Time Series Decomposition Based on Markov Random Fields

Bowen Xiao

May 16, 2018

Intrduction

The project is about Markov random fields, which is a locally adaptive nonparametric curve fitting method that operates within a fully Bayesian framework. The model assumes that each data point is generated independently with some parametirc models, like normal distributions, and the parameters are follow a Markov random fields model, which could, according to the paper, provide a combination of local adaptation and global control. For example, supposed Y_i has already been ordered by X_i, then $Y_i \sim N(\theta_i, \sigma^2)$, where $i = 1, 2, ..., n$, and $(\theta_{i+1} - \theta_i)|\tau_i \sim N(0, \tau_i^2)$, where $i = 1, 2, ..., n-1$, and τ_i are independent and identically distributed. σ^2 is a global parameter and has its own prior. This is a nonparametric model because the number of parameters is increasing with the number of data points. Specificly, my model will look like this:

$$\begin{aligned}
zsigma &\sim uniform(0,1);\\
zgam &\sim uniform(0,1);\\
ztheta1 &\sim normal(0,1);\\
zdelta &\sim normal(0,1);\\
sigma &= 5.0 * tan(zsigma * \pi/2);\\
gam &= 0.06 * tan(zgam * \pi/2);\\
theta_1 &= 5 * sd(y) * ztheta1 + \bar{y};\\
theta_{j+1} &= gam * zdelta_j + theta_j;\\
y_i &\sim normal(theta_i, sigma);
\end{aligned}$$

The idea of Bayesian Markov random fields comes from the paper in the reference[1]. This project includes the following two parts, simulation study of Gaussian Markov random fileds (GMRF) fitting and time series decomposition based on Markov random fiels. For time series decomposition, since it is count variable, in the first fitting I replace $y_i \sim normal(theta_i, sigma)$ with $y_i \sim Poisson(e^{theta_i}, sigma)$. As is well known, Bayes methods treat parameters as random variables. consequently, what we get is posterior distribution of θ. To compare with frequentist methods, I will simply use the median of θ. Furthermore, I choose the above non-centered parameterization to achieve better sampling performance, because centered parameterized model is found to have divergence issue in simulation study.

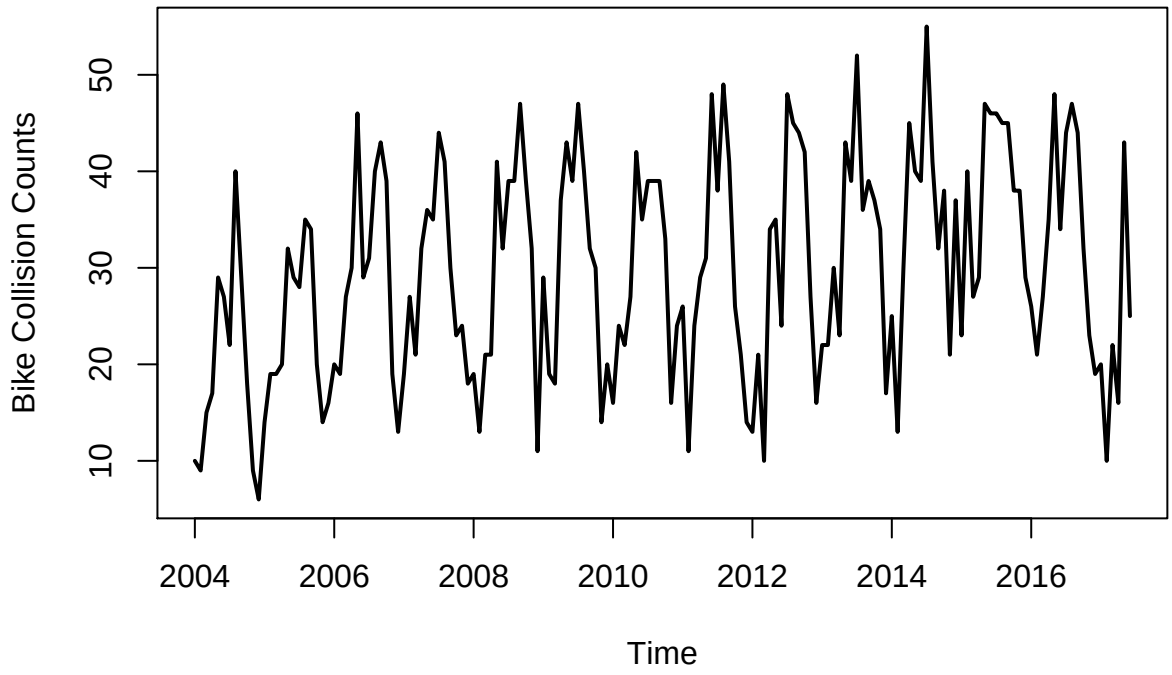
Data Description

Local Smoothness

First of all, I apply Bayesian Markov random fields into nonparametric fitting and compare it with polynomial regression and NW estimators. Data will be generated randomly: $X_i \sim^{iid} Uniform(0,1)$ and $Y_i = f(X_i) + \epsilon_i$, where f is known, $\epsilon_i \sim^{iid} N(0,1)$ and ϵs are independent with Xs. In this project, I consider two scenarios: $f(X_i) = 2X_i$ and $f(X_i) = sin(10X_i)$.

Time Series Decomposition

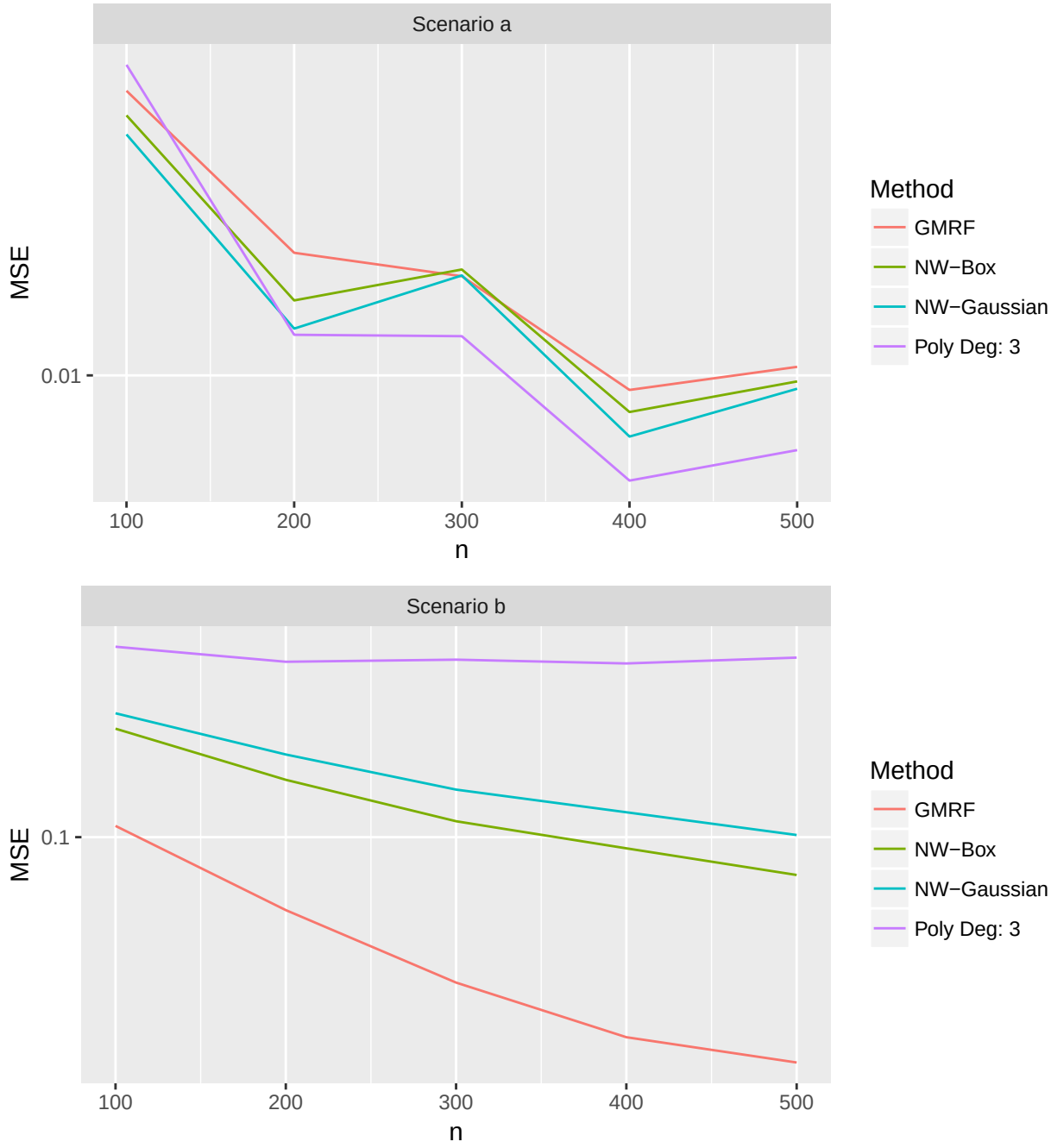
Secondly, I apply this techinique to my time series data. Specificly, I have data of bike collision records in downtown Seattle from 01/2004 to 06/2017 and I summarize them by month, so that I have 162 bike collision counts, which is shown as following. I decomposite this time series into trend part, seasonal part and random part. I treat it as an additive model and decomposit it by doing Markov random fields fitting twice. Furthermore, I compare the results with some classical parametric techniques, like ARIMA, and machine learning techniques, like RNN/LSTM, with regrard to forcasting.



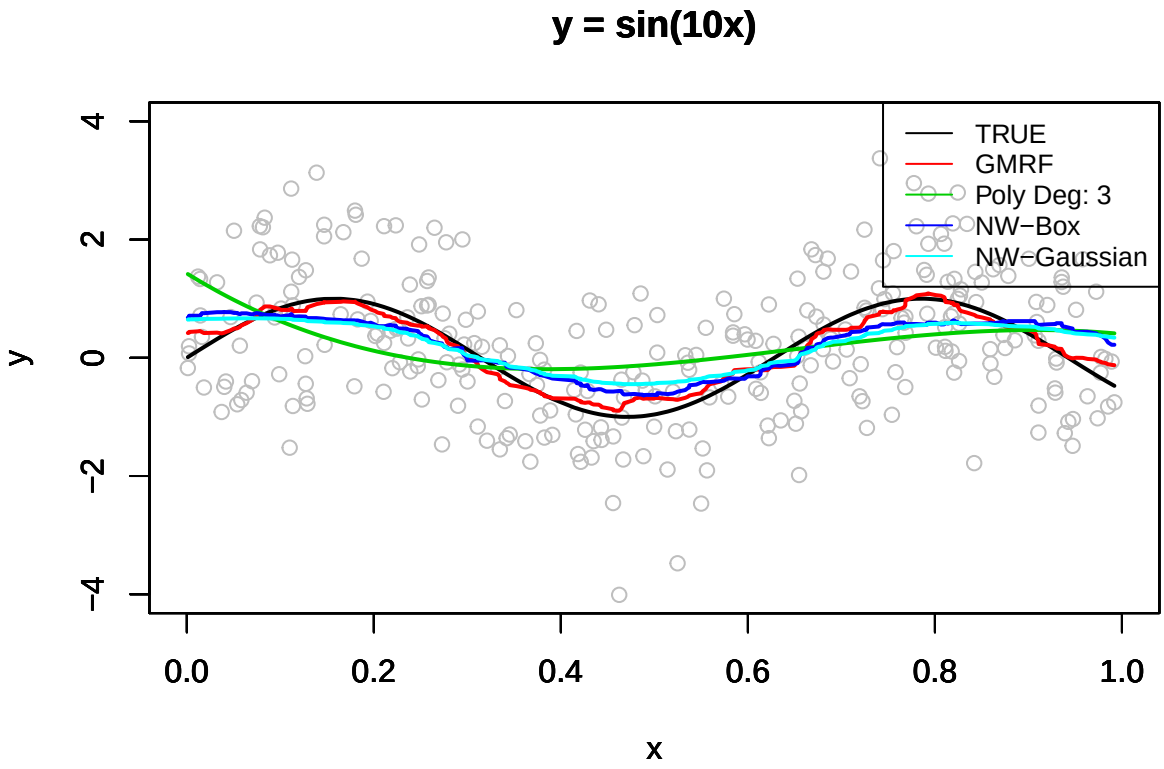
Simulation

I run the full simulation study for each of the 2 functions $f(x)$, and I calculate the MSE for $n = 100, 200, \ldots, 500$ and for each n, we report the average MSE for each n, averaged over 10 replications of the data.

My choice of bandwidth for NW estimators is motivated by the *optimal* bandwidth, which states that the optimal bandwidth should be $h_* = Cn^{-1/5}$ where the constant C depends on the function f, variance of noise, and density of x_i. In this simulation, to keep things simple, we just take $C = 1$.



As is shown, GMRF fitting performs worst in the linear scenario but appeals to have some advantages in nonlinear scenario. Actually, the idea GMRF is very similar with some well-used nonparametric estimators, like KNN and NW estimators, because they all focus on local information. The difference is GMRF always treats $\{x_i, y_i\}$ as a sequence and it seems to be more flexible with the regularization assumptions of f, which is why I consider to apply it into time series. The referenced paper shows plenty of simulation study, including smoothing functions and piecewise constant function[1]. One example of my fitting is shown as following.



Time series decomposition

I apply GMRF fitting into time series decomposition and forcasting. Firstly, I use GMRF to fit a smoothing line. I split the line by bandwidth of 12 (since I know the period is 12 month), and avarage the difference with the mean in each piece. So that I get the seasonal component. Secondly, I minus the raw data with seasonal component and do another GMRF fitting on it, which turns out to be the trend component. For the first fitting, y is Poisson distributed because it is count variable, but for the second fitting I use normal distribution.

I try three priors in first fitting: Gaussian prior, Laplace prior and Horseshoe prior. Horseshoe has 2.08% divergences, while Gaussian prior and Laplace Prior show no problems of sampling. And I compare them based on LOO-PSIS-CV (efficient approximate leave-one-out cross-validation for Bayesian models fit using Markov chain Monte Carlo, which uses Pareto smoothed importance sampling, a new procedure for regularizing importance weights.) and WAIC (a generalized version of AIC)[2].

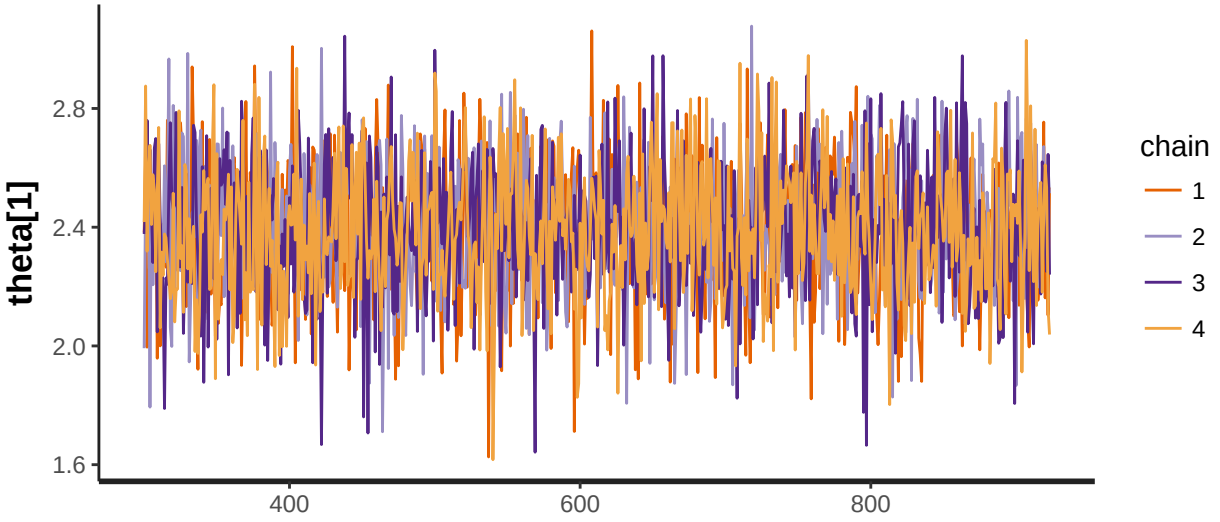
	elpd_diff	elpd_loo	se_elpd_loo	p_loo	se_p_loo	looic	se_looic
loo1	0.0000000	-514.1637	7.494875	71.21396	4.738030	1028.327	14.98975
loo2	-0.9888591	-515.1526	7.760129	72.05572	5.251740	1030.305	15.52026
loo3	-6.8918553	-521.0556	8.197306	74.44296	5.559701	1042.111	16.39461

	elpd_diff	elpd_waic	se_elpd_waic	p_waic	se_p_waic	waic	se_waic
waic1	0.000000	-500.4084	6.604535	57.45860	3.843856	1000.817	13.20907
waic2	-1.470783	-501.8792	6.832299	58.78228	4.302405	1003.758	13.66460

	elpd_diff	elpd_waic	se_elpd_waic	p_waic	se_p_waic	waic	se_waic
waic3	-7.473096	-507.8815	7.739454	61.26885	5.002189	1015.763	15.47891

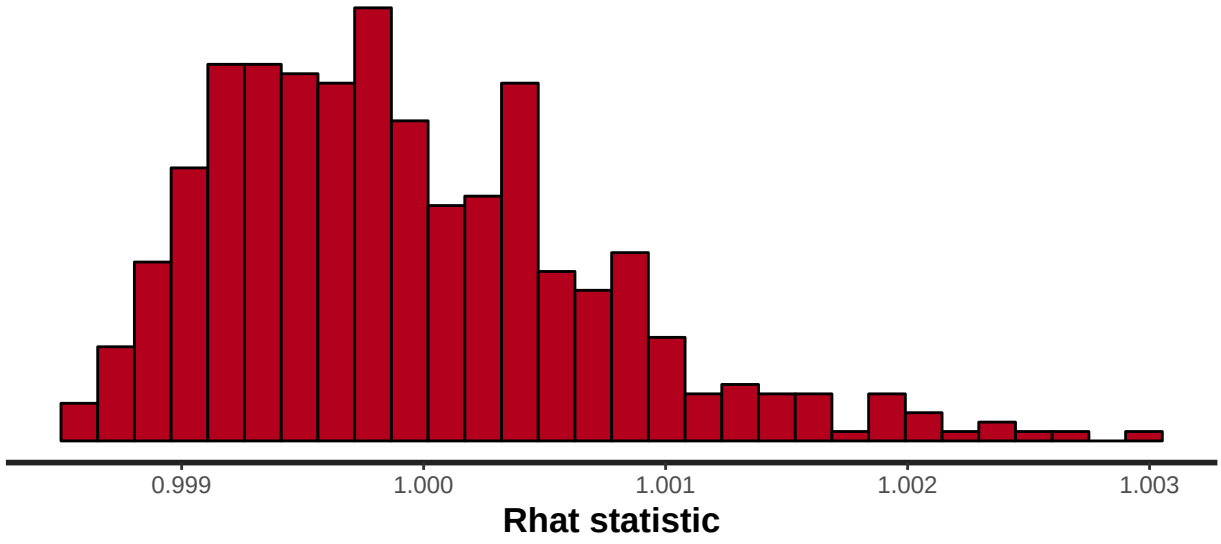
Both LOO-PSIS-CV and WAIC shows model based on Gaussian prior is the best. Thus, I focus on GMRF in the following.

The trace of $theta_1$ in the sampling can be shown as following.

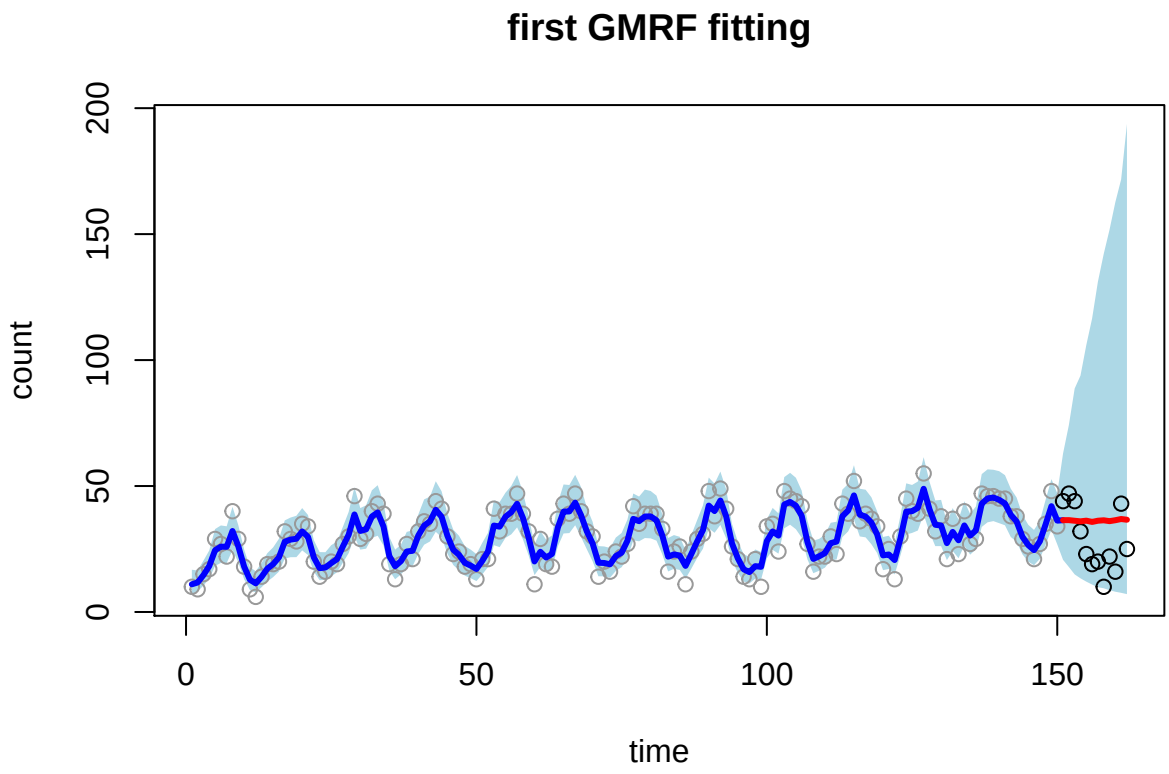


The overlapping lines show that the samples from 4 different chains are comming from one common distribution, or that, there is no violation for $theta_1$. The conclusion is also true for other parameters, because as we can see in the following, all the $\hat{R}s$ are close to 1 (The degree of convergence of a random Markov Chain can be estimated using the Gelman-Rubin convergence statistic, $\hat{R}$, based on the stability of outcomes between and within m chains of the same length, n. Values close to one indicate convergence to the underlying distribution. Values greater than 1.1 indicate inadequate convergence.)[3].

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



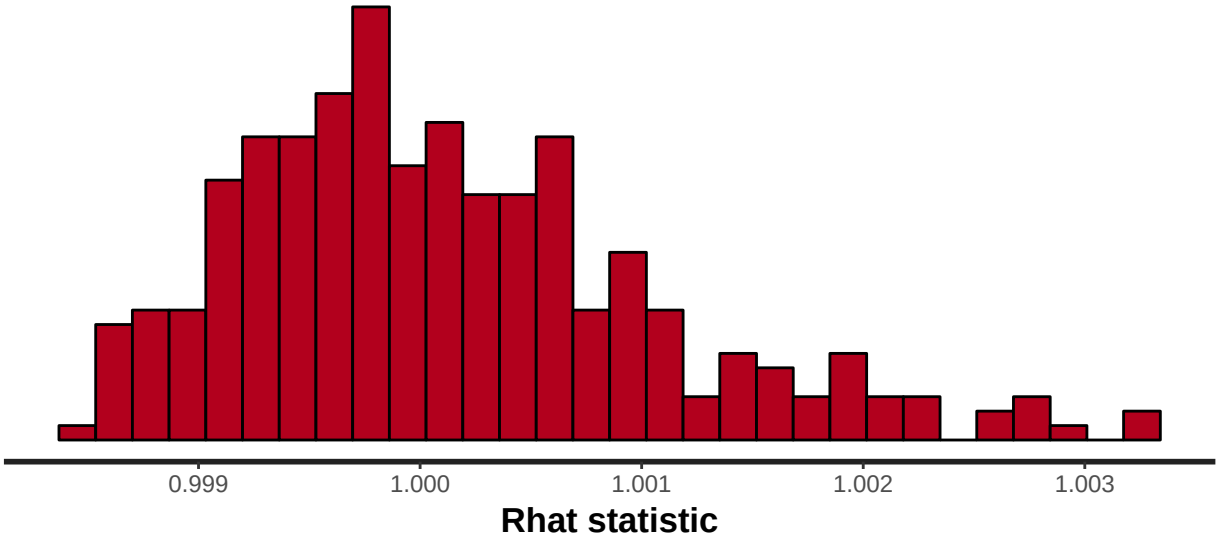
95% credible interval of θ can be shown as following.

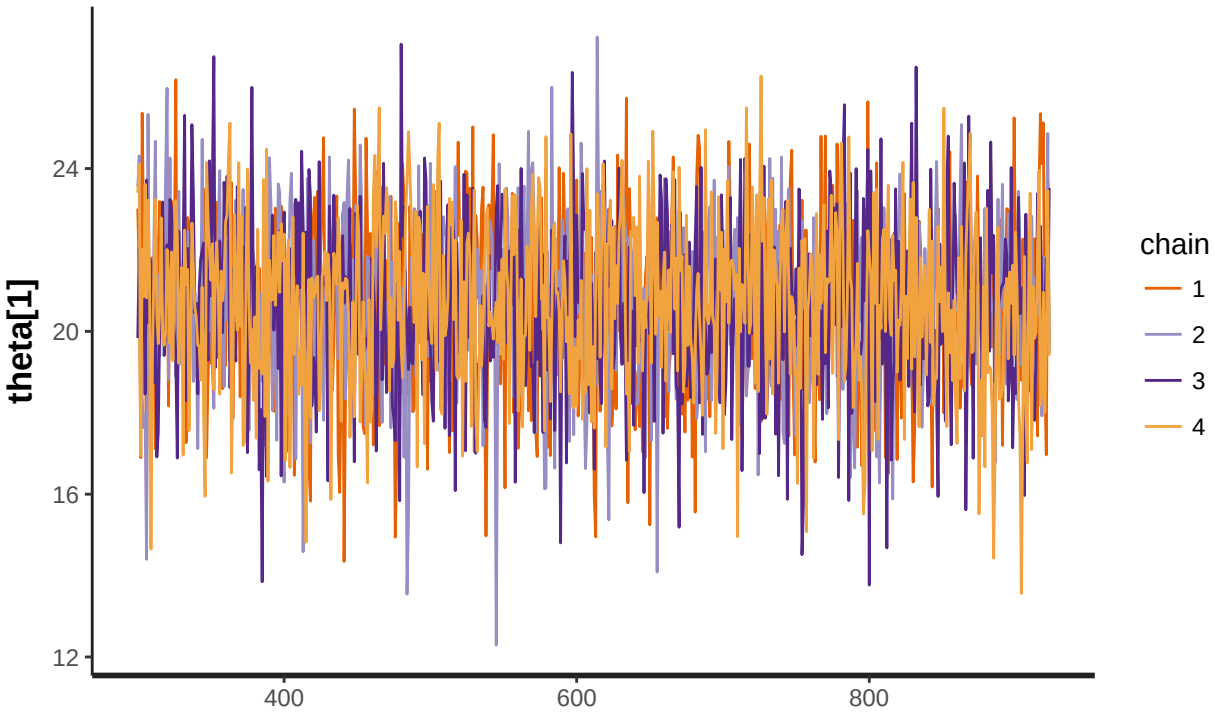


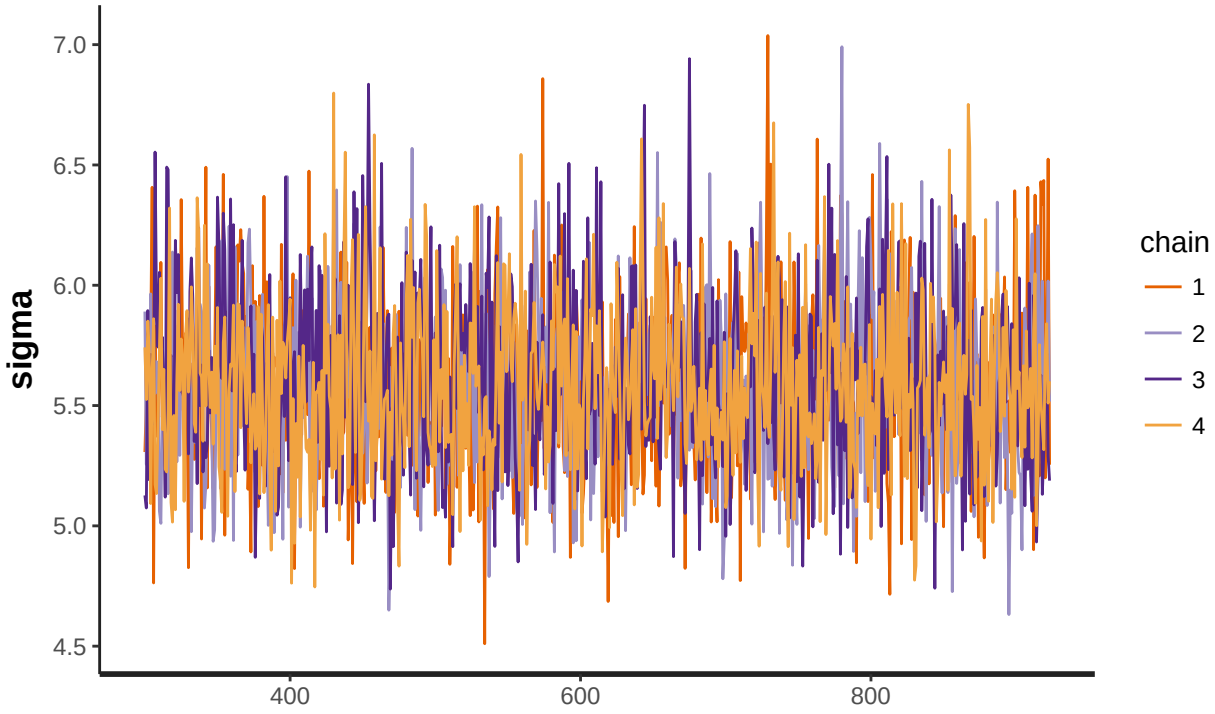
The last 12 points is simulated. As we can see, using these simulated data points to make prediction is not a wise choice. So the second fitting is necessary.

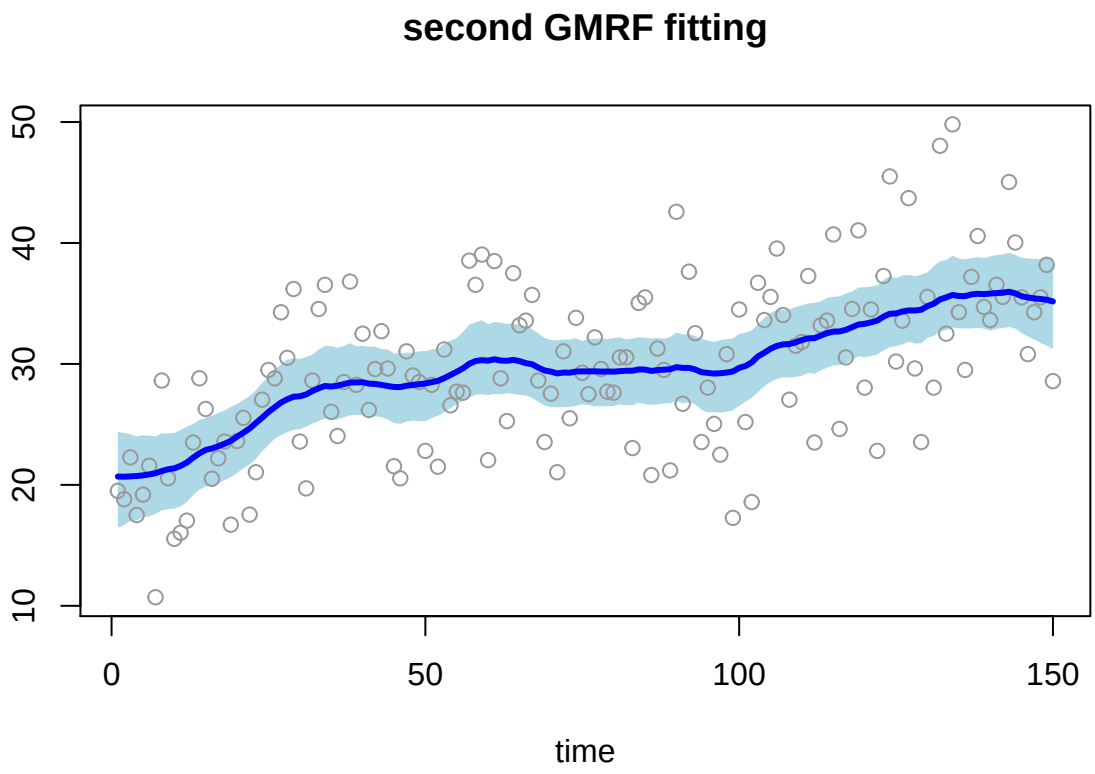
Similarly, here are the results of second fitting.

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

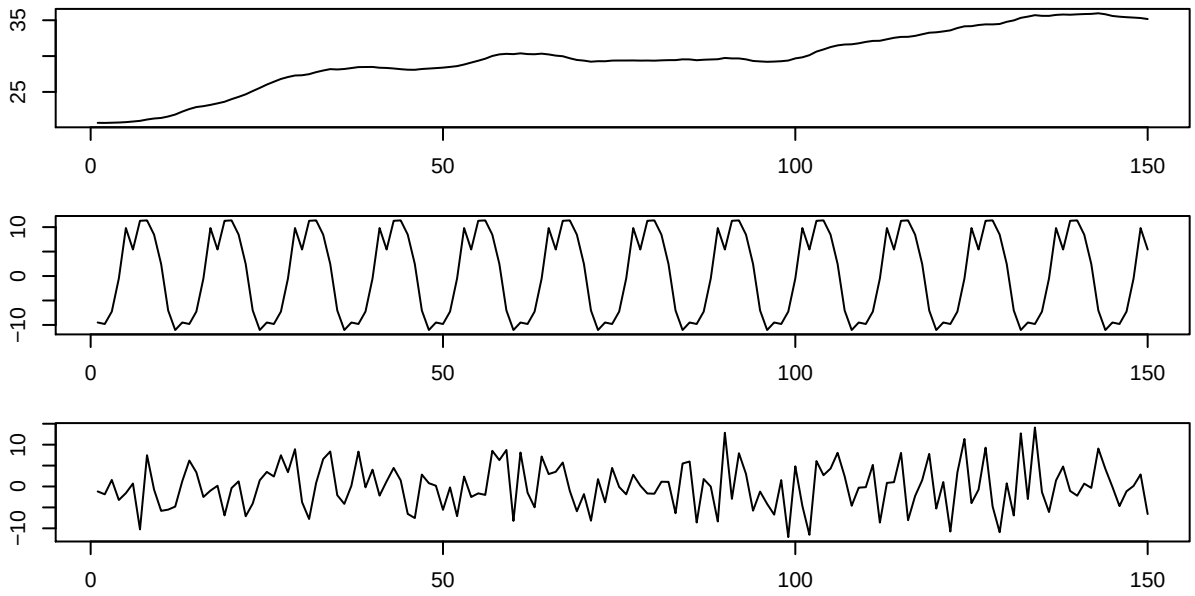



theta[1]
24
20
16
12
400
600
800
chain
1
2
3
4

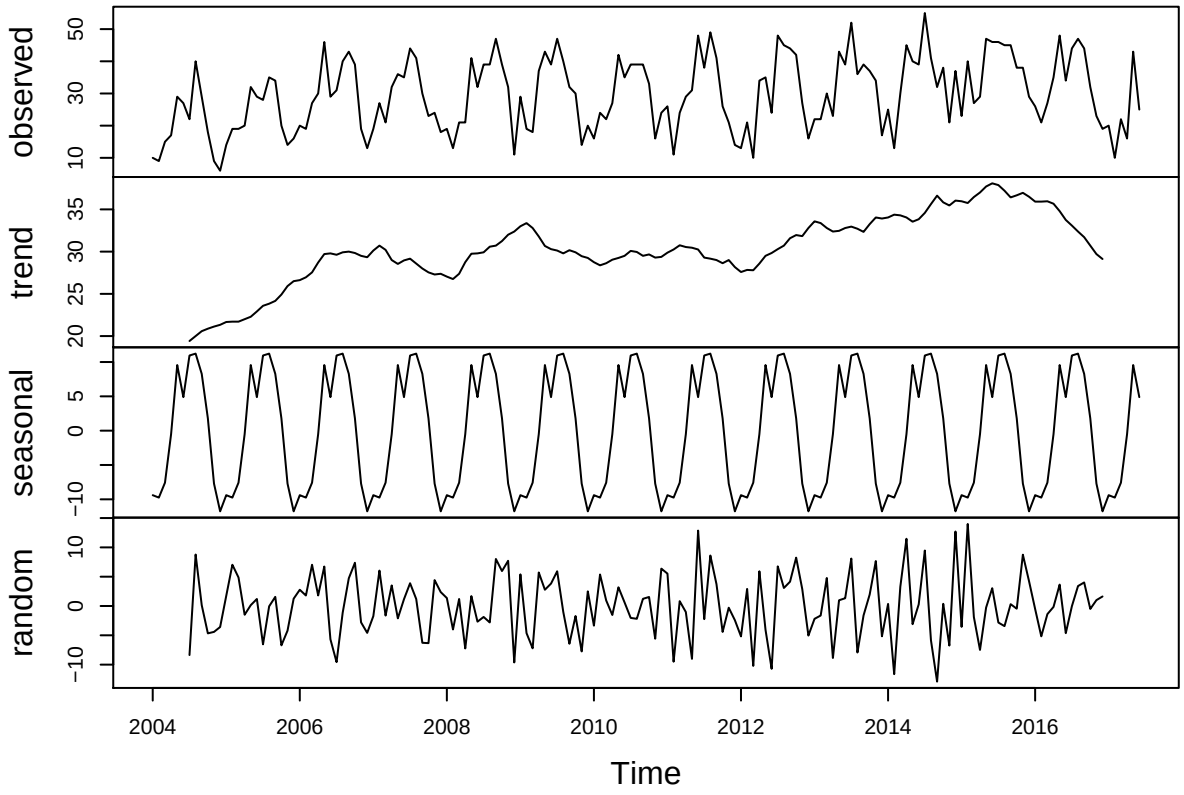

sigma
7.0
6.5
6.0
5.5
5.0
4.5
400
600
800
chain
1
2
3
4



After second fitting, my decomposition will look like the following.



As a comparison, decomposition based on ARIMA looks like the following.

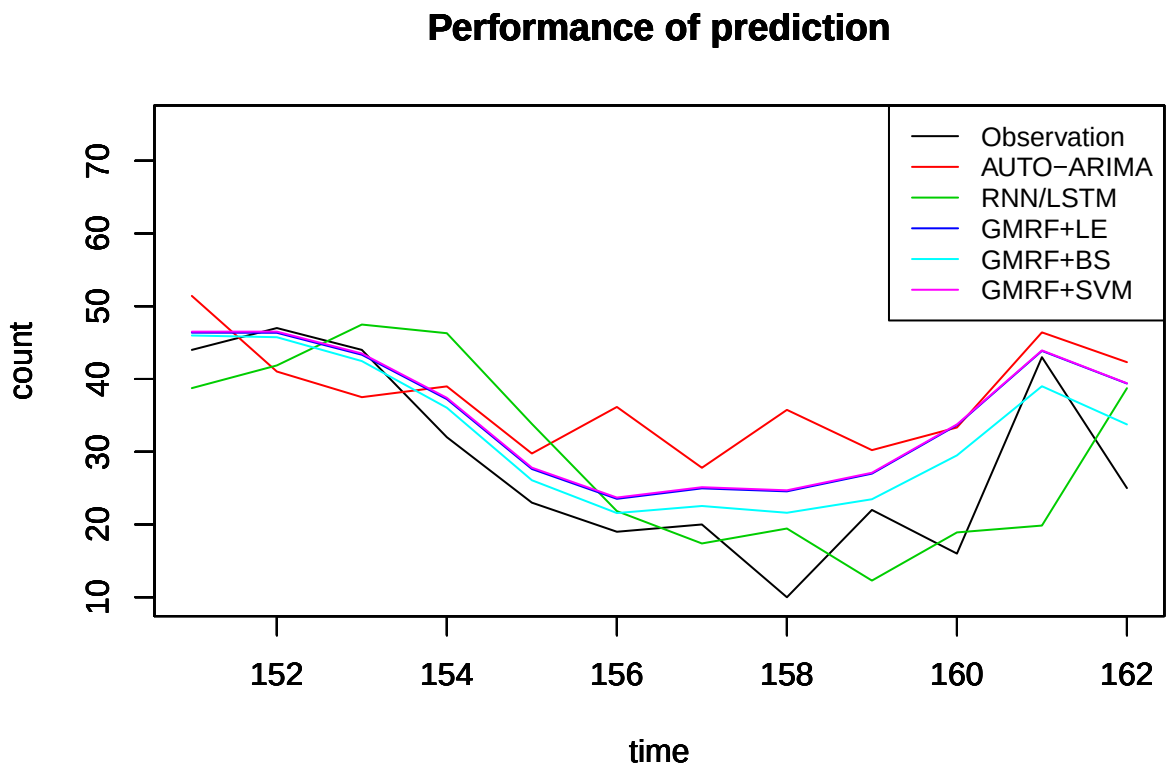


Generally speaking, the seasonal part seems to be exactly the same, but my trend line is more smoothing. I will go further and compare them with regard to forcasting the last 12 points.

Seasonal part seems to be straightforward to use in forcasting. For trend part, firstly I consider a linear extension of trend component. The linear extension is implemented by `lm` function with a step wise chosen by cross validation, that is, I use the last few points to fit a linear regression to predict the future. Secondly, I also try to fit the trend line by B-spline or SVM. And then I sum up the seasonal component with a linear extension of trend component as my prediction.

The results of the above forcasting strategy along with other model's prediction, like ARIMA and RNN/LSTM are shown as following.

Method	MSE
AUTO-ARIMS	160.4
RNN/LSTM	109.9
GMRF-Simulation	204.0
GMRF+LinearExtension	71.4
GMRF+BSpline	38.2
GMRF+SVM	72.8



As is shown above, GMRF obviously outperforms the other methods, and using B-splines to fit the trend line seems to be the best choice.

Discussion

Conclusion

As is mentioned above, GMRF is a powerful nonparametric regression method which can also be applied into time series. It has a good chance to catch various characteristic features of interests, like autocorrelation structure and periodic component. Combining GMRF and B-splines achieves really nice results, which is beyond my proposal.

In fact, priors have a shrinkage effect, which can also be seen as a kind of regularization. when it comes to MLE, ridge regression is equivalent to Gaussian prior, and Lasso regression is equivalent to Laplace prior. Under Bayesian framework, we are more flexible to choose from different kinds of priors.

When it comes to model structure, GMRF is a realization of partial-pooling model. The connections between θ_i is limited by distance, which balances local adaptation and global control. A partial-pooling model is always a good idea, bacause it is more flexible than a fully-pooling model and more controllable than a non-pooling model. It is a tradeoff of bias and variance.

Limitation

The main issue here is computational efficience. Bayesian method is time-consuming. In simulation study, for example, it takes totally about an hour to sample. And Horseshoe prior has a more extreme problem of

it than Gaussian prior and Laplace prior. Moreover, if we try funtions like $y = sin(100x)$, divergence issues raise up.

My idea is to take advantages of both fitting power of Bayesian frameworks and computational efficience of frequentist methos. firstly, we should never let a Bayesian model to deal with a too complicated function fitting. Alternatively, we may use method like B-splines with knots to do a quick pilot fitting, and then run a divide and conquer algorithm with Bayesian model. Secondly, a Bayesian method could be a correction to a frequentist method, and vice versa. Actually, `GMRF+BSpline` is an example of realization. Another idea is to leave a mangeable amount of data for Bayesian model and fit a freqentist model with the rest data. And then include the results of frequentist model as priors. Besides, A general idea of ensemble learning is also appealing to me.

Reference

[1]Faulkner, James R.; Minin, Vladimir N. Locally Adaptive Smoothing with Markov Random Fields and Shrinkage Priors. Bayesian Anal. 13 (2018), no. 1, 225–252. doi:10.1214/17-BA1050. https://projecteuclid.org/euclid.ba/1487905413

[2]Vehtari, A., Gelman, A., and Gabry, J. (2017a). Practical Bayesian model evaluation using leaveone-out cross-validation and WAIC. Statistics and Computing. 27(5), 1413–1432. doi:10.1007/s11222- 016-9696-4. (published version, arXiv preprint)

[3]Gelman, A. and D. B. Rubin (1992) Inference from iterative simulation using multiple sequences (with discussion). Statistical Science, 7:457-511