Sequence to Sequence modeling of breakpoints in time series

Amy Pitts

Marist College - DATA 440
April 10, 2019

Abstract

1 Introduction

When modeling time series data, it can be necessary to identify places or points in time where significant change occurs in the behavior of the data. By identifying these breakpoints or change points, different parts of the data can be fitted with separate, more appropriate models, allowing for noteworthy changes to be better characterized in the combined model. The summer of 2018 I spend 8 weeks working with a team at Lafayette College coming up with a Bayesian procedure that identifies the number and location of breakpoints in times series. The final product that was produced was Bayesian Adaptive Auto-Regression (BAAR), a new procedure for accurately and efficiently finding the distribution of the number and location of breakpoints in time series. I would cite my own research, however, it has not been peer-reviewed yet. The goal of this project is to attempt to create another technique that identifies breakpoints. This research project will use a sequence to sequence neural network approach to locate breakpoints. The algorithm will learn off of simulated data where the breakpoint is known and then testing the technique on re-world data. These results will be compared to the results that my REU group found over the summer as well as well regarded journals.

2 Background and History

Since breakpoints are found in numerous types of time series, there has been ample interest in developing techniques to detect them in recent decades across a wide range of fields. Existing techniques have been applied to everything from the United States Treasury bill rates [**?**] to hydrology [**?**] to climate records [**?**].

There have been various technique in detecting change-point locations. The simplest technique relies on expert opinion. This is the process experts approximating where the breakpoint location will occur based on historical knowledge. However, there is a lot of human error introduced with that technique which sparked the development of more computational methods. One widely used technique is the Bai-Perron Test [**?**] and has an accessible R package "strucchange" [**?**]. The test returns a single optimal breakpoint set but requires the user to specify a maximum number of breakpoints and minimum segment size. Another technique is Bayesian Adaptive Regression Splines (BARS), a Bayesian curve fitting technique developed by DiMatteo et al. [**?**] and implemented by Wallstrom, Liebner, and Kass [**?**]. These two methods inspired Bayesian Adaptive Auto-Regression (BAAR) procedure.

The Sequence to Sequence neural network is a technique that is typically used for language processing. The process is to train a model to convert a sequence from one domain into another. The most commonly used example is taking english words and translating them into french [**?**]. The Sequence to Sequence process is able to take sequence and convert it into another sequence of a different length [**?**]. Before other Deep Neural Networks (DNNs) also perform arbitrary parallel computation for a modest number of steps however some techniques puts a requirement on keeping the lengths of sequences the same.

3 Method

I need to expand on this. I am basing my work of off a keras blog. Primarily where they do the juvenile example of training a sequence to sequence neural network to add two number. That code can be found at the link

$$https://github.com/keras-team/keras/blob/master/examples/addition_rnn.py$$

Also the keras blog is $https://blog.keras.io/a-ten-minute-introduction-to-sequence-to-sequence-learning-in-keras.html$.

4 Implementation in python

One of the bigger stumbling blocks that I have come across is my lack of data. Therefore, I am going to train the sequence to sequence neural network on simulated time series. What this code is doing is randomly generating some specified amount of numbers between two value. The code is alternating between two set of values to create breaks in the data. The locations of the breaks are randomly generated and kept track off. For each iteration, 10 different data sets are produced the first starting with one break and then each data set after has one more break until 10 breakpoints.

```
###Setting the dataset

length = 1000 #defining the number of datapoints in the set
iterations = 10 #changing the number of iterations. not to high
   please

master_data = []
master_breaks = []

#I ran into a problem of the set of breaks was not in sense of time
#but just number of datapoints in that set
def add_one_by_one(l):
    new_l = []
    cumsum = 0
    for elt in l:
        cumsum += elt
        new_l.append(cumsum)
    return new_l

#creating the data. 1-10 breaks for each iteration
for x in range(iterations):
  for y in range(10):
    num_of_breaks = y+1

    s=[]
    for x in range(num_of_breaks):
```

```
      s.append( r.randint(1,round(length/(num_of_breaks))))
    s.append(length-sum(s)) #making sure the data is the right
       length

    numbers = []
    for x in range(num_of_breaks+1):
      #every other set should have 0-5 and 5-10 datapoint values
      if(x%2 == 0):
        l = np.random.uniform(0,5,s[x])
      else:
        l = np.random.uniform(5,10,s[x])
      numbers = numbers + [*l]
    master_data.append(numbers) #saving that dataset
    breaks = add_one_by_one(s) #changing the breakpoints
    master_breaks.append(breaks) #saving the breakpoints

#testing my creation
plt.plot(master_data[1])
master_breaks[1]
```

5 Results

6 Discussion

7 Conclusion

References

[1] I. Sutskever, O. Vinyals, Q. V. Le "Sequence to Sequence Learning with Neural Networks *Advances in neural information processing systems*, pp. 3104-3112, 2014.

[2] J. Bai, and P. Perron, "Computation and analysis of multiple structural change models. *Journal of applied econometrics*, vol.18(1), pp.1-22. 2003.

[3] I. DiMatteo, C.R. Genovese, R.E. and Kass "Bayesian curvefitting with freeknot splines. *Biometrika*, vol.88(4), pp.1055-1071. 2001

[4] M.H. Pesaran, D. Pettenuzzo, and A. Timmermann, Forecasting time series subject to multiple structural breaks. *The Review of Economic Studies*, vol.73(4), pp.1057-1084. 2006

[5] E. Ruggieri, A Bayesian approach to detecting change points in climatic records. *International Journal of Climatology*, 33(2), pp.520-528. (2013).

[6] O. Seidou, and T.B. Ouarda, "Recursionbased multiple changepoint detection in multiple linear regression and application to river streamflows. *Water Resources Research*, vol.43(7). 2007

[7] G. Wallstrom, J. Liebner, and R.E. Kass, "An implementation of Bayesian adaptive regression splines (BARS) in C with S and R wrappers. *Journal of Statistical Software*, vol.26(1), p.1. (2008).

[8] A. Zeileis, F. Leisch, B. Hansen, K. Hornik, C. Kleiber, and M.A. Zeileis. *The strucchange Package*. R manual. breakpoint in the strucchange package (2007)