Public Notices: Classification and Social Topics
CSE 519:Data Science Fundamentals
Project Progress

1 Project Overview

Public notices exist to service their local community and alert them of any necessary or helpful information happening around them. Due to this these notices inherently have a very wide range of topics(notice types). In this work we explore the effectiveness of building public notice classifiers such that given the content of the message, it will label it with a related notice type. To do this we set out with a dataset of thousands unlabeled notices, where the first step is to hand annotate a subset of them.

We start by hand annotating public notices for various reasons. First, we need to gain some domain knowledge into what these notices contain and how they're written. Secondly, after gaining domain knowledge we are interested in seeing if we could develop our own set of labels that made more sense than using some naive approaches. Another advantage of curating our own hierarchy of possible types is that since many public notices may vary from state to state we try to alleviate this and create clusters that exist on a general level. As of now, we have agreed on using a set of 6 labels(Auction, Industrial, Judicial, Family/Personal, Death, and Misc). We take our hand annotated labels and use these to bootstrap a larger dataset. By performing unsupervised learning we can now take our knowledge and attempt to spread that across the hundreds of thousands of notices that would be in-feasible to hand label.

After performing our base classification task, we want to take these public notices and see if they can give us regional insights. By grouping the notices into county of origin we will know how many of each type occurred there. Specifically, we are going to leverage our auction label as a proxy of bankruptcy. By using data gathered from Twitter, we will generate topic vectors for these counties. These topic vectors will act as inputs into a regression task where the label is the number of auction notices in that county. These predictions are then going to be correlated with the actual number that occurred, we hope to see that regional language use is highly correlated, so we can use it for downstream predictions.

2 Current Progress

2.1 Dataset Generation

As mentioned, our first step was to hand annotate some public notices so we could get an understanding of what data we were working with. At the time of writing this we have just over 300 notices labeled between group members. At this point we feel that there is enough effort put in towards understanding the nature of these notices and will be moving onto mass unsupervised labeling. We have also looked into which unsupervised technique we feel would perform well for extending our hand crafted labels to the rest of our dataset; we plan to use k nearest neighbors(k-NN). We believe k-NN will do a good job labeling the remainder of our large dataset because we have hundreds of training examples that should create natural clusters among the notices. At the time of writing, we have begun working on the k-NN code but do not have a fully functional version, we hope to finish this within the coming days.

2.2 Factor Analysis

To aid our annotation task and gain confidence in our labels we use an unsupervised technique of discovering the kinds of words/phrases that group together. The technique we use is called Factor Analysis(FA). FA works on the notion that observable variables can be reduced to fewer latent factors that share a common variance and are 'unobservable'. For example, scores on an oral presen-

tation and an interview exam could be placed under a factor called communication ability; in this case, the latter can be inferred from the former but is not directly measured itself. Moreover we use Exploratory Factor Analysis (EFA). EFA tries to uncover complex patterns by exploring the dataset.

In the FA mathematical model, p denotes the number of variables $(X_1, X_2, , X_p)$ and m denotes the number of latent factors $(F_1, F_2, , F_m)$. X_j is the variable represented in latent factors. Hence, this model assumes that there are m underlying factors whereby each observed variables is a linear function of these factors together. In equation 1, this description of FA is shown:

$$X_j = a_{j1}F_1 + a_{j2}F_2 ... a_{jm}F_m + e_j \qquad (1)$$

The factor loadings are $a_{j1}, a_{j2}, , a_{jm}$ which denotes that a_{j1} is the factor loading of j^th variable on the 1^{st} factor. The specific or unique factor is denoted by e_j. The factor loadings give us an idea about how much the variable has contributed to the factor; the larger the factor loading the more the variable has contributed to that factor. Factor loadings are very similar to weights in multiple regression analysis, as they represent the strength of the correlation between the variable and the factor.

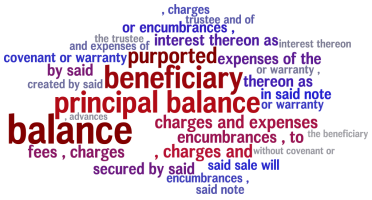
Having described the method in theory, let us describe how we did FA. We first generated a feature table which consisted of all 1,2 and 3 grams that the notices contained and assigned a normalized score to each of these words/phrases for each notice based on the occurrences. Thus we had a $notices * total1, 2, 3grams$ dimension matrix. We performed EFA on this. Now since we had labelled with 5 main categories and 1 "other" we chose 5 latent components so that we could directly compare these factors vs our labels. As we can see in the figures provided,

Now since we see this clear but not extreme discreteness among the five factors, the next challenge was, as in all of EFA, to identify how many latent factors explain most of the variance. This is conventionally done by a scree plot where in all points above or above and including the point at the "elbow" are considered to be the most contributing factors. We plotted the scree plot as seen in figure 6. There is a clear elbow that can be seen at the 6^{th} dot, backing our choice in choosing the 6 labels to categorize the public notices into. Thus our choice for labels is justified even by an exploratory and unsupervised technique and is the reason why we will further use these labels for annotation.

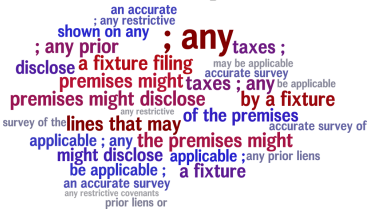
After extracting our 5 latent factors we also generated word clouds to see which words/phrases were positively and negatively contributing to each factor. This was done to see if our model had any interpret-ability when designing a mapping of factors to our notice types directly. For example, if you look at the positive aspect of component one(1a) we see words such as 'beneficiary', 'secured by', 'trustee' which seem to show a large relation with notices that discuss notification of death. Similarly, the positive component of factor 3(3a) uses language of 'management', 'commission of', 'organization' which show a mapping to our industrial factor. This exercise can be done for all the factors and we have shared all the positive/negative word clouds below.

(a) Factor 1 positive

(b) Factor 1 negative

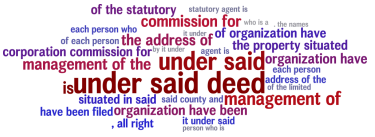


(a) Factor 2 positive

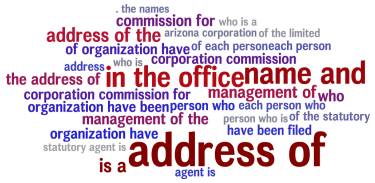


(b) Factor 2 negative

(a) Factor 3 positive

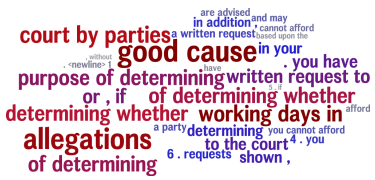


(b) Factor 3 negative

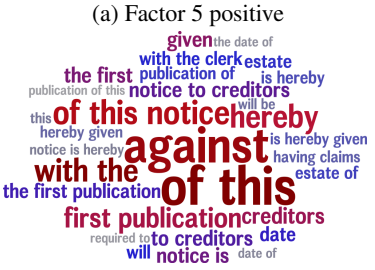


(a) Factor 4 positive

(b) Factor 4 negative



(a) Factor 5 positive



(b) Factor 5 negative

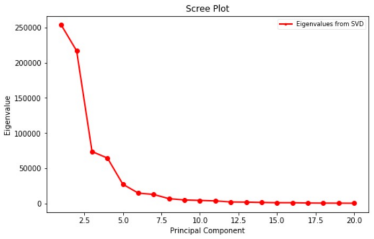


Figure 6: Scree Plot

2.3 Feature Engineering

Given that our dataset is comprised of natural language, we decided to use common techniques to convert it to feature vectors. We decided to try two different approaches because we wanted to compare which worked better for this dataset. Our first approach was using word2vec over the entire notice content and average each word vector to form a single representation. Alternatively, we used doc2vec which aims to capture the semantic meaning of a collection of words.

We chose to compare these two techniques because while doc2vec performs well there are still cases where average word2vec can do better. For example, if the documents are fairly short or the dataset is small then doc2vec may not be the best choice. Due to this, we believe word2vec to be a better choice on our small hand labeled dataset but doc2vec may end up being better after we label the larger set. Although, we do believe word2vec to still result in a good model for this task as many of the notices are short or repetitive texts.

To explore why average word2vec performed better we did look into performing T-SNE clustering. This is a technique that reduces large dimensional vectors into smaller ones that should lead to natural clustering. After we plot our averaged word2vecs using this technique we can see that this does occur around semantically similar documents that also share labels. In this case, the mapping of label number to notice type is as follows: (0, Auction), (1, Death), (2, Industrial), (3, Judicial), (4, Misc), and (5, Personal/Family).

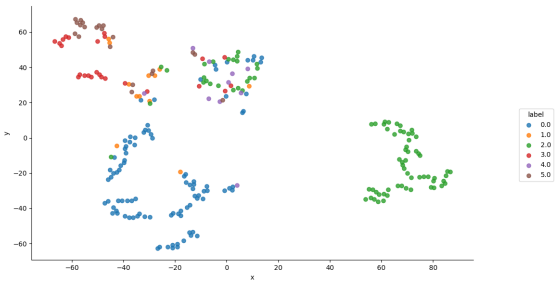


Figure 7: T-SNE Clustering plot for average word2vec vectors per notice

2.4 Baselines and Notice Classifiers

For baselines we used 3 simple models. Among these, we had a monkey classifier that will pick $\frac{1}{k}$ classes are random, a classifier that always picks the most common class, and a single variable lo-

gistic regression/SVM. The single variable classifier used just message length, this was chosen as we felt that some types of notices seemed to be significantly longer than others, so this would be a good simple baseline. We looked at total accuracy, precision, recall, and F1-score as our success metrics.

Our actual classifier models were created using sklearn. This allowed us to easily setup cross validation and splitting the data into train/test. All our models were trained using 5-fold cross validation, since our subset of data is small we took advantage of cross validation to get the most we could out of it. This done for Logistic Regression and Support Vector Machines, for each word2vec and doc2vec features. Results of our preliminary results can be found in table 1 in the appendix.

2.5 Topic Vectors

Generation of the topic vectors revolves around using Twitter data collected across American counties. We have access to a such a dataset that spans the years 2010 - 2015. First, we separated our data into different database tables by year and then began preprocessing them into features using DLATK. These intermediate features will be the input into our final topic classifier. Mainly, we needed to generate 1,2, and 3-grams from all our of our tweets. Next using a weighted lexicon containing categories for 2,000 topics applied over our ngrams we end up with our final feature vector.

The remaining steps is to take these topic vectors and use them to predict the amount of auction notices per county. We choose auction because it covers topics such as bankruptcy, which are of interest to predict due to giving insight into the quality of life of these regions. If the correlation proves large enough, we will explore a time-series application to see if we can predict how many auctions will occur next year. A task that could alert us to which areas may be struggling in the future.

As well as using these topic vectors to predict the occurrences of notice types, we also will explore their ability in models, using topics + notice info, to predict socio-economic status. For these we are interested in poverty and education levels.

3 Next Steps

- Perform unsupervised learning(K-NN) to finish labeling larger dataset
- Create Ridge Regression Model for topic vectors to predict Auction-type notice occurrences
- Do correlation analysis on ridge results and actual occurrences
- Explore time-series application of topic vectors to predict next year's amount of auctions
- Predict socio economic factors like poverty and education (from the dataset provided by the captain) by using the labels and topics as features.

4 Conclusion

We have spent time examining the contents of hundreds of notices to gain expert knowledge on our given task. After seeing the common trend among public notices we have decided to stick to 6 high-level categories as labels for our classification task. To confirm that we were on the right track with these labels, we also performed factor analysis to see the latent trends in the dataset.

Using our hand-labeled notices we built a handful of baseline models and sophisticated ones. The sophisticated models take advantage of two different techniques from natural language processing to learn semantics of words and documents. Currently, we've found that word2vec is outperforming doc2vec but we are unsure if this will continue as we expand our data.

Our next steps will be finishing up our k-nearest neighbors code and then using this technique to label the remaining large dataset (as provided recently by the captain). Once this is done we can move onto working with the topic vectors to see if they correlate with our labels per county. As of now we have made adequate progress in understanding our problem better and have implemented proof of concept models for classification. We feel as though adequate progress has been made.

5 Appendix

Model	Avg Prec	Avg Recall	Avg F1	Accuracy
Random	.12	.12	.10	.12
Common	.06	.16	.09	.40
LR(Msg Len)	.26	.34	.29	.63
SVM(Msg Len)	.49	.28	.30	.53
LR(d2v)	.40	.38	.38	.60
LR(w2v)	.66	.59	.58	.75
SVM(d2v)	.55	.54	.54	.74
SVM(w2v)	.67	.73	.67	.85

Table 1: Model Results