Predicting Stock Returns based on quarterly SEC Reports using Deep Neural Networks

Author: Daniel Winkler
Supervisors: Florian Huber & Christian Hotz-Behofsits

11 January 2019

Introduction

- Predicting stock returns is, like, really hard! (Welch and Goyal, 2007)
- Often a simple Random Walk or AR(1) outperforms complex models
- Company specific "fundamentals" improve predictions marginally
- ... so do market indicators
- **What else could be used as predictors?**

Introduction

- Predicting stock returns is, like, really hard! (Welch and Goyal, 2007)
- Often a simple Random Walk or AR(1) outperforms complex models
- Company specific "fundamentals" improve predictions marginally
- ... so do market indicators
- **What else could be used as predictors?**
- Think beyond numeric data!

Use TEXT as a predictor

- A lot of information is available in text form
- It contains insights beyond numeric indicators
- ... that might even give a hint about future developments

"The Company's financial condition and operating results have been in the past and may in the future be materially adversely affected by the Company's ability to manage its inventory levels and respond to short term shifts in customer demand patterns" (Apple 10-Q filing, 1.2.2008)

You might ask: "How TF does that work?"

- We'd like:
  $r_t = \phi r_{t-1} + \beta TEXT + \varepsilon_t, \quad \varepsilon_t \sim N(0, \sigma^2)$
- But we need a mapping: TEXT $\Rightarrow$ vector
- Current approach: Dictionaries (Loughran and McDonald, 2011)
- Problems:
  - Does not scale well
  - Pre-processing
    e.g. "Stemming": share = sharing?
  - Count positive/negative words
    What about negations? What is "positive"?
- ***NEW*** approach: Model TEXT as sequence of characters

Model TEXT as a sequence of characters

- Basic idea (Krause et al., 2016):

$$p(c_1, c_2, \ldots, c_T) = p(c_1)p(c_2|c_1)p(c_3|c_1, c_2)\ldots p(c_T|c_1, \ldots, c_{T-1})$$

- c_i being the i-th character in the TEXT
- Minimize the prediction error of the next character given all previous ones
- Can be solved efficiently using gradient descent $\rightarrow$ "Deep Learning"

Section 2

Deep Learning

Quick Introduction

- Similar to maximum likelihood (Goodfellow et al., 2016, pp. 128f.)!
- Adjust coefficient matrices until negative log-likelihood is minimized
- However, we have **multiple** coefficient matrices which are linear
- Inbetween each operation there is a non-linearity so matrices are identified
- Deep in the sense that we do not see where the change happens and why!

Gradient Descent

1. Define a loss function
   In our case the "Cross Entropy"

$$\text{loss}(c, l) = -log\left(\frac{exp\{c_l\}}{\sum_J exp\{c_j\}}\right) \tag{1}$$

2. Calculate the gradient w.r.t. the inputs
   (This shows the steepest direction!)
3. Move the weights matrices "a little bit" in the opposite direction
4. Repeat

Efficiency

- Differentiation can be automatized e.g. w/ PyTorch
- Operations are automatically parallelized over GPUs
- Relies only on the loss function (No $(X'X)^{-1}$ stuff)

Recurrent Neural Network

- Class of DL models for sequences

$$h_t = f_a(W_h h_{t-1} + W_x x_t), \qquad (2)$$

- Specifically: multiplicative long short-term memory (mLSTM)
- Achieves state of the art in sentiment discovery (Radford, Józefowicz, and Sutskever, 2017)
- I'll show that it can also "learn" about company reports

The mLSTM

1. Input

$$m_t = (W_{mx} x_t) \odot (W_{mh} h_{t-1}) \tag{3}$$
$$\tilde{c}_t = W_{hx} x_t + W_{hm} m_t \tag{4}$$

2. Update

$$f_t = \sigma(W_{fx} x_t + W_{fm} m_t) \tag{5}$$
$$i_t = \sigma(W_{ix} x_t + W_{im} m_t) \tag{6}$$
$$o_t = \sigma(W_{ox} x_t + W_{om} m_t) \tag{7}$$

3. Output

$$c_t = f_t \odot c_{t-1} + i_t \odot tanh(\tilde{c}_t) \tag{8}$$
$$h_t = tanh(c_t) \odot o_t \tag{9}$$

Econometric Framework

- Dynamic Panel regression

$$r_{kt} = \alpha_k + \phi r_{kt-1} + F_{kt}\beta_1 + S_{kt}\beta_2 + \varepsilon_{kt}, \quad \varepsilon_{kt} \sim N(0, \sigma^2) \quad (10)$$

- S_{kt} is the final cell state; F_{kt} the fundamentals
- Using shrinkage priors (Piironen and Vehtari, 2017) to get rid of unnecessary coefficients
- Estimate using Hamiltonian Monte Carlo with Stan
- Evaluated using log-predictive density and RMSE (Geweke and Amisano, 2010)

Data

- Quarterly Securities and Exchange Commission filings (EDGAR)
- Stockprices from Center for Research in Security Prices
- Fundamentals from Financial Ratios Suite (WRDS)
- 165 companies
- 2000Q2 - 2016Q4
- 38Q Estimation - 29Q Holdout

Section 4

Results

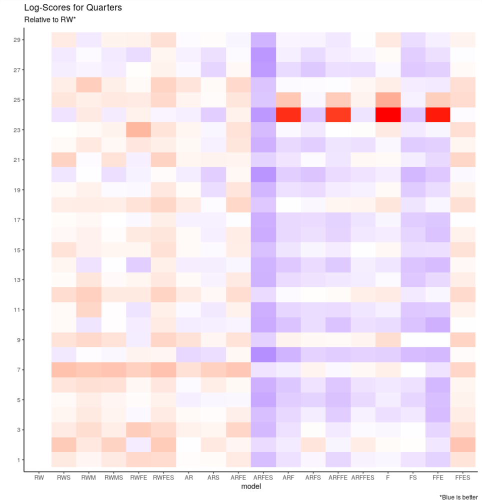

Figure: Log-Scores for Quarters relative to Random Walk

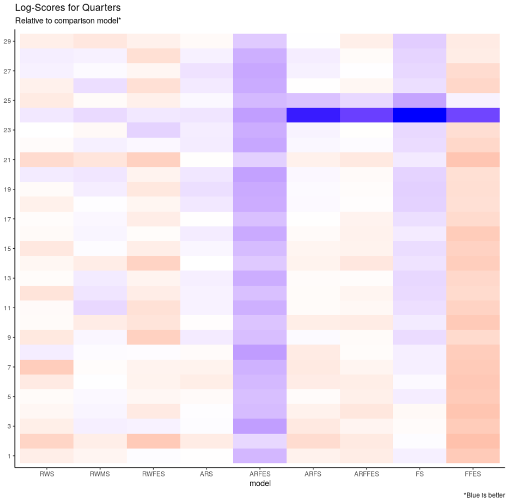

Figure: Log-Scores for Quarters relative to comparison model without text data

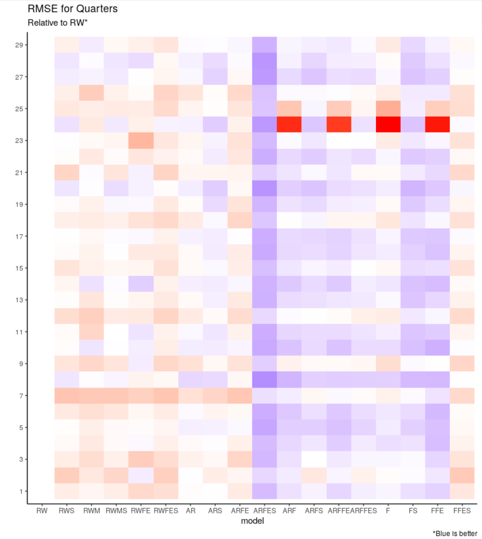

Figure: RMSE for Quarters relative to Random Walk

Relative Improvement

	Log-Scores	
Model	**Without Text**	**With Text**
RW(S)	0.00000	-0.00102
RWM(S)	-0.00122	-0.00081
RWFE(S)	-0.00062	-0.00213
AR(S)	-0.00017	0.00046
ARFE(S)	-0.00131	0.00478
ARF(S)	0.00121	0.00147
ARFFE(S)	0.00103	0.00071
F(S)	-0.00063	0.00235
FFE(S)	0.00187	-0.00157

Table: Log-Scores and RMSE relative to RW

Conclusion & Further Research

- Overall models with text information + AR best
- So mLSTM does extract relevant information
- Shrinkage is major problem for many predictors. Suggestions?
- Texts do not work for all companies. How to identify which?
- Improvement is marginal. We average over quarters.
- How about 1-2 weeks after publishing of report?
- Using only companies whose reports matter?

- Geweke, John and Gianni Amisano (2010). "Comparing and evaluating Bayesian predictive distributions of asset returns". In: *International Journal of Forecasting* 26.2, pp. 216–230.
- Goodfellow, Ian et al. (2016). *Deep learning*. Vol. 1. MIT press Cambridge.
- Krause, Ben et al. (2016). "Multiplicative LSTM for sequence modelling". In: *arXiv preprint arXiv:1609.07959*.
- Loughran, Tim and Bill McDonald (2011). "When is a liability not a liability? Textual analysis, dictionaries, and 10-Ks". In: *The Journal of Finance* 66.1, pp. 35–65.
- Piironen, Juho and Aki Vehtari (2017). "Sparsity information and regularization in the horseshoe and other shrinkage priors". In: *Electronic Journal of Statistics* 11.2, pp. 5018–5051.
- Radford, Alec, Rafal Józefowicz, and Ilya Sutskever (2017). "Learning to Generate Reviews and Discovering Sentiment". In: *CoRR* abs/1704.01444. arXiv: 1704.01444. URL: http://arxiv.org/abs/1704.01444.

Welch, Ivo and Amit Goyal (2007). “A comprehensive look at the empirical performance of equity premium prediction”. In: *The Review of Financial Studies* 21.4, pp. 1455–1508.