Yewno-Quantitative Analyst Question 1

Leo Guoyuan Liu

May 21, 2018

Question 1 Use freely available data from the web to predict/explain macroeconomic indicators. Financial/Economic/Fundamentals data are not allowed.

In this exercise, I investigate the potential of using web search data to predict an important macroeconomic indictor–unemployment rate. It is well known that people's web searches behavior reveal their needs. Now days, a large proportion of job-related information gathering is through internet. To access the job information in the internet, people commonly use search engines to locate the website. It is easy to find the most frequent words job seekers use to search. The hypothesis is that the attention trend of those key words are correlated to the unemployment rate.

Read data

The unemployment data are downloaded from Federal Reserve Bank of St. Louis https://fred.stlouisfed.org/. Search engines keywords was extracted from WordTracker's Top 500 keywrod, as Michael Ettrege 2005 did. I pick five words– recruitment, resume, employment, monster.com, job list. The weekly interest over times of these key words are collected from google trends. The interest numbers represent search interest relative to the highest point on the chart for the given region and time. A value of 100 is the peak popularity for the term. A value of 50 means that the term is half as popular.

The information obtained from google trend will be more useful when it is available ahead of official report. I try variables with lead times varying from one to four weeks. The I aggregate the data into monthly data. Then I add a two months moving average for each entry.

```
data<-read_csv('google_trend.csv')%>%
  mutate(Date=as.Date(Week,"%m/%d/%Y"))%>%
  select(Date,2:6 )

lead1<-data%>%mutate_at(vars(-Date),funs(lag(.,1)))
lead2<-data%>%mutate_at(vars(-Date),funs(lag(.,2)))
lead3<-data%>%mutate_at(vars(-Date),funs(lag(.,3)))
lead4<-data%>%mutate_at(vars(-Date),funs(lag(.,4)))

dat<-lead1%>%inner_join(lead2,by="Date", suffix=c("_ld1","_ld2"))%>%
      inner_join(lead3, by="Date")%>%
      inner_join(lead4, by="Date",suffix=c("_ld3","_ld4"))



dat<-dat%>% mutate(Date= floor_date(Date, "month"))%>%
  group_by(Date)%>%summarise_all(mean)%>%ungroup
sma2<-dat[-1,]%>%mutate_at(vars(-Date),funs(rollmean(.,2,align = "right",fill=NA)))
dat<-dat%>%inner_join(sma2,by="Date",suffix=c("","_ma2"))






ui<-read_csv('unemployment.csv')%>%
```

```
    mutate(Date=paste0(Year,sub("M","-", Period) ,"-01")%>%as.Date)%>%
    select(Date,unemployment=Value)
dat<-dat%>%inner_join(ui,by="Date" )%>%to_xts
```

Model selection

I use a Sequential Backward Reduction to select the independent variables by minimizing the AIC. After rounds of selections. There are still many variables left. Then I manually delete insignificant variables, finally I obtain a model with two variables resume_ld3_ma2 + monster_ld4_ma2.

Model performance

It has a good r-squared 0.96. The plot shows the fitted unemployment runs closely with the actual one. However, when plot the out sample test. The fit is poor. It means even with few variables, the model is still over-fitted. To get a Better model, more advanced algorithm need to be searched. I suggest models such as ARIMAX, random forest and Neural network (RNN).

```
stepAIC(lm(unemployment~., dat["/2017-6"]),trace=0)
```

```
Call:
lm(formula = unemployment ~ recruitment_ld1 + monster_ld1 + recruitment_ld2 +
    resume_ld3 + employment_ld3 + resume_ld4 + employment_ld4 +
    resume_ld1_ma2 + recruitment_ld2_ma2 + employment_ld2_ma2 +
    joblist_ld2_ma2 + recruitment_ld3_ma2 + resume_ld3_ma2 +
    employment_ld3_ma2 + monster_ld3_ma2 + joblist_ld3_ma2 +
    employment_ld4_ma2 + monster_ld4_ma2, data = dat["/2017-6"])

Coefficients:
        (Intercept)      recruitment_ld1          monster_ld1
            6.14048             -0.08148              0.03382
    recruitment_ld2           resume_ld3       employment_ld3
            0.10532             -0.11039              0.08787
         resume_ld4       employment_ld4       resume_ld1_ma2
            0.08862             -0.11923              0.07564
recruitment_ld2_ma2   employment_ld2_ma2      joblist_ld2_ma2
           -0.15920             -0.13106              0.12581
recruitment_ld3_ma2       resume_ld3_ma2   employment_ld3_ma2
            0.11220             -0.07981              0.40971
    monster_ld3_ma2      joblist_ld3_ma2   employment_ld4_ma2
           -0.19593             -0.14839             -0.21443
    monster_ld4_ma2
            0.19326
```

```
fit<-lm(formula = unemployment ~  resume_ld3_ma2 + monster_ld4_ma2,
        data = dat["/2017-6"])

summary(fit)
```

```
Call:
lm(formula = unemployment ~ resume_ld3_ma2 + monster_ld4_ma2,
    data = dat["/2017-6"])
```

```
Residuals:
     Min       1Q   Median       3Q      Max 
-0.35684 -0.11467 -0.00538  0.14920  0.27238 

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)    
(Intercept)      5.102262   0.284263  17.949  < 2e-16 ***
resume_ld3_ma2  -0.013399   0.004254  -3.149  0.00294 ** 
monster_ld4_ma2  0.037715   0.001532  24.618  < 2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.1692 on 44 degrees of freedom
  (1 observation deleted due to missingness)
Multiple R-squared:  0.9602,    Adjusted R-squared:  0.9584 
F-statistic: 530.7 on 2 and 44 DF,  p-value: < 2.2e-16
```

```r
predict(fit, dat["2017-7/"])
```

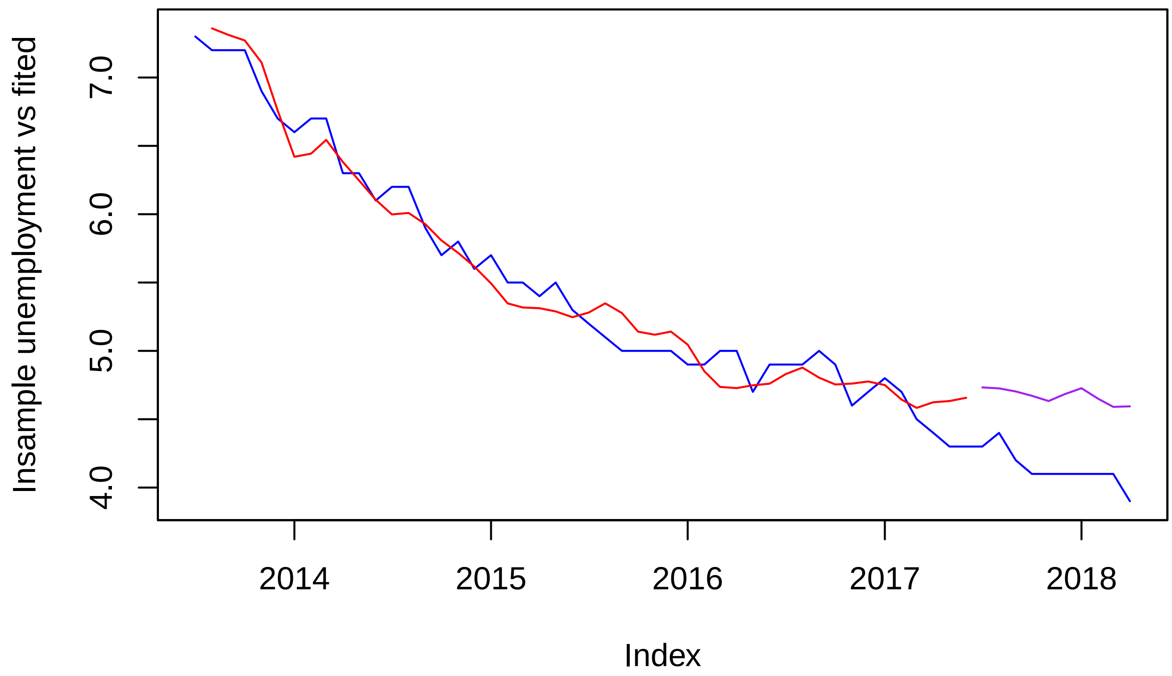
```
2017-07-01 2017-08-01 2017-09-01 2017-10-01 2017-11-01 2017-12-01 
  4.732551   4.725852   4.703111   4.671426   4.633090   4.683557 
2018-01-01 2018-02-01 2018-03-01 2018-04-01 
  4.726978   4.649578   4.590464   4.594172 
```

```r
dates=as.Date(names(fit$fitted.values),"%Y-%m-%d")

y<-dat[,"unemployment"]
y_fit=xts(fit$fitted.values, order.by=dates)

y1_fit<-xts(predict(fit, dat["2017-7/"]),order.by = index(dat["2017-7/"]))

plot(as.zoo(merge(y,y_fit,y1_fit)), ylab="Insample unemployment vs fited",
    col=c("blue","red","purple"),screens=1)
```




```r
print("rmse of in sample test")
```

```
[1] "rmse of in sample test"
```

```r
rmse(y["/2017-6"], y_fit)
```

```
[1] 0.1636789
```

```r
print("rmse of out sample test")
```

```
[1] "rmse of out sample test"
```

```r
rmse(y["2017-7/"], y1_fit)
```

```
[1] 0.5399276
```