NEW YORK UNIVERSITY

WHAT MATTERS: AGREEMENT BETWEEN CIRCUIT COURT JUDGES

XING CUI

LANYU SHANG

JUNCHAO ZHENG

May 13th, 2016

Contents

1 Introduction

1.1 What Is Federal Circuit Court?

The federal court system has changed a lot since the system created by framers. Today, there are three basic levels of federal courts: U.S. Supreme Court(1 Court); U.S. Courts of Appeals(13 Circuits including 12 Regional and one for the Federal Circuit); and U.S. District Courts. The middle level is also called Federal Circuit Courts. They are not trial courts and do not hear cases first. They hear cases that have been appealed from federal district courts, as well as appeals of decisions of federal agencies.

1.2 Who Are Circuit Court Judges?

Federal circuit judges are nominated by the president and confirmed by the Senate. Each circuit court has between eight to 40 judges as a pool, and at least three of them are selected to go over the case. Each judge has a different background such as their birth state, education, party, war experience, and other personal information.

1.3 Why Choose Circuit Court as the Target

Federal circuit courts are the intermediate appellate courts of the federal court system. It is not like the Supreme court who hears less than 100 cases a year and is also not like District Courts who hear more than millions of cases a year. In addition, due to the fact that cases heard in circuit courts are appealed from lower courts, there are no witnesses and evidence. The circuit court judges review the records from the original trial and also accept written arguments and sometimes hear oral arguments from the lawyers for each side. Currently, there are 179 judges on the Federal Circuit Courts, who are the sophisticated and experienced ones authorized by the Congress. They are located at different sites with different background. In order to ensure the justice, as well as to make sure both sides of the case are treated equally, we aim to find out the hidden connection between each judge for a case due to the personal identities and avoid some conflicts if they exist. Eventually, when judges have decided the results of the case, they issue a formal document called an *opinion*, which is the essence that is extremely important to peoples rights and what we are trying to predict who sits with whom is more likely to disagree.

2 Data Introduction & Preprocessing

We have original data in two files. One is named "100CASELEVEL_Touse.dta" consisting of 134 features and 387898 samples. It contains the case information and some objective judge information. The other one is "100VoteLevel_touse.dta" consisting of 414 features and 1163694 samples . The latter contains three entries for each case. Both data sets involve numerous numbers of missing values, which are hard to decide what to fill with. We start to look through the data carefully since data cleaning is the most vital part of data analysis, and we separate all features from the original data into three major categories: *case general information*, each *judge's biographical information*, and relevant *dummy variables*.

2.1 Data Cleaning

First, we drop some duplicated columns, such as all dummy variables that are converted directly from another feature single column. Some features including *date, month, year, MajOpinionWordCount*, and many others in the dataset, have a majority of these missing values. Others containing fewer missing values are treated as important features. For the features with few available data, we create a dummy variable indicating the data is missing or not. We fill missing values of some features in which a few data is missing by looking into the feature to find out a proper way for filling, like inserting the mean of the column. There still are some features that we could not interpret in the first place, and we treat them as unknowns and fill them with uniqueness like *-1*. In the future process, we would know these are missing values, for illustration, *Due_Process*, *Privacy*, and many others. After filling missing values, we convert comma(",") into underscore("_"), which would avoid troubles when we do MapReduce. We by now have a csv file.

2.2 Target Preparation

A list of target variables is returned from this part, as [1(judge1-judge2), 1(judge1-judge3), 1(judge2-judge3)]. Each circuit court has three records for each judge. There are two features *Judgeconcurring* and *JudgeDissenting* indicating the agreement between each two-judge pair. We assume judges who concur or dissent in this court disagree with the author. If the case has an author, lets say, is judge1. *Judgeconcurring* has a value of judge2 and *JudgeDissenting* has a value of judge3. Then in this case judge1 disagrees with judge2, denoted as 1 and judge1 disagrees with judge3, denoted as 1. However, the target variable will be set to -1 indicating that none of the two judges is the author. In this case, judge2 and judge3 have no opinion on each other, and we denote this situation as -1. Then we get a list of target variable as [1,1,-1] for this case. In other records where the author data is missing, we handle this situation by generating three variables. For example, as the same to the case above, *Judgeconcurring* has a value of judge2 and *JudgeDissenting* has a value of judge3. We generate a list of target variable as [1,1,0] as judge1 disagrees with judge2, judge1 disagrees with judge3, however, judge2 agrees with judge3 because the fact that both of them disagree with judge1 indicating they are agree with each other.

2.3 MapReduce

MapReduce is a programming model and execution framework used for processing and generating a large dataset with a parallel, distributed algorithm on a cluster. As we are handling datasets of high volume, this methodology can help to transform data in a convenient and efficient way.

In the MapReduce process, we have three major steps. First, three entries describing the same case but different judges in $100VoteLevel$ have been extended to three entries containing the case information and two judges biographical information. In judges biographical information, we add some interaction term where 1 indicating two judges have the same background feature and 0 otherwise. The interaction terms we add including *age group, party, gender, race or ethnicity, party affiliation of president, state of residence, school*, etc. Meanwhile, the target has been appended to the end of each entry that indicates whether two judges agree with each other in such cases. The target will be set to -1 indicating that none of the two judges is the author.

For example, the original dataset for each case contains the following information:

Case Information	Judge 1 Information
Case Information	Judge 2 Information
Case Information	Judge 3 Information

After first MapReduce step, the dataset will look like:

Case Information	J1 Info	J2 Info	J1 & J2 Interact	Target
Case Information	J1 Info	J3 Info	J1 & J3 Interact	Target
Case Information	J2 Info	J3 Info	J2 & J3 Interact	Target

In the second step, we extract the dates two judges have been sitting together from the result reduced in step one and output into the following format: names of two judges and a list of dates they sit together.

In the last MapReduce step, we add features whether two judges in the circuit have been sitting together before and their disagreement rate according to their previous records. The new features added are: sit_3mo, sit_6mo, sit_1yr, sit_before, $dissent\_rate$.

Eventually, we reframe the original data to a dataset of 868962 entries and 105 features.

2.4 Dummy Variable Transformation and Normalization

To have the dataset be prepared for our models in scikit-learn methods, we take a further step that is transferring each categorical feature with more than two categories to several binary variables indicating whether the sample is in such categories. Moreover, we also implement the min-max scaling method to rescale features avoiding features to be overweighted or underweighted in the model.

3 Model

3.1 Logistic Regression

Logistic Regression(LR_L1 and LR_L2) is a useful classification algorithm for highly dimensional datasets and it performs very well for binary classification problems. The basic idea for LR is that scales the output of linear regression to range [0,1], which indicates the confidence of our prediction. Under such circumstance, we implement two methods of penalty, one is l1-norm and the other is l2-norm. Also, we can choose a threshold S for classification. Any input cases with a regression result larger than S, we consider it belongs to class 1, otherwise are considered to be class 0. With the probability and the threshold, we can deal with unbalanced data by fitting a ROC curve and adjust to the threshold to get optimal false positive rate(FPR) and true positive rate(TPR).

Moreover, we design an F-test method to find out the importance of each features. An F-test is a statistical test in which the test statistic has an F-distribution under the null hypothesis. The basic idea is: (i) Fit a Logistic Regression on all features and calculate the standard deviation between real value and predicted result, denoted as std_all. (ii) For each feature, fit a Logistic Regression on all other features and recalculate the standard deviation between real-value and the new predicted

result, denoted as std_1. Then we can calculate the F-value as

$$f(1, n-m-1) = \frac{\frac{std_{all}-std_1}{std_{all}}}{n-m-1},$$

where n is the number of observation and m is the number of features(degree of freedom). Features with large F-value should be significant for target prediction while features with a F-value below some thresholds insignificant.

3.2 Random Forest

A Random Forest consists of a collection or ensemble of simple tree predictors, each capable of producing a response when presented with a set of predictor values. The main idea of random forests is using bagged (bootstrap aggregated) decision trees but modifies the tree-growing procedure to reduce the correlation between trees. By implementing this model, we divide into three parts: Random Forest with all features (RF), Random Forest with only case features (RF_case), and Random Forest using only judge features (RF_judge). This is because besides running with all features, we want to see which influences more between case and judges' information, or they are not separable.

3.3 Nave Bayesian

The Nave Bayesian(NB) model is easy to implement and relatively immune to overfitting. Besides, Nave Bayesian is well suited for data with a huge amount of features. Similar to Logistic Regression, we can also have a probability interpretation of our prediction. However, one hypothesis of Nave Bayesian model is that the features are independent of each other. We are not sure about the features in our dataset have some correlation.

We decide to build Gaussian Nave Bayesian model because we have transformed our features to binary, and Nave Bayesian tends to do well with sparsity. We would apply different values of smoothing parameter.

3.4 Confusion Matrix And Model Selection

The confusion matrix, also known as error matrix, is a table layout that allows visualization of the performance of a chosen algorithm. In a confusion matrix, there are four values: true positive(TP), false positive(FP), false negative(FN), true negative(TN). From confusion matrix, we can find out True Positive Rate(TPR) and False Positive Rate(FPR) as:

$$TPR = \frac{TP}{TP+FN} \text{ and } FPR = \frac{FP}{FP+TN}$$

In this case, we have five confusion matrices for each algorithm we used. The best one should have the highest TPR, which would mean that most true positive instances are classified to positive while the FPR is low, at a level of about 30%. In the bar chart below, Random Forest with all features achieves the optimal balance.

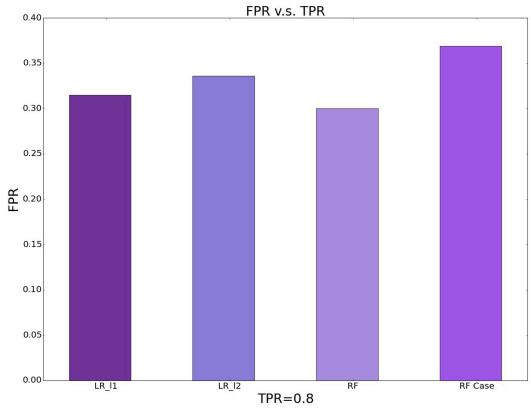


Figure 1: FPR & TPR

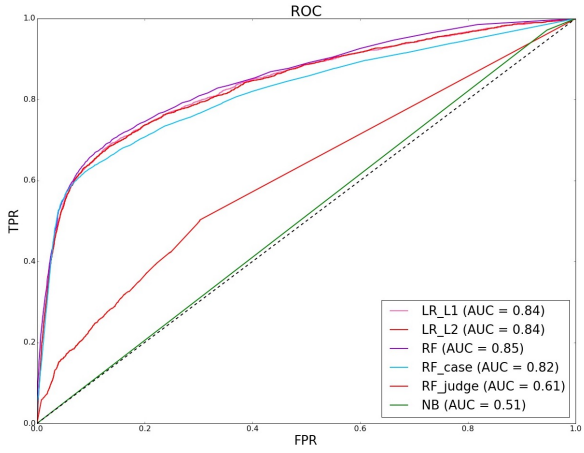


Figure 2: ROC

4 Results

4.1 Results of Models

Figure 2 shows the ROC curves of different models. It is obvious that AUC scores of RF and LR(both L1 and L2) are relatively higher than the rest. Furthermore, in figure 1, we choose expected TPR as 0.8, and we can see RF has the lowest FPR, and LR-L1 and LR-L2 are ranked as the second and third. Therefore, our final models are them.

Prediction of Each Model		
Feature Name	Top: TP, Bottom: FP	Top: FN, Bottom: TN
Logistic Regression(L1)	1862 12961	457 28169
Logistic Regression(L2)	1878 13810	441 27320
Random Forest(all)	1876 12352	443 28778
Random Forest(case)	1870 15176	449 25954
Random Forest(judge)	1166 12450	1153 28680

4.2 Results of Features Importance

Figure 3, 4, and 5 are demonstrating what features are important. In the standard Random Forest, the top 20 features are mostly case features. Judges previous dissent rates, whether they have sit together earlier, and their age similarity appear to be important in the model. Among important features in RF_case and RF_judge, it is obvious that features contributing the most are numerical features. Also, judges previous performances (i.e. their historical dissent rate and whether they have sit before) remain significant in RF_judge.

5 Error Analysis

5.1 Model Analysis

The dataset is highly unbalanced as there are only 5% target with value 1 (disagree) while 95% percent with value 0 (agree). Therefore, we should choose TPR instead of accuracy as our evaluation standard. As the agreement decided by majority of circuit courts, we focus more on what happens when two judges in a court disagree with each other, so we require a high confidence on prediction on positive results.

For each model, we draw a ROC curve on test data and gain a value of AUC. In general, we would choose the model with the highest AUC. Furthermore, we look into the ROC curve to find out (TPR, FPR) pairs. It is balance between choosing optimal TPR or FPR according to the cost

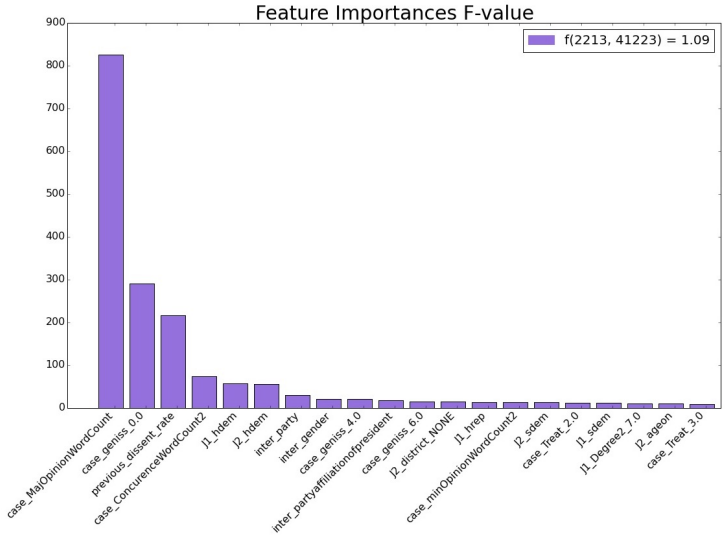


Figure 3: F-test on LR

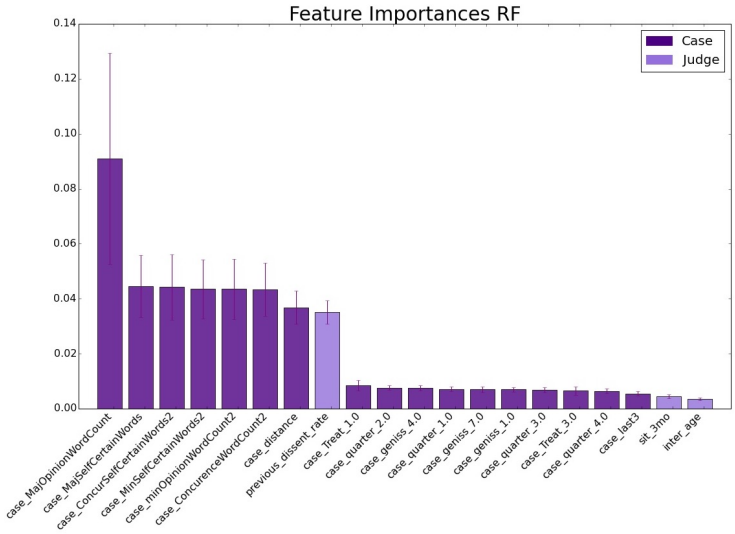
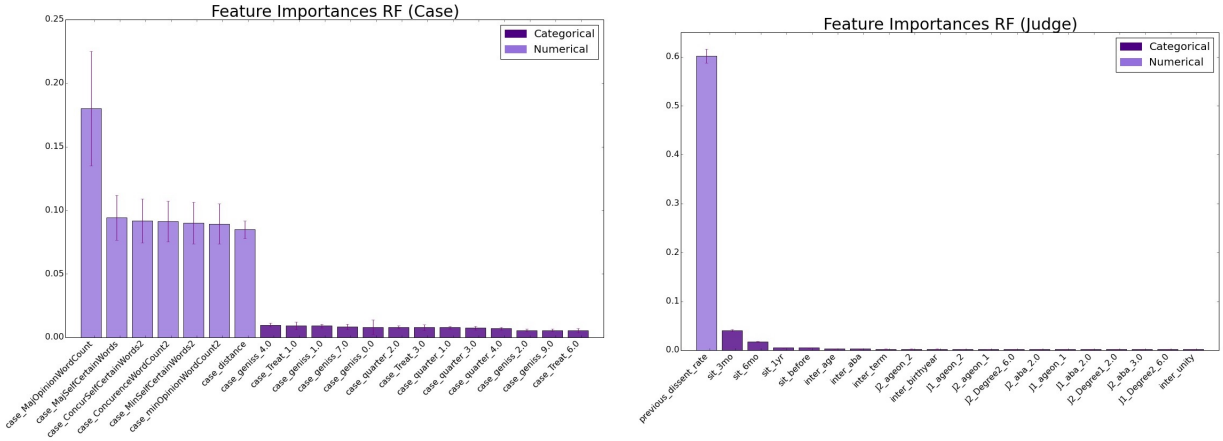


Figure 4: Random Forest



(a) (b)

Figure 5: Random Forest (a: case, b: judge)

respectively. From the analysis above, we want to choose an optimal TPR with a reasonable FPR.

ROC Curves for each model are displayed above. We can also find the confusion matrix for each model in the table of Prediction of Each Model. It can be observed from the ROC curve that Random Forest is slightly better than Logistic Regression. We choose Random Forest as our final model with parameters

$$n\_estimators = 500, max\_features = sqrt.$$

Totally we have 2214 features for prediction, so, the square root is approximately 47, which means in each tree there are 47 random features. We generate 500 decision trees, so each feature will appear in 10 or more trees on average. We would have generated more trees but the running time will expand too much. When we fix TPR to 80%, we can have a FPR = 30.0%.

As the FPR is what we looking for, we want to keep it lower as we expect, so we could have a higher value on TPR. When we fix FPR to 20%, we can have a TPR around 72%, which would be a good prediction.

5.2 Feature Ranking in Random Forest

In Random Forest, we can generate an importance ranking by computing the Gini index for each feature. In order to compare the influence between variables with cases information and variables with judges information, we generate three random forests on all features, cases features only and judges features only.

With further examination, we found that there perhaps exists one problem as most of the important features are numerical features. For example, for the Random Forest running on all features, among the top 100 features, there are 85 features with numerical values while only 15 features with categorical value. Considering the features are well normalized, we may draw a conclusion that numerical features may count more than binary features in model fitting, though we are not sure about the deduction.

5.3 Feature Ranking in Logistic Regression with F-test

In Logistic Regression model, we get a pretty nice prediction accuracy, but we are still interested in what features are relative more important. Therefore, we read a lot and choose between t-test or F-test. Due to the fact that t-test is more suitable for features less than 30, we stick with using F-test in order to find an efficient way to find important features, which is a familiar statistic way to get an F-value for each features on our Logistic Regression model. Features with an F-value larger than a basic threshold is to be considered as a significant one for the prediction. In our case, we have 43436 entries in the 20% test dataset and 2214 features. So, the degree of freedom for the model is 43436 - 2214 = 41222 and the degree of freedom for error is 2214 - 1 = 2213. With one degree of freedom, we have

$$F(2213, 41222) = 0.92,$$

which means a feature with an F value larger than 0.92 would have a probability of less than 0.005 to be a non-significant one. That is to say, we can take the features with a F value larger than 0.92 to be significant features.

Algorithm 1 Our Algorithm for F-test on Logistic Regression

1: **procedure** BY STEPS

2: Fitting a Logistic Regression on the training data with all features, we get a list of probability of our prediction on the test data$[\hat{y_1}, \hat{y_2}, ..., \hat{y_n}]$, which is also our confidence to our prediction. Also we have the list of true label of the test data $[y_1, y_2, ..., y_n]$. We calculate the standard deviation or l2-norm loss function as

$$std_{all} = \sum_{i}^{n} (\hat{y}_i - y_i)^2.$$

3: Then, for each feature, we fit a new Logistic Regression and get a new list of probability of our prediction on the test data $[\tilde{y_1}, \tilde{y_2}, ..., \tilde{y_n}]$. Now we have a new standard deviation

$$std_{feature} = \sum_{i}^{n} (\tilde{y}_i - y_i)^2.$$

4: Then we can have the F value for this feature

$$F(m-1, n-m) = \frac{std_{all} - std_{feature}}{std_{all}(n-m-1)}.$$

After running the loop for calculating the F value for 2,214 features, we find out 206 significant features. Among them here list some most important features which we think make a sense.

case MajOpinionWordCount	Words count in major opinion presents judge's decision.
case_geniss_0	Topics of cases. 0 refers to criminal case.
previous_dissent_rate	The historical disagreement rate for two judges sit together.
case_ConcurenceWordCount	Words count in concurrence presents judge's decision.
inter_party	One power feature we created indicating if the two judges belong to the same party.
inter_gender	One power feature we created indicating if the two judges have the same gender.

6 Conclusion

a. We obtain an optimal Random Forest model and set specificity(FPR) as our evaluation standard. We achieve a FPR = 0.3 and a TPR = 0.8.

b. By applying F-test, we get 206 significantly important features over 2000 features, which include: $case\_MajOpinionWordCount, previous\_dissent_rate, case\_geniss\_0, inter\_party, inter\_gender$.

c. From the result we can roughly excogitate a conclusion: a judge makes a decision mainly based on the case general information and personal information such as whether they belong to the same party or with the same gender. These features will make an influence over their judgement.

d. Moreover, the hypothesis whether the age makes a difference we raised before has been tested. From RF which focus on judges' information, the $inter\_age$ is ranked as the 6^{th} important feature. We separate age in to five ranges start from 30-year-old by ten years until 70-year-old and older than 70-year-old. This shows if judges' age are in the same range, the age might influence their thoughts because they are either relatively inexperienced or are very sophisticated with broad perspective so that they could have multiple angles for one case.

7 Future Work

7.1 Enlarge the Dataset

So far, we are running the model on 20% of the whole dataset due to a maintenance on NYU HPC server. Even though, the outcomes still look fair to us since many of our guess are true. We have faced another problem which is running the model takes really long time due to the enlarged dataset. We have tried deleting unnecessary data while the model is running, and we will try to speed up by running the model in Spark library on the whole dataset.

7.2 Pursuing Higher Accuracy

We can develop a higher accuracy by generating more power features. For example, we can get more detailed features like the historical disagreement ratio on different court topics that could push our research to a whole new level.

8 Reference

1 The Judicial Research Initiative (JuRI)a at the University of South Carolina
http://artsandsciences.sc.edu/poli/juri/sct.html
2 Berdejo, Carlos and Chen, Daniel L. Priming Ideology: Electoral Cycles without Electoral Incentives Among U.S. Judges 2013