ATLANTIS PRESS

Atlantis Highlights in Engineering (AHE), volume 2

International Conference on Industrial Enterprise and System Engineering (IcoIESE 2018)

Model Tree with Modified L1 Loss Function for Predicting Missing Attendance Data of Faculties

Mohammad Arif Rasyidi
Department of Informatics
Universitas Internasional Semen Indonesia
Gresik, Indonesia
mohammad.rasyidi@uisi.ac.id

Rachmadita Andreswari
Information System Department
Telkom University
Bandung, Indonesia
andreswari@telkomuniversity.ac.id

***Abstract*— The problem of missing attendance data in our university often arises due to the negligence of faculties. In this study, we address the problem by directly predicting the work duration of faculties. The nature of the problem require us not only to make accurate predictions, but also to minimize the rate of overestimation. To address the problem, we propose the implementation of model tree with modified L1 loss function and simple prediction result reduction. Experimental results show that our proposed method is able to lower the overestimation rate while maintaining accuracy within acceptable range.**

Keywords—model tree; loss function; prediction; attendance

I. INTRODUCTION

It is a common practice for organizations to require their employees to meet certain work hours or duration, and Universitas Internasional Semen Indonesia (UISI) is no exceptions. To monitor the attendance, employees in UISI are required to report their attendance by clocking in and out on the provided fingerprint scanners. Contrary to staffs that are expected to be punctual, faculties' attendances in UISI are monitored by their working hours or durations. These durations can be obtained by subtracting clock in time from clock out time for each faculty.

One of the problems faced by UISI is that employees, especially faculties, often fail to report their attendance either by clocking in or out because of their' negligence as well as other problems, such as device and electrical failures. When either clock in or clock out time is missing, we cannot calculate the daily work duration of the corresponding faculty. These attendance records missing either clock in or clock out times make up a large proportion of our attendance data, and therefore they cannot be left untreated.

Some ways to mitigate this problem are by setting default clock in/out times, setting a fixed daily work duration for faculties, predicting the missing data, or ignoring the problem altogether by considering the faculty to be absent from work. In this study, we try to solve the problem by predicting the missing data. Instead of predicting the missing clock times, we try to directly predict the work durations using the known clock times (either in or out) as well as other attributes, such as day of week, month, and other time related features. This problem falls into a type of machine learning task called regression: the problem of predicting numerical values using some known available data. Many prediction methods have been developed and widely used in practice for doing regression, for example linear regression [1], nearest neighbors [2], artificial neural network [3], and model tree [4], [5]. In this study, we employ linear regression and model tree, a tree and linear-based model, to tackle the problem of predicting work duration of university faculties. These methods have been extensively used in practice for solving numerical prediction problems in various fields, e.g. traffic prediction [6], [7], road accident prediction [8], and water level forecasting [9]. One specific feature of our problem is that accuracy is not the main purpose. Average error rate of under an hour is considered acceptable as per university requirement. However, the predictions should be made so that they should not overestimate the actual work duration as much as possible. Overestimation is undesirable since the fault of missing clock data often lies in the negligence of faculties and therefore it is better to underestimate than to overestimate the work duration while still keeping prediction accuracy in check.

The rest of this paper is organized as follows. Section II will give brief overview of prediction algorithms employed in our study. Our proposed method is described in section III. In section IV, we describe our experiment setup. Experimental results are presented and discussed in section V. Finally conclusion is given in the final section.

II. BRIEF OVERVIEW OF PREDICTION ALGORITHMS

A. Linear Regression

Linear regression as illustrated in Fig. 1 is one of the most popular prediction methods in machine learning because of its simplicity. It is parametric and easy to implement. In linear regression models, the target value is expressed as linear combination of the attributes or features with some predetermined weights:

$$y = f(\mathbf{x}) = w_0 + w_1x_1 + w_2x_2 + \ldots + w_jx_j \quad (1)$$

where y is the target value, $\mathbf{x} = (x_1, x_2, \ldots, x_j)$ is an instance vector with j attributes, w_0 is intercept (constant), and $\mathbf{w} = (w_1, w_2, \ldots, w_j)$ is the weight vector.

One of the common method to build linear regression model is the ordinary least square method first published by Legendre [10] and Gauss [11] which finds function $\hat{f}$ that minimizes the sum of squared error (L2 loss function):

$$\hat{f} = \arg\min_{f} \sum_{i} L_2(y_i, f(\mathbf{x}_i)) . \quad (2)$$

This problem can be solved analytically and a unique global solution exists.

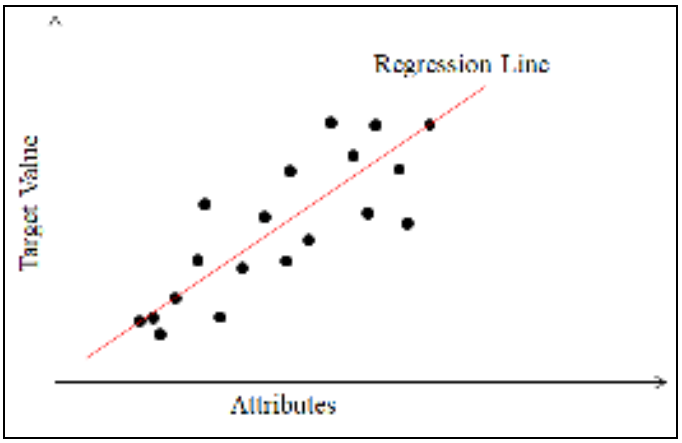

Fig. 1. Illustration of Linear Model

B. Model Tree

Decision tree is a type of predictive model that has been commonly used for both classification and regression (numerical prediction). Decision trees for regression generally can be differentiated into two categories: regression trees and model trees. Contrary to regression trees that have constants at their leaves, model tree is a type of decision tree that has nontrivial model at its leaves as illustrated in Fig. 2. In this study, we are employing M5, an algorithm developed by Quinlan [4] and improved by Wang and Witten [5] to build model trees with linear regression models at their leaves. Similar to the way decision trees for classification are build, M5 model tree uses reduction of variance to choose the best split in each node. Once the target values of all examples that reach certain node only very slightly vary or only few examples are remained, the split are terminated. A linear regression model (as explained in the previous subsection) is then built using those examples and placed at the leaf. Model simplification, pruning, and smoothing are also performed in order to avoid overfitting as well as discontinuity of adjacent linear models. This makes model trees generally superior to linear regression model since it can handle nonlinearity very well. For details on how to build the M5 model tree, one may refer to [4], [5].

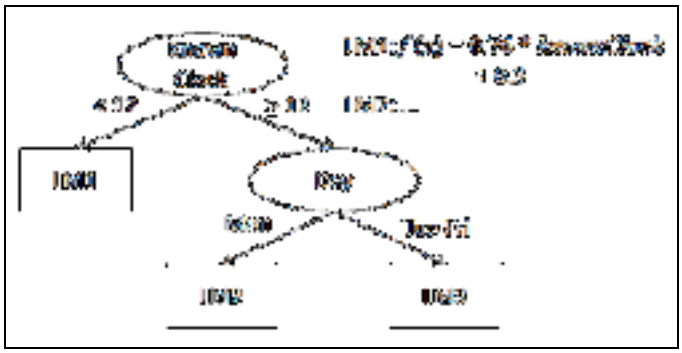

Fig. 2. Example of a model tree

III. Proposed Method

In this study, we propose three methods to handle our problem of making accurate prediction while minimizing number of estimation. Three methods are discussed below: prediction result reduction, loss function modification, and combination of both of them.

A. Prediction Result Reduction

Fig. 3 shows two regression lines with some training examples. Examples that fall below the line are those that are overestimated, while examples above the lines are those that are underestimated. Intuitively, we can push the regression line down at the cost of accuracy by simply reducing the prediction result or model intercept by some amount δ. The final prediction can then be calculated as

$$y = \max(f(\mathbf{x}) - \delta, 0) . \quad (3)$$

This method effectively lower the regression line, puts more examples above the line, and thus lowering the number of overestimation. This method is simple, it can be implemented for all prediction models, and it does not require modification to the way we build our prediction model.

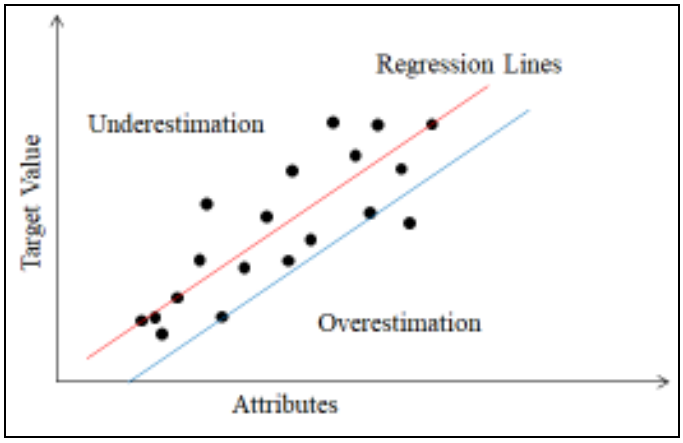

Fig. 3. Illustration of two regression lines. The blue line has lower overestimation rate than the red one.

B. Loss Function Modification

Traditionally, single linear regression model, as well those that are used in M5 model tree are built by minimizing the sum of squared error or L2 loss function:

$$L_2(y, f(\mathbf{x})) = (y - f(\mathbf{x}))^2 . \tag{4}$$

Variants exist, such as least absolute deviation that minimizes sum of absolute error or L1 loss function:

$$L_1(y, f(\mathbf{x})) = \left| y - f(\mathbf{x}) \right| . \tag{5}$$

These loss functions however, are symmetric. They punish overestimation and underestimation equally which is not suitable for our purpose.

To accommodate our problem, in this study, we use a modified L1 loss function to direct our prediction algorithm to produce models that tends to avoid overestimation. The idea is that we punish overestimation more than we do underestimation. Hence, our modified loss function is as follows

$$L_{1M}(y, f(\mathbf{x})) = \begin{cases} y - f(\mathbf{x}), & y \geq f(\mathbf{x}) \\ k(f(\mathbf{x}) - y), & \text{otherwise} \end{cases} \tag{6}$$

where k denotes a positive constant greater than or equal to 1. When k equals to 1, this is the same as least absolute deviation, while when k is less than 1, overestimation is more preferred than underestimation which is contrary to what we are trying to achieve.

Building the linear model is then straightforward. Any linear programming technique can be used to obtain the weight vector and intercept that minimize our loss function:

$$\sum_{i=1}^{n} L_{1M}(y_i, f(\mathbf{x}_i)) = \sum_{i=1}^{n} L_{1M}(y_i, \mathbf{w} \cdot \mathbf{x}_i) \tag{7}$$

which is equal to

$$\text{Minimize} \sum_{i=1}^{n} e_i \tag{8}$$

with respect to $\mathbf{w}$, u_1,..., u_n, subject to

$$\begin{aligned} e_i &\geq y_i - \mathbf{w} \cdot \mathbf{x}_i \quad \text{for } i = 1 \text{ to } n \\ e_i &\geq k(\mathbf{w} \cdot \mathbf{x}_i - y_i) \quad \text{for } i = 1 \text{ to } n \end{aligned} . \tag{9}$$

C. Combination of Loss Function Modification and Prediction Result Reduction

Reduction of prediction results can be employed in conjunction with loss function modification. Predictions that are made using models built with our custom loss function can be lowered further by reducing them with some specified amount.

IV. EXPERIMENTS

A. Data Preparation

The datasets that we use in this study are attendance logs of 64 faculties in UISI from October 1, 2016 to March 31, 2017. We take examples that are complete, meaning that there are no missing values. We divide every example into two: one with known clock in data, and another with known clock out data. We set the known clock in and out as the same feature, added several time related features such as day of week, day of month, month, and year, and set the daily work duration as the target attribute. This resulted in 64 datasets (one for each faculty, containing his/her own attendance data) with a total of 12650 examples.

B. Prediction Methods

We implement original linear regression (LR) and M5 model tree as well as our three proposed methods: reduction of prediction results, loss function modification, and their combination for each datasets. For comparison purpose, we also implement baseline predictor: historical average prediction model (HA). Historical average, as the name suggests, simply make prediction by returning the average work duration of the same day of week:

$$\overline{WorkDurati\,on}(t) = \frac{1}{w} \sum_{i=1}^{w} WorkDurati\,on(i,t) \tag{10}$$

These prediction methods are coded in Java as RapidMiner [12] extensions. For modified LR and M5, we extend the Weka [13] extension available in Rapidminer. Apache Commons Math [14] is also used for doing linear programming when building models with loss function modification.

C. Performance Measurement

To evaluate the performance of our predictors, we perform ten-fold cross validation. There are three performance measures that we use: mean absolute error (MAE) shown in (11), overestimation rate, and average of overestimation. The lower their values, the better the performance of the models.

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^{n} \left| y_i - f(\mathbf{x}_i) \right| \tag{11}$$

V. Results and Discussion

A. Performance of Ordinary Prediction Models

The results obtained by our prediction models without any modification are shown in Table I. Out of the three methods we test, model trees perform best while linear regression models are the worst. The performance of linear regression is even worse than our base predictor, historical average. As can be seen, all prediction methods are able to produce acceptable accuracy, showing average error under one hour. However, as we have suspected previously, the percentage of predictions that overestimate the work duration of faculties is noticeably high, which is undesirable as per our requirement.

TABLE I. Performance Comparison of Ordinary Predictors

Prediction Method	MAE	% Overestimation	Avg. of Overestimation
HA	43.23	35.26%	21.62
LR	45.16	36.62%	22.58
M5	38.38	35.29%	18.99

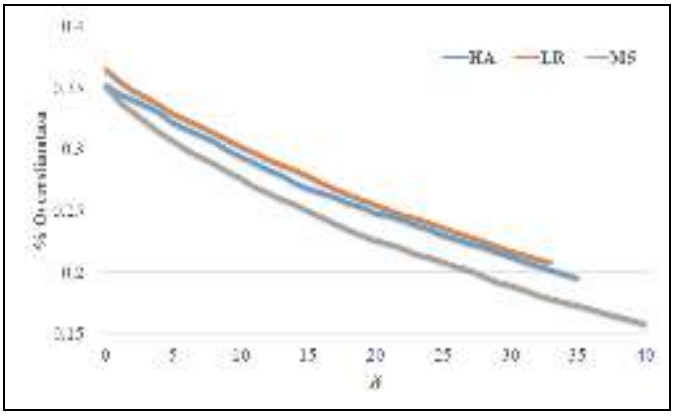

Fig. 4. Overestimation rate of ordinary prediction models with various amount of prediction reduction.

B. Performance of Models with Prediction Result Reduction

Fig. 4 shows the change of overestimation rate of our ordinary predictors with different amount of prediction result reduction δ. Only values that have MAE under one hour are displayed. We can see that once again, model tree models managed to lower the overestimation rate better than the other models.

C. Performance of Models with Modified Loss Function

Table II shows the performance comparison of model trees and linear regression models with modified L1 loss function. We can see that as the rate of overestimation decreases, the error rate increases. It is also inferred from Table I and II that when the error levels are similar, our proposed method managed to produce lower overestimation rate. The best performance is once more shown by M5 where it can decrease overestimation rate to 16.81% while keeping accuracy within acceptable range.

TABLE II. Performance Comparison of Prediction Models with Modified L1 Loss Function

k	MAE		% Overestimation		Avg. of Overestimation	
	LR	M5	LR	M5	LR	M5
1	40.77	35.62	44.95%	42.73%	29.32	23.33
2	45.63	40.06	32.01%	30.53%	19.94	16.27
3	54.04	46.74	**24.85%**	23.46%	**14.65**	12.62
4	*61.50*	53.43	*20.25%*	19.23%	*11.29*	10.05
5	*69.04*	59.05	*17.19%*	**16.81%**	*9.34*	**8.59**
6	*75.19*	*63.80*	*14.78%*	*14.40%*	*8.03*	*7.33*
7	*80.82*	*68.73*	*13.12%*	*12.93%*	*6.95*	*6.64*
8	*86.22*	*72.49*	*11.88%*	*12.11%*	*6.14*	*5.98*
9	*90.44*	*76.06*	*11.08%*	*10.77%*	*5.58*	*5.36*
10	*94.59*	*79.47*	*10.13%*	*10.06%*	*5.10*	*4.99*

D. Performance of Models with Combination of Loss Function Modification and Prediction Result Reduction

Fig. 5 and 6 show the overestimation rate of linear regression models and model trees with different values of k and δ (note that LR-i and M5-i both denote modified linear regression models and model trees with $k = i$). Here, we can see that both methods produce lowest overestimation rate when $k = 2$. We can also see that the higher the value of k, the less amount of reduction is required to lower the overestimation rate.

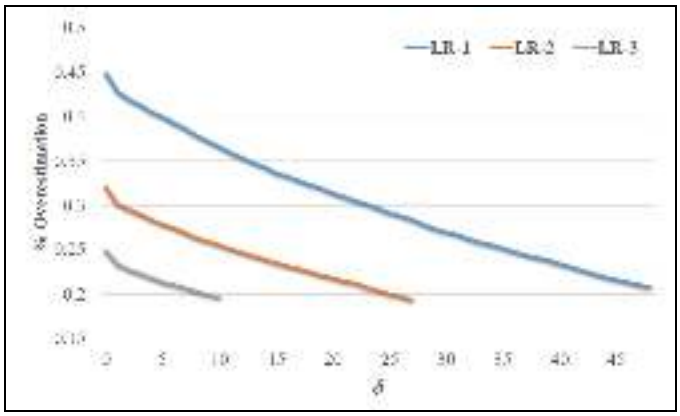

Fig. 5. Overestimation rate of linear regression models with modified L1 loss function with various amount of prediction reduction.

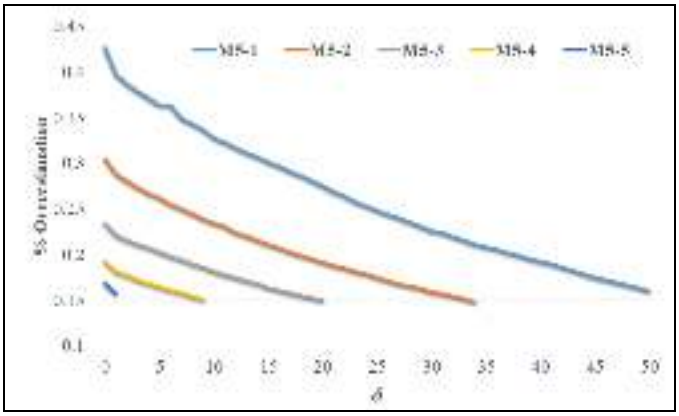

Fig. 6. Overestimation rate of model trees with modified L1 loss function with various amount of prediction reduction.

E. Performance Comparison

We compare the performance of proposed methods in Table III. We can see that when using L1 loss function modification, model trees deliver the lowest average amount of overestimation. Although the overestimation rate is higher than that of prediction result reduction, when it is combined with prediction results reduction, better overestimation rates are obtained.

TABLE III. PERFORMANCE COMPARISON OF PROPOSED METHODS

Method	Prediction Models	δ	k	% Over-estimation	Avg. of Overestimation
Pred. Result Reduction	HA	35	-	19.60%	12.34
	LR	33	-	20.86%	13.47
	M5	40	-	***15.85%***	9.55
L1 Modifi-cation	LR	-	3	24.85%	14.65
	M5	-	5	16.81%	**8.59**
Reduction + L1 Mo-dification	LR	27	2	19.33%	13.34
	M5	34	2	**14.84%**	***9.15***

VI. CONCLUSION

The problem of predicting missing work duration of university faculties requires us to minimize overestimation while still keeping prediction accuracy in check. In this study, we have proposed the implementation of model tree with modified L1 loss function and simple prediction result reduction to address the problem. The experimental results show that our proposed method manage to lower the overestimation rate to as low as 14.84% from the ±36% rate previously obtained by ordinary prediction models while maintaining accuracy within acceptable range. In future works, we will incorporate more data and try to identify whether there are some patterns or similarities of the work behavior of the faculties. We will also investigate whether we can achieve better result by grouping or clustering faculties based on their attendance behaviors and use their data instead of individual data for training the prediction models.

REFERENCES

[1] J. A. Hanley, "Simple and multiple linear regression: sample size considerations," *J. Clin. Epidemiol.*, vol. 79, pp. 112–119, Nov. 2016.

[2] M. Wauters and M. Vanhoucke, "A Nearest Neighbour extension to project duration forecasting with Artificial Intelligence," *Eur. J. Oper. Res.*, vol. 259, no. 3, pp. 1097–1111, Jun. 2017.

[3] M. Mordjaoui, S. Haddad, A. Medoued, and A. Laouafi, "Electric load forecasting by using dynamic neural network," *Int. J. Hydrogen Energy*, vol. 42, no. 28, pp. 17655–17663, Jul. 2017.

[4] J. R. Quinlan, "Learning with continuous classes," in *Proceedings of the Australian Joint Conference on Artificial Intelligence*, 1992, pp. 343–348.

[5] Y. Wang and I. H. Witten, "Induction of model trees for predicting continuous classes," in *Poster papers of the 9th European Conference on Machine Learning*, 1997.

[6] M. A. Rasyidi and K. R. Ryu, "Short-Term Speed Prediction on Urban Highways by Ensemble Learning with Feature Subset Selection," in *Database Systems for Advanced Applications*, W.-S. Han, M. L. Lee, A. Muliantara, N. A. Sanjaya, B. Thalheim, and S. Zhou, Eds. Springer Berlin Heidelberg, 2014, pp. 46–60.

[7] M. A. Rasyidi and K. R. Ryu, "Comparison of Traffic Speed and Travel Time Predictions on Urban Traffic Network," in *2014 IEEE/ACS 11th International Conference on Computer Systems and Applications (AICCSA)*, 2014, pp. 373–380.

[8] G. Singh, S. N. Sachdeva, and M. Pal, "M5 model tree based predictive modeling of road accidents on non-urban sections of highways in India," *Accid. Anal. Prev.*, vol. 96, pp. 108–117, Nov. 2016.

[9] M. Rezaie-balf, S. R. Naganna, A. Ghaemi, and P. C. Deka, "Wavelet coupled MARS and M5 Model Tree approaches for groundwater level forecasting," *J. Hydrol.*, vol. 553, pp. 356–373, Oct. 2017.

[10] A. M. Legendre, *Nouvelles méthodes pour la détermination des orbites des comètes*. Paris: Didot, 1805.

[11] C. F. Gauss, *Theoria Motus Corporum Coelestium in Sectionibus Conicis Solem Ambientum*. Hamburg: Perthes, 1809.

[12] O. Ritthoff, R. Klinkenberg, S. Fischer, I. Mierswa, and S. Felske, "Yale: Yet Another Learning Environment." 2001.

[13] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The WEKA Data Mining Software: An Update," *SIGKDD Explor.*, vol. 11, no. 1, pp. 10–18, 2009.

[14] C. M. Developers, "Apache Commons Math, Release 3.6.1." 2016.