Learning and Inference Through Graphical Models on Economic Data with a Particular Focus on Recession Indicators

Panth Patel
Laboratory for Computational Sensing and Robotics
Johns Hopkins University
Baltimore, MD 21218
panthpatel@jhu.edu

Abstract

This paper explores the use of various graphical model packages such as PC and Glasso (in R) and pgympy and pomegranate (in Python) on performing learning and inference tasks on economic data obtained from the Federal Reserve Bank of St. Louis. Structural learning is done primarily in R while Python is used to perform inference queries such as the probabilities of various features given others.

1 Introduction

This project focuses on the use of data obtained from the Federal Reserve Bank of St. Louis. A number of distinct data sets were obtained focusing on the period from 1960 through the present (2019). These data sets were then preprocessed as applicable to align the data available, seasonally adjust data, and convert level data to Year-Over-Yead Change (YOY). Once the data was preprocessed, it was ready to be used for structure and parameter learning of the underlying graph as well as for inference. Structure learning was done primarily in R using the PC and Glasso packages to determine the graph structure of the economic indicators. A Bayesian model from the Python package Pomegranate was used to perform inference on the dataset. Alternative methods for learning and inference are also discussed and demonstrated in the provided code repository.

2 Data

The Federal Reserve Bank of St. Louis has a variety of data on its website that covers a broad range of economic markers such as the unemployment rate, gross domestic income of employees, payroll numbers for different industries, and so forth. Through research on economic markers related to recessions and the researcher's intuition, a number of datasets as listed below were selected for use in this project. These datasets did not all cover the same time periods, some that had seasonality to them were not seasonally adjusted, and other factors required a significant level of processing to be done to the data before any learning or inference could be performed. These methods are discussed in section 2.2.

2.1 Dataset

The items below begin with their series ID on the FRED website and are followed by a description of their attributes (frequency, units, time period, seasonal adjustment, and so forth).

- 1. UNRATE: Civilian Unemployment Rate, Monthly, Percent, Seasonally Adjusted, 1948-Present, Node: Unemp_Gap

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

- 2. NROU: Natural Rate of Unemployment (Long-Term), Quarter, Percent, Not Adj, Q1 1949-Q4 2029 (forecasted), Node: Unemp_Gap
- 3. USREC: NBER based Recession Indicators, Monthly, 1/0, Dec 1854-Apr 2019, Node: USREC
- 4. AHEMAN: Average Hourly Earnings of Production and Nonsupervisory Employees: Manufacturing, Month, $/hr and YOY Change, Not Adj, Jan 1939-Apr 2019, Node: Avg-HEar
- 5. FEDFUNDS: Effective Federal Funds Rate, Percent, Monthly, Not Seasonally Adjusted, July 1954-Apr 2019, Node: FEDFUNDS
- 6. PCECC96 Real Personal Consumption Expenditures, level ($), Quarterly, Seasonally Adjusted Annual Rate, Q1 1947-2019, Node: Consumer_Expend
- 7. ICNSA: Initial Claims for Unemployment Insurance, Number, Not Adjusted, Weekly 1967-2019, Node: Unemp_Ins_Claims
- 8. M08297USM548NNBR: Initial Claims for Unemployment Insurance, 1000s of claims, Not Adj, Month, Aug1945-Mar 1969, Node: Unemp_Ins_Claims
- 9. PERMIT: New Private Housing Units Authorized by Building Permits, Monthly, Seasonally Adjusted Annual Rate, Jan 1960-Mar 2019, Node: PERMIT
- 10. S&P 500, Monthly, Yahoo, Jan 1950-May 2019, Node: SP_Close
- 11. UEMPMEAN: Average (Mean) Duration of Unemployment, Monthly, Seasonally Adjusted, Jan 1948-Apr 2019, lags (Its value is inverted so that it gives a higher reading during expansion and vice versa.), Node: Unemp_Gap
- 12. PAYEMS, All Employees: Total Nonfarm Payrolls, Monthly, Seasonally Adjusted, Jan 1939-Apr 2019, Economy's current state, shows net hiring, Node: PAYEMS
- 13. A4102C1Q027SBEA: Gross domestic income: Compensation of employees, paid: Wages and Salaries, Quarterly, Seasonally Adjusted, YOY %, Q1 1947-Q1 2019, Node: Gross_Dom_Income

Some of these items were believed to have a lead (ex. PERMIT, S&P 500, ICSNA) or lag (ex. PAYEMS, UEMPMEAN) relationship with recessions and thus were included in the hopes that they may help determine if a recession may be approaching or to determine if a recession is currently ongoing based on the tangentially related datasets. The USREC data series was used as the "true" label for when a recession was or was not occuring.

2.2 Pre-Processing Data

1. **Align Dates of Data:**

   First, the dates that the each data series overed had to be aligned. This was done by checking the start and end of each data set and determining that all of the data covered the period from January 1960 through the present (April 2019). As a result, the data was cleaned to remove any data outside of these periods.

2. **Convert Quarterly/Weekly Data to Monthly:**

   This step was completed in R using spline interpolation on the PCECC96, A4102C1Q027SBEA, and NROU datasets. All of these datasets were in terms of quarterly data and needed to be interpoled to obtain monthly data. The ICSNA dataset had a weekly frequency so for that, the dataset was plotted on FRED's website, the frequency of the graph was selected to show monthly values, and then the corresponding dataset was downloaded and added to the complete dataset.

3. **Seasonally Adjust Data:**

   There were a number of data series that were not seasonally adjusted. As a result, these series were plotted to determine if any seasonality existed in the series. The only non-seasonally adjusted dataset that showed seasonality were the M08297USM548NNBR and ICSNA datasets which covered the same information (Initial Unemployment Claims)

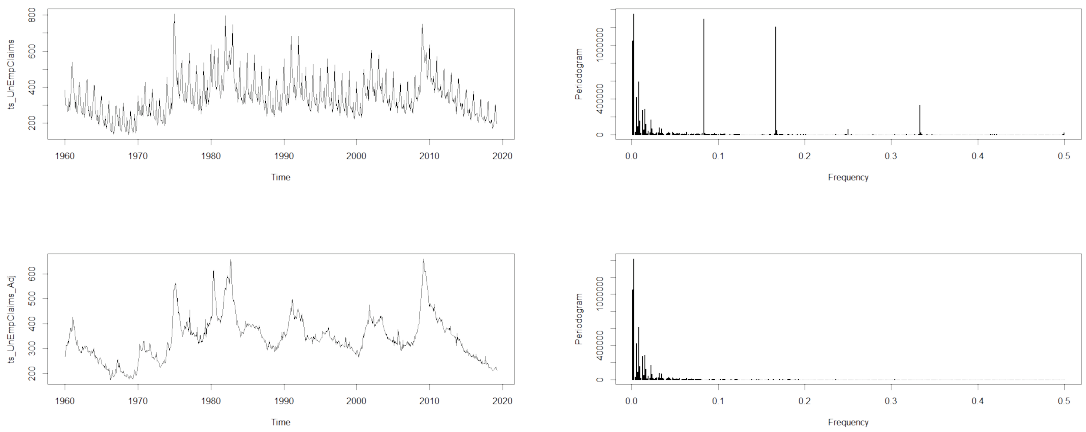

Figure 1: Figure: Top: Seasonally Unadjusted Data (left) and Periodogram (Right), Bottom: Seasonally Adjusted Data (left) and Periodogram (Right)

for two time periods. These two datasets were combined to form one complete dataset from 1960 through the present for this information. In order to address the seasonality, the "seas" package in R was used. As shown below, it is apparent that there is some seasonality to the data, particularly that the data is periodic over the course of the year as shown by the periodogram for the unadjusted data. After using the predict function from the seas package, the seasonality appeared to be adjusted as shown in the subsequent graphs.

4. **Convert Data to Year-Over-Year % Change:**

   The last step of preprocessing the data was to convert some of the data to Year-Over-Year Percent Change so that the model may interpret changes in the data instead of just the raw values. For example, take the S&P500 which, for a long time, has been continuously increasing. During periods of economic downturns however, the value of the index doesn't go down to the same level, but there is typically a negative change in the value from its immediate predecessors. Going through the data, it was determined that YOY changes would be more appropriate measures to learn for the S&P500, PCECC96, PERMIT, PAYEMS, M08297USM548NNBR, and A4102C1Q027SBEA_PC1 (the last of which was already in YOY Change from FRED). In the subsequent analysis, any data that can been converted to YOY Change will have a "_PC1" appended to the end of its name.

3 Learning

The primary focus for learning this dataset was to perform structural learning to determine how the features of the dataset related to each other. For this task a number of methods were attempted such as rsmax2 and gs from the R bnlearn package, which are a hybrid structure learning algorithm (using Max-Min Hill Climbing) and a Constraint-based structure learning algorithm (based on Grow-Shrink), respectively. The code for learning those structures may be found in the Structure_Inference.R script on this project's repository. Additionally, PC and Glasso were performed on the dataset and the results are shown below.

First-level headings should be in 12-point type.

3.1 PC

The PC algorithm provided a graph structure as shown below. The shown structure uses an alpha of 0.005, which was obtained following trials with alphas between [0.001, 0.5] and was determined to preserve the most edges between the results of the various alphas. This aligns with papers that indicate the Average Structural Hamming Distance (SHD) appears to be lower for alphas between

0.005 and 0.01 (http://www.jmlr.org/papers/volume8/kalisch07a/kalisch07a.pdf). While the structure does not reflect the relationships expected as discussed above (leading and lagging variables as related to recession), the structure appears to indicate an alternate result that should have also been expected: these variables may not have a causal relationship with recessions but the recession node (US_REC) may have a causal relationship with them (such as the Unemployment Insurance Claims increasing when a recession hits).

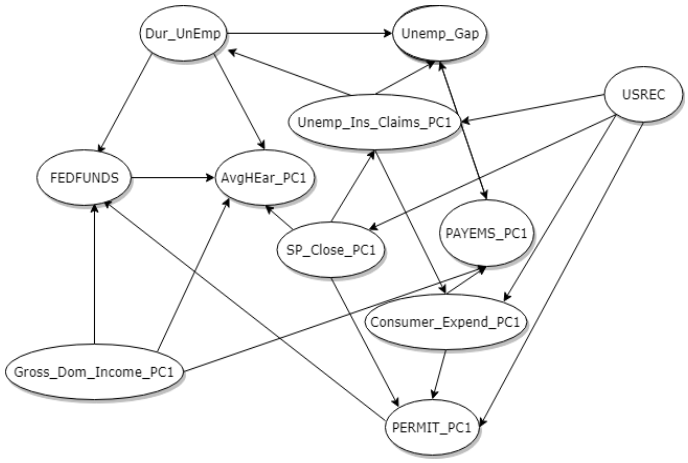

Figure 2: Figure: Network Obtained from PC Algorithm

3.2 Glasso

Prior to running Glasso, each data series was plotted on a histogram to determine if the data was gaussian. The results of that analysis is shown below. Although some of the data looks quite Gaussian (Ex. PERMIT_PC1), others are not quite so clear. As a result, the nonparanormal (npn) transform was performed on that data prior to running Glasso to ensure Gaussian distributions as shown in the resulting histograms.

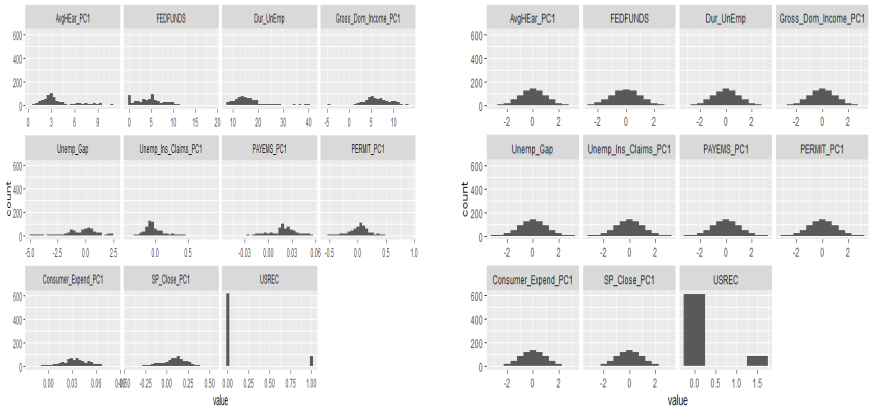

Figure 3: Figure: left: Histograms of Data, right: Histograms of Data Post-NPN Transform

When running the Glasso algorithm, the BIC score was computed in order to determine the best rho parameter to use for determining the structure of the graph. As shown in the chart below, the best rho was determined to be approximately 0.118 and that is the value that was used to construct the structure of the graph shown below.

The results from running Glasso do not entirely agree with the results of PC algorithm or the structures learned using the bnlearn package as shown in the figure below. It is suspected that this result may be due to the assumption of a Gaussian distribution on some of the data or due to the nonparanormal transform run on the data that appeared to not be Gaussian.

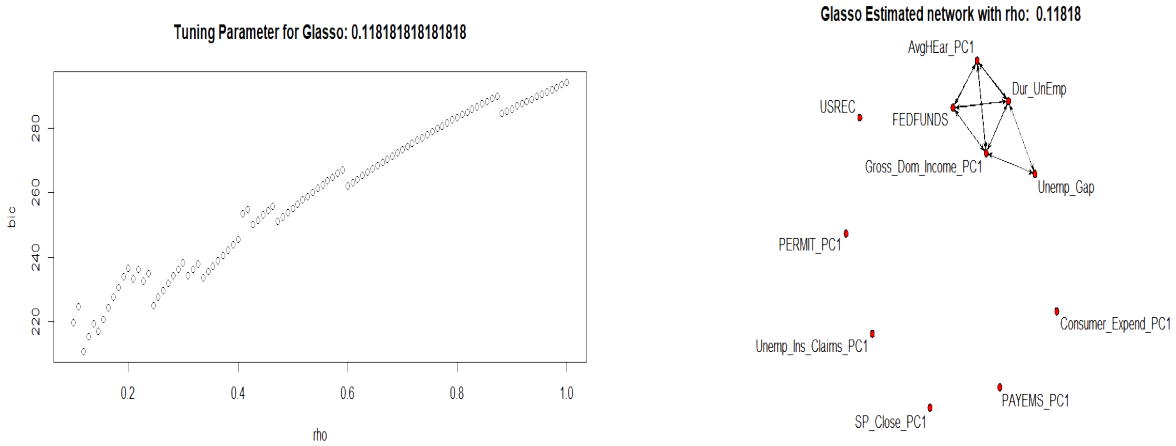

Figure 4: Figure: left: BIC Score Tuning, right: Glasso Graph

4 Inference

Inference could be performed in several different ways for this dataset. First, the dataset was used to learn a Bayesian model through the pgmpy package. The plan was to use belief propogation map_queries and maximum likelihood estimators provided by the package to perform inference tasks on the data. Possible questions would be what is the probability of a recession given that the YOY change in the S&P500 is -0.5% and the unemployment gap has grown by 5.4%. Fitting the data set to the bayesian model appeared to be intractable however unless only two features were used as the model appeared to continue training for several hours without completing. As a result, the pomegranate package was used to create a bayesian network on which inference could be performed using maximum likelihood estimation. As shown in the image below, queries could be constructed on incomplete data and the model would determine (predict) the value or, using the predict_proba function, could determine the probabilities of the data.

4.1 Belief propagation

Using belief propagation and mle, the model was able to perform inference to answer queries as those shown in the example below. Additionally, using the predict_proba function shown in the code, the probabilities of each graph given the evidence can be obtained.

```
(Pdb) answer_array = model.predict(query_array)
(Pdb) print(query_array)
[[1.45 13.7 1.118 None -0.011 -0.113 0.009 0.111 None]
 [2.0 17.0 3.822 -0.121 0.0 0.166 0.026 0.195 None]
 [2.33 15.8 6.694 -0.204 0.021 0.221 0.05 0.231 None]]
(Pdb) print(answer_array)
[array([1.45, 13.7, 1.118, 0.396, -0.011, -0.113, 0.009, 0.111, 1.0],
      dtype=object), array([2.0, 17.0, 3.822, -0.121, 0.0, 0.166, 0.026, 0.195, 0.0],
      dtype=object), array([2.33, 15.8, 6.694, -0.204, 0.021, 0.221, 0.05, 0.231, 0.0],
      dtype=object)]
(Pdb) print(np.asarray(answer_array))
[[1.45 13.7 1.118 0.396 -0.011 -0.113 0.009 0.111 1.0]
 [2.0 17.0 3.822 -0.121 0.0 0.166 0.026 0.195 0.0]
 [2.33 15.8 6.694 -0.204 0.021 0.221 0.05 0.231 0.0]]
(Pdb) print(query_array)
[[1.45 13.7 1.118 None -0.011 -0.113 0.009 0.111 None]
 [2.0 17.0 3.822 -0.121 0.0 0.166 0.026 0.195 None]
 [2.33 15.8 6.694 -0.204 0.021 0.221 0.05 0.231 None]]
(Pdb) print(np.asarray(answer_array))
[[1.45 13.7 1.118 0.396 -0.011 -0.113 0.009 0.111 1.0]
 [2.0 17.0 3.822 -0.121 0.0 0.166 0.026 0.195 0.0]
 [2.33 15.8 6.694 -0.204 0.021 0.221 0.05 0.231 0.0]]
(Pdb)
```

Figure 5: Query on Bayesian Model, Query-Array has 'None' for variables of interest (last is USREC), Answer-Array provides the predicted value for those

Due to the difficulty of training the models to perform inference, dimensionality reduction methods were considered. Using PCA, it was found that the dimensions of the data could be reduced down to four components. For the purposes of determining how the specific features we have relate to each other, dimensionality reduction is not particularly usable because we lose the actual features and can no longer perform queries such as what is the predicted change in the S&P500 given that the Federal Funds rate has changed by a given amount. However, for simply classification tasks, such as determining whether or not we are currently in a recession or may be headed towards a recession given the current economic indicators in the dataset (a binary task), we may be able to use PCA or another dimensionality reduction method such as manifold reduction.

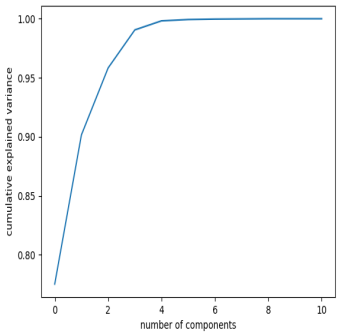

Figure 6: PCA on dataset, Elbow at about 4 Components indicating that data may be reduced to 4 dimensions

4.2 HMM classification using likelihood

From the previous information, it was determined that another possible inference task would be to test if this model is capable of determining if a recession is occuring or not (or, subsequently, if a recession is imminent). For this, two HMMs were trained, one on recession data and the other on non-recession data, and then were used to calculate the likelihood, given some segment of data for testing, of a recession. For this task, the model performed quite well and consistently yielded significantly higher likelihoods for the correct classifier (i.e. it gave a higher likelihood on the non-recession model than the recession model for data from the past 12 months, but the opposite for data such as from the period of the Great Recession). Example on data from past 12 months shown below:

```
Current Recession likelihood: -152.19998714342356
Current Nonrecession likelihood: 3.878702231074372
```

Figure 7: HMM for No-Recession has greater likelihood compared to HMM for Recession for data from past 12 months

5 Conclusion

In this paper, the PC and Glasso packages were used to perform structure learning on the economic dataset obtained from the Federal Reserve Bank of St. Louis. Additionally, the hmmlearn and pomegranate packages in python were used to perform inference in the form of classification and prediction queries on the dataset. There were several challenges with the tasks outlined above, including the difficulty of using graphical model packages, especially in python, as easily as other machine learning packages such as scikit-learn. Furthermore, training in python appeared to take significantly longer than in R, possibly due to the packages in python not being quite as powerful as those in R. Additional tasks that could be done in the future include creating a graphical model based on an individual's understand of economical data and performing inference on those graphs. The graph that the author of this paper would consider is shown below.

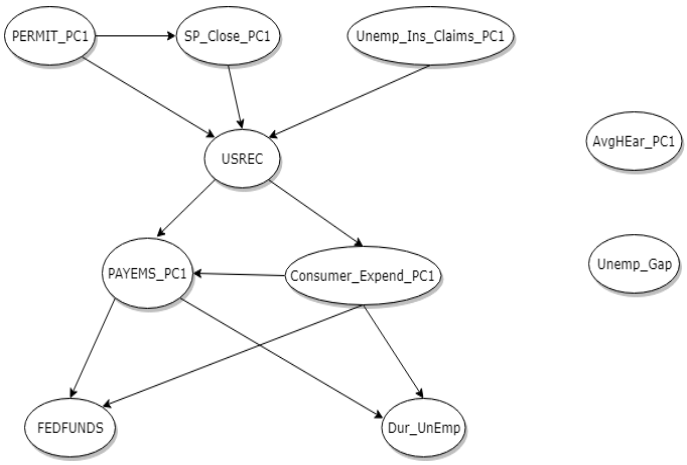

Figure 8: Graphical Model based onf lead/lag variables for recession, which could be used for subsequent inference tasks