Bayesian Ensemble Learning

for Robust Time Series Forecasting

M. Panasiuk
AI Investments
mateusz.panasiuk@aiinvestments.pl

1 Introduction

Ensemble methods are widely used to combine predictions of multiple models describing the same system. They make a foundation of multiple machine learning methods like random forests and gradient boosting machines where multiple weak submodels are combined to create a new, powerful ensemble.

In classical approaches the ensemble remains static - both in terms of the way the submodels are joined and in terms of parameters of the ensemble model. This approach works well if we can assume that the internal dynamics of the modelled system remain constant. While such an assumption can be made in numerous cases - there are systems of vital importance like financial markets or health care where it clearly is not true.

The method presented here is designed to alleviate challenges associated with evolution and lack of stability of the modelled system. It adds a flexible and dynamic component to the way submodels are combined. It allows not only a dynamic assignment of weights depending on submodels' recent performance, but also update of the parameters of the very ensemble as more observations are accumulated. In truly Bayesian fashion it further enables us to get rigorous estimates of predictions' confidence.

2 Methods

2.1 Model Structure

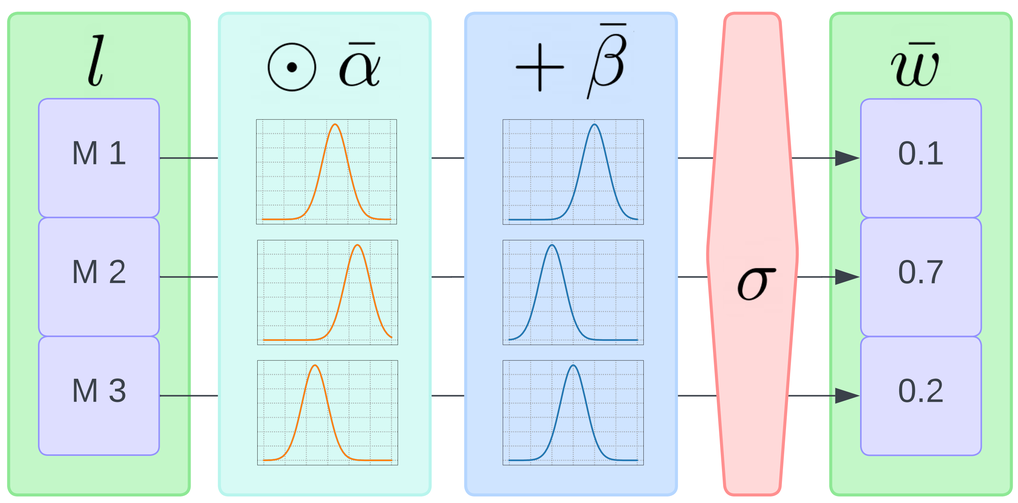
Our primary assumption is that the optimal way of combining submodels' predictions can be based on their past performance. In other words - by keeping track of submodels' errors and making use of it - we can decide what share of votes should be assigned to each of them.

Let $\bar{\alpha}, \bar{\beta} \in \mathbb{R}^m$ be the parameters of the model, with associated standard deviations $\sigma_\alpha, \sigma_\beta \in \mathbb{R}^m_{>0}$. It assigns positive weights summing to one for each of the submodels reflecting their contribution to the final prediction. The assignment of weights is based on the vector of mean past errors generated by each of the submodels $l \in \mathbb{R}^m$.

$$\bar{w} = \sigma(l \odot \bar{\alpha} + \bar{\beta}) \tag{1}$$

$$\hat{y} = \sum_{i=1}^{m} w_i \cdot \hat{y}_i \tag{2}$$

Where σ denotes softmax function and $\odot$ denotes the element-wise (Hadamard) product.



2.2 Update Rule

Bayes theorem provides a handy way of updating parameters of the model based on the new observations, while still taking into account our prior beliefs. In its basic form, it states that probabilities of parameter values given target (posterior) are proportional to probability of the target given parameters (likelihood) multiplied by probability of parameters (prior). In our case it can be expressed as:

$$\underset{\text{posterior}}{p(\bar{\alpha}, \bar{\beta}|y)} \propto \underset{\text{likelihood}}{p(y|\bar{\alpha}, \bar{\beta})} \cdot \underset{\text{prior}}{p(\bar{\alpha}, \bar{\beta})} \tag{3}$$

To maximise the probability of parameters we need to optimise for both parts of the equation - logarithm of likelihood and prior. Likelihood for binary target is a well known log loss as in 4 and prior with normal assumption can be expressed as 5, where $\theta = [\alpha, \beta]$ represents the concatenated parameter vector.

$$\log P(\hat{y}|\theta) = \sum_{i=1}^{N} (y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)) \tag{4}$$

$$\log P(\theta) = -\frac{1}{2} \sum_{k} \frac{(\theta_k - \mu_k)^2}{\sigma_k^2} + \text{const} \tag{5}$$

Sum of the above losses is minimised using conventional gradient based optimisation with every new observation. It has to be noted that it is a quick process taking only a few epochs due to limited number of parameters and relative simplicity of the function. In this context the prior distribution can be thought of as representing a memory of what the model experienced and the likelihood part may be imagined as representing the immediate experience. The goal of the process is to find parameters balancing both.

2.3 Data and Submodels

Five distinct submodels were used in ensemble - modified versions of Time Series Mixer, Deep Coupling and Transformer neural networks. The task was to predict direction of price change in the next ten hour period. Experiments were conducted for 59 foreign exchange instruments with sampling frequency of one hour.

3 Results

We have compared test performance of our ensemble against averaged prediction of all three networks treated as a benchmark. For 43 of 59 (73%) financial instruments our method provided lower error with mean improvement of 5.4% compared to the benchmark. A binomial test with $p = 0.5$ shows that this outcome is statistically significant ($p = 1.9 \cdot 10^{-4}$), suggesting that the method provides a measurable improvement over baseline performance. The effectiveness of the proposed improvement was further confirmed by a Wilcoxon signed-rank test conducted on paired error, yielding $p = 1.3 \cdot 10^{-5}$.

4 Conclusion

Bayesian approach to online ensemble learning presented in this work was demonstrated to yield meaningful improvement in significant number of examined time series. It confirms our initial intuition and provides an argument to further pursue elaborations of this concept. Natural next steps include considering different priors for the model and extension of the model to include non-linear functions (eq. 1) and dynamics or error change.

ADVISOR
Co-funded by EUREKA member countries and the European Union Horizon 2020 Framework Programme