Project Report for IEOR E4721 : Detecting regime changes for financial time series data

Vaishali Devanand, UNI : vhd2107

Abstract

In this project, we have developed Python and R models to estimate the parameters of a Markov-switching model. It should be noted that there have been several studies performing similar kind of analysis, but none of the referenced papers or textbooks have a working code in R/Python to implement such a model. The methods explored in this project are Maximum Likelihood estimation and Gibbs sampling. This project also tests the model performance using a simulator and then fits the model to the data set provided by QSquared Capital. The resulting data-driven model that will assist us in making decisions without the knowledge of the underlying asset. The applications are wide from tactical allocation to blocking trades when high volatility regimes are predicted.

Contents

List of Figures

List of Tables

1 Introduction

Compared to commonly used software for modeling Markov switching models such as GAUSS or MATLAB which can be expensive, Python and R are open source software packages which can be used free of charge. Given the extensive implementation of the estimation methods from scratch in Python and R, it is worth noting that this project is very much a work in progress given the short duration. We have attempted to compile the results and working codes that ties to the overall flow of the project. Section [2] discusses the past works in the academic literature that studies parameter estimation methods for Markov-switching auto-regressive models - mainly Maximum Likelihood and Gibbs sampling. Section [3] explains the steps and mathematical derivations in the algorithm used for developing estimation methods and for generating simulated data to test the methods. Section [4] summarizes the results from applying the model to real data.

2 Literature Review

Hamilton paper (2) proposes an innovative approach to Maximum Likelihood fitting and Nyberg et al (3) predicts whether the stock market is in bull or bear regimes using a binary dynamic time-series model. In this paper, only the probability of the state is predicted and Markov-switching is not used to incorporate the information from the last period but has used AR model to regress on the last period. We can test whether this method will work on our data. Albert and Chibb (9) proposes a method for sampling from the conditional distribution $s|\theta, \Omega_T$.Liu and Chen et al (7) uses threshold factor model for high-dimensional time-series and Osmundsen et al (6) proposes a Markov switching vector autoregression method that combines maximum likeliood estimation and Markov chain Monte Carlo methods like Hamilton MCMC and Gibbs sampling while Gao et al (5) argues that long memory and regime switching characteristics of the financial time-series data are interchangeable. Piger et al (4) focuses on other advanced methods listed below to build regime-switching models.

1. Markov-switching vector auto-regression (MS-VAR) by Hamilton (1994). This model can incorporate the case of N greater 2 regimes, as well as allow y to be a vector of random variables
2. Markov-switching models with time-varying transition probabilities (TVTP) by Filardo (1994).The transition probabilities in this model are allowed to vary depending on conditioning information.

3 Exploratory Data Analysis

Pickle file contains daily close prices of assets between Jan-1990 and Dec-2019 which is labeled as the low frequency data in the directory while the five second intra-day tick data is labeled as high-frequency data. In all of the asset classes, we observe breaks in time-series and there are also methods employed to study the discontinuity in time-series. Since the sampling frequency in the provided data is not uniform, before performing any kind of analysis it is necessary to detect the discontinuity in time-series and impute missing values. The idea behind exploratory data analysis is that simply by visualizing data, we can find a starting pointing for the regime switching model. PACF plot of asset returns points to auto-correlation of order 1 in the data.

3.1 Daily prices

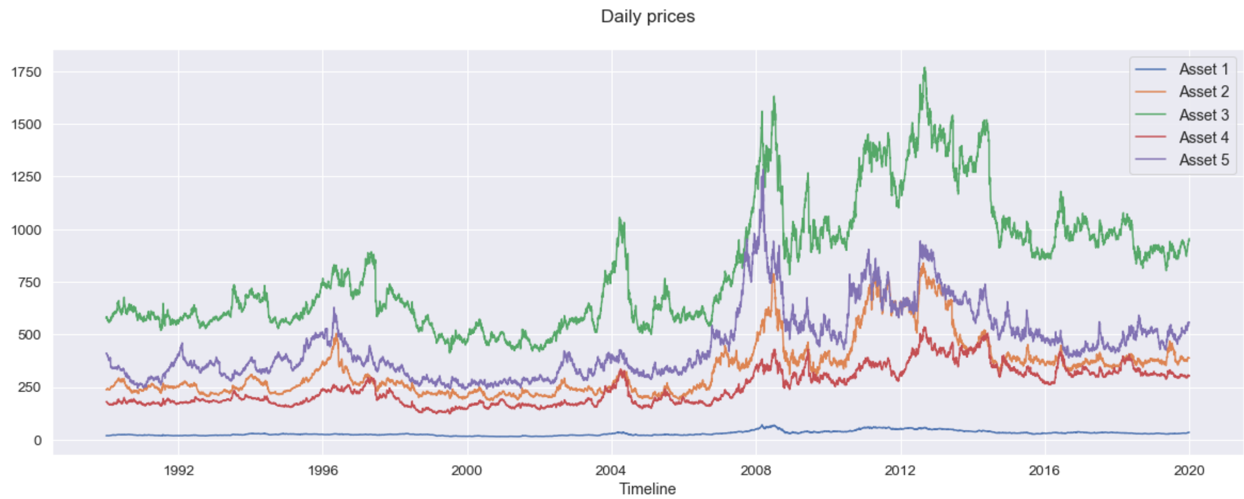


Figure 1: Daily Asset Prices

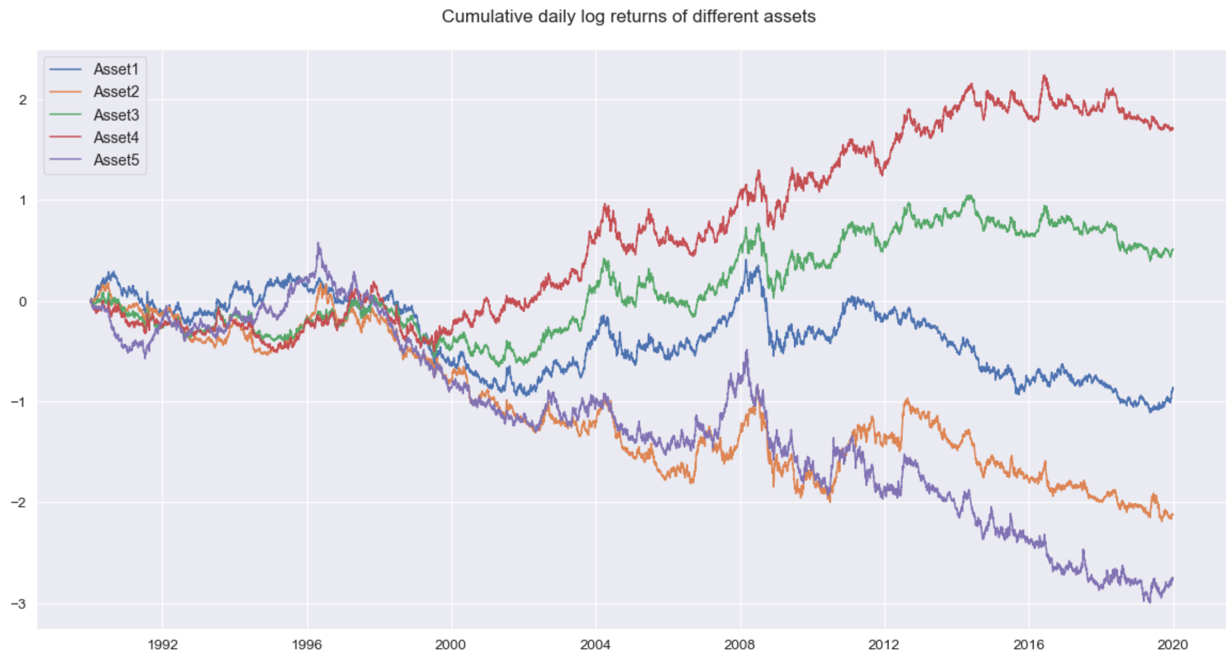


Figure 2: Cumulative log returns

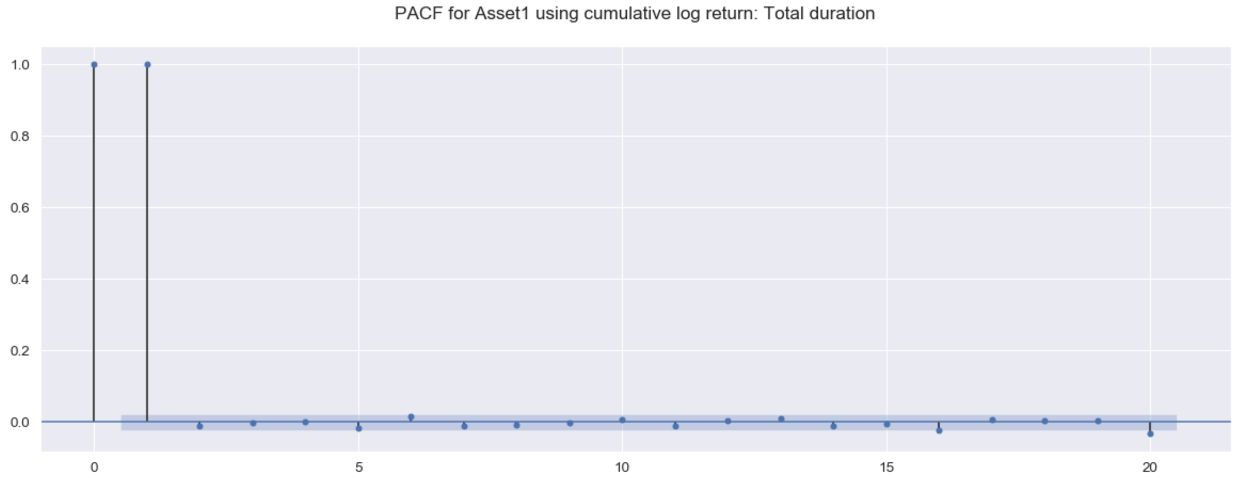


Figure 3: ACF of Asset 1 daily returns

3.2 Intra-day data

Figure 4: High frequency tick data

4 Methodology

4.1 Generalized Model

In the following sections, we will explore multiple variants of the generalized model to see which best describes the data. The variants explored in the study choose from different number of regimes and auto-regressive lag order.

$$\left.\begin{aligned} y_t &= \phi_1 y_{t-1} + \phi_2 y_{t-2} + ...\phi_r y_{t-r} + \mu_{s_t} + \sigma_{s_t}\epsilon_t \\ \epsilon_t &\sim N(0,1) \\ s_t &\in 1,2,.....,M \\ p_{ij} &= P(s_t = j|s_{t-1} = i) \\ \sigma^2_{s_t} &= \sigma^2_j \text{ if } \mathrm{s}_t = j \ \forall j = 1,....,M \\ \mu_{s_t} &= \mu_j \text{ if } \mathrm{s}_t = j \ \forall j = 1,...,M \\ b_j &= y_j \end{aligned}\right\} \tag{1}$$

There are existing packages that can estimate the MSAR parameters in Python and R. Specifically, for Python - sm.tsa.MarkovAutoregression and for R- MSwM. To test the effectiveness of the packages, we built a simulator to generate a sequence of Markov Switching time series. We then use the simulator to test if the Python and R scripts produce estimates close to the true parameters of the sample.

4.2 Parameter estimation methods

4.2.1 Maximum Likelihood Estimation

The goal of the MLE algorithm is to find the optimal parameters that maximizes the log-likelihood function below:

$$LL(\theta) = \sum_{t=1}^{T} log[f(y_t|\Omega_{t-1};\theta)] \tag{2}$$

RHS of the log-likelihood equation is computed using the conditional distribution of the observation given the state variable as below:

$$f_i = f(y_t|s_t = i, \Omega_{t-1};\theta) = (1/\sigma_{s_t}\sqrt{2\pi})e^{-((y_t-\alpha_{s_t})/\sigma_{s_t})^2} \tag{3}$$

Now, the probability of the hidden state variable given Ω_{t-1} is appended as below,

$$Pr(s_t = i|\Omega_{t-1}) = f_i/\sum_{i=1}^{N} f_i \tag{4}$$

The final step is to compute filtered probabilities using the equation below:

$$P(S_t = j|\Omega_t) = \sum_{i=1}^{N} p_{ij}P(S_{t-1} = i|\Omega_{t-1}) \tag{5}$$

since

$$Pr(s_t = i|\Omega_{t-1};\theta) = Pr(s_{t-1} = i) * p_{ii} + Pr(s_{t-1} = j) * p_{ji} \tag{6}$$

The above steps are performed iteratively for $t = 2, 3, ..., T$ after assigning initial parameters including transition probabilities p_{ij}

Since we assume the states follow a time-invariant Markov chain, the unconditional probability of initial state is assumed to be

$$Pr(s_0 = i) = (1 - p_{jj})/(2 - p_{ii} - p_{jj}) \tag{7}$$

4.2.2 Gibbs Sampling

Estimation via Gibbs sampling is more complicated than Maximum likelihood since latent variables are treated as parameters in the MCMC estimation

In the simple model, we have a parameter set and some data y. Our goal is to find the posterior distribution of $\theta|y$. Before we can compute the joint posterior distribution, we need to come up with conditional distributions.

The methodology for Gibbs sampling is as follows:

1. Use Hamilton's filtering procedure for Maximum Likelihood approach to obtain $P(S_T|\Omega_T)$
2. To generate transition probabilities, we choose beta distribution as the conjugate prior since the values lie in the interval [0,1]
3. To generate mean values, we choose Normal distribution as the conjugate prior because since this is a conditional distribution with a known variance.
4. To generate variances, we use inverted Gamma distribution as the conjugate prior
5. Once Step 1 to Step 4 are completed for a cycle, we update the counter and restart from Step 1

In Bayesian approach, both θ and s_t is considered to be random variables and the corresponding conditional distributions is used to draw samples of model parameters.

5 Results and Analysis

5.1 Standard Packages

We first identified available packages in python that can implement Markov-switching models. There were three such packages - Hamilton (1989) filter, Kim (1994) smoother and Time-Varying Transition Probability model by Filardo and Gordon, with slight variations from one another. We then tried replicating the results from seminal Hamilton(1989) (1) switching model of GNP by using the standard package.

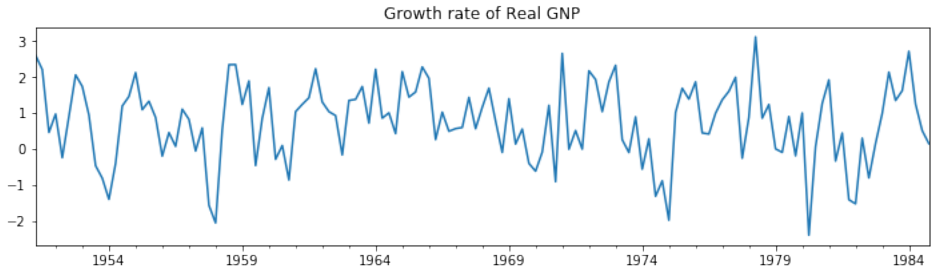
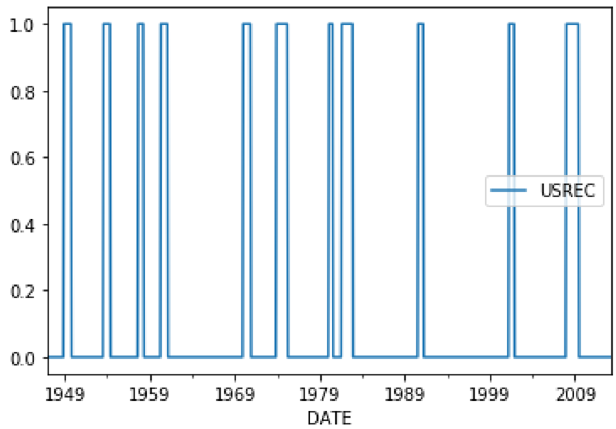
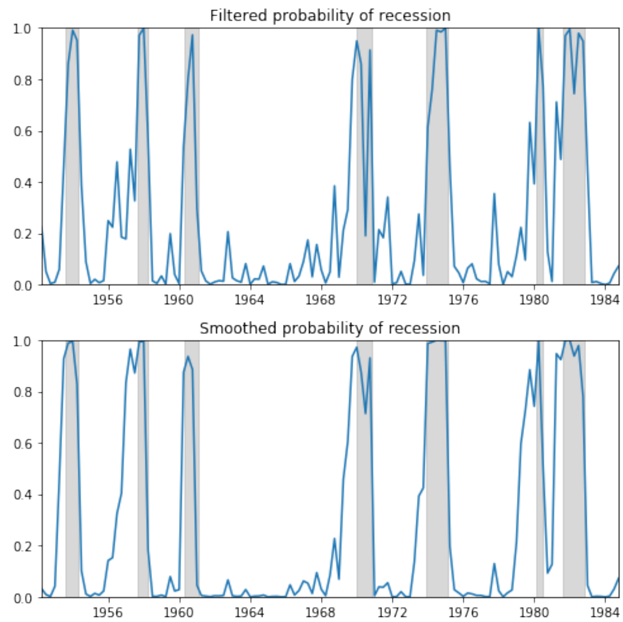


Figure 5: Hamilton GNP Data



(a) US Recession



(b) Filtered and Smoothed Probabilities

Figure 6: Hamilton GNP regimes

```
                         Markov Switching Model Results
================================================================================
Dep. Variable:                        y   No. Observations:                  131
Model:             MarkovAutoregression   Log Likelihood                -181.263
Date:                  Sun, 30 Aug 2020   AIC                            380.527
Time:                          14:03:33   BIC                            406.404
Sample:                      04-01-1951   HQIC                           391.042
                           - 10-01-1984
Covariance Type:                 approx
                             Regime 0 parameters
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.3588      0.265     -1.356      0.175      -0.877       0.160
                             Regime 1 parameters
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
const          1.1635      0.075     15.614      0.000       1.017       1.310
                           Non-switching parameters
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
sigma2         0.5914      0.103      5.761      0.000       0.390       0.793
ar.L1          0.0135      0.120      0.112      0.911      -0.222       0.249
ar.L2         -0.0575      0.138     -0.418      0.676      -0.327       0.212
ar.L3         -0.2470      0.107     -2.310      0.021      -0.457      -0.037
ar.L4         -0.2129      0.111     -1.926      0.054      -0.430       0.004
                         Regime transition parameters
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
p[0->0]        0.7547      0.097      7.819      0.000       0.565       0.944
p[1->0]        0.0959      0.038      2.542      0.011       0.022       0.170
```

Figure 7: Estimation of Hamilton GNP regimes

These models assume a time-invariant Markov chain for hidden states for identifying regimes but yields inaccurate results on the daily returns data. Due to the huge volume of data available for all assets, we will take Asset 1 as an example and explain the results which can be extended to other assets as well as seen from the code.

5.1.1 Asset 1 : 2007 - 2008

We can see the standard package is not effective when it comes to identifying regimes in the returns time series since it produces NaN values and outputs transition probability of 0.5.

Figure 8: Asset1: 2007 to 2008 returns

Figure 9: Asset1 : 2007 returns

Markov Switching Model Results

Dep. Variable:	cum_log_return	**No. Observations:**	500
Model:	MarkovAutoregression	**Log Likelihood**	1248.834
Date:	Wed, 02 Sep 2020	**AIC**	-2485.668
Time:	11:32:40	**BIC**	-2460.38
Sample:	0	**HQIC**	-2475.745
	-500		
Covariance Type:	approx		

Regime 0 parameters

	coef	std err	z	P>|z|	[0.025	0.975]
const	-0.0912	0.285	-0.32	0.749	-0.65	0.467

Regime 1 parameters

	coef	std err	z	P>|z|	[0.025	0.975]
const	-0.0912	0.285	-0.32	0.749	-0.65	0.467

Non-switching parameters

	coef	std err	z	P>|z|	[0.025	0.975]
sigma2	0.0004	2.51E-05	15.811	0	0	0
ar.L1	0.9969	0.004	271.39	0	0.99	1.004

Regime transition parameters

	coef	std err	z	P>|z|	[0.025	0.975]
p[0->0]	0.5	5.01E-09	9.98E+07	0	0.5	0.5
p[1->0]	0.5	5.84E-09	8.56E+07	0	0.5	0.5

Figure 10: Asset 1: 2007 - 08 : Estimation results, No switching coefficients

Dep. Variable:	cum_log_return	**No. Observations:**	500
Model:	MarkovAutoregression	**Log Likelihood**	1248.834
Date:	Wed, 02 Sep 2020	**AIC**	-2483.668
Time:	11:34:16	**BIC**	-2454.165
Sample:	0	**HQIC**	-2472.091
	-500		
Covariance Type:	approx		

Regime 0 parameters

	coef	std err	z	P>|z|	[0.025	0.975]
const	0.2038	6.80E-29	3.00E+27	0	0.204	0.204
ar.L1	0.7042	nan	nan	nan	nan	nar

Regime 1 parameters

	coef	std err	z	P>|z|	[0.025	0.975]
const	-0.0912	0.285	-0.32	0.749	-0.65	0.467
ar.L1	0.9969	0.004	271.39	0	0.99	1.004

Non-switching parameters

	coef	std err	z	P>|z|	[0.025	0.975]
sigma2	0.0004	2.51E-05	15.811	0	0	C

Regime transition parameters

	coef	std err	z	P>|z|	[0.025	0.975]
p[0->0]	0.9094	nan	nan	nan	nan	nar
p[1->0]	4.18E-51	nan	nan	nan	nan	nar

Figure 11: Asset 1: 2007 - 08 : Estimation results, With switching coefficients

5.1.2 Asset 1 : 2007

Dep. Variable:	cum_log_return	**No. Observations:**	248
Model:	MarkovAutoregression	**Log Likelihood**	nan
Date:	Wed, 02 Sep 2020	**AIC**	nan
Time:	11:56:59	**BIC**	nan
Sample:	0	**HQIC**	nan
	-248		
Covariance Type:	approx		

Regime 0 parameters

	coef	std err	z	P>|z|	[0.025	0.975]
const	-0.2007	nan	nan	nan	nan	nan

Regime 1 parameters

	coef	std err	z	P>|z|	[0.025	0.975]
const	-0.1993	nan	nan	nan	nan	nan

Non-switching parameters

	coef	std err	z	P>|z|	[0.025	0.975]
sigma2	0.0001	nan	nan	nan	nan	nan
ar.L1	nan	nan	nan	nan	nan	nan

Regime transition parameters

	coef	std err	z	P>|z|	[0.025	0.975]
p[0->0]	0.499	nan	nan	nan	nan	nan
p[1->0]	0.5022	nan	nan	nan	nan	nan

Figure 12: Asset 1 : 2007 : Estimation returns, No switching coefficients

Dep. Variable:	cum_log_return	**No. Observations:**	248
Model:	MarkovAutoregression	**Log Likelihood**	nan
Date:	Wed, 02 Sep 2020	**AIC**	nan
Time:	11:58:01	**BIC**	nan
Sample:	0	**HQIC**	nan
	-248		
Covariance Type:	approx		

Regime 0 parameters

	coef	std err	z	P>|z|	[0.025	0.975]
const	nan	nan	nan	nan	nan	nan
ar.L1	nan	nan	nan	nan	nan	nan

Regime 1 parameters

	coef	std err	z	P>|z|	[0.025	0.975]
const	nan	nan	nan	nan	nan	nan
ar.L1	nan	nan	nan	nan	nan	nan

Non-switching parameters

	coef	std err	z	P>|z|	[0.025	0.975]
sigma2	nan	nan	nan	nan	nan	nan

Figure 13: Asset 1: 2007 : Estimation results, With switching coefficients

Another major drawback of using the Python support for Markov-switching models is that there is no option for switching variance. So, in the next section, we try to use a custom built model that can plot the smoothed probabilities of the given observation being in regime 1 or regime 2.

5.2 Maximum Likelihood Estimation

We test the Python script containing Maximum Likelihood Estimation method on the simulated data against the MsWM package in R.

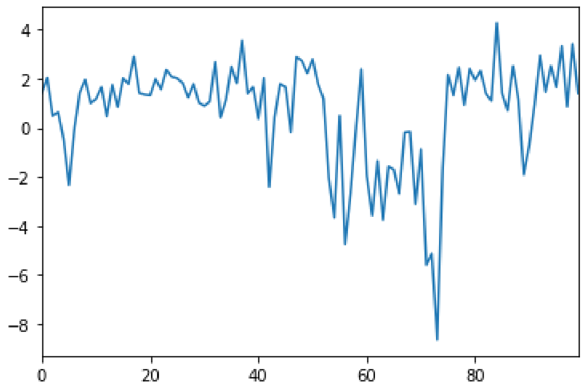


Figure 14: Simulated markov-switching Time Series

Table 1: Model comparison - Part 1

Parameters	True values	R model	Python MLE model
a1	2.0000	1.7100	1.7114
a2	-2.0000	-2.000	-2.0061
sigma1	1.0000	0.9380	0.8349
sigma2	2.0000	2.000	2.2150
p11	0.9500	0.9380	0.9401
p22	0.8500	0.8490	0.8412

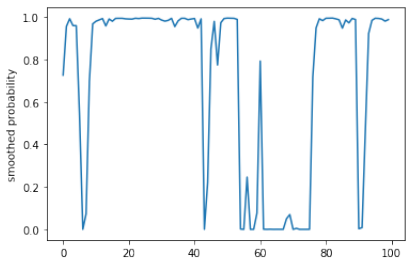


Figure 15: Smoothed Probability

The smoothed probability generated from the model resembles switching sequence in the simulated data but the model does not work for certain starting points as it does not successfully identify the optimal parameters that maximizes the likelihood function and could possibly not terminate successfully in other cases. For example if the starting point is [1,1,0.1,0.1,0.1,0.1], then the resulting estimate is [1. , 1. , 0.1, 0.1, 0.1, 0.1] which is not accurate. We have built a way around this in Gibbs estimation approach by using a bootstrapping method that uses Expectation Maximization algorithm to input viable starting points into the gibbs sampler.

On applying this model, to daily returns data by setting the order and switching parameters, we obtain the following results

For Model 1,

$$y_t = a_{s_t} + \epsilon \tag{8}$$

with only constant switching and no AR coefficient, there is no obvious regime switching as show in Figure 16:

For Model 2,

$$y_t = a_{s_t} + \phi_1 y_{t-1} + \epsilon \tag{9}$$

which is an AR(1) model with only constant switching, again there is no obvious regime switching as show in the Figure 17.

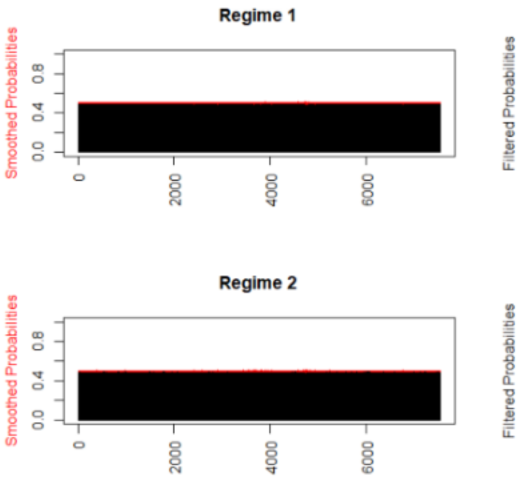


Figure 16: No AR coefficient and only constant switching

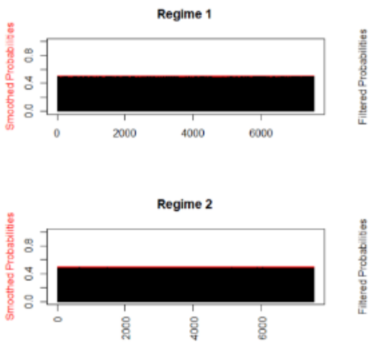


Figure 17: AR(1) model with only constant switching

But on increasing the order of this model, compiler throws an error related to Hessian matrix which is yet to be resolved.

For Model 3,

$$y_t = a_{s_t} + \phi_1 y_{t-1} + \epsilon \tag{10}$$

which is an AR(1) model with only coefficients switching. As shown in Figure 18, we can identify regimes at specific times.

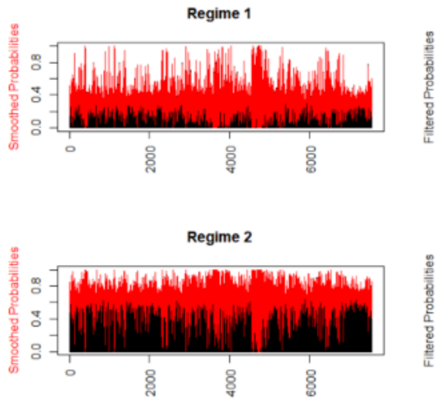


Figure 18: AR(1) model with only coefficients switching

5.3 Gibbs sampling

In the first version of the report, Gibbs source code available in R with detailed description was tested by calling discrete functions, but the model fitting doesn't work with the available data perhaps due to compatibility issues. It was noted that we will need additional time to test this package.

In the second version of the report, the bug was fixed and following results were produced.

Table 2: Model comparison - Part 2

Parameters	True values	MsWM package	Gibbs model
a1	2.000	1.7100	1.9029
a2	-2.000	-2.000	-2.055
sigma1	1.000	0.938	1.3429
sigma2	2.000	2.000	1.0167
p11	0.9500	0.9380	0.7870
p22	0.8500	0.8490	0.7871

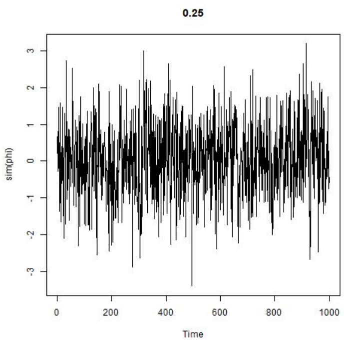
Since the simulator for Markov model for zero lag has already been tried, we will add one autoregressive coefficients in the new simulator.

This simulator was built in R to visualize the impact of switching first order lag in a simple autoregressive model with zero mean. The model is as below:

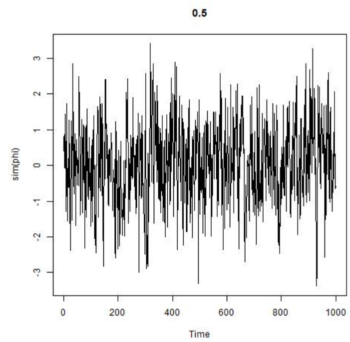
$$y_t = \phi y_{t-1} + \epsilon \tag{11}$$

where

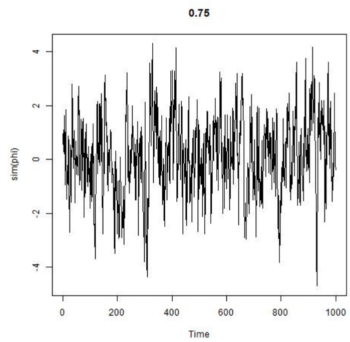
$$\epsilon \sim N(0, 1)$$



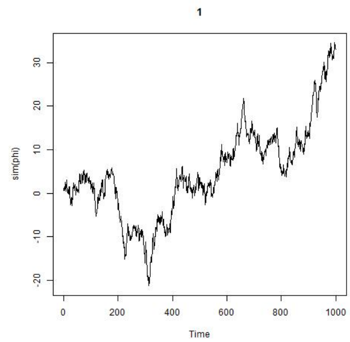
(a) $\phi = 0.25$



(b) $\phi = 0.50$



(c) $\phi = 0.75$



(d) $\phi = 1.0$

Figure 19: Visualizing the variation in first order lag ϕ

6 Conclusions

The estimation methods explored in this project works well for the simulated data. But when it comes to applying the model on real data, there are a few challenges that makes the results of the model questionable.

The first challenge is that we do not know about the data generating process and the states may not be realized unlike in the simulated process.

The second challenge to accurately estimating the model parameters is if there is relatively fewer transition among the regimes. As seen in the figures, only in chaotic periods with high volatility, our estimate is a robust and efficient predictor of the high volatility regimes.

The third challenge is the unreliable parameter estimation due to limited number of observations. This can be improved by using a high frequency time-series data.

Finally, we have shown how Markov switching models can be used to analyze financial assets when unexpected movements characterize their returns either due to changes governing the assets behavior or due to systemic changes.

7 Future work

We can further evaluate the performance of models by using goodness of fit of the model on our data through analyzing standard residuals. It is highly likely that there are outliers in the data that could result in the failure of a normality tests like Jacques Bara or Lilliefors tests . But removing some outliers may let us not reject the normality test at a desired significance level.

A further extension to the model may be to assess effectiveness of different parameter estimation methods by comparing the estimated variance from each method with VIX [aka Chicago Board Options Exchange Market Volatility Index] which is a commonly used measure of the implied volatility of the S&P500 index. In this project we use a linear autoregressive time-series model as the underlying for markov-switching. But non-linear models like Smooth transitioning Autoregressive Models, Threshold auto-regressive models, Bilinear models can be used in place of the linear model for forecasting and pattern recognition in financial data.

8 Acknowledgement

I sincerely would like to thank Mingwei Zou and Guillaume Delorme - my team members for sharing the project load and working together in preparing weekly reports, Kamyar Neshvadian and Bochen Liu for answering our questions regarding the direction of the project and Prof. Hirsa and Weilong Fu for providing alternative ideas at times when the standard methods did not work as expected.

References

[1] Hamilton, J. D. (2016). Regime Switching Models. In The New Palgrave Dictionary of Economics (pp. 1–7). Palgrave Macmillan UK. https://doi.org/10.1057/978-1-349-95121-5_2459-1

[2] Hamilton, J. (1989). A New Approach to the Economic Analysis of Nonstationary Time Series and the Business Cycle. Econometrica, 57(2). 357-384. doi:10.2307/1912559

[3] Nyberg, H. (2013). Predicting bear and bull stock markets with dynamic binary time series models. Journal of Banking Finance, 37(9), 3351–3363. https://doi.org/10.1016/j.jbankfin.2013.05.008

[4] Piger, J. (2011). Econometrics: Models of Regime Changes. In R. A. Meyers (Ed.), Complex Systems in Finance and Econometrics (pp. 190–202). Springer New York. https://doi.org/10.1007/978-1-4419-7701-4_10

[5] Gao, G., Ho, K.-Y., Shi, Y. (2020). Long memory or regime switching in volatility? Evidence from high-frequency returns on the U.S. stock indices. Pacific-Basin Finance Journal, 61, 101059. https://doi.org/10.1016/j.pacfin.2018.08.013

[6] Osmundsen, K. K., Kleppe, T. S., Oglend, A. (2019). MCMC for Markov-switching models—Gibbs sampling vs. Marginalized likelihood. Communications in Statistics - Simulation and Computation, 1–22. https://doi.org/10.1080/03610918.2019.1565580

[7] Liu, X., Chen, R. (2020). Threshold factor models for high-dimensional time series. Annals Issue in Honor of George Tiao: Statistical Learning for Dependent Data, 216(1), 53–70. https://doi.org/10.1016/j.jeconom.2020.01.005

[8] Sims, C. A., Zha, T. (2006). Were There Regime Switches in U.S. Monetary Policy? American Economic Review, 96(1), 54–81. https://doi.org/10.1257/000282806776157678

[9] Albert, J. H., Chib, S. (1993). Bayes Inference via Gibbs Sampling of Autoregressive Time Series Subject to Markov Mean and Variance Shifts. Journal of Business Economic Statistics, 11(1), 1–15. https://doi.org/10.1080/07350015.1993.10509929

[10] Kim, C.-J., Nelson, C. R. (1999). State-Space Models with Regime Switching: Classical and Gibbs-Sampling Approaches with Applications. The MIT Press.

[11] Persio, L. D., Frigo, M. (2015). Maximum likelihood approach to Markov switching models. 12, 4.

[12] Di Persio, L., Frigo, M. (2016). Gibbs sampling approach to regime switching analysis of financial time series. Journal of Computational and Applied Mathematics, 300, 43–55. https://doi.org/10.1016/j.cam.2015.12.010

A Links

A.1 Instructions to run the models

The submission folder has Python and R scripts containing the following models : Maximum Likelihood Estimation, Parameter Estimation using Gibbs Sampling. Functions to generate all the simulated data to test the models are also included in the files To test the three models separately in the order mentioned above, run the files in the following order:

1. Jupyter Notebook - MaximumLikelihood Estimation.ipynb
2. R script - MSAR Sampler and MsWM.R
3. R script - Custom built Gibbs.R

A.2 Link to Google Drive

https://drive.google.com/drive/u/1/folders/1ds8pY5hRIFA8S1qmFlBpArv5_gmBa6Nv