Dynamic Hedge Using LSTM (a report)

Xin Jing
xin.jing@mail.utoronto.ca
Sept. 28, 2017

1. Task Description:

When we hold a position in the spot market, we sometimes wish to offset the changes of our spot position by buying or selling certain amount of the corresponding futures contracts. Dynamic hedge refers to the practice that we change the position in the futures market periodically in order to offset the changes in the spot market more efficiently. The key is to estimate the risk-minimizing hedge ratios for each period.

2. Measurement of Performance:

Suppose we hold long position of one unit in the spot market, and in the next period, the change of spot price is $s^{(t)}$, and the change of the futures price is $f^{(t)}$, the risk-minimizing hedge ratio (denote it as b_t) is the number of units we want to sell in the futures market, the random return to this portfolio, $x^{(t)}$, is $x^{(t)} = s^{(t)} - b^{(t)}f^{(t)}$. The smaller $Var(x^{(t)})$ is, the better our methods of finding $b^{(t)}$ is.

3. Baseline

The baseline models were implemented in R. The packages used to implement the models were rugarch and rmgarch.
Rugarch is for univariate GARCH models, and rmgarch for multivariate GARCH models.
Both of them were created and maintained by Alexios Ghalanos.

Baseline A. No Hedging: the variance of s_t.
Baseline B. Naive Hedging: the variance of x_t if the number of units in the futures market is equal to that of units in the spot market.
Baseline C. Conventional Hedging: the variance of x_t if we use a fixed hedge ratio calculated by

$$b = \frac{Cov(s, f)}{Var(f)}$$

Cov(s, f) and Var(f) were estimated using all the past data before the time we hedge.

* For details about baseline C , please refer to my previous report on dynamic hedge or the paper *Time Varying Distribution and Dynamic Hedge with Foreign Currency Futures* written by K.F. Kroner and J. Sultan.

Baseline D. Dynamic Hedging with bivariate error-correction GARCH model assuming constant conditional correlation between s and f (i.e. CCC GARCH) : the variance of x_t if we calculate the dynamic hedge ratio in the following way:

$$b_t = \frac{Cov(s_{t+1}, f_{t+1})}{Var(f_{t+1})}$$

where Cov(s_{t+1}, f_{t+1}), Var(f_{t+1}) represent the the covariance of s and f in the next period, and the variance of f in the next period. They were estimated by the following models:

Mean Model :

$$\begin{aligned} s_t &= \alpha_{0s} + \alpha_{1s}\left(S_{t-1} - \delta F_{t-1}\right) + \epsilon_{st} \\ f_t &= \alpha_{0f} + \alpha_{1f}\left(S_{t-1} - \delta F_{t-1}\right) + \epsilon_{ft}, \end{aligned}$$

where St-1 and Ft-1 are spot price and futures price of the last period respectively.

$(S_{t-1} - \delta F_{t-1})$ was the error-correcting term. It is used to capture cointegration relationship between spot and future prices. The test for cointegration is the Engle and Granger(1987) test for integration.

In the experiment, on all the data sets I used, the conintegrating regression used to conduct the conintegrating tests invariably gave δ close to 1. Therefore, I took δ = 1.

Volatility Model:

$$\left[\begin{array}{c} \epsilon_{st} \\ \epsilon_{ft} \end{array}\right] \Bigg| \Psi_{t-1} \sim N(0, H_t),$$

$$H_t = \left[\begin{array}{cc} h_{ss,t} & h_{sf,t} \\ h_{sf,t} & h_{ff,t} \end{array}\right] = \left[\begin{array}{cc} h_{s,t} & 0 \\ 0 & h_{f,t} \end{array}\right] \left[\begin{array}{cc} 1 & \rho \\ \rho & 1 \end{array}\right] \left[\begin{array}{cc} h_{s,t} & 0 \\ 0 & h_{f,t} \end{array}\right]$$

$$\begin{aligned} h_{s,t}^2 &= c_s + a_s \epsilon_{s,t-1}^2 + b_s h_{s,t-1}^2 \\ h_{f,t}^2 &= c_f + a_f \epsilon_{f,t-1}^2 + b_f h_{f,t-1}^2, \end{aligned}$$

where $h_{s,t}^2$ represents the estimated Var(st)

$h_{f,t}^2$ is the estimated Var(ft) , is $h_{sf,t}^2$ the estimated Cov(st, ft)

The model is bivariate GARCH, therefore we were supposed to implement it with the package rmgarch. However, rmgarch didn't include constant conditional correlation GARCH, but only dynamic conditional correlation GARCH. Therefore, I used rugarch to implement this model, which guarantees the constant correlation assumption and all the other assumptions except the bivariate distribution assumption of st and ft. The result of this model should not be much different from that of the model with the bivariate distribution assumption. After all , the goal was to check constant conditional correlation model. I suggest that if time permits, you may try implementing the model with bivariate distribution assumption using other toolkits.

* For details about baseline D , please refer to my previous report on dynamic hedge or the paper *Time Varying Distribution and Dynamic Hedge with Foreign Currency Futures* by K.F. Kroner and J. Sultan.

Baseline E. Dynamic Hedge with bivariate error-correcting GARCH model assuming dynamic conditional correlation between st and ft :(i.e. DCC GARCH)

In base line D, we assumed that there is constant conditional correlation between st and ft . However, in reality, it is often not the case. In DCC GARCH, the conditional correlation is not constant, and it is calculated for each period.

This model was implemented with rmgarch.
For details about DCC GARCH implemented in R, please refer to the following document: https://cran.r-project.org/web/packages/rmgarch/vignettes/The_rmgarch_models.pdf

4. Our Proposed Model (LSTM):

Using models in GARCH families, previous researches have achieved satisfying results in dynamic hedge. However, GARCH model involves assumptions that were not necessarily true and several requirements for the data. For example, the (bivariate) normal distribution assumptions for asset returns, the linearity of the mean model and volatility model, the requirement for ARCH effects. Also, for different assets, we usually need to come up with different theories and models. For example, the error-correcting term often needs to be incorporated for hedging models in currency market because of the conintegration relationship, but not needed in commodity markets because cointegration relationship is not found in commodity market.

It is a tedious and grueling task to look into the data of different assets and construct a suitable, but usually far from perfect, model for each.

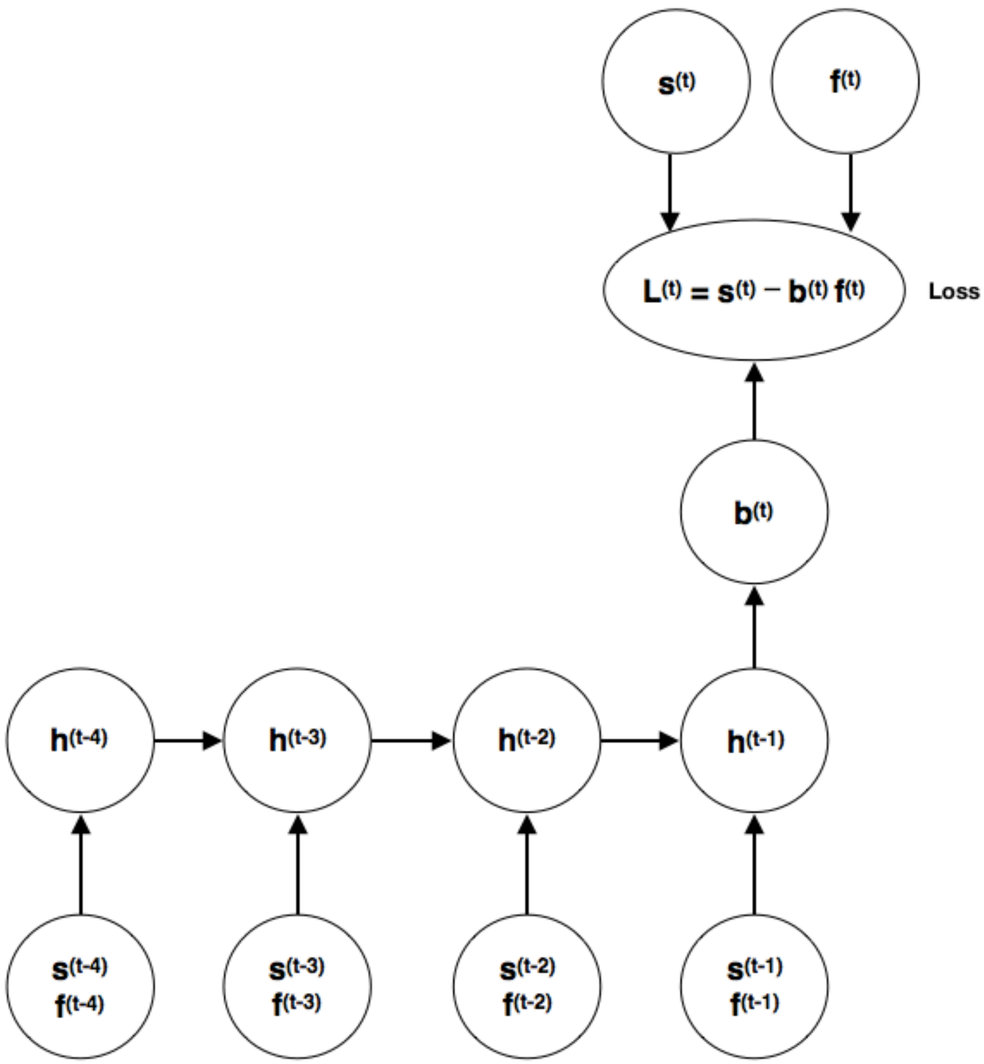
A natural idea is to modify the GARCH model by representing the mean model and/or volatility model using neural networks, which increases the ability of expression by adding non-linearity. This methods may be a good improvement for GARCH model, but not necessarily best for our hedging task, since we are still using past variances to estimate the future variances (this is what GARCH model does), from which we calculate the best ratio. However, **variance is an indirect way to express volatility, not as direct as the real changes of prices** (i.e. returns)**. We are more interested in real changes of prices because we wish to make $x^{(t)} = s^{(t)} - b^{(t)}f^{(t)}$ as small as possible**, thus making $Var(x_t)$ as small as possible.

I believe that past returns reveals enough information and that past variances derived from past returns may not be necessary. Therefore, for our task that requires us to predict the future returns of spot and futures (since $b^{(t)}_{optimal} = s^{(t)}/f^{(t)}$), we may be better off using neural networks that use **past returns as**

inputs($s^{(t-1)}$, $f^{(t-1)}$,$s^{(t-2)}$, $f^{(t-2)}$,$s^{(t-3)}$, $f^{(t-3)}$,... **)**directly, in the belief that neural networks are efficient enough to capture the information revealed by past changes of prices.

Since we still need to calculate the predicted $b^{(t)}$ after predicting the future returns($s^{(t)}$, $f^{(t)}$), we can use the predicted best hedge ratio as the output directly, skipping the calculation of it.

Therefore, we came up with the following LSTM:



Note that inputs are past returns of spot and futures, $b^{(t)}$ is the predicted best hedge ratio. The loss is equivalent to the random return to this portfolio $x^{(t)} = s^{(t)} - b^{(t)}f^{(t)}$, since we want to make $x^{(t)}$ as small as possible.

The Model was implemented with Tensorflow in Python.

It is an LSTM with 4 time steps and the only output at the last time step. After experimenting with different number of time steps, 4 is generally the best one.

The hidden size ranges from 5 to 15, depending on the assets. So far, this is the only parameter that may need to be changed for different assets.

When training the LSTM, **the batch size is 1**. Somehow it is a way to make the results stable.

The hedge ratio was calculated for each day. The portfolio was also adjusted for each day according to the hedge ratio.

5. Data:

I used past daily prices of 5 different assets. The data were from Wind. They were Australian Dollar, British Pound, Canadian Dollar, and Euro from 2010-12-06 to 2017-07-05, and S&P 500 index from 1988-1-4 to 1998-6-30 （I didn’t realize these were not up to date, but I believe the updated data would still perform well.）

For both the LSTM model and the baseline model, I chose the last 80 data of each asset as testing set and the rest of data of that asset as training set. For the LSTM model, the parameters were trained on the training set and not updated when we evaluate the performance on the testing set. However, for the GARCH models, to forecast the hedge for the following day, we need to run the model on all the previous data available, and use the latest parameters to forecast. This procedure is repeated as we roll over each of the 80 days in the testing set.

6. Results

The results of our model on the training set are not as good as those on the testing set.
However, since investors are only concerned with how well they can do in the future, not how well they could have done in the past, the good performance on the testing indicated the effectiveness of LSTM model.

With different initialization, the results given by the LSTM model are different. Therefore, I ran the LSTM model three times for each asset and took the average of the results.

As can be seen from the results, for each asset, the variance given by Dynamic Hedge with LSTM were smaller than that given by any of the other 5 baseline models.

Though making the model more complex by changing its structure and adding more features may make the performance even better, I believe that as long as the model can outperform the baselines, we can keep this simple model, since the simplicity would be more intuitive and convincing.

What to do next?

As suggested by Professor Wang, we need to make baseline models that were developed more recently, and compare their results with the LSTM model. In fact, many of the more recently proposed methods were still based on the GARCH family, with only slight changes to the previous models, but did not necessarily outperform the previous ones. Hopefully, the simple LSTM model would still beat the more recent models. If not, we first try different number of hidden states, then different time steps, and consider changing the structure of the LSTM and adding more features as the last resort.

Also, we need to run these models on various kinds of assets, since we would like to show that our proposed model is generally suitable for any kind of asset, not only the kinds of assets that satisfy the assumptions and requirements for GARCH models.