Case 2

By: Anton Ruby Larsen s174356, and Mads Esben Hansen s174434

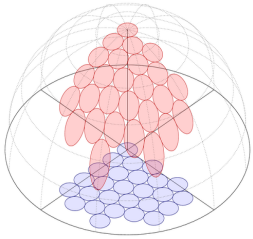
Introduction

Given speeches from the Danish parliament, we wish to generate a model that uses pattern matching and feature selection from said speeches. In turn we want to train our model to be able to recognize the jargon of a given party, and from a new speech "guess" what party the speaker belongs to.

Latent space

The most famous of all the latent space models is probably the PCA method. When doing PCA we rotate the feature space so the first axis of our space describes the must variance as possible, the second the second most possible and so on. This way we can choose a subspace containing fewer features than the original with a minimum loss in variance description.

There is however one problem with the method. It corresponds to getting a shopping list of the most frequent groceries in danish food recipes and then buying the 10 most frequent once. When you then get home you get the recipe of the dish you are to cook. There is a risk of missing some of the ingredients going into the dish. We will therefore consider another method called partial least squares. We want some new rotation R of our feature space that instead of maximising variance of uncorrelated directions maximises the covariance between the feature space and our classes we want to predict.



Results

Data

We chose to train our model on 50k speeches chosen at random throughout the period. Leaving about 100k for testing. Normally we would have used more as training data but this was simply infeasible given our hardware restrictions.

NLP

We found that vectorizing each text and pruning away all words present in more than 95% or less than 0,5% of the text gave us the most stable results.

Latent Space

We chose to go with PLS as our way of reducing our feature space. We ended up using 60 components. We had evidence that suggested we needed more, but again we ran into hardware constraints.

Method	NN on PLS	EN on PLS	NN on vocab	EN on vocab
Accuracy	39,3%	39,9%	35,2%	37,5%

References

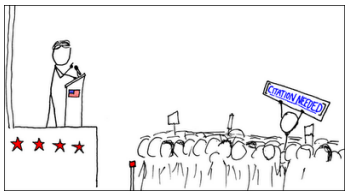
NLP figure: Wikipidean Protestor - https://xkcd.com/285/

Latent space figure: texample.net - http://www.texample.net/tikz/examples/dome/

Conclusion figure: cara.dk - https://cara.dk/klimavalg/socialdemokratiets-svar/attachment/socialdemokratiet/

NLP

Before we can begin analyse the data further, we need to do some Natural Language Processing (NLP). In order not to sacrifice efficiency we used *text2vec* library in R, which generates frequency vectors for each text. We then pruned away many redundant word i.e. stop words, very very used words, and words used by very few. This minimized number of computations needed and ensured comparability. Finally we joined all frequency vectors to form a Document Term Matrix (DTM) which we then used in our analysis.



Model selection

Elastic net

Due to the huge amount of features in text classification there is a obvious risk of suffering under ***the curse of dimensionality***. We therefore thought that elastic net (EN) was a obvious choice due to its ability to eliminate redundant features.

Neural Network

Another route is training a Neural Network (NN) to classify the speeches. NN superior ability to explain non-linear behaviour in data due to its non-linear activation functions. In general NN are prone to over-fitting the data, which is why its important to find lower dimensional manifolds that explains the data using e.g. PLS.

Conclusion

We found that using PLS does not significantly improve performance. However it greatly reduced training time of either model.

We managed to use methods and theory given in the second part of the course and create a model that was able to identify what party a given speaker is from fairly well.