Data Warehouse and Data Mining Project Report on

Markov Regime Switching GARCH

Shravan Karthik (12IT77)

Bhuvan M S (12IT16)

Nitin Jamadagni (12IT47)

Under the Guidance of,

Prof. Biju R Mohan

Department of Information Technology, NITK Surathkal

Date of Submission: 21 April 2016

in partial fulfillment for the award of the degree

of

Bachelor of Technology In **Information Technology** At

Department of Information Technology

National Institute of Technology Karnataka, Surathkal

April 2016

Contents

List of Figures

List of Tables

1 Introduction and Background

1.1 Volatility

Volatility is defined as the first difference of a series. The value of financial instruments depends on the expected volatility (covariance structure) of returns. Volatility is analogous to risk: financial institutions undertake volatility assessments as part of their risk analysis exercise. An important objective of research in time series analysis is to test hypotheses and estimate relationship amongst such variables. However, inferences drawn from a stationary analysis would not be valid if the series being examined are indeed realizations of non stationary processes. Similarly, an efficient analysis of a volatile time series cannot be accomplished without addressing the issue of time varying volatility.

1.2 GARCH

If an autoregressive moving average model (ARMA model) is assumed for the error variance, the model is a generalized autoregressive conditional heteroscedasticity(GARCH, Bollerslev (1986)) model.

GARCH models preserve the persistence of the process volatility in the sense that small variations tend to follow small variations and large variations tend to follow large variations. Incorporating GARCH models with hidden Markov chains, where each state (regime) of the chain implies a different GARCH behavior, extends the dynamic formulation of the model and enables a better fit for a process with a more complex time- varying volatility structure. However, a major drawback of such models is that estimating the volatility with switching-regimes requires knowledge of the entire history of the process, including the regime path. Incorporating GARCH models with a hidden Markov chain, where each state of the chain (regime) allows a different GARCH behavior and thus a different volatility structure, extends the dynamic formulation of the model and potentially enables improved forecasts of the volatility. Unfortunately, the volatility of a GARCH process with switching-regimes depends on the entire history of the process, including the regime path,which makes the derivation of a volatility estimator impractical.

Using a method to calculating variance in fundamental statistics, to estimate the current level of volatility, σ_n, will use a weighting scheme which will assign more weight

to recent data.

$$\sigma_n^2 = \sum_{i=1}^{m} \alpha_i u_{n-i}^2 \tag{1}$$

The variable α_i uses the amount of weight given to the observation i days ago. The restrictions on α's are the following:

1. $\alpha_i \geq 0$
2. Choose $\alpha_i < \alpha_j$, when $i > j$, were less weight is given to older observations.
3. $\sum_{i=1}^{m} \alpha_i = 1$

To understand the approach of the GARCH, one of the assumptions in the model is that there is a weighted long-run average variance rate. Therefore (1) can be written

$$\sigma_n^2 = \gamma V_L + \sum_{i=1}^{m} \alpha_i u_{n-i}^2 \tag{2}$$

where V_L is the long-run variance rate and the γ is the weight assigned to V_L.

The GARCH(1,1) model calculates σ_n^2 from the long-run variance rate, V_L, as well from σ_{n-1} and u_{n-1}. The equation is given by

$$\sigma_n^2 = \gamma V_L + \alpha u_{n-1}^2 + \beta \sigma_{n-1}^2 \tag{3}$$

where γ is the weight assigned to V_L, α is weight assigned to u_{n-1}^2, and β is the weight assigned to σ_{n-1}^2. The weights must sum up to one.

The GARCH(1,1) model is based on the fact that over time the variance tends to get pulled back to the long-run average level V_L. The amount of weight assigned to V_L is $\gamma = 1 - \alpha - \beta$. This model is similar to the Exponentially weighted moving average model (EWMA) expect that, in addition to assigning the weights that decline exponentially to past u^2, it also assigns a weight to V_L. In this paper, a EWMA model is generated and presented, however, the focus of this paper is to model using a GARCH(1,1) process.

The GARCH(1,1) indicates that σ_n^2 is based on the most recent observation of u^2 and the most recent estimate of the variance rate. The more general GARCH(p, q) model calculates σ_n^2 from the most recent p observations on u^2 and the most recent q estimates of the variance rate.

Setting $\omega = \gamma V_L$, the GARCH(1,1) equation in (2) can be written by

$$\sigma_n^2 = \omega + \alpha u_{n-1}^2 + \beta \sigma_{n-1}^2 \tag{4}$$

1.3 HMM

A hidden Markov model (HMM) is a statistical Markov model in which the system being modeled is assumed to be a Markov process with unobserved (hidden) states. In a hidden Markov model, the state is not directly visible, but the output, dependent on the state, is visible. Each state has a probability distribution over the possible output tokens. Therefore, the sequence of tokens generated by an HMM gives some information about the sequence of states. The adjective 'hidden' refers to the state sequence through which the model passes, not to the parameters of the model; the model is still referred to as a 'hidden' Markov model even if these parameters are known exactly.

2 Methodology

2.1 Workflow

The workflow has been depicted in the below figure. The complete methodology of the training process and the forcast estimation has been logically divided in to three phases. Following subsections comprehensively describe the functions in each phase.

The outerloop explains that for each forcast the training happens again by including the forcasted value in the previous iteration. This is required because, with the newly added forcast value, the MArkov Regimes might change along with their probabilities of transition. This in turn, results in differente GARCH models built on each Regime.

The overall apporach is that, different markov regimes are identified based on the training data by building a Hidden Markov Model, and the probability of transition between regimes is calculated. Then separate GARCH models are built on data points belonging to each regime. Then the current state depicted by the state of the latest point is considered. Then the estimates from each of the GARCH model built on each Regime is weighted over the respective proability of transition/switching from the current regime to the regime on the respective GARCH model is built.

We are considering a Markov Model with at most 2 states for the current project.

2.2 Phase A

In this phase, Training data is taken and Hidden Markov Model is built on it. This is provides probabilties of each data point belonging to different regimes. Then we get to know which points belong to which regime.

The Transition Probability Matrix (PMAT) contains the probabilitites of transition from one state to the other state. Since we have 2 states, we get 4 probabilties.

2.3 Phase B

In this phase, all the points belonging to each regime are considered separately because, they are considerably different according the HMM. Hence the hypothesis is that they have a different nature or behaviour, hence different GARCH model is built on each Regime.

The p - the lag of the residual terms (conditional mean terms) and q - the lag of the conditional variances are the two parameters for any GARCH model. The p and q values shold be chosen optimally for best results, as they show the balance between between generalization and specification. We can choose the optimal p and q values by experimenting with different p and q values (we have experimented with p and q values ranging from 1 to 2), then choosing that pair of p and q for which there is highest Log Likelyhood.

Once the optimal p and q values are found, the GARCH model with the optimal p and q values are built on each Regime.

2.4 Phase C

In this phase, we get the forcast from the MRS-GARCH. The forcast should be the

The current state is the the regime to which the lastest data point belongs to. The probability of transition from the current regime to each of the regime is taken. The individual GARCH model estimates from each regime is multiplied by the probability of transition from the current regime to that respective regime on which the GARCH is built on. This essentially is a probabilistic weighted average of the individual GARCH estimates of each Regime.

Since, the data is highly volatile, which cannot be effectively predicted by a single GARCH model, this MRS-GARCH uses an ensemble approach, where estimates from multiple GARCH models built on different sub-samples of data depending on the nature of the data, then an aggregated forcast is taken which would give a more efficient estimate of the forcasted value.

2.5 Evaluation

- RMSE: Root Mean Squared Error, is the squarred difference between predicted and actual values, where a square root is taken over their mean. It is a measure of error is the forcast.

- Value Forecast RMSE: The value forcasts and the data values are plotted and their RMSE is taken.

- Residuals Forecast RMSE: The data residuals and the forcast of residuals is taken. Residual referes to the differnece been current and the previous data value.

3 Results and Analysis

3.1 Data Description

The above explained methodology has been experimented on 4 different Stock Exchange time series dataset. From each of the stock exchanges, the index values for 500 days after Jan 2009. Each of the dataset is trained on 500 instances and forcast is taken cumulatively for next 5 values.

The 4 Stock Exchange datasets are:

- DAX: Deutscher Aktienindex (German Stock Index) - is a blue chip stock market index consisting of the 30 major German companies trading on the Frankfurt Stock Exchange.

- SMI: Swiss Market Index - is Switzerland's blue-chip stock market index, which makes it the most important in the country. It is made up of 20 of the largest and most liquid Swiss Performance Index (SPI) large- and mid-cap stocks.

- CAC: Cotation Assiste en Continu (French Stock Index) - is a benchmark French stock market index. The index represents a capitalization-weighted measure of the 40 most significant values among the 100 highest market caps on the Euronext Paris (formerly the Paris Bourse).

- FTSE: Financial Times Stock Exchange 100 Index - is a share index of the 100 companies listed on the London Stock Exchange with the highest market capitalization.

For each dataset, 3 graphs as presented as results. First, the state probabilities to which each point belongs to. They represent which point belongs to which Markov Regime which what probability; their implied state is taken to be the regime of the greater probability. Second, the forcasted values and the actual data are plotting to compare the relative distance between actual and forcasted values; which can be quantitatively presented by RMSE value on the plot. Thrid, the residuals of the data are plotting along with forcasted residuals of the data, to compare the relative distance; presented as RMSE value on the plot.

3.2 DAX Data

The figure below presents results with all the three plots for the DAX dataset.

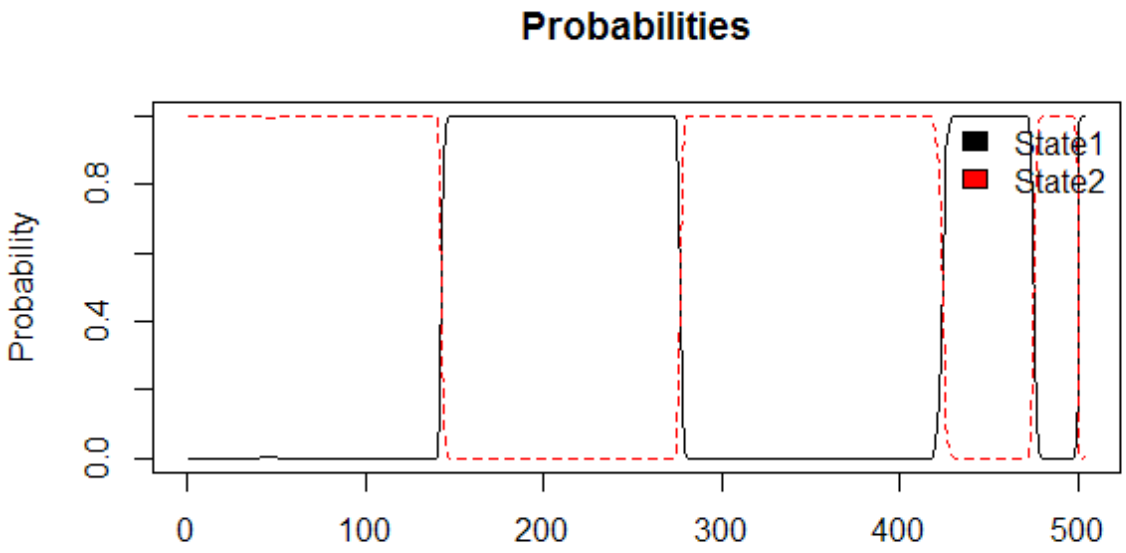

Figure 3.1: Probabilities of different points belonging to two different Markov Regimes. Results of MRS-GARCH for DAX Data.

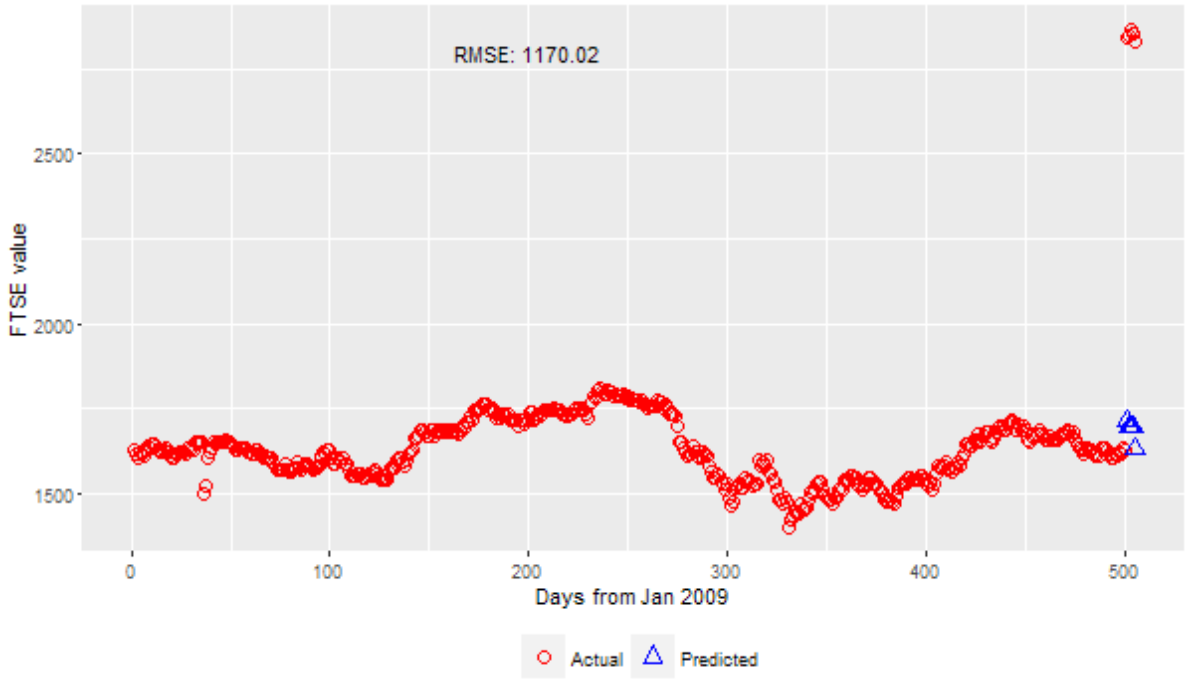

Figure 3.2: Plotting the data and the forcasts to compare them along with RMSE value. Results of MRS-GARCH for DAX Data.

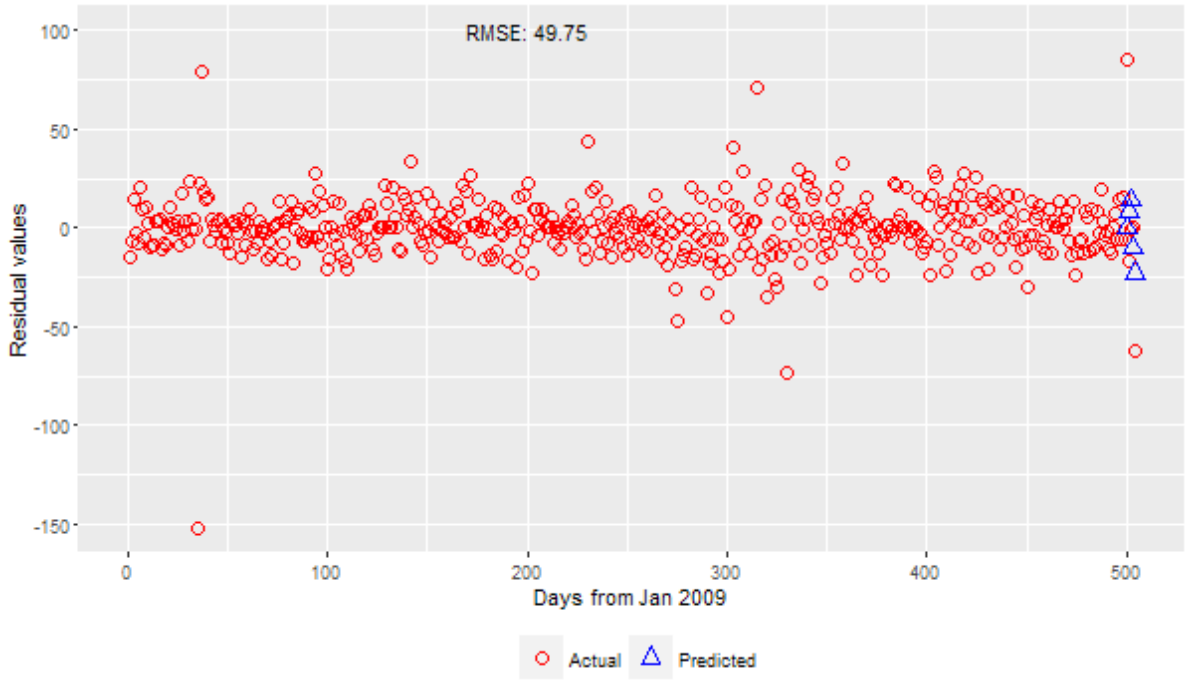

Figure 3.3: Plotting the residuals of the data and the residual forcasts to compare them along with RMSE value. Results of MRS-GARCH for DAX Data.

3.3 SMI Data

The figure below presents results with all the three plots for the SMI dataset.

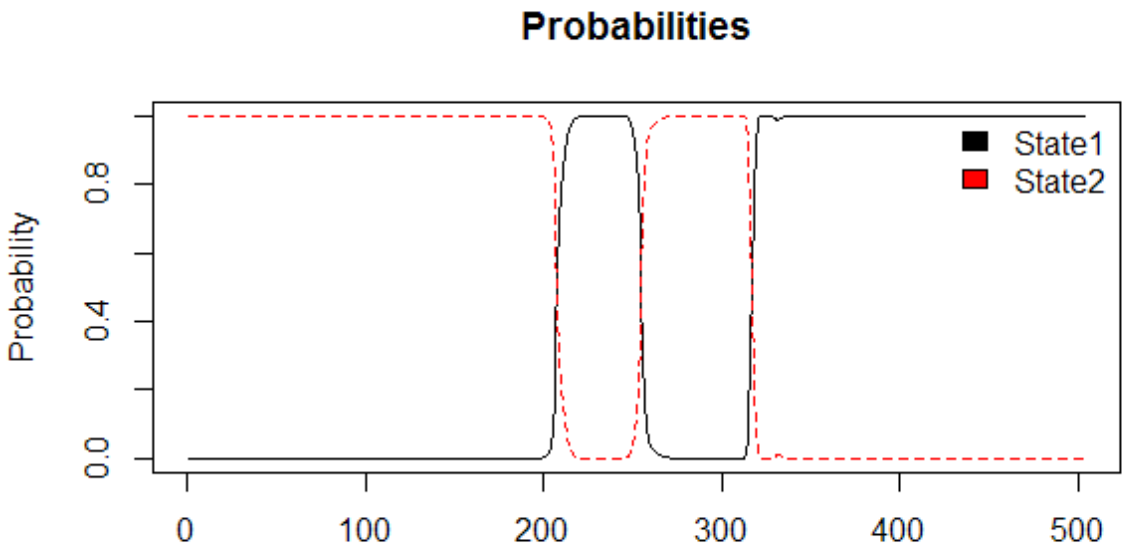

Figure 3.4: Probabilities of different points belonging to two different Markov Regimes. Results of MRS-GARCH for SMI Data.

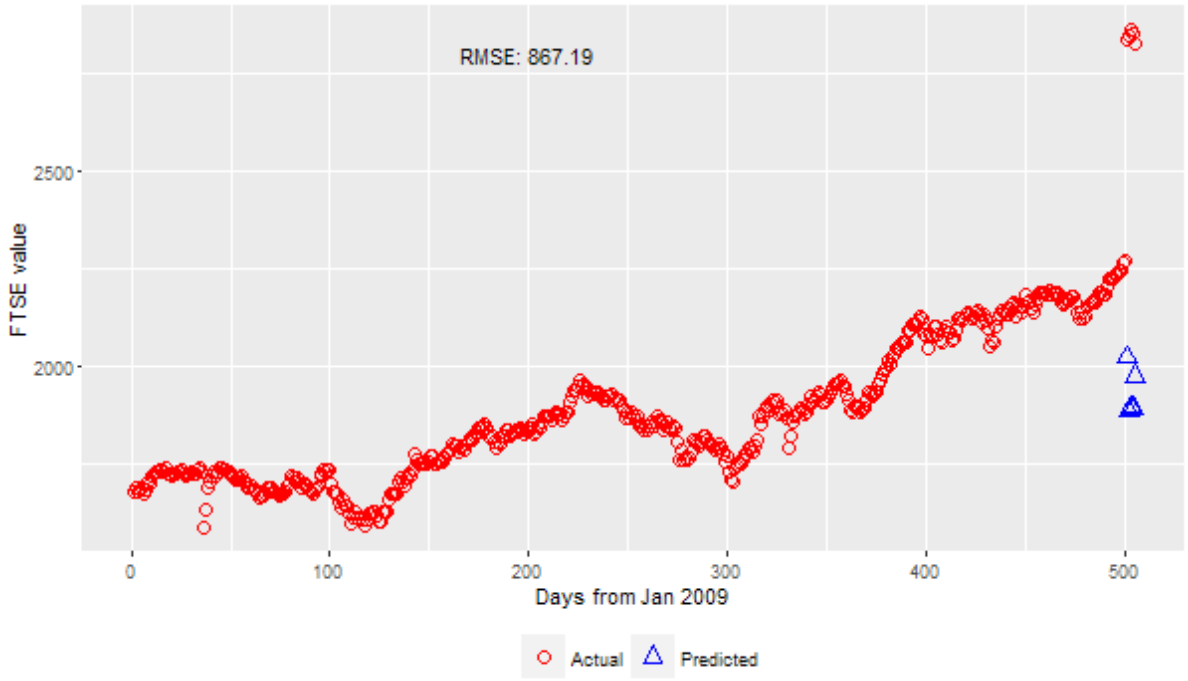

Figure 3.5: Plotting the data and the forcasts to compare them along with RMSE value. Results of MRS-GARCH for SMI Data.

Figure 3.6: Plotting the residuals of the data and the residual forcasts to compare them along with RMSE value. Results of MRS-GARCH for SMI Data.

3.4 CAC Data

The figure below presents results with all the three plots for the CAC dataset.

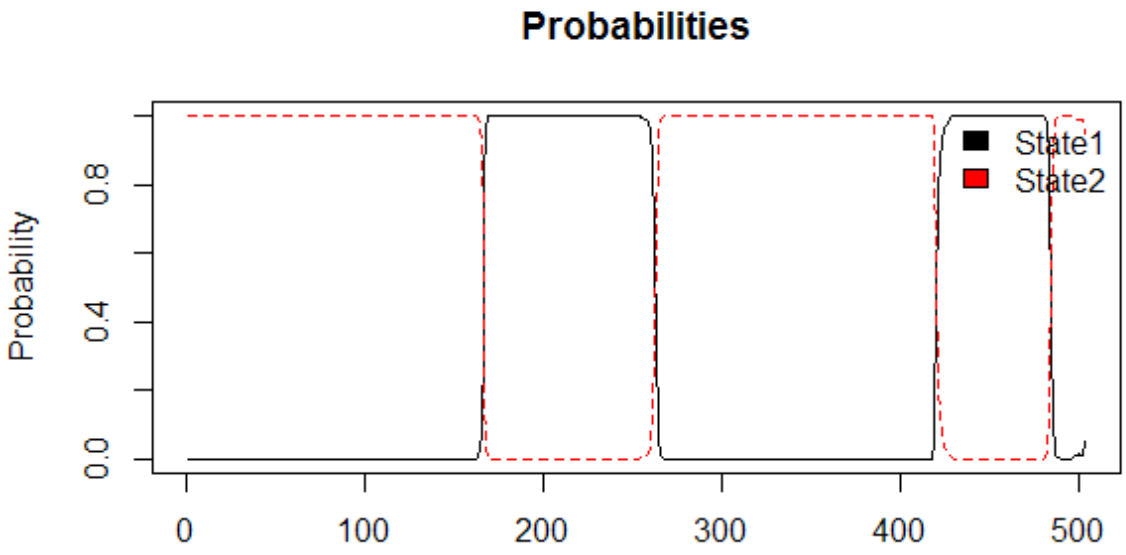

Figure 3.7: Probabilities of different points belonging to two different Markov Regimes. Results of MRS-GARCH for CAC Data.

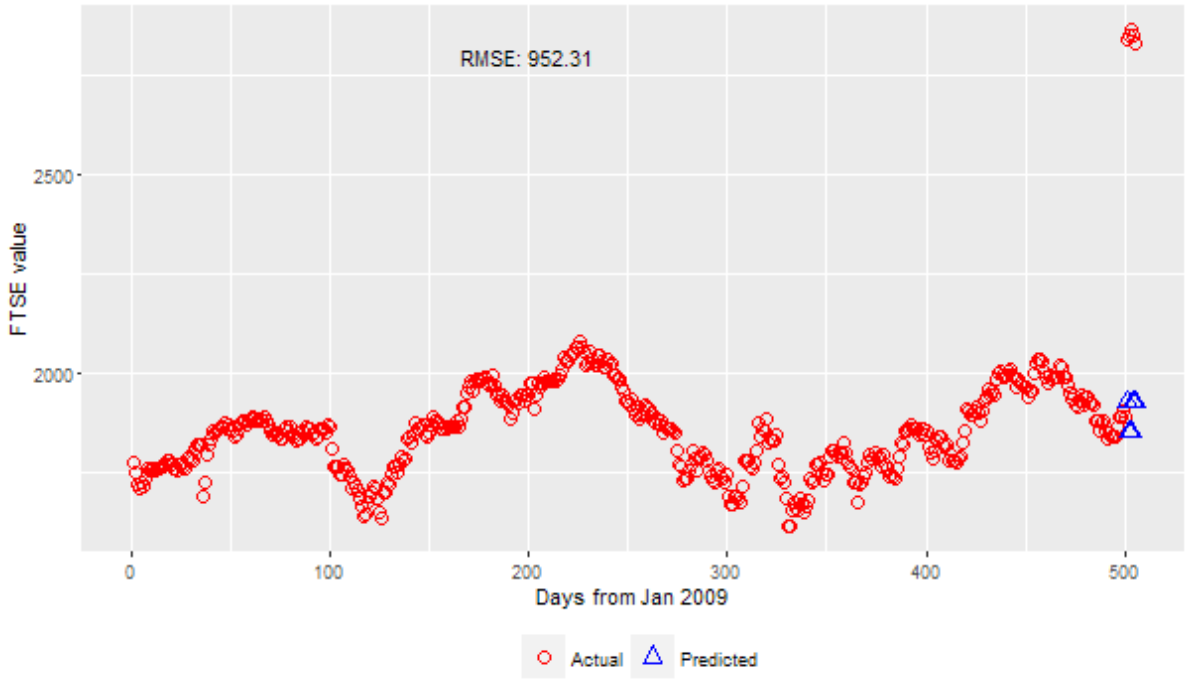

Figure 3.8: Plotting the data and the forcasts to compare them along with RMSE value. Results of MRS-GARCH for CAC Data.

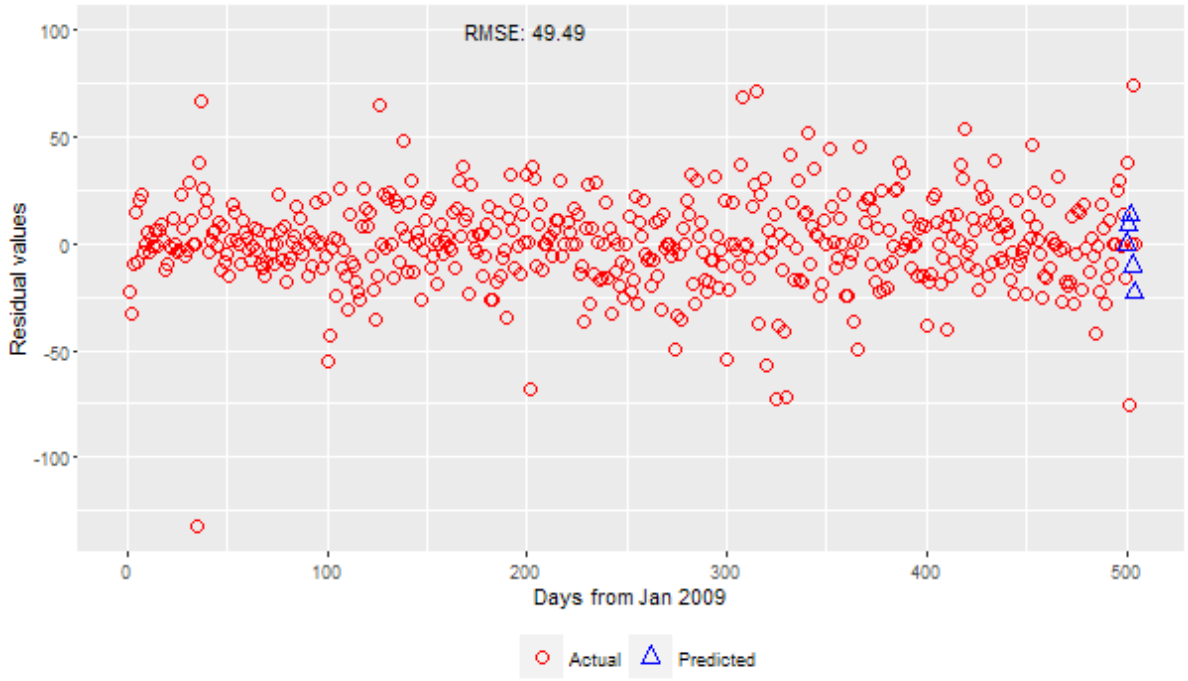

Figure 3.9: Plotting the residuals of the data and the residual forcasts to compare them along with RMSE value. Results of MRS-GARCH for CAC Data.

3.5 FTSE Data

The figure below presents results with all the three plots for the FTSE dataset.

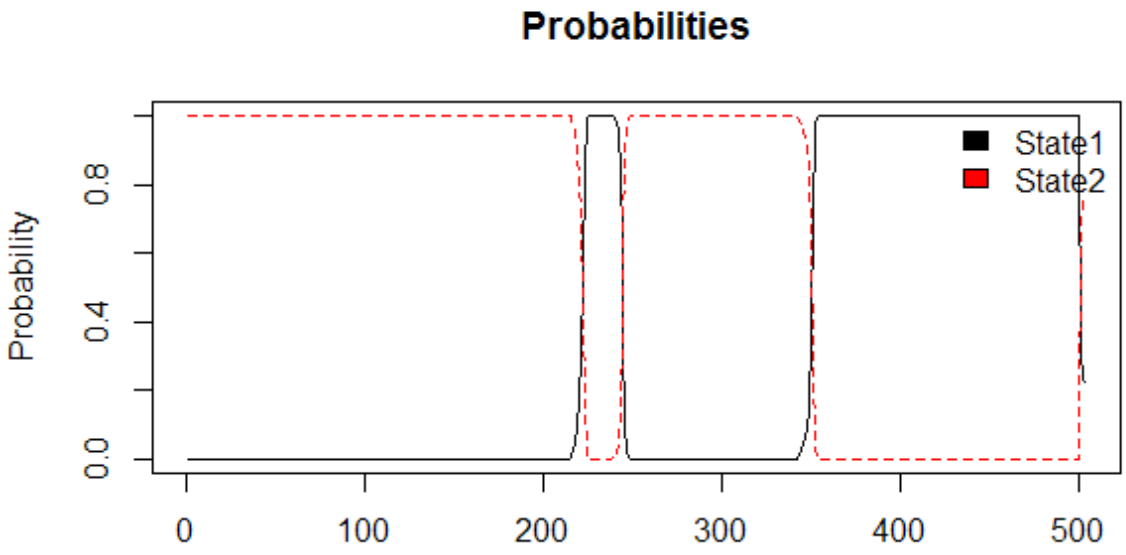

Figure 3.10: Probabilities of different points belonging to two different Markov Regimes. Results of MRS-GARCH for FTSE Data.

Figure 3.11: Plotting the data and the forcasts to compare them along with RMSE value. Results of MRS-GARCH for FTSE Data.

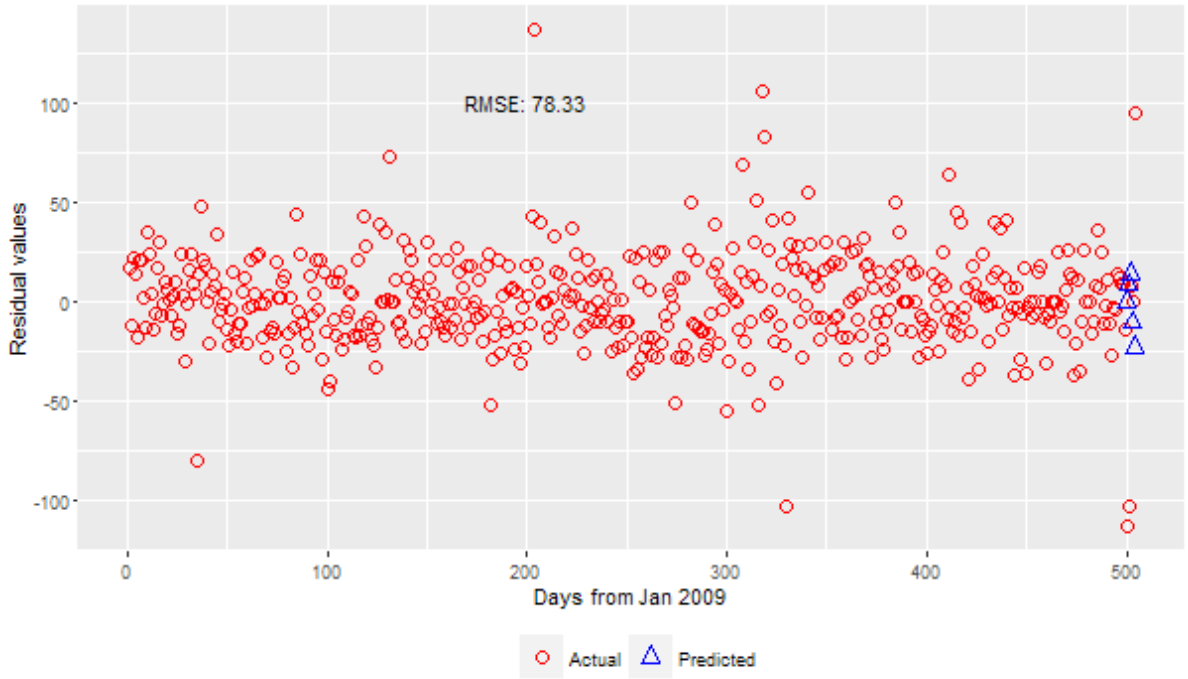

Figure 3.12: Plotting the residuals of the data and the residual forcasts to compare them along with RMSE value. Results of MRS-GARCH for FTSE Data.

4 Conclusion

The MRS-GARCH implementation explained in the report has been experimented on 4 different datasets with volatile time series. Forcasts for all the datasets from the MRS-GARCH model shows promising results with considerable RMSE values. Hence we can conclude that this model works effectively for volatile time series datasets.

5 Appendix

Code Explanation: Line by line explanation for the code attached with the project. This explains exactly what and how it is happening in the code, while the methodolofy in the report explains why each statement in the code has been written.

==========

MRS-GARCH

To install dependencies (RHmm) use: install.packages("RHmm", repos="http://R-Forge.R-project.org")

install.packages($path_to package_tar, repos = NULL, type = "source"$)

where path_to_package_tar is the path to the 'RHmm' package source (tar file) attached with the project folder.

Dependencies: 1. rugarch - GARCH model in R - To install package - install.packages('rugarch') 2. RHmm - HMM model in R - To install package please follow instructions above 3. reshape2 - (optional) required to remodel data to plot using ggplot function - To install package - install.packages('reshape2') 4. ggplot2 - Plotting graphs - To install package - install.packages('ggplot2')

= In the following description the line number is represented by '#' corresponding to the MSGARCH.R code.

==========

#8. data - It is the training data; Load the 'EuStockMarkets' data that is already available in the R. Take first 500 values as the training data. (4 denotes the selection of 4th column from the EuStockMarket dataset, corresponding to the London FTSE values.)

#10. Since the aim is to forecast upto 5 values ahead, we are looping from 1 to 5, forecasting one step ahead in each loop.

#12. Data is loaded into local variable y inside the loop.

#13. HMMFit is a function used to build a Hidden Markov Model on the data. (In our experiment we have chosen 2 states (Regimes)).

#14. Viterbi function determines which regime each point (value) belong to. (Detailed explanation - The Viterbi-algorithm computes the most probable path of states for a sequence of observations for a given Hidden Markov Model. It returns a viterbiPath - A vector of strings, containing the most probable path of states.) In line #17, this is plotted.

#15. forwardBackward function calculates the probabilities of the point (value) belonging to different regimes. This is derived from the viterbiPath vector. In line 19 this is plotted.

#21. Adding a legend to the graph being plotted.

==========

RESULTS SO FAR : Now we know which points belonging to each regime. So further we need to build separate GARCH models for points belonging to different regimes.

==========

========== #23 to #34. Calculating the probability matrix, which determines the probability of shifting from one state/regime to the other. Since we have have 2 regimes, we get 4 probabilities i.e (probability of shifting from 1 to 1, 1 to 2, 2 to 2 and 2 to 1; hence a 2x2 matrix as specified in line #34, where state11 to state 22 are the probabilities described above respectively) ========== LINE BY LINE EXPLANATION:

#23 - The variable 'diff' helps estimate points where the changes in state occur, i.e Path$state which stores the states each point belongs to is subtracted with its neighbour. Thus points where change in state occurs, the value of diff isn't 0.

#24, #25 - State2 stores indexes of points in the dataset considered, which belong to state2, similarly for state1

#26 - Compute unique values in diff, in this case the unique values are 0 (when there is no state change), -1 (change from state2 to state1), 1 (change from state1 to state2)

#27, #28 - compute indexes of change of state from state2 to state1 in #27 and #28 computes state1 to state2

#29-#32 - computes probability of state change for the PMat which is the probability matrix

34 - PMat is the probability matrix with P[i,j] corresponding to change of state from state i, to state j

==========

#lines 42 - 87: Determining the most optimal p and q parameters of GARCH for each regime Since there are 2 states, we need to build to build 2 GARCH models. Since we know points belonging to each regime, we can build different GARCH model for these regimes using those points.

In order to find the optimal p and q values for each garch models, we sample with different sets of p and q values and then pick the one for which the log likelihood (garch.fit.norm1@fit$LLH) is maximum.

==========

The nested loops in lines #42 and #44 show that we are trying out for p,q from (1,1), (1,2), (2,1), (2,2)

#46 Defines specification (the p and q values; note that since this is GARCH model we are setting the armaOrder to 0 and considering normal distribution) for the GARCH model trying out in this iteration.

#48 and #50. Using the above specification, GARCH model is built on state 2 and state 1 respectively. Now the Garch model built on State 1 is garch.fit.norm1 Now the Garch model built on State 2 is garch.fit.norm2

53 to #67. Finding the max likelihood (given by the garch.fit.norm1@fit$LLH variable) for Garch state 1, and p1 and q1 are updated. They are now the optimal values for regime 1. Similarly #70 to #84 calculate the optimal values p2 and q2 for regime 2.

==========

RESULTS SO FAR: Now the optimal values have been found. Now again building the GARCH model with optimal p and q values for both regimes.

==========

#93-94. Applying the most optimal p and q parameters for GARCH models in both the regimes and building the GARCH models. Building the GARCH model for regime 1 using the specification optimal p and q, i.e., p1 and q1 derived above.

#97-98. Building the GARCH model for regime 2 using the specification optimat p and q, i.e., p2 and q2 derived above.

Now the Garch model built on State 1 is garch.fit.norm1 Now the Garch model built on State 2 is garch.fit.norm2 =¿ These are the final GARCH models built on each regime using the optimal p and q parameters.

#100. statecurrent now holds the state to which the latest point belongs to. (states are contained in the Path$states variable, we are choosing the last one)

#102. coef1 - is the vector of coefficients in the GARCH model fit on the regime 1. these are in the order - mu (mean), omega(error), alpha1, alpha2, alpha3 ..., beta1, beta2, beta3 ... (the number of alpha coefficients is equal to p, and the number of beta

terms is equal to q)

Similarly coefficients for the GARCH built on regime 2 are found in line 103.

==========

According to the GARCH equation in order to forecast, we need to multiply square of the residuals (garch.fit.norm1@fit$residuals) (in the last p residual terms) with the alpha coefficients. A residual is the difference between a reading and the previous reading in the dataset (i.e D[n] and D[n-1])

#109 to 112. m11 - is the set of residuals^2 $for the GARCH model on first regime. \#117 to 120 m21 - is the set of residuals^2 for the GARCH model on second regime.$

==========

Similarly we need to multiply the square of the conditional variance (garch.fit.norm2@fit$var) with the beta coefficients.

#113 to 116. m12 - is the set of conditional variance for the GARCH model on first regime. #121 to 124 m22 - is the set of conditional variance for the GARCH model on second regime.

==========

#126. m1 - is the set of term values that must will multiplied with the coefficients. for example: consider p = q = 1; consider square of one residual is 0.4 and the conditional variance is 0.2. if coefficients are - mu, omega, alpha1, beta1 then m1 is - 1 , 1 , 0.4 , 0.2

then sum(coef1 * m1) will give the mu*1 + omega*1 + alpha1*0.4 + beta1*0.2; For Regime 1 then sum(coef2 * m2) will give the mu*1 + omega*1 + alpha1*0.4 + beta1*0.2; For Regime 2

#129. Forecasted value: this sum(coef1 * m1) and sum(coef2 * m2) is used in line #129; which gives the estimate of GARCH model for regime 1 and respectively. We take the weighted average of these estimates where the weights are the probabilities from moving from the current state to state 1 and state2 respectively.

PMat[statecurrent,1] is the probability of changing from current state to state 1;

PMat[statecurrent,2] is the probability of changing from current state to state 2; (statecurrent holds the current state of the latest point)

#131. The estimated forecast is appended to the data, and the whole process runs again for the next forecast going to loop started on line #10.

==========

#Plotting and getting RMSE

#137 to 150. Plotting the data and estimates Respective RMSE is found.

#154 to 168. Plotting the data residuals and estimated residuals. Respective RMSE is found.

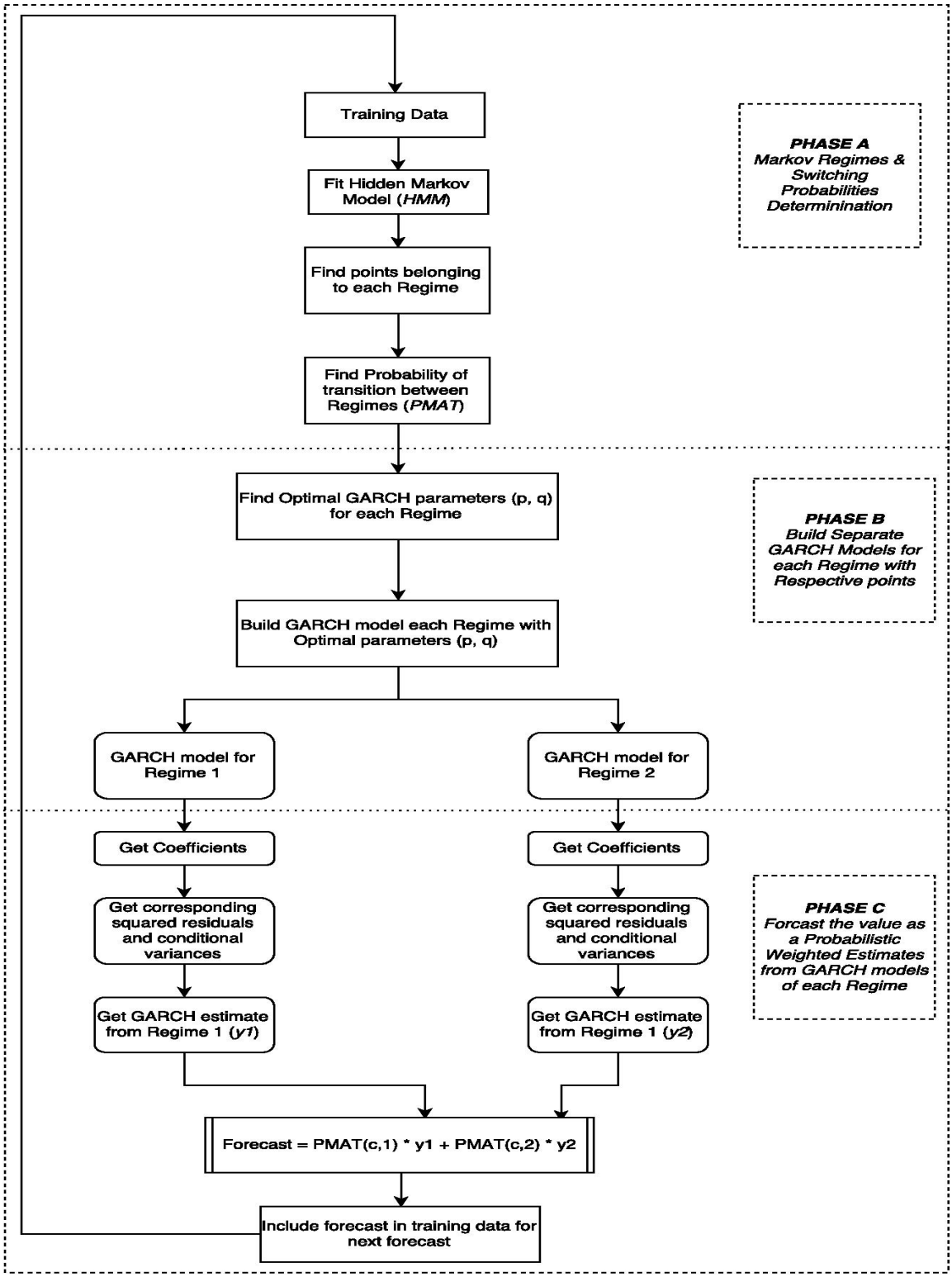

Figure 2.1: Flowchart representing the Methodology work flow.