RGP

GENETIC PROGRAMMING IN R

Acknowledgments

TODO Donec et nisl id sapien blandit mattis. Aenean dictum odio sit amet risus. Morbi purus. Nulla a est sit amet purus venenatis iaculis. Vivamus viverra purus vel magna. Donec in justo sed odio malesuada dapibus. Nunc ultrices aliquam nunc. Vivamus facilisis pellentesque velit. Nulla nunc velit, vulputate dapibus, vulputate id, mattis ac, justo. Nam mattis elit dapibus purus. Quisque enim risus, congue non, elementum ut, mattis quis, sem. Quisque elit.

Contents

Introduction

This chapter provides a detailed description of the case studies conducted in the course of this thesis. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Case Study Design and Organization

TODO general overview, concepts of problem sheet, problem instance, and method, intro to experimental desgin and planning via SPO

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Case Studies in Finance

Although the application of GP in finance is a young field, there already exist a comparatively large body of literature on this subject (TODO lit). The largest fraction of these work applies GP to evolve trading strategies for algorithmic trading. Our case studies are no exception in this regard, but are special in the fact that they apply the same GP framework in an array of vastly markets. TODO short intro to the three markets, etc.

Algorithmic Trading in Stock Markets (AppStock)

TODO

Symbol	*Company*	*Sector*
AAPL	Apple Inc.	Technology
CVX	Chevron Corp.	Basic Materials
GOOG	Google Inc.	Technology
PEP	Pepsico Inc.	Consumer Goods

Table 1: Summary Information on the Stocks included in the Case Study. This Summary Information was retreived from Yahoo Finance on March 1, 2010.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Algorithmic Trading in the FX Market (AppFX)

TODO

Objective Function The vTrader fitness function $fitness_{vt}$ assigns a numerical fitness to a signal generator function s. It is defined as follows:

$$fitness_{vt}(s) := [\underbrace{-1 \cdot pnl(s, T_{train})}_{\text{performance term}} + \underbrace{c_{mdd} \cdot mdd(s, T_{train})] \cdot balance(s)}_{\text{penalty terms}} \tag{1}$$

$$balance(s) := \begin{cases} c_{balance} & \text{if } skewness(s) > limit_{skewness} \\ 1 & \text{otherwise} \end{cases} \tag{2}$$

$$skewness(s) := \frac{abs(|s(T_{train})|_{long} - |s(T_{train})|_{short})}{|s(T_{train})|_{all}} \tag{3}$$

Parameter	*Description*	*Value*
c_{mdd}	Max-Drawdown Penalty Factor	TODO
$limit_{skewness}$	Long/Short-Skewness Limit	0.8
$c_{balance}$	Long/Short-Skewness Penalty Factor	TODO

Table 2: Default Parameter Values for the vTrader fitness function $fitness_{vt}$ used in the Case Study.

Algorithmic Trading of Emission Certificates (AppCO2)

TODO

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Case Studies in Water Resource Management

TODO

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Fill-level Prediction for Stormwater Tanks (AppStorm)

The main goal of this study is the prediction of fill levels in stormwater overflow tanks based on rainfall data in order to implement predictive control of water drain rate. Such predictions are of immense practical utility in preventing costly and damaging over- or under-loading of the sewage system connected to these stormwater overflow tanks. The task of predicting the current fill levels from the past rain data alone—not using past fill levels—is rather challenging since the hidden state of the surrounding soil influences the impact of rain in a nonlinear fashion (**??**).

Objective Function The quality of a fill level predictor is measured as the root mean squared error (RMSE) between true fill level and predicted fill level on the test dataset defined in Sec. **??**:

$$\mathrm{RMSE}(y_{pred}, y_{real}) := \sqrt{mean[(y_{pred} - y_{real})^2]} \tag{4}$$

A fill-level predictor is a function from a rainfall time series to a scalar fill level prediction. When predicting the fill level at time t, a fill level predictor has the rainfall time series up to time t available as input. A time series of predicted fill levels is obtained by iteratively applying a fill level predictor to a time series of rainfall data.

Test Data TODO update ranges

Training and test time series data for this study consist of 25, 344 data records. This data comprises measurements of the current fill level and the current rainfall at a real stormwater tank in Germany. These measurements were taken every 5 minutes, ranging from April, 21th 2007 (00:00 a.m.) to July, 17th 2007 (11:55 p.m.). We divided this dataset into a training dataset, ranging from April, 21th 2007 (00:00 a.m.) to April, 28th 2007 (00:00 a.m.) and a test dataset, ranging from April, 28th 2007 (00:05 a.m.) to July, 17th 2007 (11:55 p.m.). The results presented in the following are based on the test dataset, while all methods were trained on the training dataset. The training dataset consists of a short but balanced sample of dry and rainy days.

Making predictions on the training dataset had to be embedded into the fitness function of GP, therefore this dataset is comparatively small. We used the same training dataset in each case study to keep results comparable. While fill levels responds differently to rain depending on season, the response stays very stable within a season. This is why the size of our training dataset should not pose a problem to the methods under study, and also why our test dataset spans the late-spring/early summer season, but not more. In practice, the methods under study would be retrained for each season.

Prediction of NH4N in WTP Inflow (AppNH4N)

TODO

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetuer at, consectetuer sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst. Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

Prediction of Chemical Oxygen Demand in WTP Inflow (AppO2)

TODO

Morbi luctus, wisi viverra faucibus pretium, nibh est placerat odio, nec commodo wisi enim eget quam. Quisque libero justo, consectetuer a, feugiat vitae, porttitor eu, libero. Suspendisse sed mauris vitae elit sollicitudin malesuada. Maecenas ultricies eros sit amet ante. Ut venenatis velit. Maecenas sed mi eget dui varius euismod. Phasellus aliquet volutpat odio. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Pellentesque sit amet pede ac sem eleifend consectetuer. Nullam elementum, urna vel imperdiet sodales, elit ipsum pharetra ligula, ac pretium ante justo a nulla. Curabitur tristique arcu eu metus. Vestibulum lectus. Proin mauris. Proin eu nunc eu urna hendrerit faucibus. Aliquam auctor, pede consequat laoreet varius, eros tellus scelerisque quam, pellentesque hendrerit ipsum dolor sed augue. Nulla nec lacus.

Getting Started

This chapter provides a short real-world example of performing symbolic regression with RGP.

Installation

The newest release version of RGP is always available on CRAN. To install it just type **install**.**packages**("rgp")

Problem Definition

For a first run of RGP you need to define a fitting problem.

Listing 1: Creating a function with variables

```
makeDampedPendulum <- function(A0 = 1, g = 9.81, l = 0.1, phi = pi, gamma = 0.5) {
  omega <- sqrt(g/l)
  function(t) A0 * exp(-gamma * t) * cos(omega * t + phi)
}
```

The function above represents a damped mathematical Pendulum, the arguments are the starting Amplitude A0, gravity g, length of pendulum l, phase, radial frequency omega and damping gamma.

Listing 2: Attributes

```
dampedPendulum1 <- makeDampedPendulum(l = 0.5)
```

Here is a example of alternating the given attributes, the lenght is altered and the function is assigned to a new name.

With `xs1 <-seq(from=1, to=10, length.out=512)` a list of numbers is definded for creating a dataframe. It creates a subsequent list of numbers from 1 to 10 with 512 steps.

Listing 3: Data Frame Creation

```
dampedPendulumData <- data.frame(time=xs1,
  amplitude=dampedPendulum1(xs1) + rnorm(length(xs1), sd=0.01))
```

We create a time series dataframe with the sequence defined above. The dataframe consists of the sequence xs1 as time, the amplitude defined through our pendulum-function and a interference function. To plot the prepared data frame type `plot(dampedPendulumData, pch=20)`.

Now let's shift attention to the more interesting objects. First, we need to activate RGP using `require(rgp)`. We are going to perform a symbolic regression with our pendulum data. The time permitted for the regression are 2 minutes (2 * 60 seconds). To start the symbolic regression you need the following code.

Listing 4: Symbolic Regression

```
modelSet1 <- symbolicRegression(amplitude ~ time, data=dampedPendulumData,
                                 stopCondition=makeTimeStopCondition(2 * 60))
```

Now we got a set of models with variable fitness. To get our desired prediction we are going to use the model with the best training fitness which gives the best results.

Listing 5: Best Model

```
bestModel1 <- modelSet1$population[[which.min(Map(modelSet1$fitnessFunction,
  modelSet1$population))]]

bestModel1
```

Let's start the prediction and plot our given and predicted data.

Listing 6: Prediction

```
predictedData <- data.frame(time=xs1,
  amplitude=predict(modelSet1, newdata=dampedPendulumData))

#plot data
plot(dampedPendulumData, pch=20)
points(predictedData, pch=20, col=2)
```

The plot shows the difference between the real and the predicted data. As you see, the outcome is not ideal. To get better results we need to extend the processing time.