Preconceptions

Dataset = started with Enron corpus dataset, approximately 500k emails after cleanup ended with 16,769 emails.

Examples: $X := \mathbb{R}^d$ where d holds for the dimension of the features.

features

1. Content-Length: Length of the email body, for example:
   "Hello Dan, how are you?" the content length is 25.
2. Number of Recipients: number of recipients of the email.

The following two features is word Embedding features:

- words are converted into d-dimensional vector.
- The average of the words vector creates single d dimensional vector.

3. Email Subject Embedding, example: Subject: "Meeting Tomorrow at 10AM"

Word2Vec vectors: $V_{Meeting}, V_{Tomorrow}, V_{at}, V_{10AM} \in \mathbb{R}^d$, the final Subject Vec:

$$V_{subject} = \frac{V_{Meeting} + V_{Tomorrow} + V_{at} + V_{10AM}}{4} \in \mathbb{R}^d$$

4. Email body Embedding: Same approach as Subject Embedding:

$$V_{Body} = \sum_{i=1}^{n} \frac{V_{word\,i}}{n} \in \mathbb{R}^d \quad | \text{where } |Body| = n$$

Labels

Created 7 labels in order to categorized:

Personal

hr

Meeting and scheduling

Operations and logistics

Corporate and legal

Projects

Finance

note: We started of with 9 labels as we progress we remove two features (all details and justification in the code).

Hypothesis class H

- consises of all possible mapping from X to the selected labels.

x = features, $X \in \mathbb{R}^d$

y = Lables

$h \in H$ = trained classifier, defined as $h: X \rightarrow y$

The process ahead

Dataset

Enron dataset is a huge dataset of emails, that requires general and strict cleanup.

Owing to the fact we cannot inspect the emails manually, much noise and outliers is expected.

Lables

Given that the emails are organized at folders, after the cleanup we should remain with properly categorized emails to the respected folders.

Eventually the folders rule is to be our "ground truth" A.K.A Labels.

note: We will create new folders by merging semantic related folders (the process and details in the code)

features

Crucial part is the analysis of the email text, hence much consideration given to the NLP model that handle vast amount of data while maintaining the semantic relation of the text.

Models

Given we have labels, supervised models ofcourse will be used.

Since our objective is to categorized emails, classification models naturally comes to mind. we have chosen to analyze the data with the following:

- Logistic regression
- Support vector machine (SVM)
- Random forest

Multi class Logiseic Regression

In this model, we demand multi-class classificeion and not binary One. two approach were considered:

1. One-Vs-All

- Train 7 seperate classifiers. very expensive on high dimensional deee es exte embedding.
- choose the class with highest probabiliey.

  Sor each class K the probabilies of en email belong to class K.

  let $z = w_K^T x + b_K$ where:

  - $x \in \mathbb{R}^d$ feature vector,
  - $w_K \in \mathbb{R}^d$ weight vector for class $K \in \{1, 2, \ldots, 9\}$
  - b_K is the bias for class K
  - recall sigmoid sunceion: $\sigma(z) = \frac{1}{1+e^{z}}$.

$$P(y=k \mid x) = \sigma(z) = \frac{1}{1 + \exp-(w_K^T x + b_K)}$$

2. Softmax Regression

- Crain single classifier for all classes simultaneously
- calculate the probalieies for $x \in \mathbb{R}^d$ that input x belong to each class $K \in C$ as C holds for the set of classes.
- Probabilities Normalized by softmat, sum to 1.

- Softmax generalization the sigmoid function for multi class classification

$$P(y=K \mid X) = \frac{\exp(W_K^T X + b_K)}{\sum_{j=1}^{|C|} \exp(W_j^T X + b_j)}$$

In conclusion

as Softmax train one classifier instead |C| classifiers,

the computational cost of softmax is much faster.

Thanks to words embeddings, which increase the feature space,

computational speed takes place as primary factor,

given that we have selected the softmax approach.

Random forest

Due to the noise and imbalance nature of the dataset linear predictors struggle to capture meaningful patterns, here owing to the Random forest randomness:

- Rf is more robust to noise dataset compare to linear predictors.
- Rf is more robust to overfitting than using K decision trees with OVE(one vs all) approach.

On the other hand emails are represented using word embedding, leading to high number of features that Random forest struggles with.

We keep in mind that text based classification problems often results in complex non-linear decision boundaries.

In conclusion

We expect Rf to deliver reasonable results, however we keep in mind that due to the curse of dimensionality its performance may be overshadow by other models better suited for high-dimensional analysis.

Support Vector Machine

When considering SVM for this dataset, a clear approach emerges:

leveraging Soft SVM to take advantage of its qualities such:

- Soft SVM more robust to noisy data.
- In real world email classification, categories often overlap, Hard SVM sets in such cases while Soft SVM more flexible and robust.
- SVM is well-equipped to handle complexities of high-dimensional dataset, such as those found in Enron dataset.

In conclusion

We expect Soft-SVM will perform well due its clear advantages regarding Enron dataset.