MARKETPSYCH RESEARCH

Literature Review on Improving Stock Movement Prediction with Adversarial Training

Author of paper:
Fuli Feng, National Unversity of Singapore,
Huimin Chen, Tsinghua Unversity,
Xiangnan He, National Unversity of Singapore,
Ji Ding, University of Illinois at Urbana-Champaign,
Maosong Sun, University of Illinois at Urbana-Champaign,
Tat-Seng Chua, National Unversity of Singapore,

Date of paper: 13 Oct 2018

Link: https://arxiv.org/pdf/1810.09936.pdf

1 Proposed Methodology

1.1 Main Idea

An novel addition, Adverserial Training, to LSTM was proposed to improve prediction of inter-day price movement (up or down) for stocks. Hence, each time step here refer to a day interval. The argument for this novel addition is that traditional LSTM is not able to take care of the stochastic property of stock market and this addition is able to.

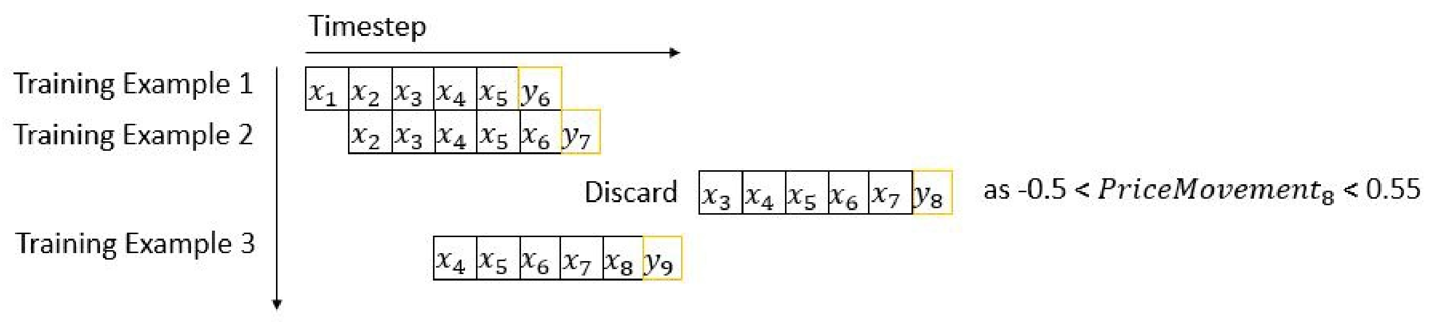
Adverserial Training is to do the following together when training:

1. train each training example normally
2. train each training example after degrading it in the bad direction

By training the model to put the degraded training example in the correct class (up or down), the training takes care of stocastic property of each training example, hence takes care of the stochastic property of stock market

1.2 Experimental Set-up

The experimental set-up is as follows:



$$y_t \begin{cases} = 1 \; if PriceMovement_t > 0.55 \\ = 0 \; if PriceMovement_t < -0.5 \\ Discarded \; if \; otherwise \end{cases}$$

$$PriceMovement_t = Price_t / Price_{t-1} - 1$$

The LSTM is configured to only remember the past 5 time-steps (x1, x2, x3, x4, x5) to predict whether price is up or down at next time step (y6). Hence the LSTM network will remember from all the training examples how should a sequence of 5 time steps should lead to the prediction for the next time step.

As illustrated in the diagram above, training examples were discarded if the price movement were not bigger than 0.55% or smaller than -0.5%. This remove arbitary examples for training and help the training to discover a more distinct separation between Up and Down classes.

1.3 Feature Engineered x variables

The x variables were selected and engineered as follows:

Table 3: Features to describe the daily trend of a stock.

Features	Calculation
c_open, c_high, c_low	e.g., $c\_open = open_t / close_t - 1$
n_close, n_adj_close	e.g., $n\_close = (close_t / close_{t-1} - 1$
5-day, 10-day, 15-day, 20-day, 25-day, 30-day	e.g., 5-day $= \frac{\sum_{i=0}^{4} adj\_close_{t-i}/5}{adj\_close_t} - 1$

1.4 Train-Validate-Test Set Up

The single Train-Validate-Test period is as such:

01-01-2014 - 31-07-2015 Train	Aug-01-2015 - Sep-30-2015 Validate	Oct-01-2015 - Dec-31-2015 Validate

1.5 Matthews correlation coefficient (MCC)

The paper evaluates with MCC whicn takes into takes into account all 4 quadrants in a confusion matrix. The MCC is in essence a correlation coefficient between the observed and predicted binary classifications; it returns a value between −1 and +1. A coefficient of +1 represents a perfect prediction, 0 no better than random prediction and −1 indicates total disagreement between prediction and observation.

$$\text{MCC} = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

1.6 Adverserial Training

During training, a training example's hidden state was made bad. The bad hidden state and the unthwarted hidden state were optimised so that both hidden states can be classified correctly to be Up or Down.

2. Results and Evaluation

2.1 Results

Table 4 and Table 5 shows the performance of the Adverserial Training (adv-ALSTM) on ACL18 dataset and KDD17 dataset.

ACL18 dataset contains end-of-day (EOD) data from Jan-01-2014 to Dec-31-2015 of 88 high-tradevolume-stocks in NASDAQ and New York Stock Exchange.

KDD17 dataset includes EOD data with longer history ranging from Jan-01-2007 to Dec-31-2015 of 50 stocks in U.S. markets. However only data from Jan-01-2014 to Dec-31-2015 were used.

Table 4: Performance of compared methods on the **ACL18** dataset. RI denotes the relative improvement of **Adv-ALSTM** compared to the associated baseline.

Methods	Acc	RI	MCC	RI
RAND	50.89±—	12.40%	-0.0023±—	—
LSTM	53.18±5e-1	7.56%	0.0674±5e-3	120.03%
ALSTM	54.90±7e-1	4.02%	0.1043±7e-3	42.19%
StockNet	54.96±—	4.08%	0.0165±—	798.79%
Adv-ALSTM	**57.20**±—	—	**0.1483**±—	—

Table 5: Performance of compared methods on the **KDD17** dataset. RI denotes the relative improvement of **Adv-ALSTM** compared to the associated baseline.

Methods	Acc	RI	MCC	RI
RAND	50.19±4e-1	5.70%	0.0038±8e-3	1276.32%
LSTM	51.62±4e-1	2.77%	0.0183±6e-3	185.79%
ALSTM	51.94±7e-1	2.14%	0.0261±1e-2	100.38%
StockNet	51.93±4e-1	2.14%	0.0335±5e-3	56.12%
Adv-ALSTM	**53.05**±—	—	**0.0523**±—	—

The results shows that Adv-ALSTM performed best in each dataset, achieving 57.2% and 53.5% in the 2 datasets.

3 Personal Takeaways

3.1 Short Time-Steps Set up

The 5 time steps set-up used in this paper could be adapted. This set-up could take away bad examples for training, thus allowing the LSTM model to converge better. This set-up also do not need continuous build-up of state to predict as it only needs to take in the past 5 time-steps to predict.

3.2 Adverserial Training

The Adverserial Training could be adapted. Adverserial Training may or may not allow the LSTM to converge better but if converged, it should present a better accuracy.

Generated on: 2018-12-20 05:35:08.834137
Code revision: 56a7a39361b685f2df4513b989f4dcea22c2c0e7