An Uncertainty Estimation Model for Algorithmic Trading Agent

Li Rong Wang[1], Thomas C. Henderson[2], Xiuyi Fan[1]

[1] Nanyang Technological University, Singapore
[2] University of Utah, USA

Abstract. We present an uncertainty estimation model for algorithmic trading agents in financial markets. These agents utilise deep learning to analyse market data and identify trading opportunities. Our model allows for optimised trade execution by providing estimated uncertainty scores along with market predictions from deep learning models. Our model is tested in trading with the S&P 500 index. Experiments demonstrate that the estimated uncertainty correlates to prediction accuracy and increased profitability can be obtained with our proposed method.

Keywords: Uncertainty Estimation, Algorithmic Trading Agent

1 Introduction

A trading agent is an intelligent software agent that uses machine learning to facilitate trading in financial markets [5]. They analyse market data, identify trading opportunities, and make trading decisions. Trading agents have become increasingly popular in recent years due to their ability to process large amounts of data quickly and make decisions based on real-time market information.

Uncertainty estimation has been attracting research attention in the machine learning community, especially with the Bayesian learning approach [6]. Having a reliable estimate of uncertainty is crucial for making informed decisions from predictions made by AI systems. This is particularly important in finance, where the decisions can have a significant impact on the end-users.

In trading, uncertainty estimation allows confidence-based decision making [4] that optimises trade execution and risk management. For example, if the algorithm has a high level of confidence in a particular trade, it may allocate more resources to that trade to maximise potential profits. On the other hand, if the confidence level is low, the algorithm may reduce the position size or avoid the trade altogether to minimise potential losses.

Although Bayesian theory offers us mathematically grounded tools to reason about model uncertainty, Bayesian learning approaches usually come with a prohibitive computational cost [3]. More recently, uncertainty estimation directly from neural network models have been explored, e.g., [7] introduce a meta-model based approach for estimation prediction uncertainty.

We present an uncertainty estimation method with an autoencoder and show its performance in trading the S&P 500 index. Our proposed model yields increased profitability when compared against a Bayesian learning approach.

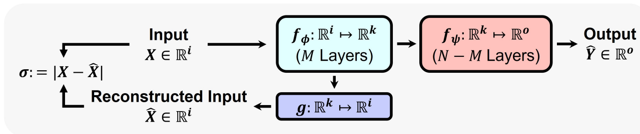

Fig. 1: Uncertainty score model architecture.

2 Uncertainty Estimation Model

To estimate uncertainty, we design an autoencoder-inspired architecture as shown in Figure 1. Consider a multi-layer perceptron (MLP) model $f_{\phi,\psi}$ with $\mathtt{N}$ layers of hidden nodes trained with input X and output Y. The model is parameterised on ϕ and ψ, which denote the first $\mathtt{M}$ and the last $\mathtt{N}-\mathtt{M}$ layers of nodes, respectively. Suppose that $f_{\phi,\psi}$ is a function $f_{\phi,\psi} : \mathbb{R}^i \mapsto \mathbb{R}^o$ and the $\mathtt{M}$th layer contains k nodes, then $f_\phi : \mathbb{R}^i \mapsto \mathbb{R}^k$ and $f_\psi : \mathbb{R}^k \mapsto \mathbb{R}^o$, such that

$$f_{\phi,\psi} = \underset{\phi,\psi}{\arg\min}\, ||Y - (f_\psi \circ f_\phi)(X)||^2. \tag{1}$$

Now, let us consider another MLP model $g : \mathbb{R}^k \mapsto \mathbb{R}^i$, such that

$$g = \underset{g}{\arg\min}\, ||X - (g \circ f_\phi)(X)||^2. \tag{2}$$

We see that f_ϕ and g form an autoencoder for X. We hypothesise that for an instance $\mathbf{x} \in \mathbb{R}^i$ and its corresponding label $\mathbf{y} \in \mathbb{R}^o$, it holds that

$$|\mathbf{y} - f_{\phi,\psi}(\mathbf{x})| \propto |\mathbf{x} - (g \circ f_\phi)(\mathbf{x})|. \tag{3}$$

Thus, we can define

$$\sigma := |\mathbf{x} - (g \circ f_\phi)(\mathbf{x})|$$

as the *uncertainty score* for the prediction $f_{\phi,\psi}(\mathbf{x})$.

In other words, the depicted model architecture provides an uncertainty score in the form of the reconstruction error σ between the actual and reconstructed input. σ is hypothesised to be concordant with the true prediction error (Eqn. 3). This provides an early indicator of a model's performance before knowing the actual outcome. Essentially, we make two assumptions in our model design:

1. a prediction model $f_{\phi,\psi}$ performs well on inputs similar to the training set and poorly on unfamiliar inputs; and
2. the decoder g trained on the same training set reconstructs inputs that are similar to the training set well; thus, achieving low reconstruction errors (small σ). For unfamiliar inputs, the decoder would not be able to reconstruct the input well, resulting in high reconstruction errors (large σ).

Note that unlike a standard autoencoder model where the encoder and the decoder are trained at once, we first train the encoder f_ϕ as part of the prediction model (Eqn. 1). With f_ϕ fully trained, we then train g separately (Eqn. 2). In this way, we ensure that there is no sacrifice to prediction accuracy for $f_{\phi,\psi}$.

3 Experiments

This experiment compares the performance of our proposed model with a state-of-the-art Bayesian learning approach, Gaussian Process Regressor (GPR) [9],

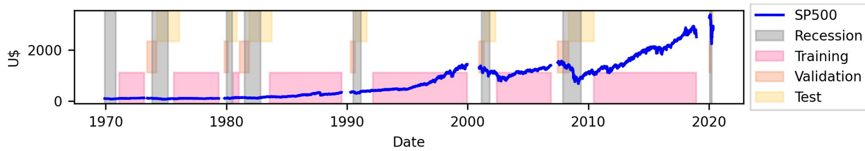

Fig. 2: Training, validation and test data splits (The S&P 500 during 1 December 1969 to 13 May 2020). Each sub-period consists of 3 consecutive data parts.

which has been used to predict prices of different assets [1]. We forecast the S&P 500 with the following features: S&P 500 index (up to one day before the forecasting date), the Fama-French 3 Factors, and Momentum Factor [2]. The data set contains values from 1/12/1969 to 13/05/2020.

As analysed in [8], the US stock market behaves differently between recessions and non-recessions and between different recessions. Thus, the data set is further divided into seven sub-periods, determined by the seven US National Bureau of Economic Research (NBER) recessions as described in [8] (c.f. Figure 2). Each sub-period is treated as a stand-alone data set as each has its own characteristics determined by the market volatility at that time. All models were trained, fine-tuned and evaluated on each of the seven overlapping sub-periods. Data in each sub-period were z-score normalised independently prior to training. We adopt cross-validation to select hyper-parameters achieving the best performance for each model. We tuned the time lag ($l \in [5, 7, 9]$) for both models and the width ($k \in [32, 64, 128]$) for our uncertainty model ($M = 1, N = 2$) with grid search.

For each period, we compare three metrics between our model and GPR:

1. prediction performance: MSE between market prices and predicted prices,
2. uncertainty reliability: correlation between prediction error and uncertainty,
3. model profitability: annualised profits under the ideal parameter.

Profits for each trade on day t were calculated with the following equation:

$$P_t := \mathbb{1}_{\hat{Y}_t > Y_{t-1}}(Y_t - Y_{t-1}) + \mathbb{1}_{\hat{Y}_t < Y_{t-1}}(Y_{t-1} - Y_t). \tag{4}$$

Eqn. 4 states that when the model predicts rising stock prices, the agent buys one unit of stocks on day $t - 1$ and sell it on day t, and vice versa. Consequentially, the derived annualised profits is such that:

$$\text{AnnualisedProfits} := \frac{252}{n} \sum_{t=1}^{n} P_t \tag{5}$$

where n represents the number of invested days. (Note that there are 252 trading days each year.) The agent invests on days when uncertainty scores fall below a threshold. For ease of comparison, we selected uncertainty thresholds generating the highest profits with a minimum investing proportion of 0.05 for both models (trade at least once a month). The results are shown in Table 1.

Our model achieves prediction accuracy comparable to that of GPR for most sub-periods while giving higher concordance between uncertainty estimation and prediction error. This implies that our model has a “better awareness” of its performance than GPR. Consequentially, it is more effective for decision making and yields increased profitability for most sub-periods for both recession and expansion test sets (five and six out of seven sub-periods respectively).

Table 1: Comparison between our model and GPR.

	69-76	**73-80**	**80-83**	**81-91**	**90-02**	**01-10**	**07-20**
Prediction Performance (Test MSE)							
GPR	**0.0245**	**0.0564**	0.2965	**0.0046**	**0.0018**	**0.0206**	0.0477
Our Model	0.1793	0.0924	**0.2683**	0.0053	0.0026	0.0646	**0.038**
Pearson's Correlation							
GPR	0.4091	**0.3505**	**0.6333**	0.0843	0.3084	0.3275	0.4350
Our Model	**0.5279**	0.3286	0.4014	**0.1142**	**0.3831**	**0.5187**	**0.4475**
Recession Model Profitability (Test U\$)							
GPR	74.83	31.46	34.35	15.81	**981.08**	**1086.59**	2052.79
Our Model	**194.23**	**97.27**	**67.90**	**378.63**	667.38	214.74	**20581.68**
Expansion Model Profitability (Test U\$)							
GPR	**56.74**	-23.65	-11.97	151.63	487.83	425.12	6680.02
Our Model	18.00	**177.34**	**5.47**	**268.49**	**1068.48**	**687.93**	**27313.02**

4 Conclusion

In this paper, we present a deep learning uncertainty estimation model for trading agent inspired by the autoencoder model. Given a prediction model, our model uses part of it as the encoder and employs a separate decoder trained with the same data set as the prediction model. For a prediction instance, the reconstruction error of the autoencoder represents the uncertainty score in predicting the instance. To demonstrate the effectiveness of our approach in market prediction, we have experimented our model with the S&P 500 market index. We show that (1) our uncertainty score highly correlates with prediction accuracy, and (2) improved profitability can be obtained with our model. In the future, we will analyse our approach from a Bayesian theoretical perspective to better understand its applicability and limitations.

References

1. Bisht, A., Chahar, A., Kabthiyal, A., Goel, A.: Stock prediction using gaussian process regression. In: Proc. of ICCMC. pp. 693–699. IEEE (2022)
2. Fama, E.F., French, K.R.: Choosing factors. Journal of financial economics 128(2), 234–252 (2018)
3. Gal, Y., Ghahramani, Z.: Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In: Proc. of ICML. pp. 1050–1059 (2016)
4. Khalvati, K., Rao, R.P.: A bayesian framework for modeling confidence in perceptual decision making. Advances in neural information processing systems 28 (2015)
5. LeBaron, B.: Agent-based computational finance. Handbook of computational economics 2, 1187–1233 (2006)
6. Magris, M., Iosifidis, A.: Bayesian learning for neural networks: an algorithmic survey. Artificial Intelligence Review pp. 1–51 (2023)
7. Navratil, J., Arnold, M., Elder, B.: Uncertainty prediction for deep sequential regression using meta models. arXiv preprint arXiv:2007.01350 (2020)
8. Wang, L.R., Fu, H., Fan, X.: Stock price predictability and the business cycle via machine learning. arXiv preprint arXiv submission ID: submit/4825925 (2023)
9. Williams, C.K., Rasmussen, C.E.: Gaussian processes for machine learning, vol. 2. MIT press Cambridge, MA (2006)