C++ Software for Temperature Forecasting

Oliver J. Boom and Mattia Guerri
Imperial College London

1 Introduction

The goal of this assignment is to predict the future temperature for a specific city. In particular, temperature forecasts for January 2020, 2030 and 2050 were requested. The city of Alice Springs, located in Australia's Northern Territory, was selected as the subject of the temperature forecasting model. A piece of C++ software was created to process the data and create a prediction for future temperature, using linear regression.
Presented here is a brief discussion of the data employed in the forecasting, the structure of our software, its usage, the method used to forecast the temperature, and the software output.

2 Input Data

A meteorological station operated by the Australian Bureau of Meteorology is located in the city. Temperature data dating back till January 1942 is available and freely accessible through the website (http://www.bom.gov.au/climate/data/). The average maximum and minimum monthly temperatures was downloaded and used as input data in the modelling.

3 Numerical Methods

The input data consists in a time series of the temperature values. The objective was to obtaining a linear model of the temperature evolution, therefore, we smoothed the time series adopting two different algorithms. Firstly a simple moving average (SMA) algorithm was applied. This is was use considering to remove the seasonality that occurs within the time series data for temperature. The SMA consists in running an averaging filter on the time series. The width of the of the filter (number of values considered in the average) can be defined by the user. It is suggested that a value of at least 12 is used to smooth the effect of seasonality. It was observed that even after the application of the SMA the result presents significant complexities. It was therefore decided to give the user the possibility to apply a further smoothing algorithm. This second method consists in an exponential smoothing, which is defined by the equation:

$$S_t = w\, y_t + (1 - w)\, S_{t-1},$$

where S_t is the smoothed value at time t, w is a weighing parameter (the higher the value the weaker the smoothing effect), y_t is the original value at time t, S_{t-1} is the smoothed value at time t-1. In this exponential smoothing the smoothed value at a certain time t is affected by all the preceding values according to a weight that decreases geometrically in time.
Then linear regression was calculated on this smoothed time series. The intercept and slope of was calculated minimizing the difference the error values using a least square approach. Extrapolating this linear model we are then able to make prediction on future temperatures.

4 Software Architecture

The software was structured with 5 Cp++ source files; main, utils, params, test and linear_regression, with each files (with the exception of main) having a corresponding header file. A brief synopsis of the structure: main: responsible for integrating the various files and orchestrating the running of the program. utils: for general functions and functions relating to loading the data. linear_regression: conducts the statistical analysis required to allow for extrapolative prediction. test: for running a test to make sure the linear regression suite is working correctly. params: the module to allow the users interaction with the code. Greater detail on the individual functionality of these respective files is given in the documentation within the files. These files are linked together via the header files, which are included in the main file. Within the header files classes and functions are declared, while in the cpp files the member functions, and generic functions are defined. Params.txt can be edited by the user to update the parameters of the model.

5 Software Usage

The software was delivered as a Microsoft Visual Studio (2017) project. In addition to those defined by ourselves, we only use classes belonging to the c++11 Standard Library. The software is therefore intended as a standalone project, with the only exception of the application gnuplot (freely available at http://www.gnuplot.info), which is necessary to make the plots. To use the software the user must run project in visual studio. They must respond to the prompts in the terminal to input the date of the temperature forecast. The temperature will be returned alongside a plot showing the extrapolation up to that date.