Predicting Candidate Responses in Presidential Debates

Seth Myers
Institute for Computational
and Mathematical Engineering
Stanford University

Stephen Kemmerling
Institute for Computational
and Mathematical Engineering
Stanford University

David Chin-lung Fong
Institute for Computational
and Mathematical Engineering
Stanford University

Abstract— **In this paper we tried to predict what a candidate in a political debate (namely George W. Bush) is going to say, based on what the moderator and the other candidate just said. In order to achieve that, we first used PCA to reduce the dimensionality of the problem, and then applied several regression models in order to do the actual predictions. While the PCA turned up some interesting patterns, the best test error (in a complex enough model) we achieved was about 70%.**

I. Introduction

Human thought is an incredibly complex thing, but simultaneously humans are creatures of habit. More specifically, most people seem to have a fixed pattern concerning how they argue on a given topic (the Authors included). While they might use different words, the points they make remain largely the same.

There is a group of people that is often seen as being particularly notorious, as far as always making the same point(s) goes: Politicians. This is made very evident during political elections when the two candidates argue over the same issues in a series of debates throughout the election without either one changing their viewpoint. We believe that learning what a politician is going to say in a given context will be easier than the general case.

We have chosen to model the presidential debates because of the availability of the data, and because the topics in such a debate are usually constrained to a loose theme, which might make predictions easier. President Bush will be our candidate of focus. He is the most recent candidate to participate in two elections (and thus six debates), so this will provide the most amount of data. The first five debates will serve as training data, and the last debate will be the test data.

Our goal is to predict candidate responses based on the questions asked by the moderator and the arguments made by their opponent. In doing so, we hope to develop a model of a candidate's thought process during such a debate.

In oder to account for the fact mentioned above, that they might make the same point using different words, and also as a form of feature selection we will first project the individual respones down into a lower dimensional *concept space*, using Principal Component Analysis. We will then try several regression models, namely Neural Networks, Linear and Polynomial regression and Support Vector regression, in order to make the predictions.

II. Approach

Due to the natural high dimensionality of written text in combination with the fact that some of the machine learning algorithms we will be using are computationally expensive, it is necessary to first map the text into a lower dimensional space. We will assume this can be done linearly. We begin by removing stop-words from the text, and then replace all words with their word stem. Next, we let X be the word vector for a speech made by a candidate or the moderator during the debate (per usual, the i^{th} element of X is the number of times word i is used in the speech). Thus, $X \in \Re^n$ where n is the number of different words used in the all of the debates. Let $\Phi \in \Re^{k \times n}$ where k is the desired dimension of the reduced text space. The goal is to find Φ such that $\hat{X}$, defined as

$$\hat{X} = \Phi X,$$

is a meaningful projection of X onto this k-dimensional space. We have run initial experimentation on values of k as high as 250 and as low as 10. From this point forward, this lower dimensional space will be referred to as the *concept* space, and the basis with which we define the concept space (i.e. the rows of Φ) will be referred to as *eigenconcepts*. It is important to note that both the input variables, i.e. the speeches made by the debate moderator and Bush's opponents, and output variables, i.e. the speeches made by Bush in response, will be projected onto the concept space. We implement two techniques for determining Φ: latent semantic analysis and principle component analysis.

Once the debates have been mapped into the concept space, we train various machine learning algorithms to predict the main content of the speech given by the candidate of focus in response to the speech delivered by the moderator and the opponent. The techniques used are:

- Linear Regression
- Polynomial Regression

- Neural Networks
- Support Vector Regression

A. Latent Semantic Analysis

With reference to Alvarez-Lacalle [1], we break down every speech made by the candidates and the moderators into eigenconcepts using a variation of latent semantic analysis. The algorithm goes as follows: first, we construct a co-occurrence matrix of all words used in five of the six debates Bush has participated in (the last debate will be reserved as our test data). This matrix will be a tally of how often any two words occur in the same speech. Next, we normalize the matrix by the expected co-occurrence of the words in a random text. We apply singular value decomposition to the co-occurrence matrix, and the resulting eigenvectors are the eigenconcepts. Each eigenconcept corresponds to a singular value as a result of the decomposition, and we can interpret each singular value as a measure of importance of the eigenconcept. Thus, we define Φ such that its rows are the eigenconcepts corresponding to the k largest singular values.

B. Principle Component Analysis

The second method we implement for deriving the eigenconcepts is Principle Component Analysis applied to the word vectors of all the speeches. Ignoring speeches of less than 20 words, our eigenconcepts become the top k eigenvectors that PCA yields.

C. Linear Regression

To investigate the structure of the data, we run the linear regression model on our data. Suppose we have the input data (speech by moderator and opponent) as X, and the output data (speech by Bush, the candidate of interest) as Y, where X, Y are in the same format as in our lecture notes, with the difference that the output being in the same dimension as the input. We define $\hat{X}$ and $\hat{Y}$ as the projections of X and Y onto the concept space, i.e.

$$\hat{X} = \Phi X$$

and

$$\hat{Y} = \Phi Y.$$

We have our hypothesis matrix M satifying the equation.

$$\hat{X} M = \hat{Y}$$

By minimizing the empirical error, or maximizing the likelihood with the error term following normal distribution, we have

$$\hat{M} = (\hat{X}^T \hat{X})^{-1} \hat{X}^T \hat{Y}$$

D. Polynomial Regression

We repeat the same procedures as in Linear Regression, with the exception that we replace our input matrix $\hat{X}$ by a extended matrix with higher order terms $\tilde{X}$, i.e. For a regression of n-th degree

$$\tilde{X} = [\hat{X}^{(1)}\,\hat{X}^{(2)}\,\cdots,\hat{X}^{(n)}]$$

where each $\hat{X}^{(i)}$ is the matrix $(\hat{X}^{(i)})_{jk} = \hat{X}^2_{jk}$. Also note that $\hat{Y}$ is not changed.

E. Neural Networks

All Neural Networks we use are feed forward networks with a symmetric sigmoid activation function. We will try several different hidden layer sizes, as well as different numbers of hidden layers. All neural networks are implemented using the "Fast Artificial Neural Network Library" (which can be found at http://leenissen.dk/fann/). In an attempt to improve results we perform several transforms on the data:

- Using only the top n eigenconcepts as output features.
- Using only the top m eigenconcepts as input features.
- Normalizing the concept vectors.
- Rescaling the individual eigenconcepts to $[-1, 1]$.
- A very simple approach to try to account for context. Politicians in debates will sometimes use their allocated time to answer or elaborate on previous questions, so we attempt to take this into account. Specifically, viewing a debate as a sequence of concept vectors, instead of trying to learn a function f, s.t.

  $$f(x^{(2k-1)}) = x^{(2k)}$$

  trying to learn a function g, s.t.

  $$g\left(\prod_{i=1}^{k} \gamma^{k-i} \cdot x^{(i)}\right) = x^{(k+1)},$$

  for $\gamma < 1$.
- Nearly all meaningful combinations of the above.

F. Support Vector Regression

If $x^{(i)}$ is the i^{th} speech made by the moderator/opponent and $y^{(i)}$ is Bush's corresponding response, then we let $\hat{x}^{(i)}$ and $\hat{y}^{(i)}$ be the corresponding projections onto the concept space. We then use support vector regression to model each component of $\hat{y}$ individually. The optimization problem

$$\begin{aligned} &\min \frac{1}{2}||w_j||^2 + C_j \sum_{i=1}^{m} \epsilon_i \\ &\text{s.t. } |w_j^T \hat{x}^{(i)} + b_j - \hat{y}_j^{(i)}| \quad < \quad \epsilon_i;\ i\text{=1,...,}m \end{aligned}$$

is performed for all $j = 1, ..., k$. A gaussian kernel of width γ_j will be used, and Leave-one-out-cross-validation will be performed to find the optimal values of γ_j and C_j. We will be using the LIBSVM library [2].

III. Main Results

A. Latent Semantic Analysis

Our implementation of this analysis has yielded word groupings within the eigenconcepts that are nonsensical (it is generally expected that latent semantic analysis yields eigenconcepts that demonstrate coherent ideas through the words associated with them). Additionally, regression models built on the LSA eigenconcepts yield extremely high error, and we attributed this to a poorly defined concept space. This failure can be attributed to the fact that this method of LSA was originally designed to reduce the dimensionality of long prose such as novels. The debate transcripts simply did not contain speeches long enough for LSA to extract meaningful word correlations.

B. Principal Component Analysis

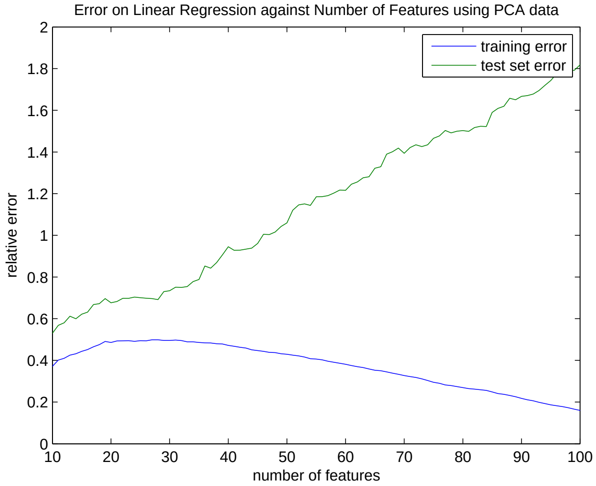
The PCA was successful. For a given eigenconcept, taking its inner product with a word vector of a single word generates a measure of association for the word to the eigenconcept. This is because the inner product is equivalent to observing the component of the eigenconcept corresponding to the word. Often, the words most positively associated with an eigenconcept are related in meaning and demonstrate a coherent idea, and these ideas are usually political issues such as stem cell research or gun control. This is also true for negatively associated words, but they describe a completely independent idea from the positively associated words. This implies that a single eigenconcept can effectively express two different political issues. See Table I for more details.

TABLE I
Sample Eigenconcepts

Eigen-concept	**Most Positively Associated Words**	**Most Negatively Associated Words**
1	school, governor, children, public, teacher, oneonon, district, privat, tuition, money, learn, classroom, educ, feder, account	iraq, war, saddam, hussein, troop, terror, weapon, world, threat, mass, afghanistan, destruct, laden, bin, qaida
4	law, gun, enforc, societi, background, school, trigger, lock, learn, lawabid, carri, age, licens, read	technolog, energi, oil, shortterm, sourc, develop, cleaner, incent, wildlif, environment, arctic, refug, oversea, deep, ga,
20	cure, embryon, bodi, ethic, scientist, michael, embryo, refug, artic, chri, oil, visit, oversea, god, embarrass	greati, program, neighbor, land, common, eat, expand, health, uniqu, care, middl, situat, difficult, practic, easi

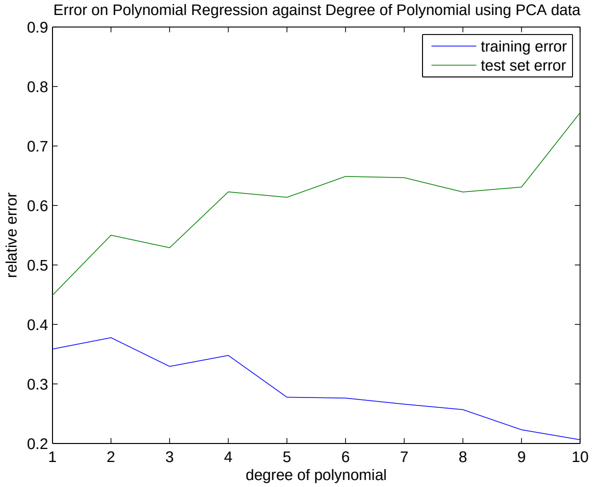
C. Linear Regression

The linear regression model was trained using different number of features (i.e. different values of k) each time. We start from $k = 10$ and increment it up to $k = 100$. We obtained a test set error of 53% for 10 features. This shows that a not-too-small portion of the speech-response mechanism was governed by a linear component. The test set error increases as the number of features that we try to predict increases, and it goes beyond 100% if we try to predict more than 50 features.



D. Polynomial Regression

The polynomial regression model was trained with 10 feature, while changing the degree of polynomial being used. We started from the linear version, up to the 10-th degree polynomial. We see that from the test set result that the linear regression is working better than all the higher degree polynomials. Therefore we conjectured that the essence of the speech-response might be better represented by the cross-term among different features, with which the essence should be captured by neural networks.



E. Neural Networks

The general behavior of the error vs. the number of eigenconcepts to predict was about the same as the linear regression. The best results in general, i.e. independent of any other transforms on the data or network type, were achieved with predicting only the top 10 eigenconcepts, but using the top 250 eigenconcepts as input data.

No significant improvement could be achieved with further lowering the number of eigenconcepts to predict. The best absolute error was achieved with rescaling each eigenconcept to $[-1, 1]$: approximately 0.025 Mean Square Error for each concept. As most concepts were close to zero in our test data, this did not produce the best relative error.

The best relative error (in the normal linear algebra sense) was achieved with normalized data, on a network with one hidden layer of 30 nodes, predicting only the top 10 eigenconcepts, but using the top 250 eigenconcepts as input features. The error was nevertheless still about 70%.

Multiple hidden layers in general overfitted the data very quickly and therefore did not improve the test set error at all. As far as the size of the hidden layer is concerned, best results were generally obtained with hidden layer sizes of about 2 to 3 times the number of eigenconcepts to predicts.

Additionally, implementing our method that accounts for candidates referring to previous questions in their responses increased the error in the predictions significantly. This is most likely due to it introducing too much noise for our comparatively little data set.

Of note is that the best test set error achieved with the neural networks was higher than that of the linear regression, at least with a low number of output concepts. However, because of the fact that the training error for neural networks goes down very quickly, whereas for the linear model it does not go to zero at all, we think that the neural network model has more promise in a situation where more data is available.

F. Support Vector Regression

The support vector regression yielded a mean square error much larger than the neural network method (on the order of 1), so we did not pursue this method further. One possible explanation for SVR's failure is that in predicting the presence of each eigenconcept independently it ignores the relational properties between eigenconcepts.

IV. Conclusion

From our implementation of a range of different machine learning techniques, we see that a significant portion of the speech-response human process has been captured computationally. The generalized error, however, remains quite high, even after experimenting with a varying number of features. The high variance of our regression models, in combination with the size of our training dataset (only 127 training examples), suggests that our results can be improved by using more data. A possible source of more training data are political interview transcripts. The reponses should be more standardized and a much larger amount of data would be available. All in all, the project succeeded in probing the tip of the iceberg of the human thinking process, as far as in a structured dialogue.

References

[1] E. Alvarez-Lacalle, B. Dorow, J. Eckmann, and E. Moses. Hierarchical structures induce long-range dynamical correlations in written texts. *Proceedings of the National Academy of Sciences of the United States of America*, 103:7956–7961, 2006.

[2] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.