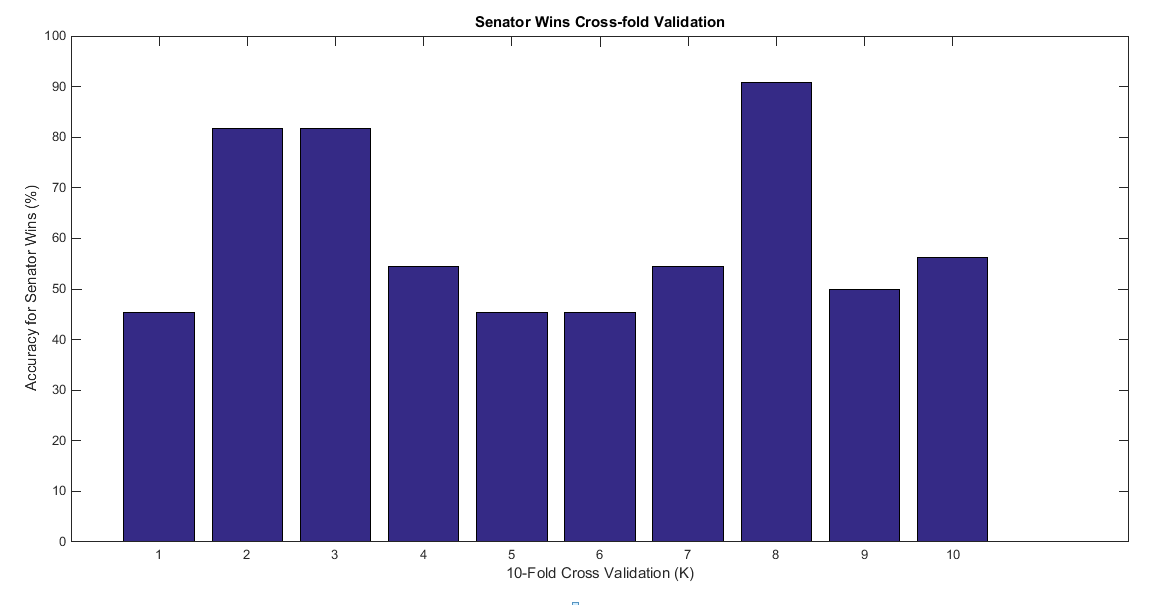
Here are the Accuracy Rates for the Governors:



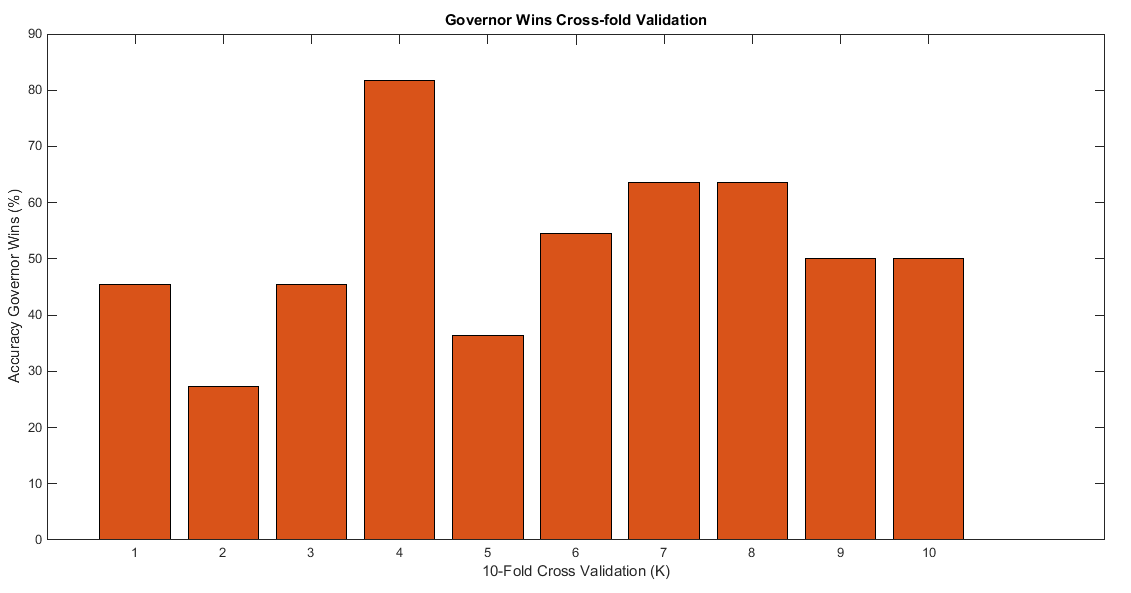
Using 10-Fold Cross Validation I was able to obtain an average accuracy of predicting senator wins by **52.99%** and governor wins by **54.55%**. Looking at these average accuracies, one can conclude that my support vector machines have a better chance of choosing a winner in the candidate race than flipping a coin, but the classifier is still not very strong. In some of the accuracy tests, our accuracies went to as low as **9.09%** (the first senator cross-fold accuracy), which means the classifier chose the wrong winner 10 out of the 11 times! So there is still much work that needs to be done in terms of improving the classifiers.

My next approach was to instead go back to my primary method of classifying, which are the binary support vectors. I decided to predict the trait scores for senators and governors by classifying them into +1 or -1 values (for each trait) and using these scores to predict the election winner. The difference here is that I am no longer using a Support Vector Regression (SVR) to estimate the trait value for the feature—I am simply classifying into +1 or -1. Then I will use these binary values as the training features for the election winning classifier.

Here were the senator win accuracy results using my new approach:



And here were the governor win accuracy results:



The average accuracy among the Senators were **60.63%**, whereas the average accuracy among the Governors were **51.82%**. These appear to be more accurate than the classifiers which were based off the support vector regressions; therefore this latter approach seems more effective in deciding wins in candidate races. The classifiers created to decide on a winning candidate however are not too amazing—60% is slight above 50% (so it is better than chance), but there is a 40% chance of still guessing the wrong candidate winner—so it is not very effective.

In terms of correlation, there were a couple of traits which seemed to have the largest positive impact in determining a winning candidate. For senators, being Old (.3217) was the most and only dominant feature for winning the senatorial candidate race. All the other traits had negligible correlation, or negatively impacted the odds of winning. For governors on the other hand, the top three traits which were correlated with winning candidate races were Dominant (.1950), Well-Groomed (.1871), and Masculine (.1793). The three worst traits were Generous (-.1950), Baby-Faced (-.1242), and Trustworthy (-.0802).

Correlation between winning candidate and social trait for Senators:

Old **= 0.3217**

Masculine **= -.1707**

Baby-Faced **= 0**

Competent **= 0**

Attractive **= Undefined**

Energetic **= -.3033**

Well-Groomed **= Undefined**

Intelligent **= -.1246**

Honest **= Undefined**

Generous **= Undefined**

Trustworthy **= Undefined**

Confident **= -.2627**

Rich **= -0.1857**

Dominant = **Undefined**

Correlation between winning candidate and social trait for Governors:

Old **= 0.1515**

Masculine **= 0.1793**

Baby-Faced **= -0.1241**

Competent **= 0.1010**

Attractive **= 0.0744**

Energetic **= 0.1538**

Well-Groomed **= 0.1871**

Intelligent **= -0.0257**

Honest **= -0.0720**

Generous **= -0.1950**

Trustworthy **= -0.0802**

Confident **= 0.1129**

Rich **= -0.0570**

Dominant **= 0.1950**