tsdataleaks: An R Package to Detect Potential Data Leaks in Forecasting Competitions

26 December 2023

Summary

Forecasting competitions are of increasing importance as a mean to learn best practices and gain knowledge. Data leakage is one of the most common issues that can often be found in competitions. Data leaks can happen when the training data contains information about the test data. There are a variety of different ways that data leaks can occur with time series data. For example: i) randomly chosen blocks of time series are concatenated to form a new time series, ii) scale-shifts, iii) repeating patterns in time series, iv) white noise is added in the original time series to form a new time series, etc. This work introduces a novel tool to detect these data leaks. The tsdataleaks package provides simple and computationally efficient algorithm to exploit data leaks in time series data. This paper demonstrates the package design and its power to detect data leakages using recent forecasting competitions data.

Statement of Need

Time series forecasting competitions have played a significant role in the advancement of forecasting practices. Typically, in forecasting competitions, a collection of time series is given to the competitors, and then the competitors submit the forecasts for the required test period of each time series. During the competition period only the training set of each time series is given to the public, and the test set is kept private from the public. Finally, competition organizers evaluate the forecast accuracy comparing the test set of each series and submitted forecasts by the competitors. Forecasting competitions helps to identifying novel methods and facilitating their performance comparison against existing state-of-the-art forecasting techniques (Hyndman 2020).

Data leakage occur when the training period of the time series includes test period data before officially release the test period of the time series. This idea is illustrated in Figure 1. A and B are two time series. The latter segment of the training set and the subsequent test set within the (B) series is derived from a training segment inherent to series (A). This type of data leak could occur when a randomly chosen blocks of time series are concatenated to form a new time series.

Competitions with data leaks will not be able to reach the original purpose. By exploiting data leakage competitors can obtain a top rank in the leader board. Such models look highly accurate within the competition environment but becomes inaccurate when applying the to a data set outside the competition environment. There is an increasing need to examine the potential data leaks in time series before the release of data to public. The tsdataleaks package is designed to identify data leaks in time series.

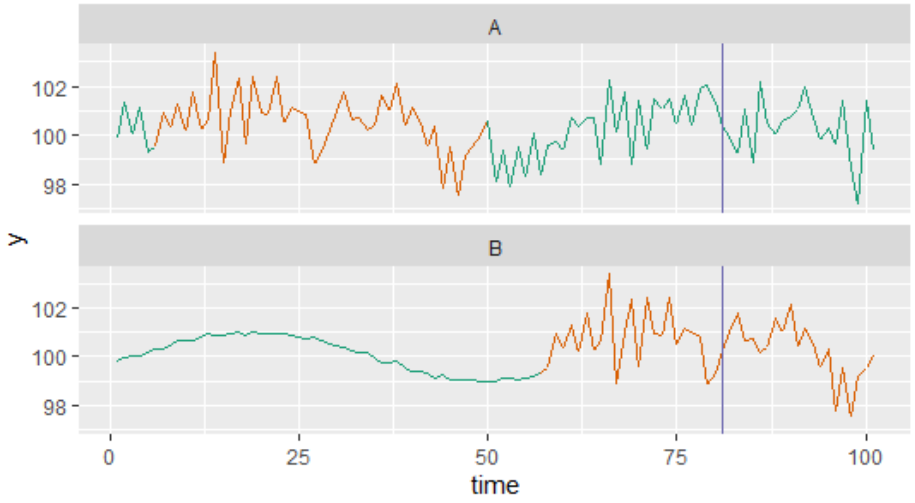

Figure 1: An example of a time series data leak. (A) and (B) are two time series. The purple verticle line separates the training and test parts of the series. The latter segment of the training set and test set of the (B) series comes from a training segment of series (A).

State of the Field in R

As of the latest information available on the Comprehensive R Archive Network (CRAN) Task View: Time Series Analysis (Rob J Hyndman 2023), there is no package available for detecting data leakages.

Algorithm

The algorithm operates as follows: it selects the final segment of the training portion from each time series in the collection, moves through all of the time series by one lag, and calculates the Pearson's correlation coefficient. Hence, the input to the algorithm are: i) the time series collection, ii) segment length, and iii) cut of value for the correlation coefficient serve as the algorithm's inputs. The algorithm returns the starting and end index of the segments that match each time series' training part of the last segment. Figure 2 illustrates the first iteration of the algorithm and a intermediate step of the algorithm is shown in Figure 3.

Figure 2: Visualization of the first iteration of the algorithm.The lst segment of the training part of the first series is coloured in purple. As the first step of the algorithm it is matched with the green section of the series 1.

Figure 3: Intermediate step of the algorithm: Identification of potential data leak.

Usage

Installation

The package tsdataleaks is available on GitHub and can be installed and loaded into the R session using:

```
devtools::install_github("thiyangt/tsdataleaks")
library(tsdataleaks)
```

Functionality

There are three functions in the package: i) `find_dataleaks`, ii) `viz_dataleaks` and iii) `reason_dataleaks`.

To demonstrate the package functions, I created a small data set with 4 time series.

```
set.seed(2020)
a <- rnorm(15)
d <- rnorm(10)
lst <- list(
  a = a,
  b = c(a[10:15]+rep(8,6), rnorm(10), a[1:5], a[1:5]),
  c = c(rnorm(10), a[1:5]),
  d = d,
  e = d)
```

The main function in the package is `find_dataleaks`. It exploits the data leakages according to the algorithm.

```
f1 <- find_dataleaks(lstx = lst, h=5, cutoff=1)
f1
```

The output of the above function is shown in Figure 4.

Next `viz_dataleaks` function visualize the results obtained in `find_dataleaks` for easy understanding as shown in Figure 5

```
$a
  .id start end
2   b     2   6

$b
  .id start end
1   a     1   5
2   b    17  21
4   c    11  15

$c
  .id start end
1   a     1   5
2   b    17  21
3   b    22  26

$d
  .id start end
5   e     6  10

$e
  .id start end
4   d     6  10
```

Figure 4: Output of find_dataleaks

```
viz_dataleaks(f1)
```

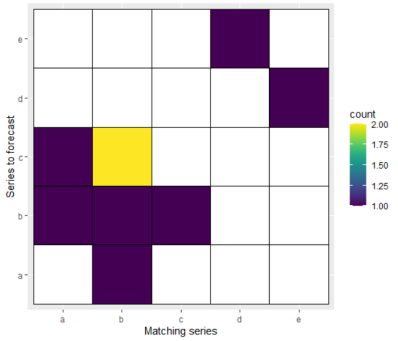

Figure 5: Output of viz_dataleaks

Finally, `reason_dataleaks` displays the reasons for data leaks and evaluate usefulness of data leaks towards the winning of the competition. The text output of the `reason_dataleaks` shown in Figure 6. The visualization is available at https://github.com/thiyangt/tsdataleaks. For example, according to the 2nd row in the output, series b last part correlates with series a index 2 to 6. Hence, series `a` segment indices 7-12 can be the series b remaining part. Hence, this identification is an useful identification. Furthermore, according to the fourth row of the same output series b last part correlates with series c segment with indices 11-15. However, we do not have observations from 16 on wards for the series c. Hence, it is not a useful identification in winning the forecasting competition.

Appication to the M1 competition data

```
library(Mcomp)
data("M1")
M1Y <- subset(M1, "yearly")
```

```
[[1]]
  series1 .id start end dist_mean dist_sd is.useful.leak       reason
1       a   b     2   6        -8       0         useful add constant
2       b   a     1   5         0       0         useful  exact match
3       b   b    17  21         0       0         useful  exact match
4       b   c    11  15         0       0     not useful  exact match
5       c   a     1   5         0       0         useful  exact match
6       c   b    17  21         0       0         useful  exact match
7       c   b    22  26         0       0     not useful  exact match
8       d   e     6  10         0       0     not useful  exact match
9       e   d     6  10         0       0     not useful  exact match
```

Figure 6: The text output of viz_dataleaks

```
M1Y_x <- lapply(M1Y, function(temp){temp$x})
m1y_f1 <- find_dataleaks(M1Y_x, h=6, cutoff = 1)
m1y_f1
```

Conclusion

The new open source R package described in this paper enable, i) explot data leakages, ii) identify the reasons for data leakage as exact match or add a constant, iii) determining whether the data leakages identified are useful in winning the forecast competition and iv) visualise the results.

References

Hyndman, Rob J. 2020. "A Brief History of Forecasting Competitions." *International Journal of Forecasting* 36 (1): 7–14.

Rob J Hyndman, Rebecca Killick. 2023. "CRAN Task View: Time Series Analysis." https://cran.r-project.org/web/views/TimeSeries.html.