homework9

April 18, 2019

1 Homework 9 - Berkeley STAT 157

Your name: XX, SID YY, teammates A,B,C (Please add your name, SID and teammates to ease Ryan and Rachel to grade.)

Please submit your homework through gradescope

Handout 4/16/2019, due 4/23/2019 by 4pm.

This homework deals with sequence models for numbers. It builds on Homework 8 in terms of modeling. The main difference to last week is that we're modeling *real valued numbers* of stocks rather than characters.

This is teamwork.

1.1 1. Time Series Model

The goal is to develop multivariate regression models where the numbers are *nonnegative* and where changes are *relative*. That is, a stock price can never assume negative values and for convenience we assume that the companies listed do not go bankrupt, i.e. their stock price will never be zero. Moreover, we assume that we can ignore quantization of prices, i.e. the fact that stocks aren't traded at arbitrary prices in $\mathbb{R}$ but only at fractions of a cent (see this link for a backstory).

The prices x_{st} for a security s at time t typically reported at a given date are `(open, high, low, close, volume)`. Here `open` denotes the price when the market opens, `high` the highest price that it was traded for during that day, `low` the lowest, and `close` is the price of the security at closing. Lastly `volume` is an indicator for how many units were sold at that day. We index the respective values with $x_{st} = (o, h, l, c, v) \in \mathbb{R}^5$. To process them we transform x_{st} as follows:

$$z_{st} := (\log o, 10 \cdot (\log h - \log o), 100 \cdot (\log l - \log o), \log c, \log v)$$

Moreover, we assume that z_{st} is obtained as part of a regression problem with squared loss, i.e. for an estimate $\hat{z}_{st}$ we compute the loss as

$$l(z_{st}, \hat{z}_{st}) = \frac{1}{2} \|z_{st} - \hat{z}_{st}\|^2$$

1. Why is converting values into logarithms (and logarithms of ratios) a good idea? Explain this for each variable.
2. Why would we want to rescale the ratios by 10?
3. Explain why this model assumes a *lognormal* distribution of prediction errors between the values of the securities z_{st} and their estimates $\hat{z}_{st}$. That is, rather than being drawn from a Gaussian, they're drawn from another distribution. Characterize it (hint - exploit the connection between squared loss and the normal distribution).

4. Now assume that we have not just one security but the top 500 stocks over some period of time. Why might it make sense to estimate the share prices jointly?

1.2 2. Load Data

1. Obtain data from the S&P500 for the past 5 years and convert it into a time series. You can get the data either from Kaggle www.kaggle.com/camnugent/sandp500 or crawl it directly using the Python script given here: github.com/CNuge/kaggle-code/blob/master/stock_data/getSandP.py. Your dataset will contain tuples of the form `(date, open, high, low, close, volume, Name)`.
2. Import this data into an NDArray dataset where you have a vector containing `(open, high, low, close, volume)` for each security. That is, this is a 2,500 dimensional vector and you have 5 years' worth of data.
3. Preprocess the data into logarithmic representation as outlined in problem 1.
4. Split the data into observations for the first 4 years and a dataset for the last year.
5. Load data into an MXNet dataset iterator.
6. Why do you need to do this as opposed to splitting into random segments?

1.3 3. Time Series Implementation

1. Implement a model similar to `RNNModel` of section d2l.ai/chapter_recurrent-neural-networks/rnn-gluon.html suitable for regression. It should take as input vector-valued data, such as the time series mentioned above and it should output vector-valued data (of some other dimensionality).
2. Train the model on the first 4 years of data using plain RNN, GRU and LSTM cells (for a single layer). How well can the model
   - Predict the stock value the next day on the last 1 year of data (price at opening).
   - Plot how the quality of the model degrades as we apply it throughout the year (i.e. we ingest all the data up to day t and predict forward at day $t + 1$).
   - Predict the stock value the next week on the last 1 year of data (price at opening).
3. Train the model on each stock separately (with much lower dimensionality) and compare the performance of the above model with the one you get by using each stock separately.
4. Improve the model using better features, e.g. the fact that time is not uniformly spaced (Saturday, Sunday and holidays do not see any trades). For that use the day of the week as an additional input feature.
5. Improve the model further by using a deeper RNN, e.g. with 2 layers.

Note, there are many cases where we might want to know the *sequence* of stock prices over a period of time rather than just knowing the value, say one month from now. This is relevant e.g. for options pricing where investors can bet on or bet against volatility of a stock price. For a detailed description of this see e.g. en.wikipedia.org/wiki/Options_strategy.