Inter Asset Class Correlation

for Futures using LSTM time series prediction for multiple Driving series

Devansh Chandak
Varun Madhavan

Ticker Embedding Model

- Given, the hourly data for futures for each RIC across 5 asset classes for 23 years, we had to create a word embedding model so that we can pass it through the LSTM to be created later
- The features chosen were Asset Class, Country/Exchange, Price Bucket ID, Volume Bucket and the Open Interest Bucket ID
- Each of the 3 numeric features were divided into equally sized buckets with the bucket being represented by an integer ID

Bucket IDs

5 types of bucket IDs were created

Sl. No. / No. Of Buckets	Price	Volume	Open Interest	Asset Class	Country
1	10	10	10	5	0
2	10	0	10	5	11
3	5	5	5	5	11
4	0	0	0	5	0
5	18	17	18	0	0

Sentences

- Sentences were created for each type of bucket ID by concatenating the RICs sorted by (date, time) and then by RIC, occuring in each Bucket ID. Example :

Bucket	Sentence
000::00::00::Commodities	BOc1 BOc1 LCc1 BOc1 BOc1 BOc1 BOc1 BOc1 BOc1 B
100::00::00::FX	RAc1 RAc1 RAc1 RAc1 RAc1 RAc1 RAc1 RAc1 RAc1 RA
200::00::00::Fixed Income	10TBc1 10TBc1 10TBc1 10TBc1 10TBc1 10TBc1 10TBc1
300::00::01::Commodities	FCc1 FCc1 FCc1 FCc1 FCc1 HGc1 BOc1 BOc1 BOc1 BC
400::00::01::FX	BRc1 BRc1 BRc1 BRc1 BRc1 BRc1 BRc1 BRc1 BRc1 BF
500::00::01::Fixed Income	10TBc1 10TBc1 10TBc1 10TBc1 10TBc1 10TBc1 10TBc1
600::00::02::Commodities	LHc1 LHc1 LHc1 LHc1 HOc1 LHc1 LHc1 BOc1
700::00::02::FX	MPc1 MPc1 MPc1 MPc1 MPc1 MPc1 MPc1 MPc1 MPc1
800::00::03::Commodities	HOc1 LHc1 LHc1 LHc1 LHc1 LHc1 LHc1 BOc1
900::00::03::FX	MPc1 MPc1 MPc1 MPc1 MPc1 MPc1 MPc1 MPc1 MPc1
##00::00::04::Commodities	BOc1
##00::00::04::FX	MPc1 MPc1 MPc1 MPc1 MPc1
##00::00::05::FX	MPc1 MPc1 MPc1 MPc1 MPc1 MPc1 MPc1 MPc1 MPc1
##00::00::06::FX	MPc1 MPc1 MPc1 MPc1 MPc1 MPc1 MPc1
##00::00::09::FX	MPc1

This was done for all 5 types of bucket IDs and the result was concatenated into a single csv and passed as corpus to our language model

How we made the embeddings -

- We tried to use our own language model using the CBOW architecture which theoretically should have done better.
- But, word2vec does better than FastText because subword information and the neighbours of each ticker is not really of any use and also training time is much higher
- So we ended up using word2vec implemented using gensim and passed our corpus created to generate a vector for each #RIC to be passed to the LSTM, with inherent similarities between similar tickers based on the features used

LSTM

- LSTM was fed a 1*822 vector for each ticker for each timestamp across the period from 2003 to 2010 and we will test it on 2010 onwards. The data prior to 2003 was neglected since the fill rate was too less to be learnable
- LSTM was constructed such that our model acts as a classifier. Expected Return was divided into 5 equally sized classes. Class of the Expected Return was the target variable
- Important features among the 822 :

  -> Average Expected Return per asset class per hour

  -> Bid, Ask, Price, Return, Expected Return

  -> Return Class

  -> 1*800 vector created from the language modeller

  -> Trailing Volatility

Preprocessing for LSTM

- LSTM needs equally sized sequences so we had to ensure that each day has equal number of data points and also, continuity across time for each ticker is maintained
- Mode of the number of hours traded per day across tickers across time was found out to be 23
- So each ticker's data has been padded to 23 for each day (Padded with 0 makes sense since it will not affect the weights of the LSTM. Padded data was assigned a different class and is ignored by the Loss function
- For continuity, the prices at the expiration of a future (mostly one month) and the opening of the next future was made equal such that the Return does not change

- A variant of LSTM was also tried in which we initialize the hidden state to be the vector for each ticker and pass only a 1*22 vector for each timestamp

Ongoing Work

- LSTM with attention is also being tried which improves the long memory of the LSTM and is supposed to give better results on general NLP tasks
- Instead of passing information for one ticker in each batch, we will also try passing a 56*822 vector (all tickers for each timestamp together)
- After getting the classifier outputs we will study the correlation between the asset classes, by studying the correlation between the average return per hour per asset class for all 5 asset classes and the Expected Return column
- ***Ultimate aim is to study the non-linear interaction on Excess Return***