Ensembling a Linear Regression Model with an Error Mitigation Component

Submitted: 8th March 2018; accepted: 27th March 2018

Artur Nowosielski, Piotr A. Kowalski, Piotr Kulczycki

DOI: 10.14313/JAMRIS_1-2018/5

Abstract:
This paper presents a proposal of a model error mitigation technique based on the error distribution analysis of the original model and creating the additional model that tempers the error impact in particular domain areas identified as the most sensitive. Both models are then combined into single ensemble model. The idea is demonstrated on the trivial two-dimensional linear regression model.

Keywords: *ensemble model, error mitigation, regression model, linear regression, FPA, RSS, RSE*

1. Introduction

Error measurement is one of the fundamentals of mathematical modelling. By definition, error is a measure of a modelled parameter value disturbance from its expected value. The expected value is a theoretical value for the infinite population, which means it's impossible to get its exact value. For the machine learning purposes trained with a finite collection of samples, expected values are calculated from that sample set and model error is measured against that empirical expected value. Depending on particular modelling technique, model may have systematic error, that is, error dependant on one of the input parameters. Such an error is sometimes referred to as bias or skew. Also, model may yield higher error in some particular ranges of input parameter domains. In such cases, model can be extended by a component that mitigates the error impact, that also depends on the input parameter or parameters. Of course, such a component may be included directly to the original model. However, there is a variety of cases when this should not or could not be done. For example, model may be closed, immutable component, provided by some external service or a legacy one. Also, recalculating the whole model may be expensive in terms of computing power or data may start to get burden with a skew after the model got trained. The already running model can be hard to reconfigure on production environment or its reconfiguration may cause downtime whereas it may be required to work with no down-time, for example because of the service level agreements. For those reasons, this paper presents alternative approach: building a separate error model and combining it with the original model in the ensemble model, that sums the output of the actual phenomenon model with a fix provided by the error mitigation model. It needs to be stated that presented approach is just a problem mitigation rather than fix for the root-cause. This is just another model and may have the same problems as any other model. However, thanks to being limited to subset of input variables and being focused on another dimension of the modelling goal (difference between training sample instead of absolute value) there are cases when it performs well. Model ensembling has proven to be an effective way of combining multiple models for the sake of increasing the output accuracy over the single-technique models [2] [3]. However, a typical use case is to combine multiple models made with different techniques and then judge which output is the best or aggregate all the outputs into single model response, for example by taking average, weighted average or sum of multiple components. This approach is popular in combining classification and clusterization models.

2. Model

In order to present the proposed approach, a simple linear regression model will be discussed briefly. The bike sharing system data set [1] is used. The set presents a number of a municipal bike rentals (registered and occasional users) in the Washington metropolitan area. Input parameters include weather conditions (temperature, wind, humidity) and time of year and day. Floating point parameters are normalized to a range of $[0, 1]$. The data file includes 17379 entries representing hourly registered data points. As there is no outliers nor incomplete entries, each record contains 16 features. Output variable is a total count of bike rentals, that spans both casual and registered users. For the idea demonstration, a single input variable of hours of day is used.

Model accuracy is measured with two common error metrics: the Residual Sum of Squares (or sum of squared errors; RSS) and the Residual Standard Error (RSE). The RSS is more convenient to use as the optimization goal for the algorithm, however its values are unintuitive when it comes to human interpretation, because they are orders of magnitude higher than actual output values. It is calculated as follows:

$$RSS = \sum_{i=0}^{n}(\epsilon_i) = \sum_{i=0}^{n}(r_i - m_i)^2, \tag{1}$$

where $\boldsymbol{r}$ is the bike rentals vector and $\boldsymbol{m}$ is the vector of model output values calculated for the same input data as corresponding r_i samples. The RSE is calculated on

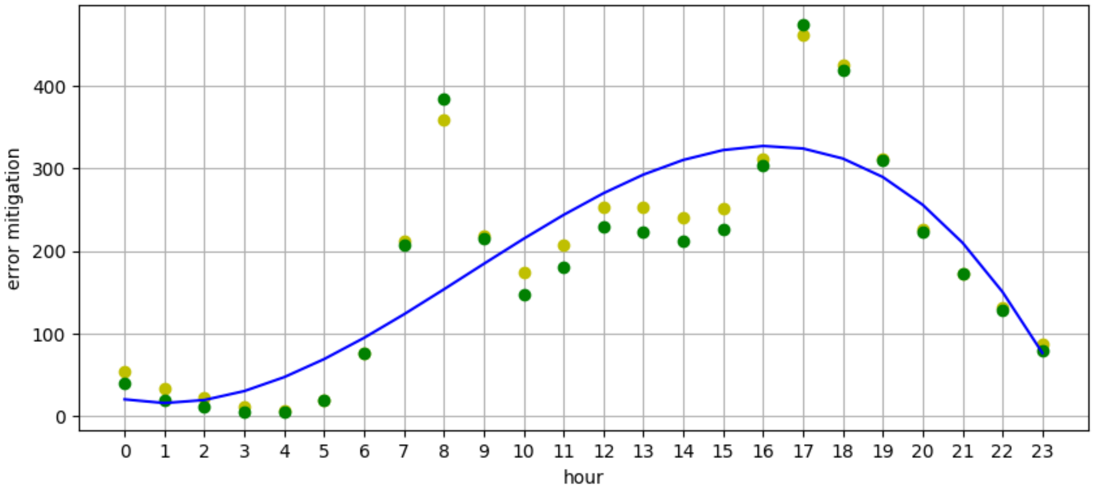

Fig. 1. Average and median bike rentals count and modelling curve

the basis of the RSS with the following formula:

$$RSE = \sqrt{\frac{RSS}{n-p-1}} \qquad (2)$$

Where n is a number of samples and p is a number of optimized parameters in a model, that is regression equation coefficients. The whole $n-p-1$ value is often referred to as a number of degrees of freedom.

Figure 1 presents average and median number of rentals by each hour (yellow and green dots, respectively), as well as a curve representing the model output and the RSE error per each hour (red dots). The model is a linear regression model calculated as a 3rd order polynomial and thus has four parameters. The following equation expresses a general model formula:

$$f_m(x) = w_3x^3 + w_2x^2 + w_1x + w_0 \qquad (3)$$

where ***w*** if the coefficients vector. Coefficients are optimized with the Flower Pollination Algorithm [4] and the formula with supplied coefficients is:

$$\begin{aligned} f_m(x) = &-0.18048221x^3 \\ &+ 4.647178x^2 \\ &- 8.96642054x \\ &+ 20.47717682 \end{aligned} \qquad (4)$$

The RSE of this model is 122.58.

3. Error Model

The first step in error distribution analysis is a visual assessment of error metric value. Figure 2 presents a curve representing a number of rentals per each hour of day estimated by the model together with red dots marking the RSE value per hour. It is visible that error is relatively higher in rush hours, that is, at 7 - 8 am and 5 - 7 pm.

In case of more sophisticated models with multiple input variables, they do not have equal impact on the output variable. More formally, input variables' impact can be compared by calculating and comparing some kind of an input importance measure. The measure depends strongly on selected modelling method. For example, in a model where all the inputs are embedded linearly into a model equation, input importance can be directly inferred by taking an absolute coefficients values. Similarly in neural networks, where a notion of input weight is one of fundamental concepts.

Similarly as in the actual domain modelling, two approaches can be distinguished when it comes to error modelling. The first one is based on a detailed analysis of the error distribution over the input parameters. This approach is useful when error distribution can be easily aligned to a common known function, such as a logarithm or linear. The second one is a black-box approach, where error distribution is not a subject of a detailed analysis, but a metaheuristic algorithm is applied to align best function or best coefficients to a predefined class of functions, for example polynomial function.

Error value distribution analysis lets to choose which input variables should be involved in the error model. However, both error metrics discussed previously are mean metrics, which means they miss important information about a sign of the difference between empirical samples and estimated value and also about the sign of the estimated value itself. Another critical question while considering a proper function for the purpose is choosing an appropriate benchmark, that is whether it should refer to average values of the training samples set or to the median or, possibly, some other measure. This decision depends on the model output value variance. In case of highly-variable values, any central tendency measure may be inappropriate and quartiles or n-th deciles could work better.

Error mitigation function has multiple desired features. Obviously, it should decrease error value at least at some sensitive points, while not increasing it at the same time throughout whole domain. The function should not fit to the training data too precisely. Too strict alignment to the training data may result in yielding worse results when applied to a real data. There are overfitting-prevetion techniques, such as cross-validation that can be used for both submodels.

As stated previously, error is higher in rush hours. Example error mitigation component formula is:

$$f_e(x) = \begin{cases} 0, & \text{if } x \in [0,7) \\ -200(x-8)^2 + 250, & \text{if } x \in [7,10) \\ -40, & \text{if } x \in [10,14) \\ -60, & \text{if } x \in [14,17) \\ 100, & \text{if } x \in [17,19) \\ 0 & \text{if } x \in [19,23) \end{cases} \tag{5}$$

where x is the input variable. Ranges, constant values and function coefficients were chosen arbitrarily. Both models are in fact independent of each other and can be created using any technique. Figure 3 displays the error mitigation function graph.

4. Ensemble Model

When the error model is ready it needs to be combined with the original model of the discussed phenomenon. There are multiple ensembling methods. However, in the discussed case, both sub-models have clearly defined roles. The initial model is responsible for estimating the actual result and is focused on dealing with all the input data. There is also the error model, that tries to mitigate the original model's skew and it estimates the error, not the value itself. That makes most of commonly used techniques, such as voting, stacking, blending or bucketing unapplicable for the purpose. The presented application uses simple sum function that sums the basic model output and the error mitigation model output. The general formula is:

$$f(X) = f_m(X) + f_e(X) \tag{6}$$

where $f_m()$ is the original model, $f_e()$ is the error model and thus $f()$ is the ensemble model. $\boldsymbol{X}$ is the input samples vector.

The RSE of the ensemble model is 114.22, and is 7% lesser than RSE of the initial model.

5. Summary

This paper presented an idea of ensembling a linear regression model together with additional model that decreases average error in particular parts of the input variable domain by adding/subtracting a constant or function depending on input variable value to/from an estimated output value. The trivial two-dimensional example is used to demonstrate the idea.

AUTHORS

Artur Nowosielski* – Findwise Sp. z o.o., 00-023 Warsaw, Poland, e-mail: artnowo@gmail.com, www: Findwise.

Piotr A. Kowalski – AGH University of Science and Technology, 30-059 Cracow, Poland, e-mail: pakowal@ibspan.waw.pl, www: Faculty of Physics and Applied Computer Science.

Piotr Kulczycki – AGH University of Science and Technology, 30-059 Cracow, Poland, e-mail: kulczycki@ibspan.waw.pl, www: Faculty of Physics and Applied Computer Science.

*Corresponding author

REFERENCES

[1] H. Fanaee-T and J. Gama, "Event labeling combining ensemble detectors and background knowledge", *Progress in Artificial Intelligence*, 2013, 1–15, 10.1007/s13748-013-0040-3.

[2] A. Janusz, T. Tajmajer, and M. Świechowski, "Helping AI to Play Hearthstone: AAIA'17 Data Mining Challenge". In: M. Ganzha, L. Maciaszek, and M. Paprzycki, eds., *Proceedings of the 2017 Federated Conference on Computer Science and Information Systems*, vol. 11, 2017, 121–125, 10.15439/2017F573.

[3] Q. H. Vu, D. Ruta, and L. Cen, "An ensemble model with hierarchical decomposition and aggregation for highly scalable and robust classification". In: M. Ganzha, L. Maciaszek, and M. Paprzycki, eds., *Proceedings of the 2017 Federated Conference on Computer Science and Information Systems*, vol. 11, 2017, 149–152, 10.15439/2017F564.

[4] X.-S. Yang. "Flower Pollination Algorithm for Global Optimization". In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, volume 7445 LNCS, 240–249. 2012.

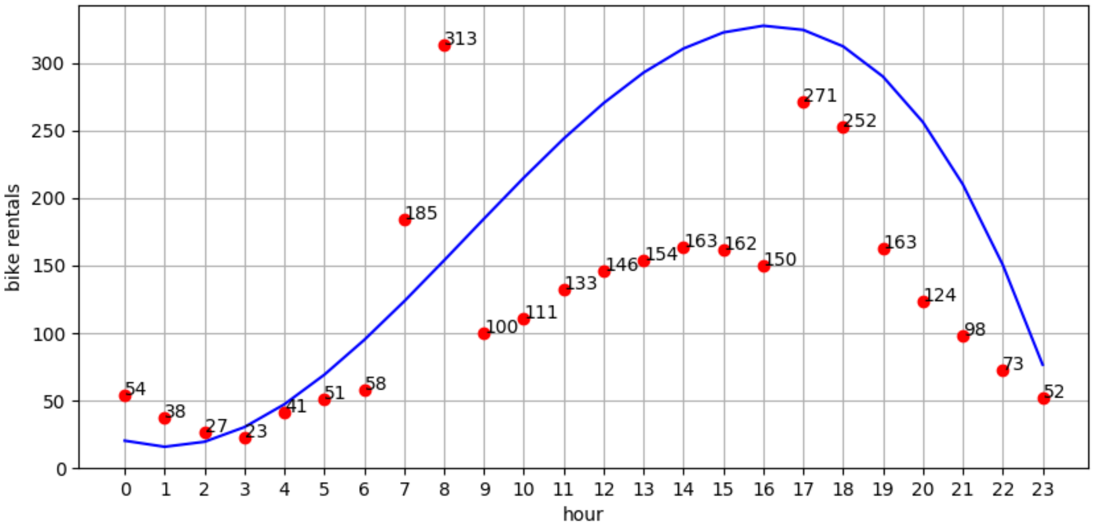

Fig. 2. RSE per each hour and modelling curve

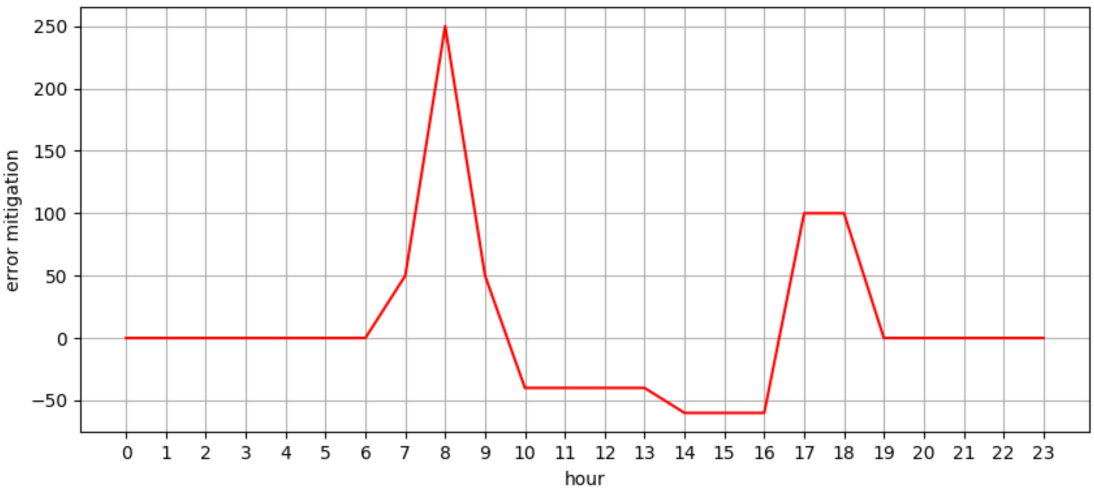

Fig. 3. Error mitigation function plot

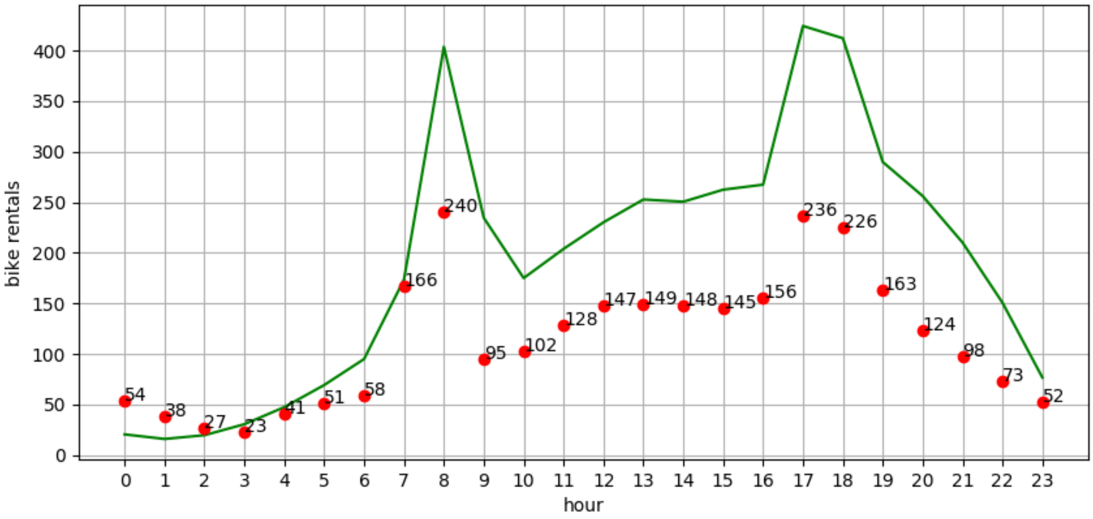

Fig. 4. Ensemble model curve and RSE values per hour of day