Adari Lohit- [ladari@kent.edu](mailto:ladari@kent.edu)

Abhinav thupli- [athupili@kent.edu](mailto:athupili@kent.edu)

Kai Li- [Kli17@kent.edu](mailto:Kli17@kent.edu)

INTRODUCTION

**(Volatility prediction in Stock Exchange)**

**Volatility**

Volatility is defined as the first difference of a series the value of financial instruments depend on the expected volatility (covariance structure) of returns. volatility is analogous to risk: financial institutions undertake volatility assessments as part of the risk analysis exercise. an important object of research in time series analysis is to test hypothesis 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 nonstationary processes. similarly, an efficient analysis of a volatile time series cannot be accomplished without addressing the issue of time varying volatility.

**Applications**

volatility prediction plays a significant role in Stock Exchange because it helps investors make informed decisions about trading and investment strategies. here are some applications of volatility prediction in Stock Exchange:

1. Risk management: volatility prediction helps investors identify the level of risk associated with particular stock this information can be used to create risk management strategies to minimize loss and maximize profit.
2. Portfolio optimization: volatility prediction can be used to optimize portfolios by identifying stocks that have low volatility and high expected returns. this information can be used to create a portfolio that maximizes returns while minimizing risk.
3. Trading strategies: volatility prediction can be used to develop trading strategies that take advantage of market volatility.
4. Option pricing: volatility is a critical component of option pricing models. Accurate volatility prediction can help traders price options more accurately which can lead to more profitable trades.
5. Forecasting: volatility prediction can be used to forecast future market trends. this information can be used to make informed investment decisions and adjust training strategies accordingly.

**Description of project**

**GARCH**

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

GARCH models preserve the persistence of process volatility in the sense that small variation tends to follow small variations and large variations tends to follow large variations. Incorporating GARCH models with hidden Markov chains, where each date of 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, include the regime path. incorporating GARCH models with a hidden Markov chain, where each state of chain a laws are different GARCH behavior in this are different volatility structure, extends the dynamic formulation of the model and potentially enables improved forecast of the volatility. using a method to calculating variance in fundamental statistics, to estimate the current level of volatility.

**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 give 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.

**Workload Distribution**

1)LOHIT ADARI- coding and selection of techniques for the project report

2)ABHINAV THUPLI- selecting the dataset and working on the background of the topic, flowcharts.

3)Li Kai- supported with the coding part and theory

**BACKGROUND**

1. (Reference papers)

<https://www.sciencedirect.com/science/article/pii/S1059056022001150?casa_token=GbIsMN3FJ7kAAAAA:-Nx__p6J3BeKFfGbKmqLGEGs-ozTImk9PU9lEu6kU_8v5TbqewPYUO3elnheB0qhgOvzgGGq>

<https://ieeexplore.ieee.org/abstract/document/8626097>

<https://koreascience.kr/article/JAKO201915658234490.page>

<https://www.sciencedirect.com/science/article/pii/S1057521900000375?casa_token=tKHpg8eu1rEAAAAA:l5rGhcgR0JA2ptQX8AgQBAzL-lqqdifr60QkiXY5Dla3V1gDZ5s6rWUznSy7gUfqq844KHPC>

1. (Software tools)

We will be using R programming for the statistical analysis and libraries used are:

**Reshape2, RHmm, Ggplot2, Rugarch**

1. (Programming Skills)

Should have knowledge on Data Structure, transforming the data, functions, For Loops.

**Problem Definition and tackling the problem:**

The complete methodology of the training process and the forecast estimation has been logically divided into 3 phases.

The outer loop explains that for each forecast the training happens again by including the forecasted value in the previous iteration. this is required because, with the newly added forecast value the Markov regimes might change along with their probability of transition. This intern returns in different GARCH models built on each Regime.

The overall approach is that, different Marco regimes are identified based on 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 estimate from each of the GARCH but it is model built on each regime is weighted over the respective probability of transition from the current regime to the regime of the respective GARCH model is built.

As I explained we have divided the project into three phases:

**Phase A**

in this phase, training data is taken, and hidden Markov model is built on it. this is providing probabilistic of each data point belonging to different regimes. then we get to know which point belongs to which regime.

The transition probability matrix contains the probability of transition from one state to another state since we have two states, we get four probabilities.

**Phase B**

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

**Phase C**

In this phase, we get the forecast from MRS-GRCH. The forecast should be the current state is a regime to which the latest data points belong 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 perspective regime on which the GARCH is built on. this essentially is a probabilistic weighted average of the individual GARCH estimate of each regime.

**The Proposed Technique**

RMSE: root mean squared error, is this squared difference between predicted and actual values, where is square root is taken over there mean. it is a measure of error is the forecast.

Value forecast RMSE: they value forecasts, and the data values are plotted and their RMSE is taken.

Residuals forecast RMSE: the data residuals and the forecast of residuals is taken residuals refers to the difference between current and previous data value.

**Challenges**

The challenges we experience while doing the project is the data is highly volatile, which cannot be effectively predicted by a single GARCH model, this MRS-GARCH Users and ensemble approach. where estimates from multiple GARCH models built on different subsamples of data depending on the nature of the data then an aggregated forecast each taken which would give a more efficient estimate of the forecasted value.

**Summary**

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

**The proposed Techniques**

**Framework:**

So, we are going to experiment 4 different Stock Exchange time series data set. from each of the stock exchanges the index values for 500 days after Jan 2009. each of the data set is trained on 500 instances and forecast is taken cumulatively for next five values.

**Techniques:**

The techniques we're going to use is to state probabilities. Present which point belongs to which Marco regime. And will be finding the relative distance between actual and forecasted values which can be quantitatively represented by RMSE value on the plot.

The residuals of the data are plotting along with forecasted residuals of the data, to compare the relative distance, presented as RMSE value on the plot.

**Pseudo code:**

#Determining the most optimal p and q parameters of GARCH for each regime

p1 = 0

q1 = 0

p2 = 0

q2 = 0

for(p in 1 : 2)

{

for(q in 1 : 2)

{

garch.spec.norm = ugarchspec(variance.model=list(garchOrder=c(p,q)),mean.model= list(armaOrder=c(0,0)),distribution.model = "norm")

garch.fit.norm1 = ugarchfit(spec=garch.spec.norm,data=data[state2],solver.control=list(trace = 1))

garch.fit.norm2 = ugarchfit(spec=garch.spec.norm,data=data[state1],solver.control=list(trace = 1))

{

if(!exists('max1'))

{

max1 = garch.fit.norm1@fit$LLH

p1 = p

q1 = q

}

else

{

if(garch.fit.norm1@fit$LLH > max1)

{

max1 = garch.fit.norm1@fit$LLH

p1 = p

q1 = q

}

}

}

{

if(!exists('max2'))

{

max2 = garch.fit.norm2@fit$LLH

p2 = p

q2 = q

}

else

{

if(garch.fit.norm2@fit$LLH > max1)

{

max2 = garch.fit.norm2@fit$LLH

p2 = p

q2 = q

}

}

}

}

}

#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.

garch.spec.norm = ugarchspec(variance.model=list(garchOrder=c(p1,q1)),mean.model= list(armaOrder=c(0,0)),distribution.model = "norm")

garch.fit.norm1 = ugarchfit(spec=garch.spec.norm,data=data[state2],solver.control=list(trace = 1))

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

garch.spec.norm2 = ugarchspec(variance.model=list(garchOrder=c(p2,q2)),mean.model= list(armaOrder=c(0,0)),distribution.model = "norm")

garch.fit.norm2 = ugarchfit(spec=garch.spec.norm2,data=data[state1],solver.control=list(trace = 1))

statecurrent = Path$states[length(Path$states)]

coef1 = garch.fit.norm1@fit$coef

coef2 = garch.fit.norm2@fit$coef

m11 = c()

m12 =c()

m21 = c()

m22 =c()

for( p in 1 : p1)

{

m11[p] =(garch.fit.norm1@fit$residuals[length(garch.fit.norm1@fit$residuals) - (p - 1)]) ^ 2

}

for( p in 1 : q1)

{

m12[p] =(garch.fit.norm1@fit$var[length(garch.fit.norm1@fit$var) - (p - 1)])

}

for( p in 1 : p2)

{

m21[p] =(garch.fit.norm2@fit$residuals[length(garch.fit.norm2@fit$residuals) - (p - 1)]) ^ 2

}

for( p in 1 : q2)

{

m22[p] =(garch.fit.norm2@fit$var[length(garch.fit.norm2@fit$var) - (p - 1)])

}

m1 = c(1,1,m11,m12)

m2 = c(1,1,m21,m22)

futureval = ((PMat[statecurrent,1] \* sum(coef1 \* m1)) + (PMat[statecurrent,2] \* sum(coef2 \* m2))) ^ 0.5 + mean(data)

data = c(data,futureval)

}

RMSE = mean(((data[500:505] - EuStockMarkets[500:505,4])^2))^0.5

RMSE

xaxis = c(1:505,501:505)

yaxis = c(data,EuStockMarkets[501:505,4])

col = as.character(c(rep(1,500),rep(2,5),rep(1,5)))

df = data.frame(X = xaxis, Y = yaxis, Z = col)

plot = ggplot(df,aes(x = X, y = Y, col = Z,shape = Z)) + geom\_point(size= 3)

plot = plot + scale\_shape\_manual(values = c("1" = 1, "2" = 2),labels = c("1" = "Actual","2" = "Predicted"))

plot = plot + scale\_color\_manual(values = c("1" = "red", "2" = "blue"),labels = c("1" = "Actual","2" = "Predicted"))

plot = plot + annotate("text", label = paste("RMSE:",round(RMSE,2)), x = 200, y = 2800) + theme(legend.position = "bottom",legend.title = element\_blank())

plot = plot + xlab("Days from Jan 2009") + ylab("FTSE value")

plot

residueactual = EuStockMarkets[501:505] - EuStockMarkets[500:504]

residuefcst = data[501:505] - data[500:504]

RMSE = (mean((residueactual - residuefcst)^2)) ^ 0.5

xaxis = c(1:504,500:504)

yaxis = c((data[-1] - data[-length(data)]),(EuStockMarkets[501:505,4] - EuStockMarkets[500:504,4]))

col = as.character(c(rep(1,504),rep(2,5)))

df = data.frame(X = xaxis, Y = yaxis, Z = col)

plot = ggplot(df,aes(x = X, y = Y, col = Z,shape = Z)) + geom\_point(size =3)

plot = plot + scale\_shape\_manual(values = c("1" = 1, "2" = 2),labels = c("1" = "Actual","2" = "Predicted"))

plot = plot + scale\_color\_manual(values = c("1" = "red", "2" = "blue"),labels = c("1" = "Actual","2" = "Predicted"))

plot = plot + annotate("text", label = paste("RMSE:",round(RMSE,2)), x = 200, y = 100) + theme(legend.position = "bottom",legend.title = element\_blank())

plot = plot + xlab("Days from Jan 2009") + ylab("Residual values")

plot

**Visual Applications**

So, from the stock data we have selected few stocks in one is called DAX dataset.

So, the below image represents the probability of different points belonging to two different Markov regimes. Results of MRS-GARCH for DAX data

**Chart

Description automatically generated**

In the below image is the plotting the data and the forecasts to compare them along with RMS value

**Chart, line chart

Description automatically generated**

Below image is the plotting the residuals of the data and the residual forecasts to compare them along with RMSE value.

Chart, scatter chart

Description automatically generated

Below figure represents result with the three plots for SMI dataset.

Probabilities of different points belonging to two different Markov regimes

Chart

Description automatically generated

Plotting the data in the forecast to compare them along with RMSE value.

Chart, line chart, scatter chart

Description automatically generated

Plotting the residuals of the data and the residual forecasts to compare them along with RMSE value

Chart, scatter chart

Description automatically generated

**CAC Data**

The figure below represents result with all 3 plots for the CAC datasets.

Chart

Description automatically generated

Plotting the data in the forecast to compare them along with RMSE values.

Chart, scatter chart

Description automatically generated

Plotting the residuals of the data and the residual forecast to compare them along with RMSE values.

Chart, scatter chart

Description automatically generated

**FTSE Data**

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

Probabilities of different points belonging to two different Markov regimes.

**Chart

Description automatically generated**

plotting the data in the forecast to compare them along with RMSE values.

Chart, scatter chart

Description automatically generated

Plotting the residuals of the data and the residual forecast to compare them along with RMSE values.

Chart, scatter chart

Description automatically generated

**Experimental Evaluation**

**Description of data sets:**

So, we have been experimented on four different Stock Exchange time series data set. from each of the stock exchanges the index value for 500 days after Jan 2009. each of the data set is trained on 500 instances in forecast taken cumulatively for next five values.

The four Stock Exchange data sets are:

**DAX**: Deutscher Aktienindex (German stock index) - is a blue-chip stock market index consisting of the thirty 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 - in mid-cap stocks.

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

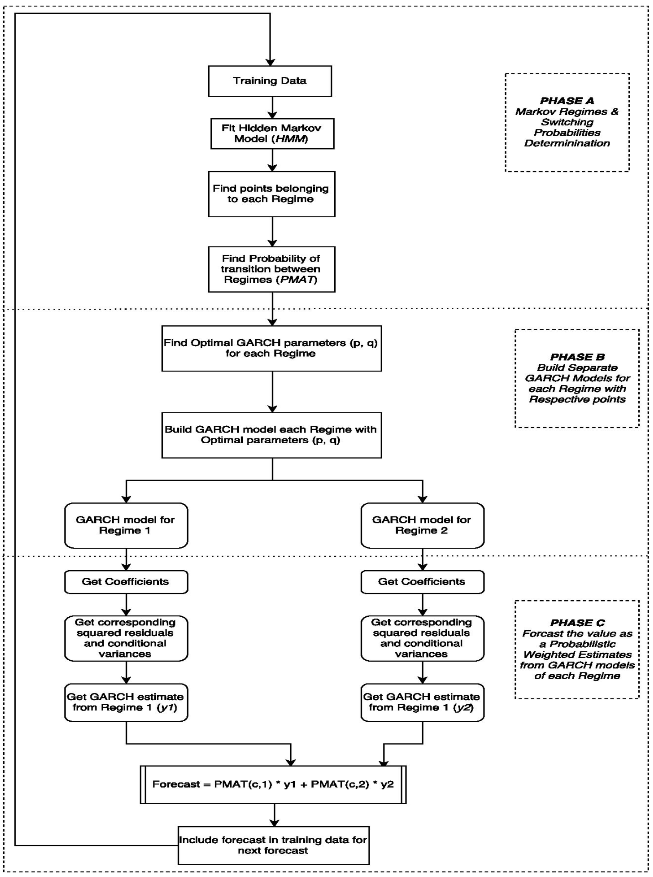
**FTSE**: Financial time Stock Exchange 100 index- is a share index of 100 companies listed on the London Stock Exchange with the highest market capitalization.

For each data set three graphs are represented as results first the state probabilities to which each point belongs to. they represent which points belong to which Markov regime and probability.

Second, the forecasted values and the actual data are plotting to compare the relative distance between actual and the forecasted values; which can be quantitatively represented by RMSE value on the plot.

Third, the residuals of the data are plotting along with forecasted residuals of the data, to compare the relative distance; presented as RMSE value on the plot.

**Flowchart representing the methodology workflow.**

****

**References**

Peiming Wang, Xinyi Liu, Dimitris Margaritis, Stock market volatility and equity returns: Evidence from a two-state Markov-switching model with regressors.

Zhuo Qiao, Yuming Li and Wing- Keung Wong, Regime-dependent relationships among the stock markets of the US