1. Abstract

Goal: effectively regularise (stop overfitting) time-series models and how to adaptively tune forgetting rates (in case of .

2. Introduction

Time series data analysis is of foremost importance for forecasting, anomaly detection and decision making. However, learning from time-series data is challenging because of non-stationarity and a presence of structural breaks within the real world data(e.g. Stock market). A common approach in learning from time-series data is to gradually forget the past data and adapt the newest, but there are two problems with implementing controlled information decay.

Firstly, there is a requirement for a method to set and adaptively tune the forgetting rate (if we don't forget enough we cannot adapt to changes, if we forget too much, we might overreact to noise).

Second, there is a requirement for appropriate regularisation, since the size of the effective data is small when we forget the past and there is a high risk of overfitting.

Therefore, in the article Second Order Techniques for Learning Time-series with structural breaks Takayuki Osogami has proposed a novel technique for adaptively changing the regularisation strength and the forgetting rate in O(n^2) complexity in order to account for changes posed by structural breaks.

*It is difficult to optimise weights of a linear model with L2 normalisation (as discussed in the 2010 article). Additionally, the L2 norm is sensitive to coordinate transformations (coordinates may change scale as time-series advances). Thus Osagami proposed his own regularizer which overcomes these obstacles and works in O(n**2).

The main contributions of this article are thus are:

1) Ensemble learning method that uses the "following best hyper forgetting rate" to adaptively tune hyperparameters of a non-stationary time-series.
2) Regularisation that is invariant to linear transformations.

3. Problem

Forgetting rate for the L2 model is not a hyperparameter and therefore cannot be changed after every time step. So if the structural brakes are present in the time-series then the model can either not adapt to the rapid changes or overestimate the future values.

4. Regularized Recursive Least Squares:

This section is based on a study of how to adaptively choose gamma and lambda in O(n**2). These hyperparameters determine the loss function and thereby minimizers give the least predictive error per step. The values of lambda and gamma may change over time due to structural breaks, thus we would need to recompute g and H. To do so we train a number of models such that they minimize our function with varying forgetting rate and regularization coefficient. We then track the CSR of the prediction given by each model discounted by the forgetting rate (eta).

Generally, to learn a non-stationary time series we first make a prediction y_hat_t = f_/theta_t(x_t), where x_t is a feature vector of observations before t. We then observe y_t that occurred in reality and use its value to update the parameters /theta_(t+1) and use them to make a consecutive prediction regarding y_hat_(t+1).
In order to find the optimal parameter weights /theta* We use a WMSE method. {**Insert formula for WMSE**}. // Why WMSE??? → V krasnom opisanno.

The minimizer of WMSE is given by /theta_(t+1) = H_t ^-1g_t where g_t is the negative gradient at the origin and H_t is the Hessian.
{**Insert formulas for g_t and H_t**}

We can use the Sherman-Morrison lemma to compute the H_t^-1 in O(n**2) where n is the dimension of x_t. (tak-je est' method of using pseudo-inverse when matrix H_t is not invertible).

The standard L2 regularisation has the following form. {Insert L2 regularisation}. Where the minimizer is given by {insert formula for theta} and {Insert formula for Hessian}. The inverse of such a matrix cannot be computed recursively. (since it's not a Hessian).

Therefore the authors propose the following loss function {insert new L2 regularizer}. Where minimizer is given by {insert minimization}. → Imenno formula 7 since it does not take into account and regularise the intercept.

Two important properties about the regularizer are proved:
The authors thus propose and proof the following lemma about the O(n^2) complexity of their algorithm.

Additionally they propose and proof a lemma about the invariance to linear transformations of their regularizer.

5. Following best Hyper Forgetting Rate:

Since the optimal parameters can be revamped in O(n^2), the next step is to find the optimal values of gamma and lambda. The paper proposes to train a small number of models with different forgetting rate gamma and regularization-coefficient lambda for each model. Then the tracked cumulative squared error(CSE) helps to pick the best model for the next step.

$$\mathrm{CSE}_t^{(i)}(\eta) \equiv \sum_{d=0}^{t-1} \eta^d \, (\hat{y}_{t-d}^{(i)} - y_{t-d})^2 \qquad (8)$$

$$= \eta \, \mathrm{CSE}_{t-1}^{(i)} + (\hat{y}_t^{(i)} - y_t)^2. \qquad (9)$$

, where eta is the hyper forgetting rates which are recommended by the paper in the range from 0.9 to 1.
Then the minimum of all CSE is selected and it is considered to be the best performing model at t + 1.

6. Experiments:

Experiments were conducted to investigate three important questions:

1) How does the proposed regularization compare against the L2 regularization?
2) Can the proposed algorithm adaptively tune hyperparameters?
3) How does the proposed approach compare against existing methods for predicting non-stationary time series?

Question 1:
~ Refer to Osagami 2020.
Conclusion: Overall there is a favourable conclusion about the effectiveness of the proposed regularisation. Although the effectiveness depends on particular data, experiments show that the proposed regularisation works in O(n^2) time with the expected effect of regularisation. Additionally, on certain occasions it outperforms the L2 regularisation.

Question 2:
To answer this question Osogami has used synthetic 2000 point time series with changepoints. The time-series were generated according to an AR model and a structural break was set at point t=1000 and two different statistical properties of the time-series before and after the changepoint.
The time-series is learnt using the proposed algorithm and its predictive error is compared against the baseline with fixed gamma and lambda.
Osagami has plotted two graphs showing RMSE 100 steps before and after the structural break figures 2a and b. Overall the graphs portray that algorithm 1 performs as well as the best performing static choice of gamma and lambda.
Finally, figure 2c shows the values of gamma and lambda as the time-series progresses. We can therefore observe how gamma and lambda change as time-series moves further away from a structural break. (Dostatochno medlenno converges btw).

Question 3:
To answer this question the proposed algorithm was compared against existing ML methods for non-stationary time-series. 10 year data of historical stock price for indices were used. For each time-series Osagami calculated the absolute value of the daily return {insert formula} and predicted this return for the consecutive day. A figure that compared the proposed algorithm against vSGD,

Almeida, HGD, Cogra, Adam, RMSProp and Adagrad. The MSE was normalised relative to that of the naive prediction that the absolute daily return stays unchanged from the previous day.
The conclusion is that the algorithm gives significantly smaller predictive errors for all financial indices and all the baselines under considerations.
However, this outperformance could have been caused by minimization of the WMSE with the hyperparameters that minimize the CSE and the hyperparameters which minimize CSE. But the other baselines do not minimize this metric. Therefore no conclusion may be drawn (Check other metrics).
Furthermore, the proposed algorithm requires additional computational cost than baselines. (7.1 seconds for order 12 model against less than a second for the other models.)

.

Conclusion:

It was empirically shown that the proposed algorithm outperforms the L2 regularization for time-series with **drastic variability.** In addition, it can easily tune hyperparameters after each time step and can go beyond the baselines in forecasting financial time-series.