

# ClassCast: A Tool for Class-Based Forecasting

Florian Heimgaertner, Thomas Sachs, and Michael Menth

University of Tuebingen, Chair of Communication Networks,  
Sand 13, 72076 Tuebingen, Germany  
{florian.heimgaertner,menth}@uni-tuebingen.de,  
thomas.sachs@student.uni-tuebingen.de

**Abstract.** We present ClassCast, a tool for class-based forecasting. It partitions an input time series into class-specific time series. It uses conventional prediction methods for each of these time series and maps class-specific values to classified future time indices. It is a simple means to account for non-linear factors in time-series for the purpose of prediction.

**Keywords:** Forecasting, prediction, tool

## 1 Introduction

Various approaches have been discussed for load forecasting in the domain of electrical energy [1, 2]. Especially methods based on machine learning, such as artificial neural networks [3] or support vector machines [4], have attracted interest. However, when looking at time series  $y_j$  of energy consumption of administrative buildings, a clear dependency on working and non-working days and on the weather can be observed. Therefore, a simple forecast method accounting for these influencing factors can outperform sophisticated forecast methods neglecting that influence. This basic forecasting concept is known as similar-days method [5]. A very simple forecast is the average energy consumption specific for working and non-working days. The average prediction may be further adapted to cold and warm working and non-working days. We propose class-based forecasting as a generalization of the similar-days method. Cold/warm and working/non-working are an example for a classification  $c_j$  of the time index  $j$ . Besides basic calendar information, additional data in the time series that is known for both the past and the future can be used for classification, e.g., scheduled operating times of specific devices. Essentially, we compute class-specific means  $\bar{Y}(c)$  with  $c \in \mathcal{C}$  and  $\mathcal{C}$  being the class range. These class-specific means may be used for prediction  $y_j = \bar{Y}(c_j)$  of future time indices  $j$  which are classified as  $c_j$ .

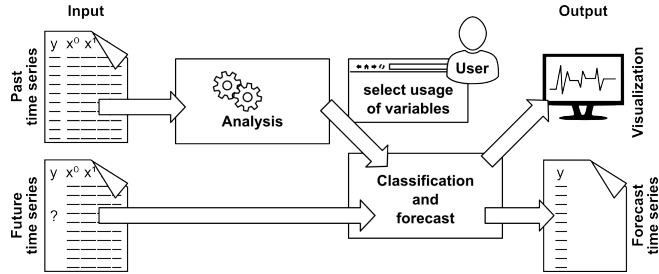


Fig. 1. Operation of ClassCast.

This class-based forecast method can be adapted to more complex forecasting methods. We exemplify this for linear regression. It predicts future values  $y_j$  with a dependence on a regressor  $x_j$ :  $y_j = \beta_0 + \beta_1 \cdot x_j$ . Class-specific forecast implies that linear regression is applied to a class-specific subsets of the time series with  $(y_j^c) = (y_j)_{j:c_j=c}$  for  $c \in \mathcal{C}$ . Based on those time series, class-specific forecast models  $\{(\beta_0^c, \beta_1^c) : c \in \mathcal{C}\}$  are derived for each class  $c \in \mathcal{C}$  which predict  $y_j = \beta_0^{c_j} + \beta_1^{c_j} \cdot x_j$  for  $j : c_j = c$ .

We have developed ClassCast [6], a tool for class-based forecast supporting prediction based on mean values and linear regression.

This work is structured as follows. Section 2 provides a high-level description of ClassCast. Section 3 explains the concept of the tool and describes the classification and forecasting mechanism. In Section 4, we present implementation details and the user interface of ClassCast. Section 6 concludes the paper.

## 2 Tool Description

The input of ClassCast is an annotated time series  $y_j$ . The data series consists of past values of the *dependent variable*, i.e., the variable to be forecasted. Annotations are values of one or multiple *independent variables*  $x_j^i$ ,  $0 \leq i \leq n$  that are supposed to influence the values of the dependent variables. An example for a dependent variable is the total energy consumption while independent variables could be day of week, time of day, outside temperature, or the state of a monitored device.

The second part of the input is the future time series. The future time series consists of values for the independent variables but does not contain values for the dependent variable. The independent variables of the future time series are used to forecast values of the dependent variables using the approaches presented in Section 3.

Figure 1 depicts the operation of ClassCast. Past time series  $y_j x_j^i$  and future time series  $x_j^i$  are supplied to the tool. For each independent variable of the input time series, the user selects whether it is discarded, used as classifier, or (if applicable) as regressor. This selection determines the forecasting method. If one or multiple independent variables are selected as regressors, linear regression is used for forecasting. Otherwise, class-specific averages are used for forecasting.

The future values for the dependent variables  $y_j$  are computed based on the combination of values for the independent variables  $x_j^i$  of the future time series. The forecast can be saved to a file or visualized for further analysis.

### 3 Concept

The past data set for the forecast consists of the time series  $(y_j)_{j=s, \dots, -1}$  where  $y_j$  are the dependent variables and  $j$  are the time series indices. The index  $s < 0$  denotes the start index of the time series and is negative as it relates to the past. Additionally, the input data set contains  $n$  series of independent variables  $(x_j^i)_{j=s, \dots, -1}^{i=0, \dots, n-1}$ . The future data set starts at the index 0 and contains values for the independent variables  $(x_j^i)_{j=0, \dots, m}^{i=0, \dots, n-1}$  where  $m$  denotes the forecasting horizon. ClassCast uses information from the past and future data set to compute the forecast time series  $(y_j)_{j=0, \dots, m}$ .

The basic concept of ClassCast is classifying time series entries according to the values of the independent variables. The independent variables are elements of an  $n$ -dimensional annotation space  $\mathcal{X} = \mathcal{X}_0 \times \dots \times \mathcal{X}_{n-1}$ . A function  $f : \mathcal{X} \mapsto \mathcal{C}$  maps the annotations to a  $q$ -dimensional class space  $\mathcal{C} = \mathcal{C}_0 \times \dots \times \mathcal{C}_{q-1}$ . The mapping function  $f$  of our tool is limited in the way that any annotation dimension is mapped to at most one class dimension or not used for the mapping. Furthermore, any class dimension is determined by exactly one annotation dimension so that  $q \leq n$  holds. We use the mapping function  $f$  to calculate class-specific model parameters and to determine the class of future independent variables  $x_j$  as a base for class-based forecasting. Class-specific means are calculated by

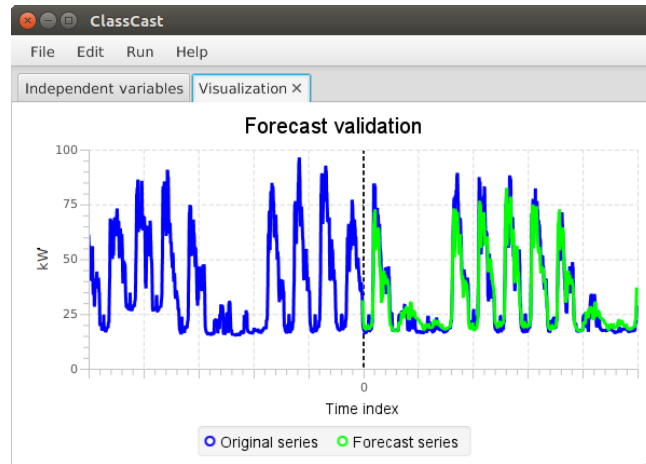
$$\bar{Y}(c) = \frac{1}{|\mathcal{J}_c|} \cdot \sum_{j \in \mathcal{J}_c} y_j, \text{ for } c \in \mathcal{C}, \mathcal{J}_c = \{j : f(x_j) = c\}$$

If at least one independent variable is selected as regressor, the past time series is partitioned into class-specific subsets  $(y_j^c) = (y_j)_{j: f(x_j)=c}$ . Linear regression can be used to compute regression parameters  $\beta_0, \beta_1$  for each class-specific partition.

Independent variables can be given as symbolic or numeric values. The scale of measure [7] for the independent variables is derived from the input data type. Symbolic values (e.g., “Monday”, “Tuesday”, ...) are interpreted as nominal scale, integers as ordinal scale, and real numbers as interval scale. By default, ClassCast uses independent variables on nominal and ordinal scale as classifiers and independent variables on interval scale as regressors. If an independent variable containing real numbers is manually selected as classifier, classes need to be defined based on intervals.

### 4 Implementation

ClassCast is implemented in Java and features a graphical user interface (GUI) based on the JavaFX [8] framework. Input data series are processed in comma-



**Fig. 2.** Validation of an electrical load forecast.

separated values (CSV) format. The tool automatically detects separators (comma, semicolon, tab), line break encoding, and input data types.

Values for independent variables can be given as strings, integer numbers, or floating point numbers. Strings are interpreted as nominal scale, integers as ordinal scale, and floating point numbers as interval scale. The first line of the input data file is interpreted as heading and is used for visualization only.

If the past data set is insufficient for forecasting, ClassCast presents a dialogue to the user for selecting an independent variable to discard. E.g., if the past data series is a time series recorded in winter, a forecast for summer cannot be computed using class-specific means without discarding the independent variables representing the weather.

ClassCast can be used both as an interactive tool with a GUI and as a non-interactive command line tool. In non-interactive mode, a CSV file is given as command line argument. Configuration like interval size for floating point classifiers can be given as option. The forecast data series is written to standard output in the same CSV format as used for the input data series.

## 5 Forecast Validation

ClassCast implements a very generic forecast approach. The forecast quality depends on the availability of sufficient data for the input time series, the existence of a strong dependence of dependent variables  $y_j$  on independent variables  $x_j^i$ , and the selection of appropriate classifiers. Therefore, ClassCast includes an interactive validation feature. For validation, the input data series is split in two parts. The first part is used to forecast the second part of the data series. The forecast can be visually compared to the actual measurement values and the mean square error can be computed to quantify the forecast quality.

Figure 2 shows a screenshot of the interactive validation, with an electrical load time series sampled from a school building in Germany. The forecast is computed as class-specific means with day of week and time of day as classifiers. The measured loads are plotted in blue, the forecast for the second part of the time series is plotted in green.

## 6 Conclusion

In this paper, we presented ClassCast, a simple tool for class-based forecasting. ClassCast classifies entries of time series according to the values of the independent variables and predicts future values for dependent variables using class-specific means or linear regression. The conception of ClassCast was motivated by energy load forecasting. However, ClassCast is agnostic to input data semantics. Therefore, it can be applied for forecasting time series in other domains under the following two conditions. First, time indices can be classified based on annotated information (independent variables). Second, this annotated information has a major impact on the values of interest (dependent variables).

**Acknowledgement.** The research leading to these results received funding from the German Ministry for Economic Affairs and Energy under the ZIM programme (Zentrales Innovationsprogramm Mittelstand), grant reference no. 16KN039521. The authors alone are responsible for the content of this paper.

The authors thank Prof. Joachim Grammig for fruitful discussions and helpful advice.

## References

1. Feinberg, E.A., Genethliou, D.: Applied Mathematics for Restructured Electric Power Systems: Optimization, Control, and Computational Intelligence pp. 269–285 (2005)
2. Campbell, P.R., Adamson, K.: Methodologies for Load Forecasting. In: International IEEE Conference on Intelligent Systems (2006)
3. Park, D.C., El-Sharkawi, M., Marks, R., Atlas, L., Damborg, M.: Electric Load Forecasting Using an Artificial Neural Network. *IEEE Transactions on Power Systems* 6(2), 442–449 (1991)
4. Chen, B.J., Chang, M.W., et al.: Load Forecasting Using Support Vector Machines: A Study on EUNITE Competition 2001. *IEEE Transactions on Power Systems* 19(4), 1821–1830 (2004)
5. Mu, Q., Wu, Y., Pan, X., Huang, L., Li, X.: Short-Term Load Forecasting Using Improved Similar Days Method. In: IEEE PES Power and Energy Engineering Conference (APPEEC) (2010)
6. Heimgaertner, F., Sachs, T., Menth, M.: ClassCast. <https://github.com/uni-tuekn/classcast> (2018)
7. Stevens, S.S.: On the theory of scales of measurement. *Science* 103(2684), 677–680 (1946)
8. Oracle Corporation: JavaFX API Documentation. <https://docs.oracle.com/javase/8/javafx/api/> (2015)