Wisdom of Crowds on Refining Survey of Professional Forecasters

Renjun Cheng rc3426@columbia.edu Columbia University

Xinzhi(Cindy) Liang xl3150@columbia.edu Columbia University

Helen Wang hw2811@columbia.edu Columbia University

Thomas Michel tm3206@columbia.edu Columbia University

Abstract

This paper aims at finding the best model to predict RGDP using the Survey of Professional Forecasters (SPF) dataset. After analyzing the performance of multiple models, we identified the contribution weighted model (CWM) as the better performing model.

I. Introduction

In this paper, we're aiming to use three different types of models to predict RGDP (the output is either a point or probability distribution prediction, depending on the nature of model used) from the SPF dataset, and use scoring rules to compare their performance to determine the best model.

The paper is structured in the following way: first, we will talk about our intuition and inspiration behind the project, as well as existing literature. In the methodology chapter, we will provide a detailed description of our dataset, our analysis, and the cleaning process. Then, we will analyze the models we selected (benchmark models, neural network, and contribution and contribution weighted models), and the reason behind the choices. In the last section, we will use evaluators to generate results, interpret the results and compare the performance.

II. Background and Related Work

There is a large amount of existing literature with different models in an attempt to predict the GDP, most of which are linear or nonlinear econometric models that rely on different explaining variables. Our project is different from these literatures in that we rely on purely experts' probability predictions. In other words, we are focusing

on the Wisdom Of Crowds (WOC) in our analysis.

The concept of WOC is to improve judgment or forecast by mathematically combining multiple predictions from groups of individuals. The WOC approach was first identified by Surowiecki in his 2004 paper, and since then there have been many applications.

Our project is mainly inspired by the paper by Budescu and Chen. In their paper, they introduced the contribution weighted model (CWM), the idea of which is to identify the experts in the crowds, in hopes to extract more accurate predictions. They have identified the contribution weighted model to be the better performing model compared to several other benchmark models in two scenarios. For our project, therefore, we want to implement CWM to model our dataset and to compare its performance with other more traditionally and commonly used models adopted by many researchers in forecasting. We will cover more about the technicalities of CWM in our Architect section.

III. Methodology

Dataset

For this project, we applied our knowledge to the Survey of Professional Forecasters (SPF) dataset. The SPF dataset is one of the most common publicly available macroeconomic forecasting tools in the US. It is widely used by business owners, individual investors, and others who do not have the resources to make these predictions themselves. However, research has proven that not only are the forecasts from the SPF historically inaccurate, but also that the forecasters themselves demonstrate significant overconfidence in their predictions. This can result in serious consequences for the groups who may use these forecasts. And part of our goal is to refine the SPF predictions.

Data Cleaning

The Survey of professional forecasters' dataset is very organized and cleaned. The only problem we are facing is: the data we need is from over 6 different excel sheets instead of one. Therefore we used python and pandas to extract all the data we needed: the forecasters' id and the bins with their

probability predictions, and the actual data on what they have predicted.

Exploratory Data Analysis

This part focuses on analyzing the training data from the Survey of Professional Forecasters and obtaining some interesting characteristics or facts of the SPF data. We will dive down to explore the following analysis based on the training data set obtained from the previous data cleaning procedures.

Hit Rate Analysis:

We want to compare the forecasters' peak confidence in terms of the average of the highest probabilities assigned to certain bins to the average hit rate over all forecasts. This can tell us in general what is the relationship between people's confidence in their predictions versus the correct rate of their predictions.

The "One-Sample Test"result picture below shows the results of the t-test we performed by setting x as the maximum probability of each prediction and μ as the average hit rate. In this case, the null hypothesis is that the mean of the hit rate which is equal to 0.3620046. For the purpose of this example, we will set our significance (alpha) level to 0.05. The Sig. column displays the p-value for the test which is less than 2.2e-16 and is definitely less than 0.05. This suggests that the null hypothesis is rejected in favor of the alternative hypothesis, and the peak confidence level of the forecasters is significantly different from the average hit rate.

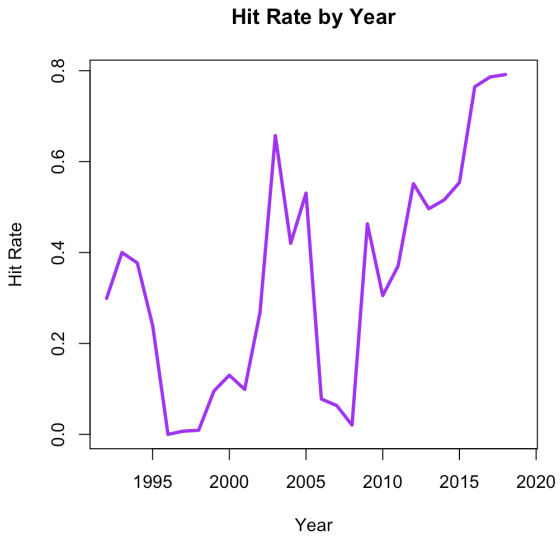
```
        One Sample t-test

data:  training$MAX
t = 57.199, df = 5057, p-value < 2.2e-16
alternative hypothesis: true mean is not equal to 0.3620046
95 percent confidence interval:
 0.5100556 0.5205644
sample estimates:
mean of x
  0.51531
```

The calculated average highest probability or peak confidence is 0.51531 and the average hit rate is 0.36200461. Combining the result obtained from one sample t-test, we can conclude that forecasters are in general more confident about their predictions.

Hit Rate by Year Graph:

Graphing hit rate by year provides a good visualization of the prediction trend over a long period of time. This is helpful in a way that we can use it to identify some sudden changes and dive deeper into the reasons behind them. Therefore, in later analysis, we

will be able to avoid some bias or take some influential social factors into account as we build the models.

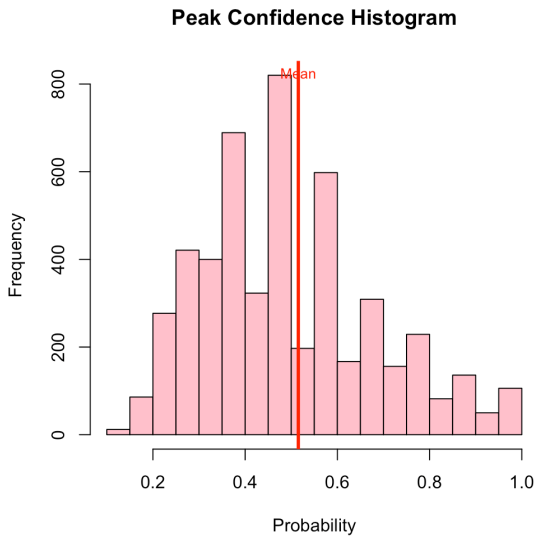


Overall, the hit rate by year shows an increasing pattern. However, there were two significant drops in 1997 and 2008, which can be the results of the Asian Financial Crisis in 1997 and the Great Recession in 2008. Forecasters did not expect the sudden impact on the economic market at the time, and therefore the hit rates were especially low.

Peak Confidence Histogram:

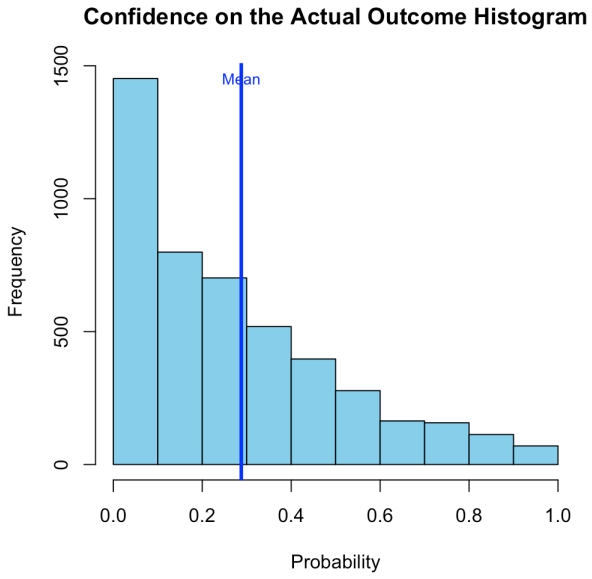
By graphing the histogram of peak confidence and the mean peak confidence, we can see that the distribution is slightly skewed to the right which indicates that the majority of the forecasters tend to be less than 50% confident in their predictions. The portion of forecasters who are to the right side of the mean peak confidence value might be worth studying since their high confidence level might be based on some private information. Therefore, it can be helpful to study each group of forecasters.

Confidence on the Actual Outcome Histogram:



We are also curious about the distribution of forecasters' confidence on the actual outcome. This shows how confident forecasters were on the actual result when they made their predictions. From the graph we can see that the distribution is right skewed indicating that most forecasters were not so confident about the actual outcome given that their mean peak confidence level

is higher than their mean actual confidence level and the more rightly skewed actual confidence histogram.



IV. Architect

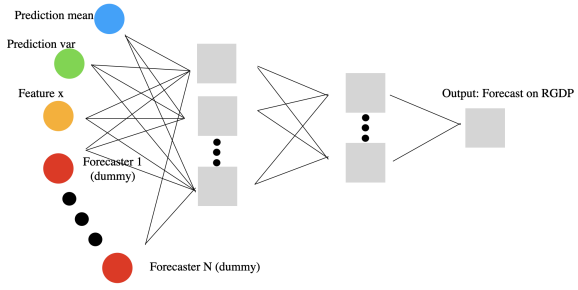
Benchmark Models

For the project, we included several traditionally used models as the benchmark models. The models include unweighted average, linear regression, and random forest. For linear regression and random forest, we used one hot encoding for forecasters to make them become dummy variables to enhance the model performance. The main purpose of using these models is to compare their performance with CWM, which is the major model we are focusing on.

Note here the linear regression model and random forest model can only produce point forecasts, while unweighted produce both point and probability forecasts. We will cover more on this in the performance section.

Neural Network (Multilayer Perceptron)

In our paper, we included a Neural Network model as well. We choose a multilayer perceptron model which is a class of feedforward artificial neural network. We used one hot encoding to make all forecasters as dummy variables with their prediction average and prediction variance as features for our mode. The MLP structures looks like the graph below:



The blue and green dots represent the prediction average and prediction variance, yellow dots are the other potential features but for this model we have nothing there. And the red dots represent forecasters which is most important here. I expect our MLP model to learn forecasters' behavior to generate a 'refined' prediction on RGDP. The output of the model is a point prediction and will be evaluated by accuracy using MSE.

The Contribution Weighted Model (CWM)

Our usage of CWM is inspired and guided by David V. Budescu and Eva Chen's paper *Identifying Expertise to Extract the Wisdom of Crowds*.

The paper developed a more effective aggregation method by building a weighted model based on an individual forecaster's contribution to the crowd. This method overperforms many other prediction methods due to its ability to assign different weights to individual judgments based on their quality so that the experts' judgments will be highly weighted and the poorly performing individuals will be eliminated from the crowd. To quantify the effects of Wisdom of the Crowd (WOC) forecasting applications and take the probability judgment context into account, a proper scoring rule is found to be an appropriate measure of the quality of the aggregate and the forecasters. Budescu and Chen's paper used a quadratic scoring rule, and they mentioned that the proposed approach and procedure can be applied to all other scoring schemes. Here is some detailed procedure regarding the CWM algorithm given our data set background:

First, we let N denote the number of events forecasted, 1992-2020 year 1-4 quarter total of 29 * 4 = 116 events. and Let R_i be the number of categories used in forecasting event i (where i = 1 , … , N). For our events, there were originally 15 bins for each event but we chose to reduce it to two. The reason is, if we use the original 15 bins, each bin has a probability range of 1%. There are many events that the forecast prediction that is very close to the actual but classified as incomplete. Here is an example: A forecaster's prediction has a mean of 4.55% change in RGDP, and the real RGDP change is 4.4% if we use the original bins, bin 6's range is 3.5% to 4.5%. That will lead to the forecaster with a 4.55% prediction which is

outside of bin 6. This is not a single case and that is why I changed to a binary measure. If the forecaster's prediction's mean value is within +/- 0.5%, we will consider he did well and have 1 in the correct bin. Else, he will have a 0 in the correct bin, which makes these 15 bins problems into a binary question. It both increased our result and reduced the algorithm complexity. Next, let m_{ir} be the aggregated mean probability of the crowd for each outcome, r (where r = 1 , … , R_i) of each event (i = 1 , … , N), and let o_{ir} be the binary indicator of the actual outcome for each instance (1 = occur or 0 = not occur). The crowd's score for the event: Si , is calculated by following formula:

$$S_i = a + b \sum_{r=1}^{R_i} (o_{ir} - m_{ir})^2$$

Then, we have the performance of the crowd is aggregated across all events, based on the quadratic score:

$$S = a + b \sum_{i=1}^{N} (\sum_{r=1}^{Ri} (o_{ir} - m_{ir})^2)$$

The quadratic score is unique up to a linear transformation. We used the same constants as Budescu and Chen's paper: a = 100 and b = 50 to yield scores ranging from 0 to 100, where 0 indicates the worst as possible performance and 100 indicates perfect performance. Because after I change our events to binary, the algorithm should be almost identical to their study one.

The contribution of each forecaster, C_f (where f = 1 , … , F), is calculated as the average difference between the crowd's scores based on the mean forecasts (m_r), with and without the j_{th} forecaster, across all N_j events answered by the forecaster. We allow for the possibility that not all forecasters forecast all the events by setting Nj <= N. The formula to generate contribution as follow:

$$C_f = \sum_{i=1}^{Nj} (S_i - S_i^{-f}) / N_f$$

For more detail on our contribution to each forecaster, we have a detailed analysis in the result part to further discuss.

After we obtain our C_f , we will get our **weighted aggregated model**, **CWM**, which employs only forecastors with positive C_f in forecasting new events. These C_f are normalized to generate weights such that the aggregated prediction of the crowd is the

weighted mean of the positive contributors' probabilities.

We also created a **contribution model** that is equally weighted for positive contributors.

V. Result

For all models, we did an 80/20 split. Using first 80% of our data training models and 20% of our data validation the model performance.

Evaluation

For each model, we yielded a point forecast and a probability distribution forecast of GDP. While for a point forecast, a loss function would be sufficient to provide evaluation, for a probability forecast scoring rules would be more reasonable matrices.

Therefore, for the sake of this project, we are implementing two different ways of evaluation: MSE and quadratic scores

Accuracy MSE

For the loss function to evaluate point forecast, we chose Mean Square Error (MSE). That is:

$$\frac{1}{N}\sum_{i=1}^{N}(y_i - \widehat{y}_i)^2$$

where y is the actual GDP, $\widehat{y}$ is the predicted value, and N denotes the total number of original values in the data set. We chose MSE because it is considered as one of the popular model evaluation metrics and can intuitively measure model effectiveness.

We used MSE to evaluate point predictions of all models, and the result is presented in the following table. We will analyze the results in the performance sections.

Models	MSE
CWM	1.147520
Contribution Model	1.239755
Unweighted Model	1.398145
MLP	2.716616
Linear_Regression	2.773127
Random_Forest	3.977798

Scoring Rules

In this paper, we are using two scoring rules: quadratic and logarithm.

The quadratic score is this:

$$2p_k - \sum_{k=1}^{K} p_{k'}^2$$

where p_k is the probability associated with the bin inside of which the actual GDP lands in, and the summation part is the sum of all other unrealized probabilities squared.

We choose to use this specific scoring rule because we have a discrete distribution, and this rule is appropriate in this situation. Since the quadratic scoring rule was used in Budescu and Chen's paper, for the sake of consistency we are especially interested in the quadratic scores.

We applied the quadratic scoring rule to the unweighted model, CWM, and contribution models only, given that the other models cannot produce probability forecasts. See the following table for respective scores.

Models	Quadratic Scoring
CWM	0.264905
Contribution Model	0.254293
Unweighted Model	0.243337

Benchmark Performance

Among our benchmark models, unweighted average is the best performing model by all three evaluators. This is not surprising, since it is consistent with the Budescu and Chen paper's result.

What must be noted is that Random Forest, a widely adopted machine learning model, has very poor performance compared to others. We believe this is due to the fact that while Random Forest can fit very well with many feature inputs, here the dataset is untraditional, and the features we're using are the prediction probabilities themselves. Therefore each event should be clustered very closely and it would be difficult to find a pattern and yield accurate predictions.

This result is really indicating that these traditionally used weighting methods cannot really efficiently apply to the wisdom of crowds type of datasets.

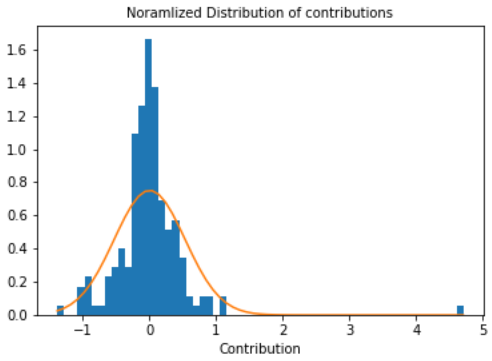
MLP Performance

The MLP performance is quite disappointing. The accuracy evaluated by MSE is lower than the unweighted model. However, it is slightly better than linear regression and random forest.

CWM Performance

After running the CWM model we found quite interesting contributions. Indeed, the average of the contributions was around 0, the maximum up to 4.7 (forecaster 461 with 100% accuracy), the minimum just about -1.37 (forecaster 517 with accuracy 0%), and the sample standard deviation is about 0.53. The contributions seem quite "normally distributed" around 0. Doing so, the model was able to get the best out of the crowd putting more weight on good forecasters and less on poor ones.

The contributions look like the following :



On this graph we observe the distribution of the contributions as well as the cdf of a normal distribution with parameters μ and σ computed from the data set.

Observing the data we can see that the best forecaster, according to the model, really differs from other forecasts in terms of contribution. Indeed after analyzing the data collected we noticed that this forecaster (ID 461) has 100% of accuracy on his/her forecast. The reason for this exceptional result is simple, forecaster 461 did only one bet. We could interpret this singularity as a limit of the model.

We therefore thought about fixing a minimum number of forecasts per forecaster for the model to take them into account, but the data set being limited, unlike David V. Budescu and Eva Chen's paper we weren't able to filter the forecasters that way. Nevertheless we tried to exclude this particular forecaster to see any improvement in the model's accuracy. Yet eliminating the outlier produced worse results. The new MSE became 1.16418 when it was 1.147520 (the model would still be the best, but slightly less accurate). We therefore kept the outlier in our model.

VI. Conclusion

To conclude on this work, using two evaluating methods on the SPF adata set, we were able to compare different models of prediction on GDP forecasts. The major

model we focused on is the contribution weighted model. The CWM model, as expected from David V. Budescu and Eva Chen‘s paper, was the most accurate model applied to this dataset.

CWM outperformed the contribution model, MLP, and the benchmark models under different scoring rules (MSE and quadratic scoring). The biggest surprise came from the inaccuracy of the random forest model that ranked last with the MSE scoring rules.

VII. Future Work

The result of our CWM is decent. However, during our data processing and modeling, we find we could improve in the following field:

The first one is that we have many forecasters with less than two predictions, the best idea could be to remove those. However, we have limited data points. Over the past 30 years, there have been 172 forecasters making predictions. And only 82 forecasters make predictions with a positive contribution score. If we remove the forecasters with a small number of predictions, we will have too few data points. Therefore if we can keep working on this, we will test the model on other robust data, for example, unemployment rate prediction could be better.

The second one is the hyperparameter tuning for MLP and random forest. These models are not the major models for this paper, we only spend a little time on the hyperparameter tuning. If we have more time, we could perform fine-tuning on hyperparameters that could potentially increase model performance.

VIII. References

David V. Budescu, Eva Chen (2015) Identifying Expertise to Extract the Wisdom of Crowds. Management Science
61(2):267-280.
https://doi.org/10.1287/mnsc.2014.1909

Survey of professional forecasters. Federal Reserve Bank of Philadelphia. (n.d.). Retrieved December 24, 2021, from *https://www.philadelphiafed.org/surveys-and-data/real-time-data-research/survey-of-professional-forecasters*

Jia, Y. (n.d.). IEOR4630 Prediction Markets Probabilistic forecasts and proper scoring rules. *https://courseworks2.columbia.edu/*

courses/136605/files/folder/Slides?preview=11915608

“Multilayer Perceptron.” Multilayer Perceptron - an Overview | ScienceDirect Topics,*https://www.sciencedirect.com/topics/computer-science/multilayer-perceptron.*