



Time Series Methods in the R package MLR

Steve Bronder
Columbia University

Abstract

The MLR package is a unified interface for machine learning tasks such as classification, regression, cluster analysis, and survival analysis. **mlr** handles the data pipeline of pre-processing, resampling, model selection, model tuning, and prediction. This paper details new methods for developing time series models in the **mlr**. Standard and novel tools such as auto-regressive and LambertW transform data generating processes, fixed and growing window cross validation, and forecasting models in the context of univariate and multivariate time series. Examples from forecasting competitions will be given in order to demonstrate the benefits of a unified framework for machine learning and time series.

Keywords: time series, model building, tuning parameters, R.

1. Introduction

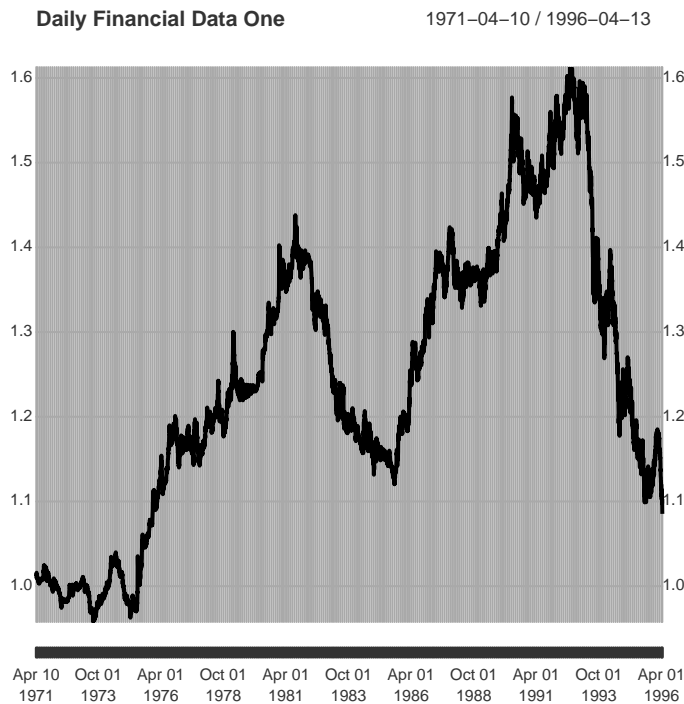
There has been a rapid development in time series methods over the last 25 years [Gooijer and Hyndman \(2006\)](#). Time series models have not only become more common, but more complex. The R language [R Core Team \(2015\)](#) has a large task view with many packages available for forecasting and time series methods. However, without a standard framework, many packages have their own sub-culture of style, syntax, and output. The **mlr** [Bischl, Lang, Richter, Bossek, Judt, Kuehn, Studerus, and Kothhoff \(2015\)](#) package, short for Machine Learning in R, works to give a strong syntactic framework for the modeling pipeline. By automating many of the standard tools in machine learning such as preprocessing and cross validation, **mlr** reduces error in the modeling process that is derived from the user.

While there are some time series methods available in the **caret** from [Jed Wing, Weston, Williams, Keefer, Engelhardt, Cooper, Mayer, Kenkel, the R Core Team, Benesty, Lescarbeau, Ziem, Scrucca, Tang, and Candan. \(2015\)](#), development of full on forecasting models in **caret** is difficult due to computational constraints and design choices. The highly modular structure of **mlr** makes it the best choice for implementing time series methods and models. This

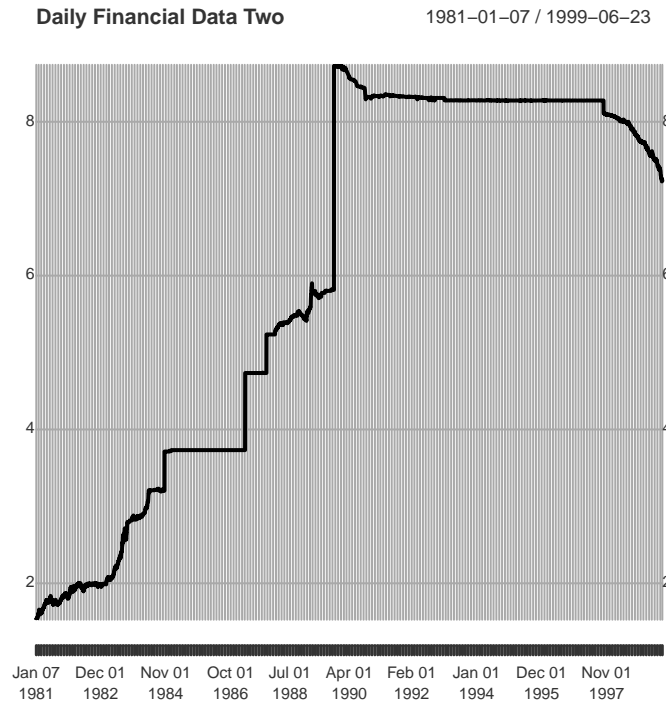
paper will show how using **mlr**'s strong syntactic structure allows for time series packages such as **forecast** Hyndman and Khandakar (2008) and **rugarch** Ghalanos (2015) to use machine learning methodologies such as automated parameter tuning, data preprocessing, model blending, cross validation, performance evaluation, and parallel processing techniques for decreasing model build time.

2. Forecasting Example with the M4 Competition

Professional forecasters attempt to predict the future of a series based on its past values. Forecasting can be used in a wide range of fields. Examples include forecasting stock prices, Granger (1992), weather patterns Allan H. Murphy (1984), international conflicts Chadeaux (2014), and earthquakes Yegulalp (1974). In order to evaluate **mlr**'s forecasting framework we need a large set of possible time series to make sure our methods generalize well.¹ The Makridakis competitions Makridakis and Hibon (2000) are forecasting challenges organized by the International Institute of Forecasters and led by Spyros Makridakis to evaluate and compare the accuracy of forecasting methods. The most recent of the competitions, the M4 competition, contains 10,000 time series on a yearly, quarterly, monthly, and daily frequency and areas such as finance, macroeconomics, microeconomics, and industry. For our purposes we will look at two particular daily financial series, one with 9136 observations from April 10th, 1971 to April 13th, 1996 and another with 6742 observations from January 7th, 1981 to June 23rd, 1999. Each series has a forecasting of 328 and 242 periods into the future, respectively.



¹Very goofy sentence need to fix



These two series were chosen for their large time features and stark contrast. Our data set should be large enough that the tuning method can take multiple windows of the data. Some series in M4 only contain 12 observations, which is not enough data to accurately train a model. These two time series were chosen as they are the two largest ones in the M4 competitions data set. We can see figure one is what most people imagine when they think of a time series. Figure two shows a series which appears to have a sort of step feature. The stark difference between the time process of the two series will allow us to investigate whether the methods in **mlr**'s forecasting framework can find the appropriate model. The data can be found in the package **M4comp** [BenTaieb \(2016\)](#) under sets 'M4[28]' and 'M4[29]'.

3. Forecasting Tasks

From a development perspective, it is a good idea to form a rigorous framework to define the data and relevant information about a particular predictive task. For forecasting tasks, this is handled in **mlr** by the function `makeForecastRegrTask()`. The forecasting task inherits from `makeRegrTask()`, but has two noticeable differences in parameters.

data: Instead of a data frame, an `xts` object from **xts** [Ryan and Ulrich \(2016\)](#) containing the time series.

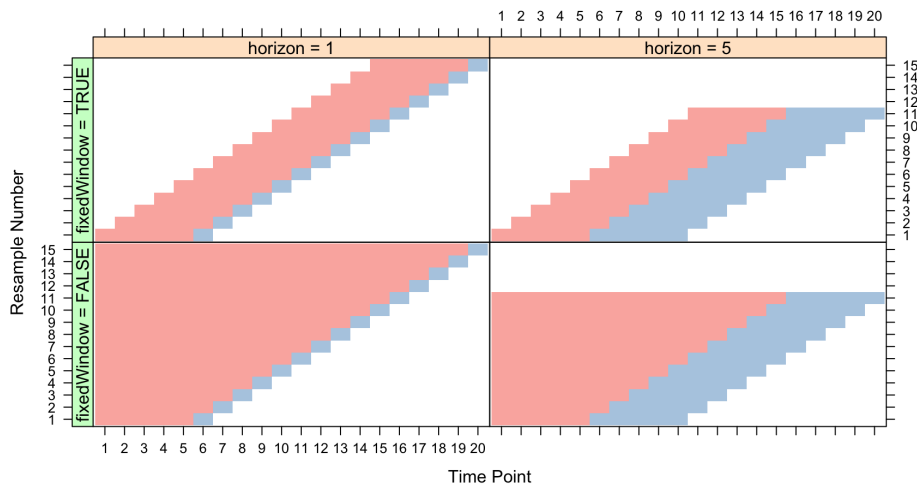
frequency: An integer with the number of periods your time series contains. For example, daily data with a weekly periodicity has a frequency of 7 while daily data with a yearly frequency have a frequency of 365.

```
Fin.task1 = makeForecastRegrTask(id = "M4 Finance Data One", data = m4train1, target = "ta
```

4. Resampling with Time

Overfitting models is one of the most common problems in prediction. Resampling schemes such as cross-validation, bootstrapping, etc. are common in machine learning. When there is a time component to the data, windowing schemes are necessary so that we correctly resample while still validating the time component of the model². Growing and fixed window resampling such as from [Hyndman and Athanasopoulos \(2014\)](#) are now available in the `resampling()` function of **mlr**.

Figure 1: Resampling with a window scheme as exemplified by caret



²crap

References

- Allan H Murphy RLW (1984). “Probability Forecasting in Meterology.” *Journal of the American Statistical Association*, **79**(387), 489–500. ISSN 01621459. URL <http://www.jstor.org/stable/2288395>.
- BenTaieb S (2016). *M4comp: Data from the M4 Time Series Forecasting Competition*. R package version 0.0.1, URL <https://CRAN.R-project.org/package=M4comp>.
- Bischl B, Lang M, Richter J, Bossek J, Judt L, Kuehn T, Studerus E, Kotthoff L (2015). *mlr: Machine Learning in R*. R package version 2.7, URL <https://CRAN.R-project.org/package=mlr>.
- Chadefaux T (2014). “Early warning signals for war in the news.” *Journal of Peace Research*, **51**(1), 5–18. doi:10.1177/0022343313507302. <http://jpr.sagepub.com/content/51/1/5.full.pdf+html>, URL <http://jpr.sagepub.com/content/51/1/5.abstract>.
- from Jed Wing MKC, Weston S, Williams A, Keefer C, Engelhardt A, Cooper T, Mayer Z, Kenkel B, the R Core Team, Benesty M, Lescarbeau R, Ziem A, Scrucca L, Tang Y, Candan C (2015). *caret: Classification and Regression Training*. R package version 6.0-62, URL <https://CRAN.R-project.org/package=caret>.
- Ghalanos A (2015). *rugarch: Univariate GARCH models*. R package version 1.3-6.
- Gooijer JGD, Hyndman RJ (2006). “25 years of time series forecasting.” *International Journal of Forecasting*, **22**(3), 443 – 473. ISSN 0169-2070. doi:<http://dx.doi.org/10.1016/j.ijforecast.2006.01.001>. Twenty five years of forecasting, URL <http://www.sciencedirect.com/science/article/pii/S0169207006000021>.
- Granger CW (1992). “Forecasting stock market prices: Lessons for forecasters.” *International Journal of Forecasting*, **8**(1), 3 – 13. ISSN 0169-2070. doi:[http://dx.doi.org/10.1016/0169-2070\(92\)90003-R](http://dx.doi.org/10.1016/0169-2070(92)90003-R). URL <http://www.sciencedirect.com/science/article/pii/S016920709290003R>.
- Hyndman R, Athanasopoulos G (2014). *Forecasting: principles and practice*. OTexts. ISBN 9780987507105. URL <https://books.google.com/books?id=gDuRBAAQBAJ>.
- Hyndman RJ, Khandakar Y (2008). “Automatic time series forecasting: the forecast package for R.” *Journal of Statistical Software*, **26**(3), 1–22. URL <http://www.jstatsoft.org/article/view/v027i03>.
- Makridakis S, Hibon M (2000). “The M3-Competition: results, conclusions and implications.” *International Journal of Forecasting*, **16**(4), 451 – 476. ISSN 0169-2070. doi:[http://dx.doi.org/10.1016/S0169-2070\(00\)00057-1](http://dx.doi.org/10.1016/S0169-2070(00)00057-1). The M3- Competition, URL <http://www.sciencedirect.com/science/article/pii/S0169207000000571>.
- R Core Team (2015). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria. URL <https://www.R-project.org/>.
- Ryan JA, Ulrich JM (2016). *xts: eXtensible Time Series*. R package version 0.10-0, URL <https://github.com/joshuaulrich/xts>.

Yegulalp TM (1974). “Forecasting for Largest Earthquakes.” *Management Science*, **21**(4), 418–421. ISSN 00251909, 15265501. URL <http://www.jstor.org/stable/2629612>.

Affiliation:

Steve Bronder
Quantitative Methods in the Social Sciences
Columbia University in the City of New York
International Affairs Building, MC3355
420 W 118th St, Suite 807
New York, NY 10027
E-mail: sab2287@columbia.edu
URL: [insert.url](#)