Asset Management: Class Project

Alessandro Stolfo, Metod Jazbec, Emanuele Palumbo

Problem and motivation

Given a set of assets, **portfolio optimization** is the process of selecting the best portfolio, according to some objective: typically this means maximizing factors such as expected return, and minimizing costs like financial risk. The selection of the best portfolio is based on information extracted from past data. One central issue to consider in portfolio optimization is thus to limit the error that comes with estimation of relevant parameters (e.g. mean of asset returns) from historical data. We analyze different estimation techniques, with particular focus on Hidden Markov Models and LSTM neural networks.

Initial results

To help us decide which model to focus on, we initially run all of them and compared their performance on the same investing period.

Model	Sharpe Ratio	Max. D.	Ann. Return
Sample mean	0.085	0.961	2.1%
Equal weights	0.426	0.356	3.8%
James-Stein	0.441	0.254	2.7%
HMM	0.719	0.112	2.8%
HMM (with reg.)	0.738	**0.105**	2.8%
LSTM	**0.842**	0.150	**7.9%**
LSTM (seq2seq)	0.275	0.949	3.5%

Tab. 1: Investing period from 2004-12-31 to 2015-12-28. Weekly data. Weights are recalculated every 52 weeks, using data from the past 52 weeks (exception is LSTM seq2seq, where we re-balance every 10 weeks).

LSTM

We used many-to-many LSTM architecture (see fig. 1), keeping the design of labels and inputs as proposed in the lectures, i.e. $\bar{x}_t = [x_{t-N}, \dots, x_t]$, $y_t = mean(y_t, \dots, y_{t+\texttt{future}})$. At first re-balancing data entire past data is used to train the model. Later on, only data from the most recent past window is used to fine-tune the weights (instead of re-training the model from the scratch at each re-balancing date).

Our experiments with LSTM show that promising results can be achieved, but performance is very dependent on both LSTM hyper-parameters, as well as parameters in the mean-variance optimization (start of investing period, frequency of re-balancing etc.). We further observed the lack of connection between re-balancing period from optimization pipeline and `future` parameter from labels' design: we found that smaller values for `future` parameter yield better results, even though intuitively the values of both should be approximately the same.

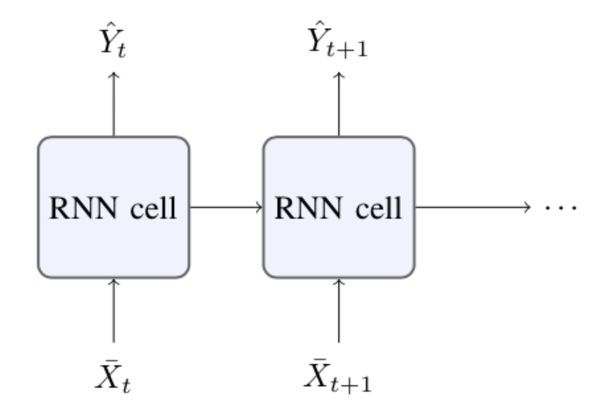

Fig. 1: Many-to-many LSTM architecture.

Sequence-to-sequence LSTM

In addition we tried out sequence-to-sequence model using encoder-decoder LSTM architecture. Here, input sequence $x_{t-M}, \dots, x_t$ is first fed to the encoder in order to obtain the last hidden state, which can be thought of as representation of the history up to now. This state is then used to initialize decoder, with which we generate predictions $\hat{x}_{t+1}, \dots, \hat{x}_{t+N}$. To come up with the final estimation of the mean vector, we simply take the mean of predictions generated by decoder. During training, the so called "teacher-forcing" is deployed, where we feed decoder ground-truths $x_t, \dots, x_{t+N-1}$ instead of its own predictions.

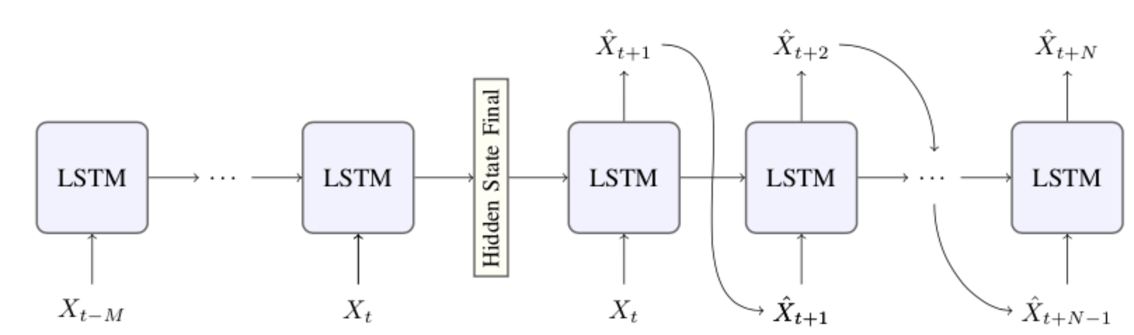

Fig. 2: Sequence-to-sequence architecture (prediction pipeline). M denotes the length of encoding sequnce, N the length of prediction/decoding interval.

Even though this approach was shown to achieve state-of-the-art results when it comes to multivariate time series forecasting [1] (which is essentially the goal of the proposed pipeline), the results we obtained were worse than the ones with many-to-many architecture. Often the model was not even able to outperform baselines (i.e. sample mean, equal weights).

Start of investing period	Sharpe Ratio	Max. D.	Ann. Return
2008	0.803	0.139	9.7 %
2010	0.232	0.302	3.2%
2012	0.452	0.161	5.2 %
2013	-0.189	0.256	-2.9 %
2014	0.402	0.053	3.7%

Tab. 2: Runs of LSTM model for different starts of investment periods.

Hidden Markov Models

Two different versions of HMMs were tested. First variant is an instance of the plain algorithm, where we fit the model on the given data of the 14 assets. In the second approach, we first fit the model on dataset consisting of 4 assets, that we thought are more representative of the regimes present in the economy: volatility index ("VIX"), the "S&P500" index, the "DAX" index and the price of the U.S. 5-year Treasury bond ("US5Y"). Obtained annotations (in figure 3) of the past dates are used to segment initial dataset (14 assets) into separate clusters. In both versions, only data belonging to the next predicted most likely state/regime is used to compute estimate of the mean vector.

Model	Sharpe Ratio	Max. D.	Ann. Return
Sample mean	0.149	1.136	3.3%
Equal weights	0.334	0.387	3.0%
HMM	0.375	0.266	2.4%
HMM (with reg.)	**0.886**	0.428	**3.9%**

Tab. 3: Results for investing period from 2004-12-31 to 2015-12-28. Data considered on daily basis. Weights are recalculated every 12 weeks (60 business days), using data from the past 104 weeks (520 business days).

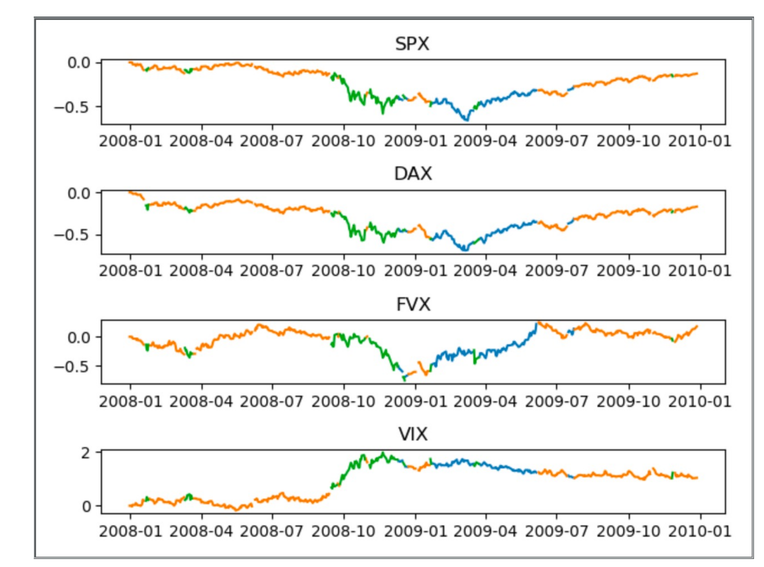

Fig. 3: Annotated past dates for selected indices used to fit HMM.

Conclusions

After extensive experiments with variants of both LSTM and HMM models, we decided to stick with the latter as the final model. Even though LSTM usually obtained higher annualized returns, we deemed the results too dependent on the parameters of the mean-variance optimization pipeline to consider the model fully trustworthy. Hence we decided to go with more conservative, yet more stable approach of HMM.

We would point of two possible areas of improvement. First one would be coming up with the better evaluation/back-testing pipeline, with the goal of getting results that are less susceptible to changes in the parameters of the optimization engine. Second one would be to put more attention into making LSTM produce more stable results (performing multiple train-validation data splits, deploying techniques to combat possible over-fitting issues like drop-out or regularization, using exponential decay for learning rate, standardizing data etc.).

References

[1] Salinas D., Flunkert F., and Gasthaus J. "DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks". In: *arXiv* (2019).