Has models' forecasting performance for China inflation changed over time, and when?

Mingwei Li

Abstract

We evaluate various models' relative performance in forecasting future China inflation on a monthly basis. There are 4 candidate models: (1) Benchmark Model (2) Single-Variable Model (3) Time-varying Linear Model. (4) Machine-learning Model. G-R test is adopted to address the problem that the models' relative performance can be varying over time.

We show that the models' relative performance has changed dramatically over time. Single-Variable Model with variable $Treasury\_10$ performs best among the 4 models; Machine-learning Model has the worst performance among these 4 models. In addition, this paper discusses why Machine-learning Model doesn't perform well in forecasting future China inflation.

The code of this essay is at this github project.

Keywords: Inflation Forecasts, Relative Performance, G-R Test

Contents

1 Introduction **3**

2 Theoretical Econometric Method **4**

2.1 Fluctuation Test 4

2.2 One-time Reversal Test 6

3 Data and Model **7**

3.1 Data 7

3.2 Model 7

4 Model Results **9**

4.1 Benchmark Model 9

4.2 Comparison between Benchmark Model and Single-Variable Model 9

4.3 Comparison between Benchmark Model and Time-varying Linear Model 11

4.4 Comparison between Benchmark Model and Machine-learning Model 12

4.5 One-Time Reversal Test 13

4.6 Use U.S Federal Fund Rate as A Predictor 14

5 Conclusion **15**

1 Introduction

This paper investigates whether the relative performance of competing models for forecasting Chinese inflation has changed over time.

Forecast of important economic variables like CPI has become more and more important in recent studies. Several classical econometric models can be used to predict future CPI. Nonlinear machine-learning model seems to have the potential to solve this prediction problem. In order to find the best model, we need to compare the out-of-sample performance between different candidate models. Classical metrics like mean-squared-error can be used to evaluate model performance, but it ignores the fact that model performance changes over time. Therefore, G-R fluctuation test is adopted in our study to calculate relative out-of-sample performance of different models at different times.

In our study, we put up 4 models: (1) Benchmark Model. (2) Single-Variable Model. (3) Time-varying Linear Model. (4) Machine-learning Model. These models are used to fit on rolling windows of 24 months and make out-of-sample forecast from 2004-01 to 2020-02. G-R test is used to compare the out-of-sample performance of these models.

We find that Single-Variable Model with variable $Treasury\_10$ performs significantly better than Benchmark Model; Machine-learning Model has the worst performance among these 4 models.

The rest of the paper is organized as follows. Section 2 describes Fluctuation Test and One-time Reversal Test, the econometric method we adopt in the empirical analysis. Section 3 describes the data set and the details of the 4 models. Section 4 focuses on comparing the performance of different models. Section 5 concludes.

2 Theoretical Econometric Method

We need to compare the out-of-sample forecasting performance of 2 competing models in our research. This section demonstrates Fluctuation test and One-time Reversal test. **This section is totally based on Giacomini and Rossi (2010)** [1].

2.1 Fluctuation Test

The test is straightforward. Sample path of relative out-of-sample errors, together with critical values, is plotted. If relative out-of-sample errors cross critical values, then one of the models outperformed its competitor at some point in time. Figure 1 is an example. We can see model 1 performs significantly better than model 2 in 2018m1.

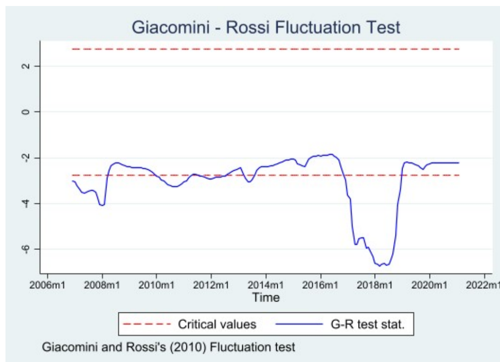


Figure 1: An Example of Fluctuation Test

Fluctuation Test tests the null hypothesis that the two models have equal out-of-sample performance at each point in time, against the alternative that 2 models' out-of-sample performance is different at least at one point in time.

To illustrate the principle of Fluctuation Test, we consider two h step ahead forecasts for the variable y_t, which we assume for simplicity to be a scalar. We use θ and γ to represent the parameter of the first and second model, respectively. We assume that the size of whole sample is T and the size of fitting sample is R. For a general loss function L, we thus have a sequence of $T-(R+h)+1$ out of-sample forecast loss differences, which can be written in the following way. y_t is realized future value; $\widehat{\gamma_{t-h,R}}$ represents the model estimated at $t-h$ with fitting sample size R(estimated on $t-h-R+1, ..., t-h$); $L^{(1)}$ is loss function like rooted mean square error.

$$\{\Delta L_t(\widehat{\theta_{t-h,R}}, \widehat{\gamma_{t-h,R}})\}_{t=R+h}^{T} = \{L^{(1)}(y_t, \widehat{\theta_{t-h,R}}) - L^{(2)}(y_t, \widehat{\gamma_{t-h,R}})\}_{t=R+h}^{T}$$

The local relative loss at time t for the two models as the sequence of out-of-sample loss differences computed over centered rolling windows of size m is as follows:

$$m^{-1} \sum_{j=t-m/2}^{t+m/2-1} \Delta L_j(\widehat{\theta_{j-h,R}}, \widehat{\gamma_{j-h,R}}), t \in \{R+h+m/2, ..., T-m/2+1\}$$

The procedure of Fluctuation Test is as follows:

(1) We first define statistics $F_{t,m}^{OOS}$, which depends on estimated model parameters $\hat{\theta_{j-h,R}}$ and realized y_t.

$$F_{t,m}^{OOS} = \hat{\sigma}^{-1} m^{-1/2} \sum_{j=t-m/2}^{t+m/2-1} \Delta L_j(\hat{\theta_{j-h,R}}, \hat{\gamma_{j-h,R}})$$

$\hat{\sigma}^2$ is a HAC estimator of σ^2 which is defined as follows. $q(P)$ is a bandwith function grow with P.

$$\hat{\sigma}^2 = \sum_{i=-q(P)+1}^{q(P)-1} (1 - |i/q(P)|) P^{-1} \sum_{j=R+h}^{T} \Delta L_j(\hat{\theta_{j-h,R}}, \hat{\gamma_{j-h,R}}) \Delta L_{j-i}(\hat{\theta_{j-i-h,R}}, \hat{\gamma_{j-i-h,R}})$$

(2) Under the null hypothesis $H_0 : E[\Delta L_t(\hat{\theta_{t-h,R}}, \hat{\gamma_{t-h,R}})] = 0, \forall t \in \{R+h, R+h+1, ..., T\}$, which means 2 models perform equally well.

$$F_{t,m}^{OOS} \Rightarrow [\mathcal{B}(\tau + \mu/2) - \mathcal{B}(\tau - \mu/2)]/\sqrt{\mu}$$

Where $t = [\tau(T - (R+h) + 1)], m = [\mu(T - (R+h) + 1)]$ and $\mathcal{B}$ is a standard univariate Brownian motion. The critical values for a significance level α is k_α, which is defined as:

$$Pr\{sup_\tau |[\mathcal{B}(\tau + \mu/2) - \mathcal{B}(\tau - \mu/2)]/\sqrt{\mu}| > k_\alpha\} = \alpha$$

The null hypothesis is rejected against the two-sided alternative $E[\Delta L_t(\hat{\theta_{t-h,R}}, \hat{\gamma_{t-h,R}})] \neq 0, \exists t \in \{R+h, R+h+1, ..., T\}$ when $max_t |F_{t,m}^{OOS}| > k_\alpha$

Critical values for testing H_0 against the one-sided alternative $E[\Delta L_t(\hat{\theta_{t-h,R}}, \hat{\gamma_{t-h,R}})] > 0, \exists t \in \{R+h, R+h+1, ..., T\}$(model θ is worse than γ) can be obtained from

$$Pr\{sup_\tau [\mathcal{B}(\tau + \mu/2) - \mathcal{B}(\tau - \mu/2)]/\sqrt{\mu} > k_\alpha\} = \alpha$$

In this case, we reject when $max_t F_{t,m}^{OOS} > k_\alpha$.

In my research, I would use STATA command **giacross** to conduct fluctuation test.

2.2 One-time Reversal Test

One-time Reversal Test is more complicated than Fluctuation Test, but they share the same principle. We first define 2 statistics LM_1 and $LM_2(t)$. LM_1 measures the performance difference of 2 models over the **whole** test data set($\{R+h, R+h+1, ..., T\}$). $LM_2(t)$ is the difference of the model performance difference between $[R+h, t]$ and $[R+h, T]$. $LM_2(t)$ can be used to test whether there is a time reversal in relative model performance at time t. For simplicity, we define prediction length $P = T - (R+h) + 1$

$$LM_1 = \hat{\sigma}^{-2} P^{-1} [\sum_{j=R+h}^{T} \Delta L_j(\theta_{j-\hat{h},R}, \gamma_{j-\hat{h},R})]^2$$

$$LM_2(t) = \hat{\sigma}^{-2} P^{-1} (t/P)^{-1} (1 - t/P)^{-1} [\sum_{j=R+h}^{t} \Delta L_j(\theta_{j-\hat{h},R}, \gamma_{j-\hat{h},R}) - (t/P) \sum_{j=R+h}^{T} \Delta L_j(\theta_{j-\hat{h},R}, \gamma_{j-\hat{h},R})]^2$$

The estimated variance $\hat{\sigma}^2$ of model performance difference:

$$\hat{\sigma}^2 = \sum_{i=-q(P)+1}^{q(P)-1} (1 - |i/q(P)|) P^{-1} \sum_{j=R+h}^{T} \Delta L_j(\theta_{j-\hat{h},R}, \gamma_{j-\hat{h},R}) \Delta L_{j-i}(\theta_{j-i-\hat{h},R}, \gamma_{j-i-\hat{h},R})$$

We then define $\Phi_P^*(t) = LM_1 + LM_2(t)$, and $QLR_P^* = sup_t \Phi_P^*(t), t \in \{[0.15P], ...[0.85P]\}$. Under the null hypothesis:

$$H_0 : E[\Delta L_t(\theta_{t-\hat{h},R}, \gamma_{t-\hat{h},R})] = 0, \forall t \in \{R+h, ..., T\}$$

We have $QLR_P^* \Rightarrow sup_\tau [\frac{\mathcal{BB}(\tau)^2}{\tau(1-\tau)} + \mathcal{B}(1)^2]$, where $t = [\tau P]$, and $\mathcal{B}(.)$ and $\mathcal{BB}(.)$ are, respectively, a standard univariate Brownian motion and a Brownian bridge.

The procedure of Fluctuation Test is as follows:

(1) Test the hypothesis of equal performance at each time by using the statistic QLR_P^* we defined before for a significance level α.

(2) If the null is rejected, compare LM_1 and $sup_t LM_2(t), t \in \{[0.15P], ...[0.85P]\}$ with the following critical values: (3.84, 8.85) for $\alpha = 0.05$. If only LM_1 rejects then there is evidence in favor of the hypothesis that **one model is constantly better than its competitor**. If only LM_2 rejects, then there is evidence that there are **instabilities in the relative performance of the two models but neither is constantly better over the full sample**. If both reject, then it is not possible to attribute the rejection to a unique source.

(3) Estimate the time of the break by $t^* = argmax_{t \in \{[0.15P], ..., [0.85P]\}} LM_2(t)$.

(4) The time path of the underlying relative performance is:

$$\begin{cases} \frac{1}{t^*} \sum_{j=R+h}^{t^*} \Delta L_j(\theta_{j-\hat{h},R}, \gamma_{j-\hat{h},R}), t \leq t^* \\ \frac{1}{P-t^*} \sum_{j=t^*+1}^{T} \Delta L_j(\theta_{j-\hat{h},R}, \gamma_{j-\hat{h},R}), t > t^* \end{cases}$$

As there is no available STATA package for one-time reversal test, I would use **Python** to draw the time path of the underlying relative performance.

3 Data and Model

3.1 Data

Our data is collected from CSMAR data base(China Stock Market & Accounting Research Database). The time span is 2002/01-2021/02, and frequency is month. Table 1 shows part of our original data.

Table 1: Part of Data

Time_index	cpi_index	Int_Rate	Treasury_1Yr	Treasury_5Yr	Treasury_10Yr	CSI_300	M2	M1	M0
2002/01	1	2.32	2.13	2.77	3.36	1221.76	159639.27	60576.06	16725.89
2002/02	1.011	2.32	2.18	2.68	3.19	1244.603	160935.59	58702.87	16641.55
2002/03	0.997857	2.11	1.96	2.49	3.01	1313.466	164064.57	59474.83	15544.63
2002/04	0.994863429	2.08	1.98	2.32	2.68	1353.607	164570.56	60461.31	15864.18
...	...	...	...	...	...	...	...	...	...
2021/01	1.637983773	1.662	2.46	2.96	3.15	5351.9646	2213047.33	625563.81	89625.24
2021/02	1.647811676	1.9416	2.65	3.07	3.24	5336.7609	2236030.26	593487.46	91924.6

Description and Unit of data are listed in table 2.

Table 2: Description of Data

Data Item	Description and Unit
Int_Rate	Interbank Offered Rate: Overnight(%)
Treasury_1Yr	One year national debt rate(%)
Treasury_5Yr	Five year national debt rate(%)
Treasury_10Yr	Ten year national debt rate(%)
CSI_300	CSI 300 index(Chinese stock index)
M2	M2 currency supply(10^8 RMB)
M1	M1 currency supply(10^8 RMB)
M0	M0 currency supply(10^8 RMB)

Following the work finished by Rossi and Sekhposyan(2008) [2], the prediction target is defined as follows:

$$y_{t+h} = 1200ln(CPI_{t+h}/CPI_t)/h - 1200ln(CPI_t/CPI_{t-1}), h = 12$$

CPI_t is cpi_index, and the corresponding code is available at here. All time series have been transformed to stationary series before modeling.

3.2 Model

There are 4 models in our research. The prediction target is:

$$y_{t+h} = 1200ln(CPI_{t+h}/CPI_t)/h - 1200ln(CPI_t/CPI_{t-1}), h = 12$$

(1) Benchmark Model: We define $y_t = 1200ln(CPI_t/CPI_{t-1})$, and the benchmark model is as follows. The corresponding code for Benchmark Model is available at here.

$$y_{t+h} = \beta_0 + \beta_1 y_t + \epsilon_t$$

(2) Single Variable Model: We define $y_t = 1200ln(CPI_t/CPI_{t-1})$, and x_t is a single explanatory variable like M_0. The single variable model is:

$$y_{t+h} = \beta_0 + \beta_1 y_t + \alpha x_t + \epsilon_t$$

For instance, both $y_{t+h} = \beta_0 + \beta_1 y_t + \alpha Treasury\_1Yr_t + \epsilon_t$ and $y_{t+h} = \beta_0 + \beta_1 y_t + \alpha M_{0,t} + \epsilon_t$ are single variable models. The corresponding code for single variable model is available at here.

(3) Time-varying linear Model: We define $y_t = 1200ln(CPI_t/CPI_{t-1})$, and $\vec{x}_t$ is a vector of other explanatory variables. The elements of $\vec{x}_t$ can vary at different times. The time-varying linear Model is:

$$y_{t+h} = \beta_0 + \beta_1 y_t + \vec{\alpha} \times \vec{x}_t + \epsilon_t$$

For instance, if we want to predict y_{t+h} when $t = 2004/01$, we may fit the model $y_{t+h} = \beta_0 + \beta_1 y_t + \alpha_1 M_{0,t} + \alpha_2 M_{1,t} + \epsilon_t$; if we want to predict y_{t+h} when $t = 2008/01$, we may fit the model $y_{t+h} = \beta_0 + \beta_1 y_t + \alpha_1 Treasury\_1Yr_t + \epsilon_t$. Explanatory variables change at different time points. The details of time-varying linear model are listed below. The corresponding code for time-varying linear model is available at here.

Algorithm 1 Time-varying Linear Model Algorithm

Input $all\_feature\_names$
for $i = 2004m1$ to $2020m2$ **do**
  Define $current\_feature\_names = all\_feature\_names$
  while True **do**
    Fit linear model on $current\_feature\_names$(Fitting period is $[i-24, i-1]$)
    if Every feature of $current\_feature\_names$ is significant(p-value≤ 0.05) **then**
      Break
    else
      Delete all insignificant features from $current\_feature\_names$
    end if
  end while
  Make out-of-sample prediction regarding time point i
end for

Not like classical linear econometric model, the following machine-learning model can capture nonlinear relationship.

(4) Machine-Learning Model: Classical tree model XGBRegressor is adopted to make out-of-sample forecast. Feature selection and parameter tunning are based on cross-validation score. $\vec{x}_t$ is a vector of other explanatory variables. The corresponding code for machine-learning model is available at here.

$$y_{t+h|t} = Tree\_Model(y_t, \vec{x}_t)$$

Algorithm 2 Machine-Learning Model Algorithm

Input $all\_feature\_names$
for $i = 2004m1$ to $2020m2$ **do**
  Define $current\_feature\_names = all\_feature\_names$
  Fit $XGBRegressor$ on $current\_feature\_names$(Fitting period is $[i-24, i-1]$)
  We get top K important features based on $XGBRegressor$
  for $k = 1$ to K **do**
    Fit $XGBRegressor$ on top k features (Fitting period is $[i-24, i-1]$)
    Use hyperopt(Python Package) to get the parameters with the lowest CV-error.
  end for
  Get k_{best} features and $param_{best}$ based on CV-error.
  Use k_{best} and $param_{best}$ to make out-of-sample prediction regarding time point i
end for

4 Model Results

4.1 Benchmark Model

Benchmark model uses $y_t = 1200ln(CPI_t/CPI_{t-1})$ as the single explanatory variable.

$$y_{t+h} = \beta_0 + \beta_1 y_t + \epsilon_t$$

Figure 2 demonstrates the R-squared of Benchmark Model. We can see R-squared is higher than 0.90 in most times, which means benchmark model performs very well. Therefore, it's difficult to make improvement on the benchmark model.

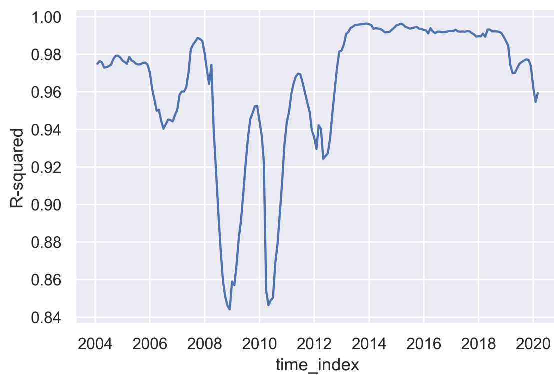


Figure 2: R-squared of Benchmark Model at Different Times

4.2 Comparison between Benchmark Model and Single-Variable Model

We define $y_t = 1200ln(CPI_t/CPI_{t-1})$, and x_t is a single explanatory variable. Single-Variable model is as follows:

$$y_{t+h} = \beta_0 + \beta_1 y_t + \alpha x_t + \epsilon_t$$

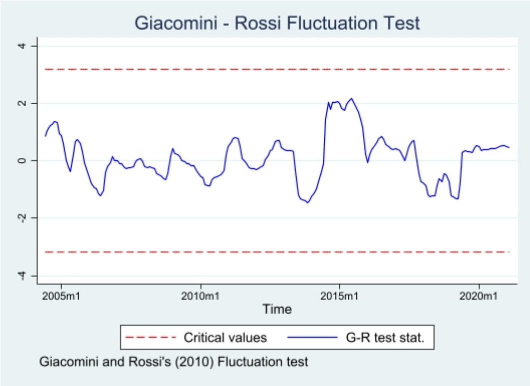
There are 8 possible single explanatory variables, and giacross fluctuation test results are listed in figure 3. Blue line is an estimator for the difference of out-of-sample performance between 2 models. Red dashed lines are upper and lower critical bound. The tesi logi is as follows:

(1) Blue line penetrates upper red dashed line$\Rightarrow$ model 1 performs significantly worse than model 2 at that time.

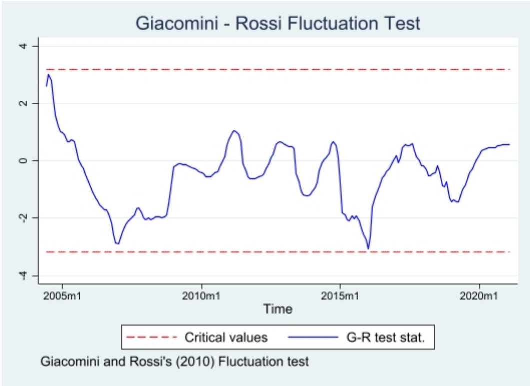
(2) Blue line penetrates lower red dashed line$\Rightarrow$ model 1 performs significantly well than model 2 at that time.

(3) Blue line lower than 0$\Rightarrow$ model 1 performs well than model 2, but this advantage is not significant.

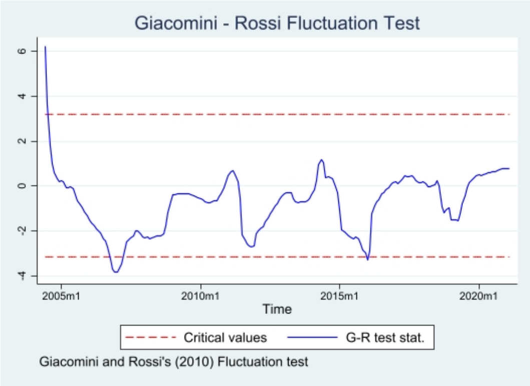
We can see only single variable $Treasury\_10$(sub-figure(d)) can improve benchmark model significantly in 2006 and 2007. Meanwhile, $Treasury\_1$ and $Treasury\_5$ can make insignificant improvement on benchmark model.



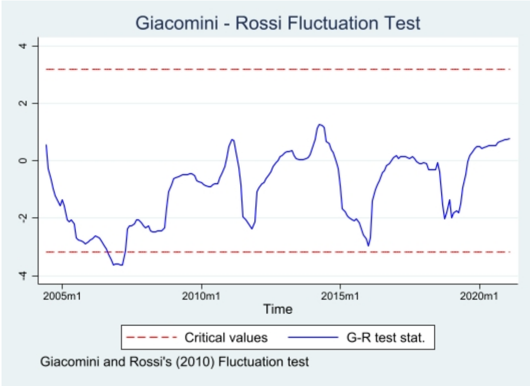
(a) Single Variable:Int_Rate



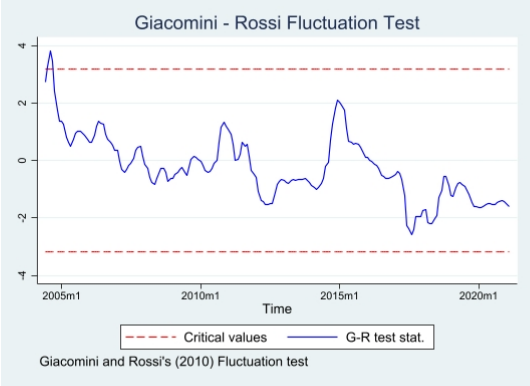
(b) Single Variable:Treasury_1



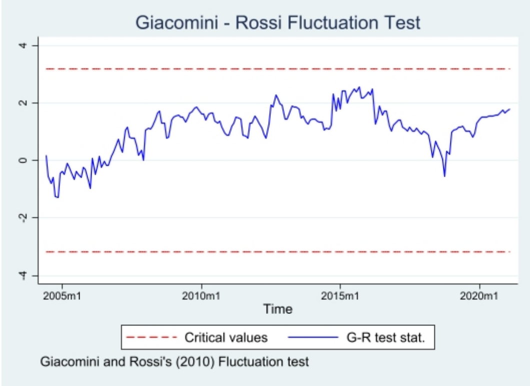
(c) Single Variable:Treasury_5



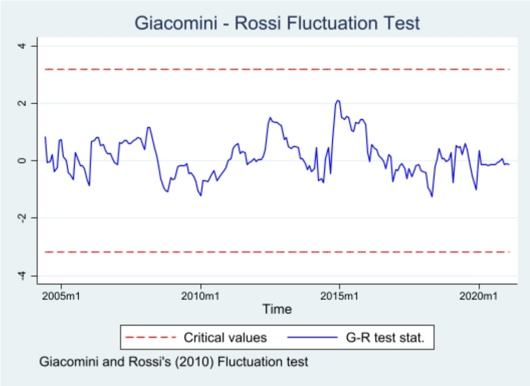
(d) Single Variable:Treasury_10



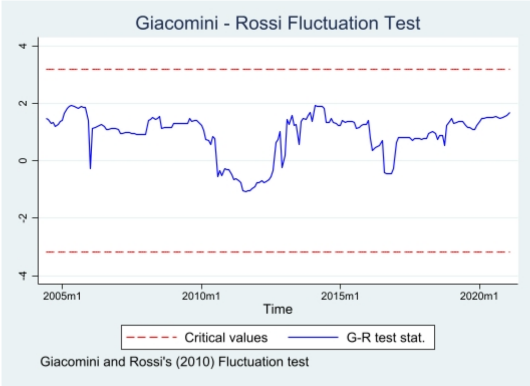
(e) Single Variable:CSI_300



(f) Single Variable:M2



(g) Single Variable:M1



(h) Single Variable:M0

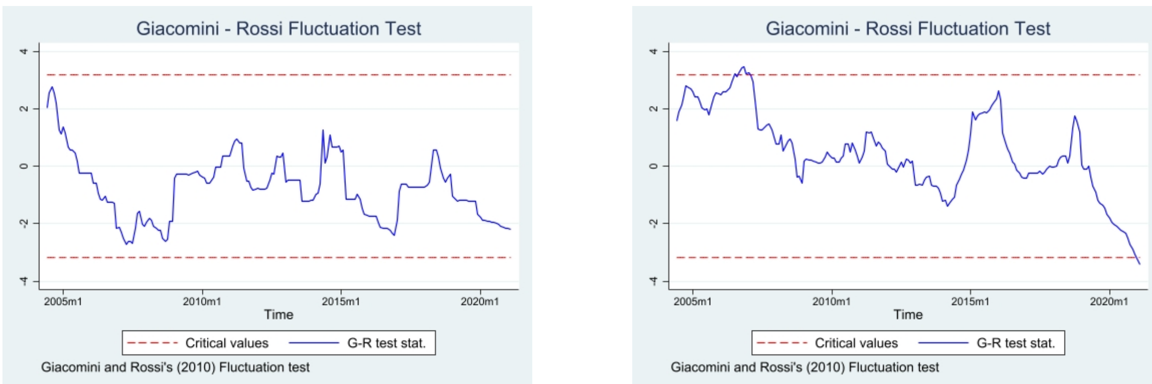
Figure 3: Giacross Fluctuation Test Results

4.3 Comparison between Benchmark Model and Time-varying Linear Model

Time-varying linear Model is defined in the following way. The elements of $\vec{x}_t$ can vary at different times.

$$y_{t+h} = \beta_0 + \beta_1 y_t + \vec{\alpha} \times \vec{x}_t + \epsilon_t$$

Figure 4 is the fluctuation test results for time-vary linear model. From sub-figure(a), we can see time-varying linear Model performs better than benchmark model in most of the time, but the advantage is not significant. When comparing with $Treasury\_10$, the best single-variable model, time-varying linear Model is significantly worse in 2007, but significantly better in 2020.



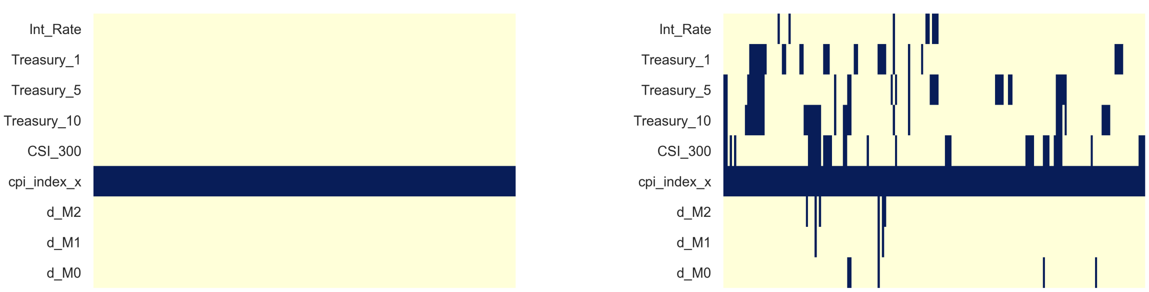
(a) Time-varying Linear Model V.S Benchmark

(b) Time-varying Linear Model V.S Treasury_10(single variable)

Figure 4: Giacross Fluctuation Test Results for Time-varying Linear Model

Figure 5 demonstrates the features of time-varying linear model and benchmark model. x-axis stands for time index and y-axis represents different features. Color blue means particular feature is utilized at a certain time. Color yellow means that feature is not used at that time.

We can see (1) benchmark model uses single feature cpi_index_x the whole time period. (2) time-varying linear model adopts $Treasury\_1, Treasury\_5$ more frequently than currency variables like M_0 and M_1. (3) Features other than cpi_index_x are used more frequently in the first half period than in the last half period, which means they tend to lose their predictive ability in the last half period.



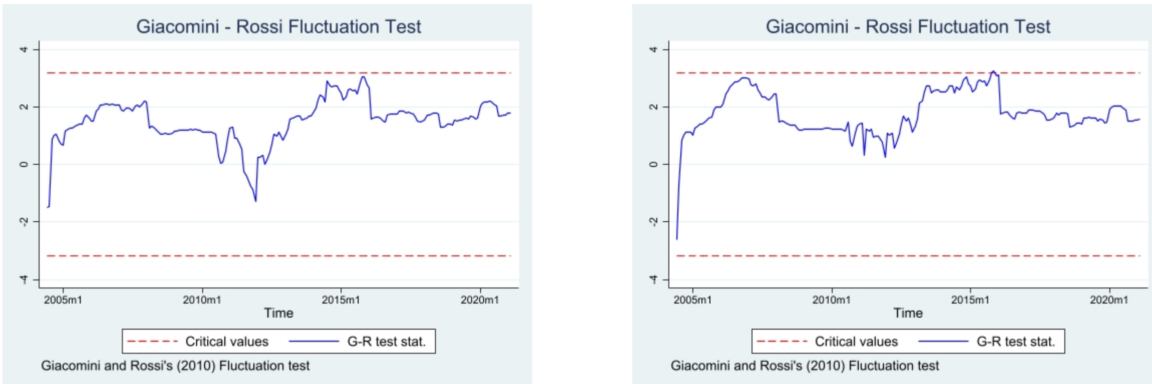
(a) Features of Benchmark Model

(b) Features of Time-varying Linear Model

Figure 5: Features of Time-varying Linear Model and Benchmark Model

4.4 Comparison between Benchmark Model and Machine-learning Model

According to figure 6, machine-learning model is worse than benchmark model in most of the time, but this disadvantage is not significant. In 2016, machine-learning model is significantly worse than single-variable($Treasury\_10$) model.

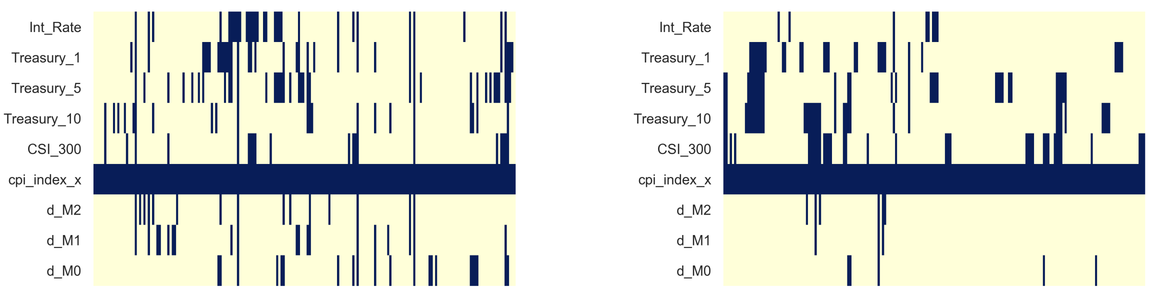


(a) Machine-learning Model V.S Benchmark

(b) Machine-learning Model V.S Treasury_10(single variable)

Figure 6: Giacross Fluctuation Test Results for Machine-learning Model

According to figure 7, Machine-learning Model uses more features than Time-varying Linear Model. Too many features would involve noise, and that's the reason why meachine learning model performs poorly in our research.



(a) Features of Machine-learning Model

(b) Features of Time-varying Linear Model

Figure 7: Features of Time-varying Linear Model and Machine-learning Model

Figure 8 demonstrates feature importance based on $XGBRegressor$ model. We can see cpi_index_x(y_t) is the only feature that performs well over the whole time period.

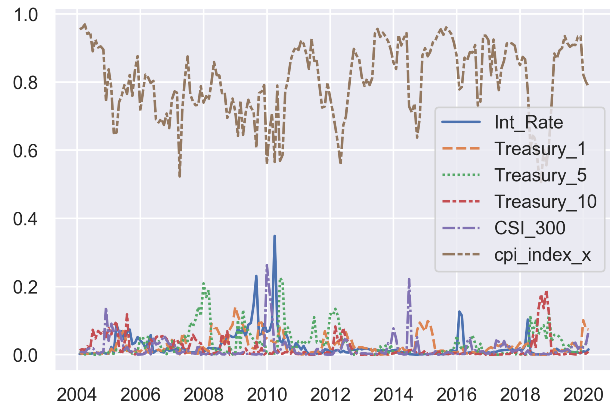


Figure 8: Feature Importance based on Machine-learning Model

4.5 One-Time Reversal Test

Results of One-Time Reversal Test are listed in table 3. We define $LM_2(t) = [\sum_{j=R+h}^{t} \Delta L_j(\theta_{j-\hat{h},R}, \gamma_{j-\hat{h},R}) - (t/P)\sum_{j=R+h}^{T} \Delta L_j(\theta_{j-\hat{h},R}, \gamma_{j-\hat{h},R})]^2$, which is different from previous section for the convenience of calculation.

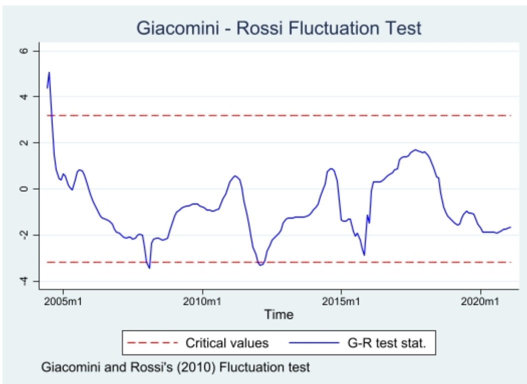
Table 3: Results of One-Time Reversal Test

Model Type	Break Point	Performance
Int_Rate_forecast	Apr-10	worse after break point
Treasury_1_forecast	Sep-08	worse after break point
Treasury_5_forecast	Sep-08	worse after break point
Treasury_10_forecast	Aug-08	worse after break point
CSI_300_forecast	Aug-04	better after break point
d_M2_forecast	Nov-09	better after break point
d_M1_forecast	Jan-10	worse after break point
d_M0_forecast	Feb-09	better after break point
machinelearning_out_of_sample	Dec-11	worse after break point
timevary_forecast	Oct-08	worse after break point

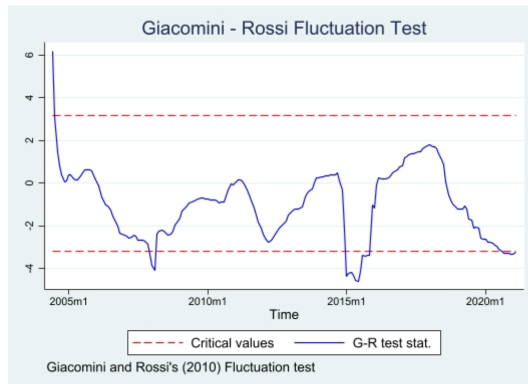
From table 3, we can see those break points mostly distribute around 2008 and 2009. The global financial crisis happened at that time.

4.6 Use U.S Federal Fund Rate as A Predictor

In this part, U.S federal fund rate is adopted to predict China CPI. From sub-figure (a), we can see single variable model of U.S federal fund rate is better than benchmark model in most of the time. In 2008 and 2012, G-R statistics pass through the lower bound, which means U.S federal fund rate is significantly better than benchmark. What seems interesting is sub-figure(b), which demonstrates the comparison between U.S federal fund rate and China Interest Rate. **Different from our common sense**, U.S federal fund rate performs significantly better than China Interest Rate in predicting China CPI in 2008 and 2015! Such phenomenon may be due to financial crisis in 2008 and China stock market crash in 2015. It requires further study to interpret this phenomenon.



(a) Single Variable of Federal Fund Rate V.S Benchmark



(b) Single Variable of Federal Fund Rate V.S China Interest Rate

Figure 9: Giacross Fluctuation Test Results for U.S Federal Fund Rate as A Predictor

The code of this part is available at here

5 Conclusion

In order to predict Chinese inflation rate 12 months ahead, we put up 4 models: (1) Benchmark Model. (2) Single-Variable Model. (3) Time-varying Linear Model. (4) Machine-learning Model. These models are used to fit on rolling windows of 24 months and make forecast from 2004-01 to 2020-02. G-R test is used to compare the out-of-sample performance of these models.

We find:

(1) The R-squared of Benchmark Model is higher than 0.90 most of the time, which means Benchmark Model performs very well and it's very hard to make improvements on Benchmark Model.

(2) Single-Variable Model of $Treasury\_10$ outperforms Benchmark Model. According to G-R test, Single-Variable Model of $Treasury\_10$ ≈Time-varying Linear Model>Benchmark Model>Machine-learning Model. Model A>Model B means model A performs better than model B. ≈ means 2 model's performances are similar.

(3) cpi_index_x(y_t) is the only feature that performs well over the whole time period. Most of the predictive power comes from this single feature. Machine-learning Model is good at capturing nonlinear relationships and combining different features. In our study, as most of the predictive power comes from cpi_index_x, we can't make full use of the feature combination ability of Machine-learning Model. Moreover, other features would bring a lot of noises to machine-learning model, that's the reason why machine-learning performs poorly in our study.

(4) U.S federal fund rate performs significantly better than China Interest Rate in predicting China CPI in 2008 and 2015.

References

[1] Raffaella Giacomini and Barbara Rossi. Forecast comparisons in unstable environments. *Journal of Applied Econometrics*, 25(4):595–620, 2010.

[2] Barbara Rossi and Tatevik Sekhposyan. Have economic models' forecasting performance for us output growth and inflation changed over time, and when? *International Journal of Forecasting*, 26(4):808–835, 2010.