Association for Information Systems

AIS Electronic Library (AISeL)

International Conference on Information Systems 2020 Special Interest Group on Big Data Proceedings

Special Interest Group on Big Data Proceedings

12-14-2020

Automated feature engineering based on Bayesian optimization for time series

Konstantin Danilov

Automated feature engineering based on Bayesian optimization for time series

Konstantin Danilov , National Research University Higher School of Economics, Lead data scientist, Moscow, Russia, danilovkostya@yandex.ru

Abstract

Nowadays, the forecasting time series task is relevant in solution of a wide range of problems in various spheres of human activities. One of the possible variants to provide prediction is to construct a forecasting model. The main criterion for the forecasting model quality is its accuracy. Researchers have resorted to different approaches to achieve the necessary accuracy of the forecasting model, including feature engineering.

This paper presents an automated feature engineering method based on Bayesian optimization for time series data. The process of selection an optimal set of features in order to minimize the objective function is described. The developed method has an ability to create new features based on existing ones by using diverse algebraic operations. The proposed method considers any machine learning model as a black box, that allows applying different algorithms: linear regression, decision trees, neural networks, etc.

The experiments demonstrated the high efficiency of the proposed approach. A comparative analysis showed that the developed algorithm in most cases was superior to human-made custom feature engineering. The accuracy of machine learning models is greatly improved with high-quality feature engineering. Mean squared error and coefficient of determination were applied to calculate quality metrics of machine learning models.

Testing the developed method took place on open time series data from different subject areas (energy, manufacturing, air pollution), which provided reliable verification.

Keywords — automated feature engineering, Bayesian optimization, forecasting time series, automated machine learning

Introduction

Feature engineering is the process of an extraction features from raw data via data mining techniques. It plays a significant role in machine learning algorithms because generated features can be used to improve model performance [1]. In this research paper the novel feature engineering method based on Bayesian optimization for time series forecasting task is proposed.

Time series forecasting task is an import part of machine learning and it is a relevant problem in various fields. Analysis of economic, social, geophysical and manufacturing processes involve the data with time instants. Nowadays, the concept of the Internet of things is gaining popularity and significance. Consequently, it generates enormous amounts of information, usually presented in the form of time series.

In addition to it, the forecasting task arises when it is necessary to create automatic control systems and decision support systems. Prediction methods are widely used to study systemic relationships and patterns of object functioning. They are also an extremely important tool in the analysis of complex applied systems, work with information, and targeted human exposure to research objects in order to increase the efficiency of their functioning.

The task of forecasting time series can be solved by creating a forecasting model that adequately describes the studied process. The forecasting model is a functional dependence that approximates the considered time series with some error.

At present, there are many models targeting at forecasting time series: regression and autoregressive models, neural network models, models of exponential smoothing, models based on Markov chains, models based on decision trees, etc. More than that, there are some tasks where a combination (ensemble) of different models provides the ability to achieve a more accurate forecast of time series. [3,4]

Most machine learning models are based on generated features that were obtained by means of raw data exploration. Therefore, the process of creating feature space is an essential and important step for building a machine learning model.

In many cases, scientists try to generate features based on domain knowledge during the machine learning model construction. The range of domain knowledge is different in each case and, consequently, the quality of the generated set of features also varies. It has a direct impact on the quality metrics and performance of machine learning model [1].

The transformation of the existed features in order to boost the accuracy of the machine learning model is also included in feature engineering process. The result of the feature engineering step is a feature space which provides the best performance of the forecasting model.

In this paper, a novel method for feature engineering based on Bayesian optimization for solving the problem of forecasting time series is proposed.

Problem formulation

The word ‘forecast’ derives from the Greek ‘ρόγνωσις’, which means foresight, prediction of what is likely to happen in the future, based on the information that you have now. Forecasting is understood as prediction with the help of scientific methods. The forecasting is a special scientific study of the specific prospects for the development of a process. According to [2], processes which must be predicted are most often described by time series, that is, by a sequence of values of certain quantities obtained at certain points in time. The time series includes two mandatory elements - the time stamp and the value, which is obtained in one way or another and corresponding to the specified time stamp. Each time series is considered to be a selective implementation from an infinite population generated by a stochastic process, which is influenced by many factors [2].

Time series can have different nature of origin, which contributes to the acquisition of individual characteristics. As a rule, when analyzing a time series, four of its components are distinguished [2]:

1) Trend. A smoothly changing component that describes the proper influence of long-term factors (population growth, change in the structure of age composition, etc.)

2) Cycling. The time series component describes relatively long periods of rise or fall. It consists of cycles that vary in amplitude and length.

3) Seasonality. A component of a time series describes a behavior that changes regularly over a given period of time (year, month, week, day).

4) Noise. The random component of the time series, which includes the remaining unexplained fluctuations in the time series.

Most time series forecasting models try to take into account the components listed above for more efficient operation. As a rule, the procedure for a time series forecasting model can be decomposed into the following components [3]:

1. Time series preparation. Classic approaches to improve data quality are applied. For example, processing outliers and missing values in time series.

2. Feature engineering. This stage is one of the key one, since the construction of the feature space is formed where a forecasting model will be created on. Features can be nominal, binary, categorical, ordinal, etc.

3. Feature selection. There is a filtering and selection of features to build a forecasting model.

4. Training and testing of the forecasting model. Training and testing of the forecasting model on prepared features is performed.

5. Tuning hyperparameters. This stage includes the selection of the necessary hyperparameters of the forecasting model in order to achieve a more accurate forecast.

6. Validation of the forecasting model. The constructed model is verified on a holdout sample of data.

The set of values $\{Y_1, Y_2, Y_3, ...Y_T\}$, which is represented as the values of a parameter varying in time, is called a time series. In this context each value corresponds to the value of the parameter at a particular time $\{t_1, t_2, t_3, ...t_n\}$. The time series whose values must be predicted is called the target time series.

Let's assume that there are values of the target time series at discrete time moments $t = 1,2,...,T$, and there is also a set of features, whose values are available at discrete time instants $t = 1,2,...,T$ and are represented as time series. Let's denote the target time series $Y(t) = Y_1, Y_2, Y_3, ...Y_t$ and the set of features $X(t) = X_1(t), X_2(t), X_3(t), ...X_m(t)$, where $X_m(t)$ − feature, t − time moment, m − number of features.

It is supposed that there is the following functional dependency between the feature set and the target time series:

$$Y(t) = F(x_1(t), x_2(t), ..., x_m(t)) + \varepsilon_t \tag{1}$$

Dependency (1) is called a forecasting model. At time T, values of the target time series $Y(t)$ are determined at time instants $T+1, T+2, ..., T+K$. The value $K (K \in R)$ sets the prediction depth. The time moment T is called the forecast moment [2].

It is required to find the optimal set of features where the deviations between the real values of the target time series and the predicted ones tends to the minimum for a given K. The deviations can be calculated as a mean squared error:

$$\sum_{i=1}^{m} \frac{1}{m} \left(Y_i(t) - \tilde{Y}_i(t) \right)^2 \rightarrow \min_t, \tag{2}$$

where $Y_i(t)$ is the actual value of the target time series, $\tilde{Y}_i(t)$ is the predicted value of the target time series.

Automated machine learning pipeline

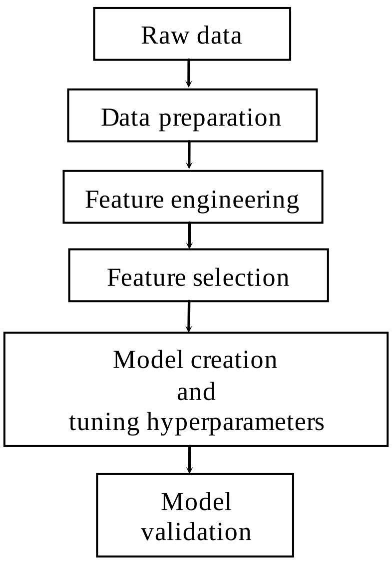

Fig. 1. Machine learning pipeline

The basic machine learning pipeline is demonstrated above (Fig. 1). The input (raw) data comes in a structured form with marked features and a target variable. The data packet is delivered to the preparation step, where the problem of detecting and processing outliers / missing values is solved. The data preparation phase may complete the automatic machine learning process if invalid gaps or large numbers of outliers are detected. Features are generated on pre-processed data where there are no any outliers and missing values. The structure of the incoming information at the stage of data preparation is the same to the creation of additional features. The feature engineering block provides intellectual feature transformation in order to improve the performance of the defined machine learning model [4, 5].

After that, the data goes to the feature selection stage, where the most important features are searched for solving the assigned problem. The analysis of various machine learning algorithms and the selection of the most optimal one are executed at the model creation and tuning hyperparameters step. Model validation is an additional verification of constructed model on a holdout data. It helps to prevent overfitting or underfitting [7, 9].

Data preparation

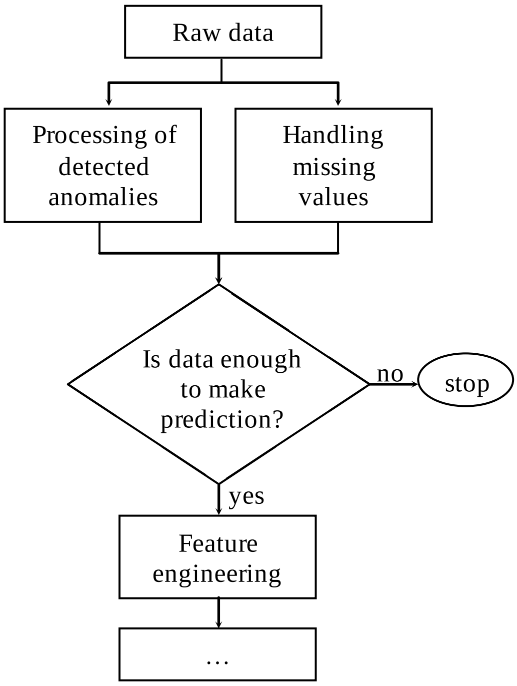

Fig. 2. Data preparation in automatic mode

The data preparation process is one of the most important one in automatic machine learning, as it makes adjustments to the incoming information. This stage is rather difficult to automate, since raw data requires custom processing. The main components of data preparation are processing of detected anomalies and handling missing values. Undoubtedly, these two steps are not the main ones, but they vitally affect the final quality of the machine learning model. Handling missing values has a strong impact on the next stage (feature engineering), because the application of methods such as rolling window, taking lag value, trigonometric transformation are highly dependent on the missing values [8, 11].

The result of data preparation is a constructed dataset that will be used in the subsequent steps. At this point, it is also possible to make adjustments to the target variable, if such a need arises. As a rule, the chain of actions at the data preparation stage is based on an explanatory analysis.

Feature engineering and feature selection

As a rule, data scientists use the accumulated experience within the scope of feature engineering and rely on domain knowledge. For sure this stage is one of the key one, as it affects the performance of the machine learning model. There are various approaches to the transformations of feature space on time series. The most popular of them are the following [10]:

- Taking lag values
- Application of trigonometric functions
- Multiplication of binary features
- Differentiation of lag values
- Taking the seasonal component of a time series
- Taking the trend component of a time series
- Rolling window
- Raising to the Nth power

After generating features, there is a step which includes an assessment of the each feature importance on the target variable. Various metrics can be used to assess the impact of a feature on a target variable: Pearson correlation, Spearman correlation, statistical tests, feature importance based on decision trees, etc.

This article proposes a novel method for the feature engineering based on Bayesian optimization. The algorithm of its operation will be described in detail below. The verification results of the proposed method confirmed its effectiveness.

Model creation and tuning hyperparameters

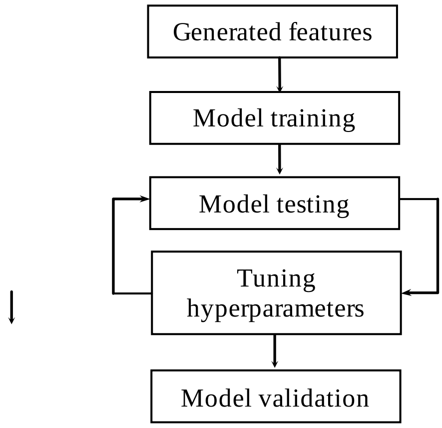

Fig. 3. Model creation and tuning hyperparameters

This stage is rather successfully automated, since the steps are known and transparent. It is necessary to divide the dataset into three parts: training, testing and validation. The training dataset will be used in the learning process, the testing one is used to evaluate the accuracy of the trained model and finally the validation dataset allows checking the adequacy of the selected hyperparameters. Undoubtedly, there are some more techniques for testing the model to avoid overfitting or underfitting, such as cross-validation.

The architecture of the model can be based on various techniques: decision trees, neural networks, linear regression, etc. The interpretability of model can be achieved using such approaches as SHAP (SHapley Additive exPlanations).

Most models have sets of hyperparameters that are needed to tune for more efficient and optimized performance. In order to achieve this goal, such methods as Grid search, Random search, Bayesian optimization are applied.

As a result of this stage, a trained machine learning model with tuned hyperparameters is obtained. After that, it is needed to test the model by using a validation dataset. This process provides reliable verification.

Proposed feature engineering method

The proposed feature engineering method is based on the optimization model which tries to minimize the deviations between predicted and actual values by means of predefined possible feature transformations. Schematic diagram of the method is depicted in figure 4.

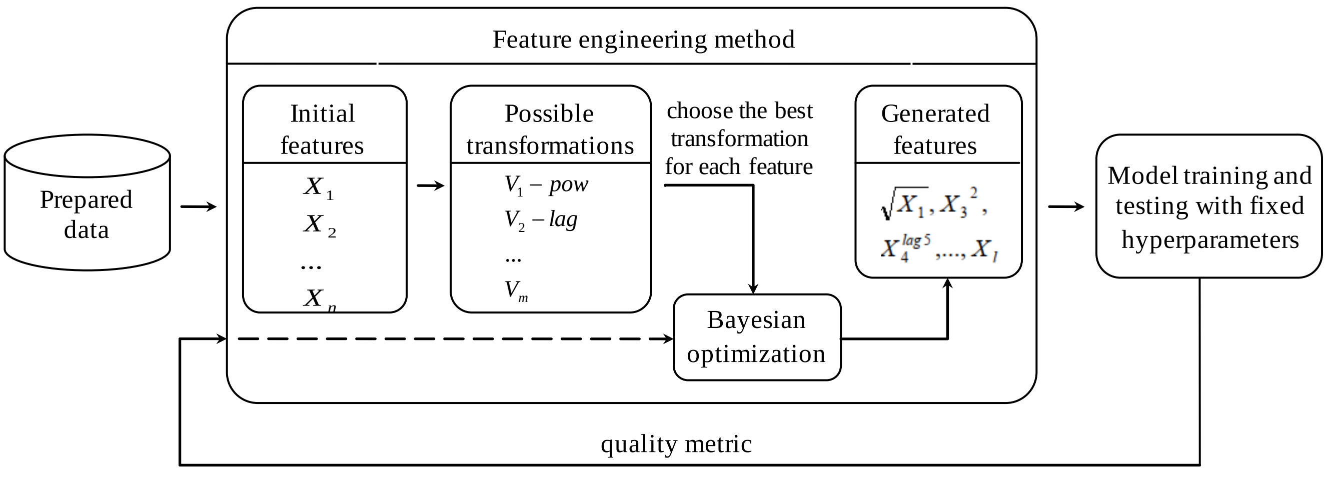

Fig. 4. Feature engineering method

The feature generation method consists of the following main blocks: initial features, possible transformations, optimization model, and generated features. Initial features are a feature set that has been manually generated based on a domain expertise. Handling missing values and processing of detected anomalies have to be performed on the previous data preparation step. All features should be represented as a time series and available at the same time moments as a target variable.

Possible transformations are a set of transformations that can be applied to initial features. They are defined by human and cannot be changed during the optimization process. For example, possible transformations may be the following: number of lag value, Nth power to raise, trigonometric functions, multiplications of features, size of rolling window, etc.

Optimization model minimizes the predefined objective function. The objective function represents a quality metric of a machine learning model. Usually, it can be mean squared error, mean absolute error, coefficient of determination, etc.

Optimization model is based on the Bayesian theory. The main advantage of the Bayesian approach to the optimization problem is the ability to work with objective function as a black-box. It doesn't need to know how the objective function looks like. It manages this by constructing its own (surrogate) model of the objective function. Bayesian optimization is also fairly robust to noisy objective function evaluations.

Quality metric and prepared data come to the input of feature engineering method. The output of the feature engineering method is a set of generated features, which can be used in machine learning model creation. Optimization process is iterative and requires model training and testing each time. Generally, fifty iterations are enough to provide effective optimization and improve machine learning model performance.

Architecture of the model and its hyperparameters should be fixed. The algorithm of the feature engineering method is as follows:

1. Perform explanatory data analysis and prepare data to construct initial features.
2. Define target variable and check the availability of features at the same time moments as a target.
3. Split data and train/test machine learning model with default hyperparameters.
4. Find the best hyperparameters for machine learning model to fix them in optimization process.
5. Provide some definitions before optimization process:
    a. Possible transformations for each feature
    b. Optimization model parameters. The meaningful parameters for optimization model are a number of iterations and the initial inputs to start optimization process.
    c. Quality metric for machine learning model.
6. Run optimization process with defined number of iterations
7. Verify the set of generated features

Thus, Bayesian optimization helps to generate new features, applying possible transformations to the initial ones. For example, new feature X^N can be constructed based on initial feature X and transformation "raise to power". Frequently, the main problem is finding the best N to achieve better result when the process is carried out manually. Proposed feature engineering method tries to solve this problem by using optimization model.

The result of the feature engineering method is a set of features that provides the best quality metric for machine learning model with fixed hyperparameters.

Experiments

Verification of the proposed feature engineering method was carried out on 3 different datasets [6]: Air quality, appliances energy prediction and SML. These data were taken from open sources.

Three persons were involved who solved the time series forecasting task. Each of these people has enough knowledge about machine learning model construction and feature engineering process.

It was decided that the first person would apply linear regression to create machine learning model, second one –gradient boosting, third one – neural networks.

Testing the machine learning model was carried out using the cross-validation method. The mean squared error was used as a quality metric. The table below reflects the experiment results.

Table 1. Result of experiments

Person	Dataset	Model	Human-made feature engineering (MSE)	Proposed approach with 50 iterations (MSE)
1	Air quality	Linear regression	6.71	3.65
2	Air quality	Xgboost	0.69	0.67
3	Air quality	Neural network	1.32	1.15
1	Appliances energy prediction	Linear regression	12.51	9.6
2	Appliances energy prediction	Xgboost	5.32	3.89
3	Appliances energy prediction	Neural network	5.6	5.57
1	SML	Linear regression	9.8	3.9
2	SML	Xgboost	0.98	0.87
3	SML	Neural network	0.65	0.74

The experimental results demonstrate the effectiveness of the proposed feature generation method. In many cases, the optimization model was able to generate such features that that it turned out to surpass the human-made feature engineering. The addition of new features played a significant role in constructing the linear regression model. Models based on neural networks and gradient descent also showed a positive result when using the generated features by the proposed method.

Conclusion

Feature engineering step is one of the most important parts of machine learning pipeline during the model construction. The proposed feature engineering method has the ability to work automatically.

According to the results of the experiments, it may be concluded that the proposed feature engineering method demonstrates high efficiency and surpasses human-made feature engineering in most cases.

It is a well-known fact that models based on linear regression are often applied to solve a wide range of problems, since they are interpretable and can be expressed in the form of an understandable formula. The proposed feature engineering method allows linearizing feature space and achieves better accuracy.

Certainly, it is necessary to increase the number of persons and expand the list of used datasets for a more reliable testing of the method. In the development of the method, it seems possible to use the analysis of residuals, which will make feature engineering process more intellectual.

The main disadvantage of the method is the need to retrain the model at each iteration, which can take a lot of time. Nevertheless, fifty iterations were enough to find the optimal set of features during the experiments.

References

[1] Gilad Katz, Eui Chul Richard Shin, and Dawn Song, “Explorekit: Automatic feature generation and selection”, In 2016 IEEE 16th International Conference on Data Mining (ICDM), IEEE, 2016, pp. 979–984

[2] George E. P. Box, Gwilym M. Jenkins, Gregory C. Reinsel, Greta M. Ljung, “Time Series Analysis: Forecasting and Control”, June 2015, pp. 406-409

[3] Morariu N., Iancu E., Vlad S., “A neural network model for time series forecasting”, Romanian Journal of Economic Forecasting. 2009, No. 4, pp. 213 – 223.

[4] Udayan Khurana, Deepak Turaga, Horst Samulowitz, Srinivasan Parthasrathy, “Cognito: Automated Feature Engineering for Supervised. Learning”, 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW). 12-15 Dec. 2016, pp. 2-4

[5] James Max Kanter, Kalyan Veeramachaneni, “Deep feature synthesis: Towards automating data science endeavors”, 2015 International Conference on Data Science and Advanced Analytics (DSAA), 2015, pp. 1-3

[6] Machine learning repository, 14.02.2020, [Online], Available: https://archive.ics.uci.edu/ml/datasets.php

[7] Franziska Horn, Robert Pack, Michael Rieger, “The autofeat Python Library for Automated Feature Engineering and Selection”, unpublished

[8] Andrew Y Ng., “Feature selection, l1 vs. l2 regularization, and rotational invariance”, In Proceedings of the twenty-first international conference on Machine learning, ACM, 2004, p. 78

[9] Udayan Khurana, Horst Samulowitz, and Deepak Turaga. Feature engineering for predictive modeling using reinforcement learning. In Thirty-Second AAAI Conference on Artificial Intelligence, 2018

[10] Miron B Kursa, Witold R Rudnicki, “Feature selection with the boruta package. J Stat Softw”, 2010, pp. 1–13,

[11] Shaul Markovitch and Dan Rosenstein, “Feature generation using general constructor functions”, Machine Learning, 2002, pp. 59–98