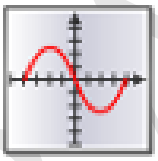
ADAMS

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

Module: adams-gnuplot



Peter Reutemann

December 24, 2014

DRAFT

©2009-2013

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 Generating output **9**
2.1 Data 9
2.2 Scripts 11

3 Running gnuplot **13**

4 Spreadsheets **15**

Bibliography **17**

List of Figures

1.1 Surface plot. 7
1.2 Gaussian distributions. 7

2.1 Flow for outputting spreadsheet data in Gnuplot format. 9
2.2 The generated Gnuplot data file. 10
2.3 Command-lines of example scriptlets for plotting data. 11
2.4 Flow for outputting spreadsheet data in Gnuplot format and generating a script for plotting it. 12
2.5 The generated plot. 12

3.1 Flow that executes Gnuplot with the generated data and script file. 13

Chapter 1

Introduction

Gnuplot[2] is an extremely versatile command-line plotting tool, that allows you to plot all kinds of graphs (e.g., plots with error bars, surface plots, plotting formulas). It has been under development since 1986 and is also used in popular scientific software like GNU Octave[1].

ADAMS offers you tools for turning data into Gnuplot data and script files, allowing you to take advantage of its advanced visualization techniques.

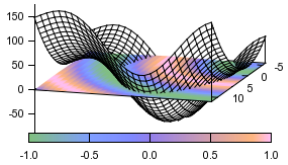


Figure 1.1: Surface plot.

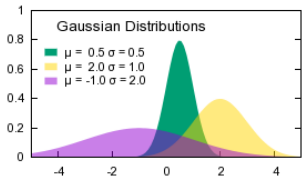


Figure 1.2: Gaussian distributions.

[1] `http://www.gnu.org/software/octave/`

Chapter 2

Generating output

ADAMS offers you Gnuplot support for outputting the data in Gnuplot format and also generating scripts for plotting the data. The following sections explain this in more detail.

2.1 Data

The most important step is to get the data from ADAMS into a format that Gnuplot can handle. Any data that is in spreadsheet format, can be output using the *SpreadSheetWriter* sink in conjunction with the *GnuplotSpreadSheetWriter* writer. Figure 2.1 shows a very simple flow[1] that dumps the spreadsheet data obtained from a CSV (comma-separated values) files as Gnuplot data file. This file can be plotted using the Gnuplot command-line interface. The raw data file is shown in Figure 2.2.

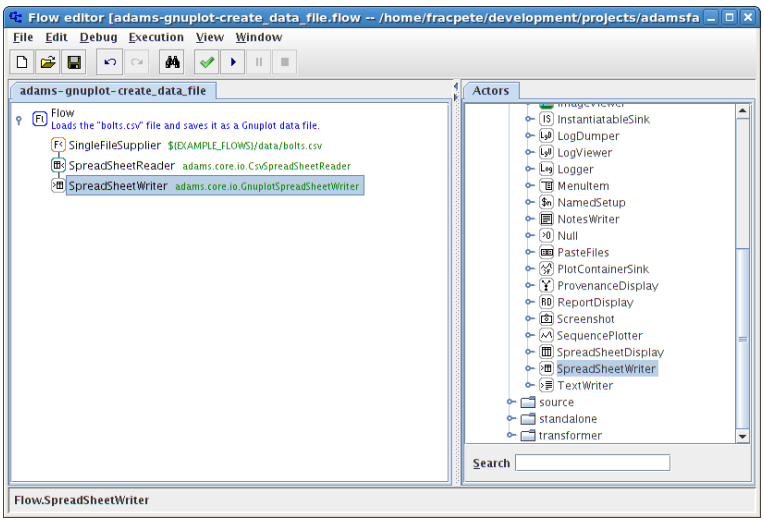


Figure 2.1: Flow for outputting spreadsheet data in Gnuplot format.

[1] adams-gnuplot-create_data_file.flow

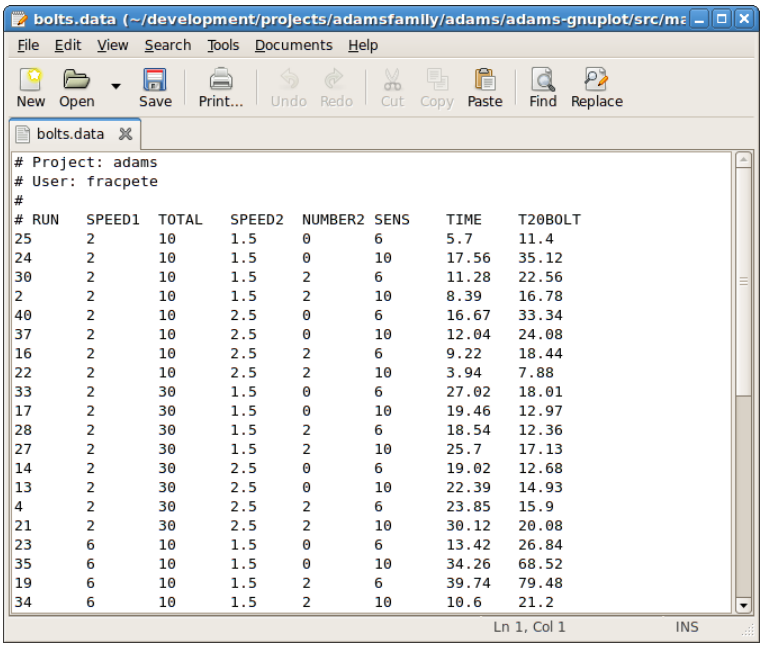


Figure 2.2: The generated Gnuplot data file.

2.2 Scripts

Outputting the spreadsheet data as Gnuplot data file is only the first step and only recommended if you already have an existing script for plotting the data. With ADAMS, you can also automate the process of generating a file containing the commands for plotting the data. The *GnuplotScript* sink can be used to generate a script file for plotting the data.

ADAMS offers already some basic scriptlets for generating parts of the script file:

- *Initialize* – Initializes the plot, e.g., sets plot and axes titles, output terminal.
- *CustomScriptlet* – Use this "free text" scriptlet if you need more advanced plotting features that the other scriptlets don't offer.
- *MultiScriptlet* – Groups multiple scriptlets and combines the output of all of them.
- *Pause* – Adds a "pause" statement to the flow, usually used as last scriplet.
- *SimplePlot* – Allows you to plot several columns as predefined plot types (points, lines, error plots, ...).

In Figure 2.3 you can see the command-lines of all the scriptlets that are used in Figures 2.4 and 2.5. Here a dataset for supervised learning is used to plot two attributes versus the class attribute.[2]

```
adams.core.gnuplot.Initialize -title "bolts dataset" -x-label "target variable" -y-label "input variables"
adams.core.gnuplot.SimplePlot -cols 8:2 -plot-type POINTS -plot-name "speed1 vs t20bolt" -first-plot
adams.core.gnuplot.SimplePlot -cols 8:7 -plot-type POINTS -plot-name "time vs t20bolt"
adams.core.gnuplot.Pause -waiting-period 5 -message "Press <Enter> to close the plot..."
```

Figure 2.3: Command-lines of example scriptlets for plotting data.

[2] adams-gnuplot-generate_gnuplot_script.flow

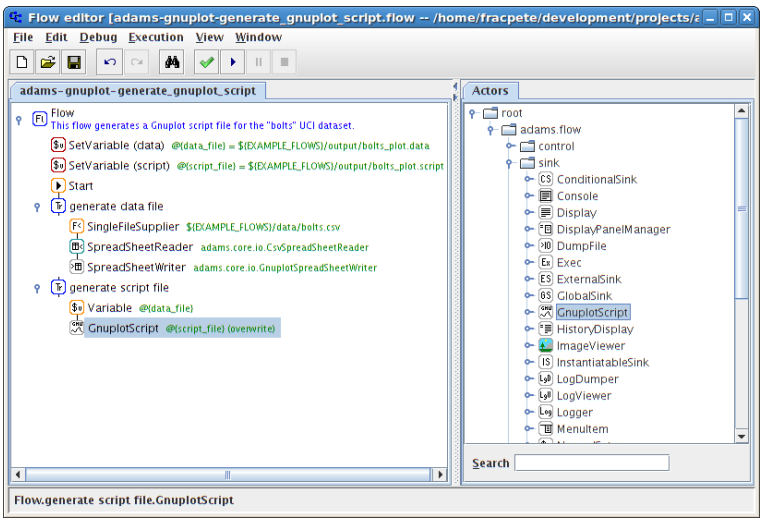


Figure 2.4: Flow for outputting spreadsheet data in Gnuplot format and generating a script for plotting it.

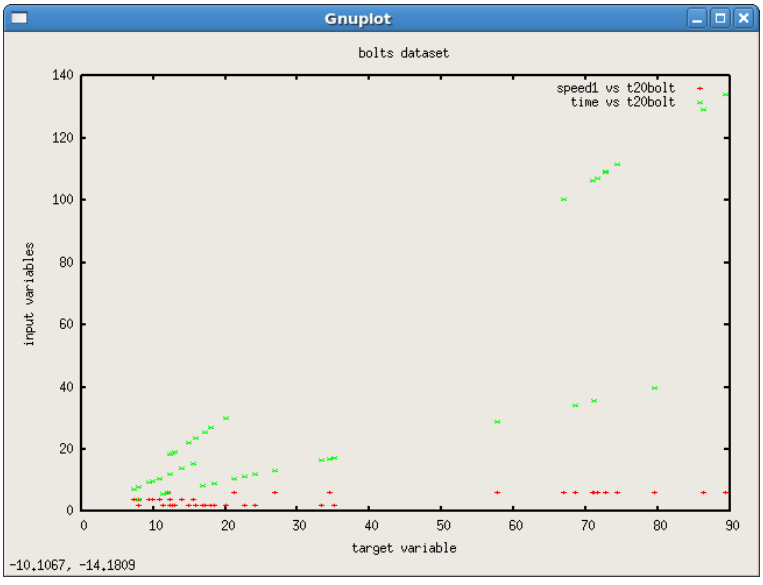


Figure 2.5: The generated plot.

Chapter 3

Running gnuplot

Simply generating the data and script files might not be sufficient in all circumstances, but executing the script itself can be necessary. Using the *Gnuplot* standalone, you can simply execute any script file. Figure 3.1 shows a flow that displays the data file with the generated script file.[1]

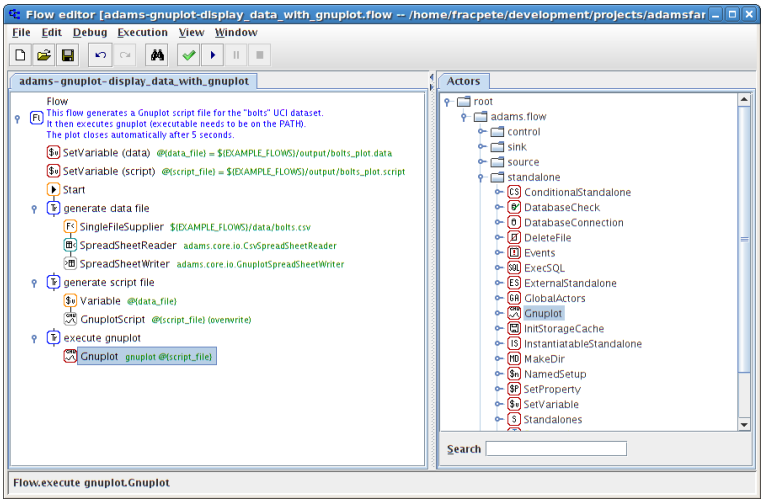


Figure 3.1: Flow that executes Gnuplot with the generated data and script file.

NB: If the *gnuplot* binary should not be available on the system's path, you need to specify the full path to it in the *binary* option.

[1] adams-gnuplot-display_data_with_gnuplot.flow

Chapter 4

Spreadsheets

ADAMS allows you to load Gnuplot data files as spreadsheet objects and also save spreadsheet objects as Gnuplot data files again.

- **read** – use the *adams.data.io.input.GnuplotSpreadSheetReader* class in conjunction with the *SpreadSheetFileReader* transformer.
- **write** – use the *adams.data.io.output.GnuplotSpreadSheetWriter* class in conjunction with the *SpreadSheetFileWriter* sink.

Bibliography

[1] *ADAMS* – Advanced Data mining and Machine learning System
`https://adams.cms.waikato.ac.nz/`

[2] *Gnuplot* – Portable command-line driven graphing utility
`http://gnuplot.sourceforge.net/`