

OPENCOURSEWARE
ADVANCED PROGRAMMING
STATISTICS FOR DATA SCIENCE
Ricardo Aler



R and C++ integration

The purpose of this topic is to show how to improve the efficiency of some R operations by programming them in a compiled language : C++.

R is an interpreted language and this makes its execution slow. This is not usually a problem, because R is used as a glue language, in order to put together functions which belong to external libraries. Those libraries have been programmed in Fortran, C, or C++, all of them fast (compiled) languages. Given that R spends most of its time executing those libraries, R inefficiencies do not show in most cases. However, there are some situations, such as executing long loops, for which R is known to be slow, and efficiency might benefit from moving those parts to functions programmed in a compiled language. Package RCpp offers a simple and seamless way of integrating both languages.

First it will be explained what is meant by interpreted versus compiled, dynamic versus static typing, and at what situations R is slow (mainly loops and function calling). And therefore, it is worth considering moving those parts to a compiled language like C++.

Instead of explaining the complete C++ language, RCpp will be introduced by means of examples. In particular, the following types of functions will be explained, with both the R syntax and the C++ syntax :

1. No inputs, scalar output
2. Scalar input, scalar output
3. Vector input, scalar output
4. Vector input, vector output

Finally, the concept of RCpp syntactic sugar will be explained : syntactic sugar is a way of making writing C++ programs more simple, using a simplified yet powerful syntax.

To finish with, the two ways of integrating RCpp and R are exemplified in the slides : writing single functions, or writing and compiling a source Cpp file.

Associated Material :

- Slides and some exercises (check the lecture notes).
- One of the labs is about using Rcpp ("Programming KNN with $K=1$ in Rcpp")