

MetaOS R integration. A how-to guide.

Sergio Alvarez (setelena@gmail.com), Luis F. Canals (luisf.canals@gmail.com)

July 2, 2011

1 Abstract

2 MetaOS viewed from Python scripts.

The following example shows how Python scripts invokes a R code in the simplest way: load of R environment and R code, initialization of R variables and objects, example of loop to call several times to a function R in the R-environment and new call at the end of the loop:

```
from java.lang import Long
from java.lang import Double

interpreteR = R(["correlation.r"])

interpreteR.eval("corre<-correlator()")
for i in range(1,200):
    interpreteR.eval("corre$memo(" + str(i) + "," + str(i) + ")")

print interpreteR.eval("corre$show()")

interpreteR.end()
```

As it's seen from the code, a R source code named *correlation.r* where a class *correlator* is defined. The class should have got the methods *memo* and *show*, as in this example:

```
correlator <- function() {
  xVals <- c()
  yVals <- c()

  memo <- function(x, y) {
    xVals <- union(xVals, x)
    yVals <- union(yVals, y)
  }

  show <- function() {
    r <- lm(xVals ~ yVals)
    print(r)
  }

  return(list(memo=memo, show=show))
}
```

3 Generalization

The following code (*rintegration.py*) uses the same described principle in the previous section but letting the name of R source containing the class

as a runtime parameter:

```
from java.lang import Long
from java.lang import Double

interpreteR = R(["correlation.r"])

interpreteR.eval("corre<-correlator()")
for i in range(1,200):
    interpreteR.eval("corre$memo(" + str(i) + "," + str(i) + ")")

print interpreteR.eval("corre$show()")

interpreteR.end()
```

In this case, interface for R class has been modified, to create a simple predictor with two methods, *learn* and *predict*. An example of predictor based on linear regression might look like this:

From the example, we write down the interface all R-class should satisfy to be compatible with *rintegration.py* requirements (in pseudocode Java-like):

```
interface PredictorInR {
    // Learns a new relation x->y
    void learn(double x, double y);

    // Tries to predict the value of y from x
    double predict(double x);
}
```

4 Usage from command line and visual interface

The code in *rintegration.py* can be invoked even from command line or from visual interface. In both cases, file acting as a source of prices, the main Python script (*rintegration.py* in this case) and the R file containing the source code of the class with the predictor following the interface described in the interface 3.

5 License terms

This is a GNU Licensed Document. Modifications and copies of this document must follow the GNU License, referring to authors and to the original document. All other rights are reserved by the authors.