STAT 447 Project Final Report

Jessie Liang (52819596)

2025-04-08

0. GitHub repo link

https://github.com/jessie-liang/447-project.git

1. Problem formulation

Due to various factors affecting Canadian economic environment (e.g. policy making, international relation, economic development, etc), the consumer price index (CPI) has been climbing yet fluctuating over the past 20 years. CPI is an index describing changes in prices faced by Canadian consumers by comparing the cost of a representative basket of goods and services through time (Statistics Canada, 2019). CPI is widely used to measure living costs of residents and provide insights of inflation. Therefore, it would be beneficial for all economic agents if future CPI index could be predicted accurately so as to warn people to take necessary and timely actions. The problem that this project aims to solve is to build a reasonably accurate prediction model that helps forecast future CPI values, through a Bayesian approach.

CPI data is essentially a time series, where dependence among neighboring observations cannot be ignored. To incorporate this characteristic into our Bayesian framework, state-space model and ARMA model are used for modelling in this project. One of the main challenges is to elaborately capture all features that the time series exhibits, including an upward trend and cyclical variations, and then yield predictions that also show these features. The steps to tackle this problem are: (1) Split the original dataset into a training set and a testing set. (2) Decompose the training set using LOESS method (Cleveland et al., 1990) into a trend component, a seasonal component and remainders. (3) Model the trend component using the state-space model and calculate predictions of this component. (4) Model the remainders component using ARMA model and calculate predictions of this component. (5) Obtain point forecasts by summing three parts, namely predictions of the trend component, the corresponding values of the seasonal component, and predictions of the remainders component. (6) Obtain 95% credible intervals by calculating the 2.5% and 97.5% quantile of the posterior samples of point forecasts. (7) Compare the prediction results with the original testing set.

2. Literature review

3. Data analysis

4. Discussion

5. Bibliography

1. Statistics Canada. (2019, May 28). Consumer price index portal. Www.statcan.gc.ca. https://www.statcan.gc.ca/en/subjects-start/prices_and_price_indexes/consumer_price_indexes

2. Cleveland, R. B., Cleveland, W. S., & Terpenning, I. (1990). STL: A seasonal-trend decomposition procedure based on loess. Journal of Official Statistics, 6(1), 3. Retrieved from https://www.proquest.com/scholarly-journals/stl-seasonal-trend-decomposition-procedure-based/docview/1266805989/se-2

6. Appendix

```
library(tidyverse)
library(forecast)
cpi <- read.csv("../data/consumer_price_index.csv")[910:1209,]
year <- rep(2000:2024, each = 12)
month <- rep(1:12, times = 25)
cpi <- data.frame(year, month, cpi)
cpi$cpi <- as.numeric(cpi$cpi)
cpi.ts <- ts(cpi$cpi, start = c(2000,1), frequency = 12)
```

```
training <- window(cpi.ts, c(2000,1), c(2022,6), frequency = 12)
test.ts <- window(cpi.ts, c(2022,7), c(2024,12), frequency = 12)
cpi.stl <- stl(training, s.window = "periodic")
trend <- cpi.stl$time.series[,"trend"]
seasonals <- cpi.stl$time.series[,"seasonal"]
remainders <- cpi.stl$time.series[,"remainder"]

auto.arima(remainders)
```

```
## Series: remainders
## ARIMA(2,0,2) with zero mean
##
## Coefficients:
##          ar1      ar2      ma1      ma2
##       1.3263  -0.4929  -0.6481  -0.2082
## s.e.  0.0969   0.0912   0.1097   0.1150
##
## sigma^2 = 0.08896:  log likelihood = -54.86
## AIC=119.73   AICc=119.95   BIC=137.72
```

```
# the result shows the most appropriate model for remainders is: ARMA(2,2)
```

```
require(rstan)

fit = stan(
  seed = 447,
  file = "my_stan_3.stan",
  data = list(N_train=270,
              N_test=30,
              y_train=as.numeric(trend)),
  iter = 2000,
  refresh = 0
)

samples <- extract(fit)

point_forecast <- rep(NA, 30)
lower_bound <- rep(NA, 30)
upper_bound <- rep(NA, 30)
for (i in 1:30) {
  summary <- summary(fit, pars = c("predictions"))$summary
  point_forecast[i] <- summary[i,1]
  lower_bound[i] <- summary[i,4]
  upper_bound[i] <- summary[i,8]
```

```
}
df_trend <- data.frame(1:30, point_forecast, lower_bound, upper_bound)

trend_posterior <- samples$predictions
trend_posterior <- as.data.frame(trend_posterior)
```

```
require(rstan)

fit = stan(
  seed = 447,
  file = "my_stan_4.stan",
  data = list(T=270,
              N_test=30,
              y=as.numeric(remainders)),
  iter = 2000,
  refresh = 0
)

samples <- extract(fit)

point_forecast <- rep(NA, 30)
lower_bound <- rep(NA, 30)
upper_bound <- rep(NA, 30)
for (i in 1:30) {
  summary <- summary(fit, pars = c("predictions"))$summary
  point_forecast[i] <- summary[i,1]
  lower_bound[i] <- summary[i,4]
  upper_bound[i] <- summary[i,8]
}
df_remainders <- data.frame(1:30, point_forecast, lower_bound, upper_bound)

remainder_posterior <- samples$predictions
remainder_posterior <- as.data.frame(remainder_posterior)
```

```
trend_estimates <- ts(df_trend$point_forecast, start = c(2022,7), frequency = 12)
seasonal_estimates <- window(seasonals, start = c(2000,7), end = c(2002,12))
seasonal_estimates <- ts(as.numeric(seasonal_estimates), start = c(2022,7), frequency = 12)
remainder_estimates <- ts(df_remainders$point_forecast, start = c(2022,7), frequency = 12)

posterior <- trend_posterior + remainder_posterior
for (i in 1:30) {
  posterior[,i] <- posterior[,i] + as.numeric(seasonal_estimates)[i]
}
quantiles <- apply(posterior[],2,quantile,probs=c(0.025,0.975))
lower.bound <- quantiles[1,]
upper.bound <- quantiles[2,]

point.forcast.ts <- trend_estimates + seasonal_estimates + remainder_estimates
lower.bound.ts <- ts(as.vector(lower.bound), start = c(2022,7), frequency = 12)
upper.bound.ts <- ts(as.vector(upper.bound), start = c(2022,7), frequency = 12)
ts.plot(test.ts, point.forcast.ts, lower.bound.ts, upper.bound.ts,
        gpars = list(xlab = "year",
                     ylab = "monthly CPI index",
                     main = "Forcasted series compared to original series",
```

```
                    col = c("black", "red", "blue", "blue")
                    ),
        ylim = c(145, 170)
        )
legend("topleft", legend = c("point forecast", "real data", "prediction interval"),
col = c("red", "black","blue"),lty = c(1,1,1), cex=0.8)
```