

ADAMS

Advanced **D**ata mining **A**nd **M**achine learning **S**ystem

Module: adams-timeseries



Peter Reutemann

March 4, 2015

©2013-2014



THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato



Except where otherwise noted, this work is licensed under
<http://creativecommons.org/licenses/by-sa/3.0/>

Contents

1	Introduction	7
2	Flow	9
3	Tools	13
3.1	Timeseries Explorer	13
3.2	Weka Explorer	17
4	Setup	19
	Bibliography	21

List of Figures

1.1	Image of random data plus trend, with best-fit line and different smoothings[4].	7
3.1	Timeseries Explorer displaying wine sales, raw and smoothed with a Savitzky-Golay filter.	13
3.2	Wizard: start page.	14
3.3	Wizard: connection parameters.	14
3.4	Wizard: SQL queries for obtaining timeseries data and associated meta-data.	15
3.5	Wizard: selected IDs of timeseries to load.	15
3.6	Timeseries loaded from database.	16
3.7	WEKA Explorer displaying forecasts for Australian wine sales. .	17

Chapter 1

Introduction

According to Wikipedia¹, a timeseries is *is a sequence of data points, measured typically at successive points in time spaced at uniform time intervals*. ADAMS offers standalone tools, like the Timeseries Explorer, and flow components to loading, processing and saving of timeseries data. Performing predictions using the Weka timeseries forecasting module, is possible as well. The following sections explain the available functionality in depth.

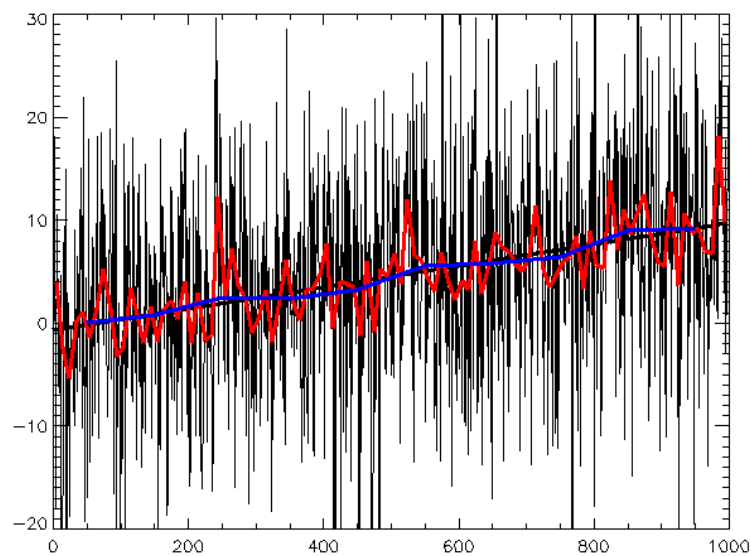


Figure 1.1: Image of random data plus trend, with best-fit line and different smoothings[4].

¹https://en.wikipedia.org/wiki/Time_series

Chapter 2

Flow

The ADAMS flow allows you to comprehensively load, process and save time-series data, as well as make future predictions.

The following data conversions are available:

- *SpreadSheetToTimeseries* – generates a timeseries from two spreadsheet columns (timestamp and value).
- *TimeseriesToArray* – extracts the values of a timeseries to be further analyzed with array statistics, for instance.
- *TimeseriesToSpreadSheet* – turns a timeseries back into spreadsheet.
- *TimeseriesToWekaInstances* – generates data suitable for use with WEKA.
- *WekaForecastContainerToArray* – extracts the WEKA forecasts as an array for further processing.
- *WekaForecastContainerToTimeseries* – turns the WEKA forecasts into a proper timeseries data structure.¹
- *WekaInstancesToTimeseries* – generates a timeseries from two attributes of a WEKA dataset (timestamp and value).²

The following filters are available:

- *Derivative* – calculates a derivative from the timeseries data.
- *EquiDistance* – ensures that the data points are evenly spaced, can also interpolate to create timeseries with the same number of points.
- *FastWavelet* – applies a wavelet transform to a timeseries.
- *LOWESS* – applies LOWESS smoothing [6].
- *Round* – allows the rounding of the values associated with timestamps in the timeseries.
- *SavitzkyGolay* – transforms the timeseries using the Savitzky-Golay filter (smoothing and differentiation) [5]
- *SetStart* – shifts all data points relatively to a new starting point for the timeseries.
- *Window* – extracts a specified window (or the inverse) from a timeseries.

The following outlier detectors are available:

¹adams-timeseries-build_and_use.forecaster2.flow

²adams-timeseries-build_and_use.forecaster2.flow

- *MinPoints* – requires the timeseries to have a certain number (absolute or percentage) of values above a specified value.

The following smoothers are available:

- *LOWESSBased* – uses a LOWESS filter for smoothing [6].
- *SavitzkyGolayBased* – uses the *SavitzkyGolay* filter to smooth the timeseries.[5]
- *SlidingWindow* – wrapper smoother that applies the base smoother to a sliding window.

The following sources are available:

- *TimeseriesDbReader* – reads timeseries data from any JDBC database (required columns: ID, timestamp, value).
- *WekaForecasterSetup* – contains the configuration for a WEKA forecaster, i.e., classifier and how to the meta-dataset is generated.³
- *WekaForecasting* – uses a trained and primed Weka forecaster model from storage to generate a fixed number of predictions.⁴

The following transformers are available:

- *MakeForecastPlotContainer* – turns predictions generated by a WEKA forecaster into plotable data.⁵
- *SpreadSheetRowToTimeseries* – similar to the *TimeseriesDbReader* source, this transformer generates a timeseries container each of the rows in a spreadsheet.
- *SpreadSheetToTimeseries* – similar to the *TimeseriesDbReader* source, this transformer generates one or more timeseries containers from specified columns in a spreadsheet.
- *TimeseriesDbReader* – reads timeseries data from any JDBC database (required columns: timestamp, value) using the incoming token as identifier for the timeseries.
- *TimeseriesFeatureGenerator* – extracts features from a timeseries and generates, e.g., spreadsheet data from it.
- *TimeseriesFileReader* – loads timeseries data from disk.⁶
- *TimeseriesFileWriter* – writes timeseries data to disk.⁷
- *TimeseriesFilter* – applies one of the aforementioned filters to the timeseries passing through.
- *TimeseriesInfo* – allows to generate some basic information for a timeseries container.
- *TimeseriesReportDbReader* – adds all key-value pairs obtained from an SQL query to the report of the timeseries passing through.
- *TimeseriesSplit* – splits the incoming timeseries into subsets using the specified algorithm.

³adams-timeseries-build_and_use_forecaster.flow, adams-timeseries-sliding_window.flow,
adams-timeseries-use_saved_forecaster.flow

⁴adams-timeseries-build_and_use_forecaster.flow, adams-timeseries-sliding_window.flow,
adams-timeseries-use_saved_forecaster.flow

⁵adams-timeseries-build_and_use_forecaster.flow, adams-timeseries-sliding_window.flow,
adams-timeseries-use_saved_forecaster.flow

⁶adams-timeseries-load_and_display.flow, adams-timeseries-load_from_csv_and_save.flow

⁷adams-timeseries-load_from_csv_and_save.flow

- *WekaPrimeForecaster* – primes a forecaster obtained from a callable actor.⁸
- *WekaTrainForecaster* – trains a callable forecaster on the incoming dataset.⁹

The following sinks are available:

- *TimeseriesDisplay* – displays one or more timeseries.¹⁰

⁸adams-timeseries-build_and_use.forecaster.flow, adams-timeseries-sliding_window.flow,
adams-timeseries-use_saved_forecaster.flow

⁹adams-timeseries-build_and_use.forecaster.flow, adams-timeseries-sliding_window.flow,
adams-timeseries-use_saved_forecaster.flow

¹⁰adams-timeseries-build_and_use.forecaster2.flow

Chapter 3

Tools

3.1 Timeseries Explorer

The *Timeseries Explorer* allows you to load timeseries data from files and directly from a database (using JDBC). Figure 3.1 shows a timeseries loaded from a WEKA ARFF file, containing data on Australian wine sales over time.

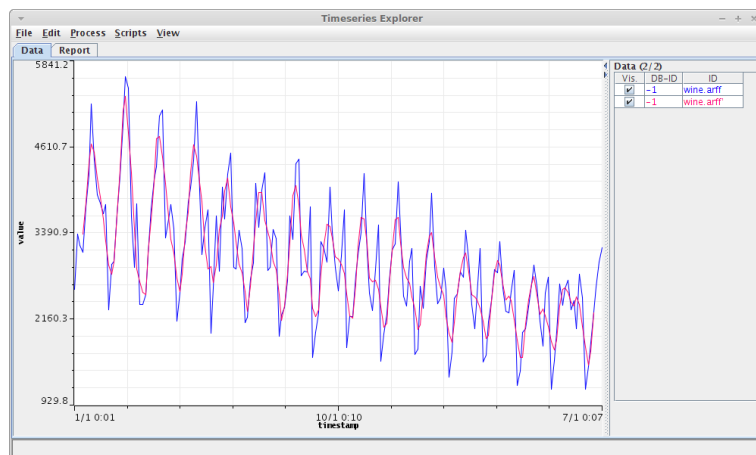


Figure 3.1: Timeseries Explorer displaying wine sales, raw and smoothed with a Savitzky-Golay filter.

Figures 3.2 to 3.6 show the wizard for loading timeseries data from a database using JDBC.

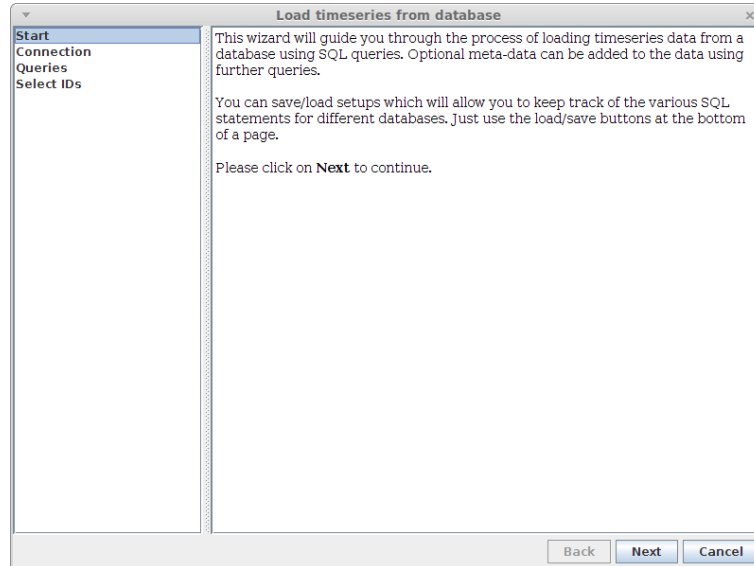


Figure 3.2: Wizard: start page.

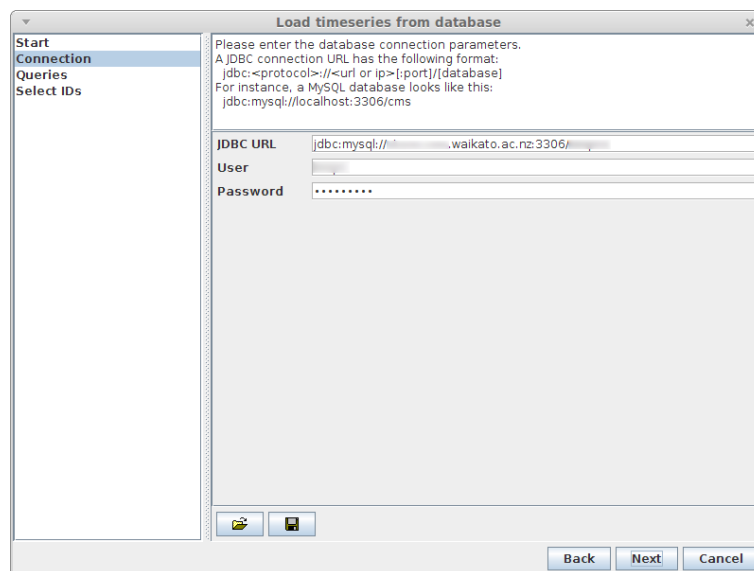


Figure 3.3: Wizard: connection parameters.

Load timeseries from database

Please enter the queries to retrieve data from the database. You can use the placeholder '{ID}' in your SQL statements for reading the timeseries data and meta-data. If no meta-data available, leave the meta-data statements empty. Otherwise, either use the statement that returns multiple rows of key-value pairs or the one that returns a single row with all the meta-data columns (uses the column name as key).

Listing the IDs

```
select distinct({ID})
from temperature
where year(time) = 2013
```

Timeseries data

```
select {ID}, time, temperature
from temperature
where {ID} = '{ID}'
```

Meta-data (key-value)

Meta-data (row)

```
select *
from {ID}
where {ID} = '{ID}'
```

Back Next Cancel

Figure 3.4: Wizard: SQL queries for obtaining timeseries data and associated meta-data.

Load timeseries from database

Please select the IDs of the timeseries to load.

43988660
43988615
43988530
43988455
43988448
43988431
43988134
43988127
43988103
43988097
43988035
43988004
43987946
43987861
43987793
43987786
43987595
43987588
43987564
43987557
43987526
43987519
43987472
43987281
43987267

Back Next Load

Figure 3.5: Wizard: selected IDs of timeseries to load.

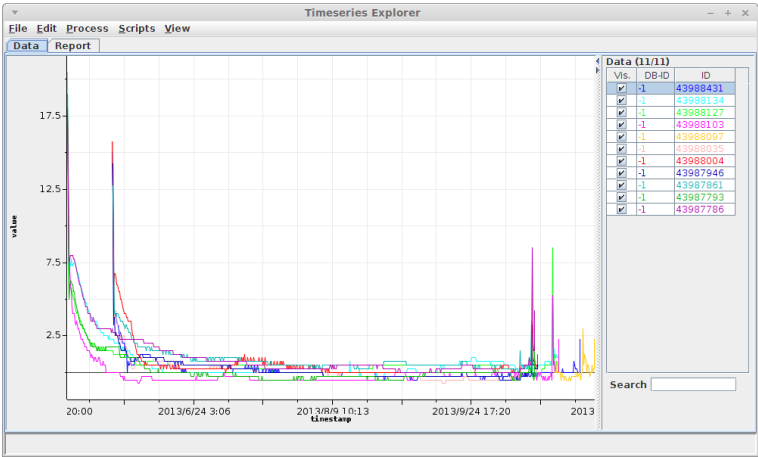


Figure 3.6: Timeseries loaded from database.

3.2 Weka Explorer

Using the *Timeseries Forecasting* package[3], you can perform basic timeseries analysis in the Weka Explorer as well. The *Forecast* tab in the Explorer allows you to configure a forecaster, specifying the attribute to forecast, what periodicity to use, etc. Figure 3.7 shows a screenshot of a forecast for Australian wine sales of fortified wine. An overall downward trend, despite seasonal ups and downs, is clearly visible.

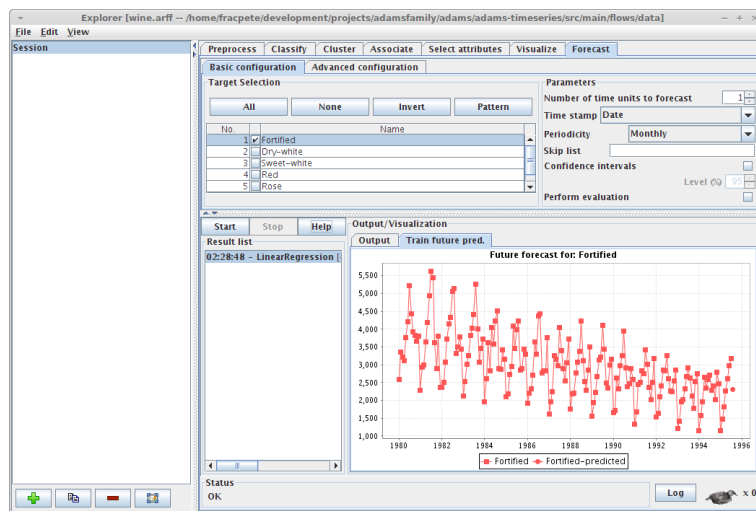


Figure 3.7: WEKA Explorer displaying forecasts for Australian wine sales.

Chapter 4

Setup

The following properties file contains the default format strings for periodicity types:

```
adams/data/timeseries/Periodicity.props
```


Bibliography

- [1] *ADAMS* – Advanced Data mining and Machine learning System
<https://adams.cms.waikato.ac.nz/>
- [2] *WikiPedia* – Time series
https://en.wikipedia.org/wiki/Time_series
- [3] Time Series Analysis and Forecasting with Weka
<http://wiki.pentaho.com/display/DATAMINING/Time+Series+Analysis+and+Forecasting+with+Weka>
- [4] *WikiPedia* – Image of random data plus trend, with best-fit line and different smoothings
<https://en.wikipedia.org/wiki/File:Random-data-plus-trend-r2.png>
- [5] *WikiPedia* – SavitzkyGolay filter
https://en.wikipedia.org/wiki/Savitzky%E2%80%93Golay_filter_for_smoothing_and_differentiation
- [6] *WikiPedia* – locally weighted scatterplot smoothing (LOWESS)
<https://en.wikipedia.org/wiki/Lowess>