Applying bias-adjustment to SEAS5 seasonal forecasts for I-CISK living lab Spain, station-based data

Contact: thomas.bosshard@smhi.se

Introduction

The processing of seasonal forecasts and tailored bias-adjustment towards station data has been developed at SMHI. However, SMHI cannot sustain the operational production of this system beyond the project's end. Here, we explain how the processing and bias-adjustment is done. The idea is that anyone with access to the necessary input data could build an operational production for that specific living lab.

Limitations

The methodological instructions can be applied to all similar kinds of data. However, the provided parameter files are strictly speaking only valid for SEAS5 forecasts and the station-based reference data that were used for calibration. Also, the resolution of the original SEAS5 forecast might impact the results. Differences to the produced bias-adjusted seasonal forecast data within the ICISK-project have to be expected if the input data is different (e.g. has been downloaded at a different resolution).

Data requirement

Both the parameter files for bias-adjustment and the grid definition files for the spatial interpolation with cdo can be download from here until January 18 2025:

http://exporter.nsc.liu.se/6a66e206394a48fb90fe6103315d2106,
rsync://exporter.nsc.liu.se/6a66e206394a48fb90fe6103315d2106

There is a readme in the downloadable data that describes the data files in more detail.

Processing steps

1) Interpolation to station location
   The downloadable grid description files are compatible with CDO remapping function [https://code.mpimet.mpg.de/projects/cdo].
   The command syntax is:
   ```
   cdo remapbil,#GRIDDESCRIPTIONFILE# #INPUTFILE# #OUTPUTFILE#
   ```

   Placeholders:
   #GRIDDESCRIPTIONFILE#: Path to the grid description file as downloaded according to section X. Please make sure that you use the grid description file that matches the variable. For precipitation, there are more stations available than for temperature.
   #INPUTFILE#: Path to SEAS5 data file that should be interpolated
   #OUTPUTFILE#: Path to where the interpolated SEAS5 data should be saved to.

Bilinear interpolation is used and if one prefers another tool to do the interpolation, one can extract the station latitudes and longitudes from yvals and xvals in the grid description file, respectively.

2) For temperature variables only: Derive diurnal temperature range
The bias-adjustment method requires diurnal temperature range (TNG) as and input. This is derived by subtracting the daily minimum temperature (TASMIN) from daily maximum temperature (TASMAX). In CDO, this would be

```
cdo sub #TASMAXFILE# #TASMINFILE# #TNGFILE#
```

3) Bias-adjustment of the variables precipitation (PR), daily minimum temperature (TASMIN) and diurnal temperature range (TNG)

The bias-adjustment uses a Q-Q relation to correct the seasonal forecast data from steps 1) and 2). Here, we are sketching how this bias-adjustment could be done in python. We assume that data is already read in and structured in a way that one can loop over the months and stations in the forecast. Parts where you would have to add your own code are indicated by …

```
from scipy.interpolate import splev
…
for m in months:
      #Read in the npz-files for with the corresponding parameters
      param=…
      #param should have shape (number_of_stations,1,1450)
      for st_i in stations:
                  #Select the corresponding data
                  d=…
                  #Extract relevant parameters by first defining the indices
                  #where they are located in param
                  ind_degree = 1 # Index of degree of spline fit
                  ind_no_knots = 2 #Index where the number of spline knots is
                  #saved
                  ind_knots = 3 #Index where the knots start
                  n = param[st_i,0,ind_no_knots]  #number of knots
                  ind_coeffs = ind_knots + n #indices range for coefficients
                  # of spline
                  ind_dmin = ind_coeffs + n #Indices where minimum value for
                  #spline fit is stored
                  ind_dmax = ind_dmin + 1 #Indices where maxmimum value for
                  #spline fit is stored
                  ind_ssr_threshold = ind_dmax + 1 #Index where the wet-day
                  #threshold is stored

                  k = int(param[st_i,0,ind_degree])
                  t = param[st_i,0,ind_knots:ind_coeffs]
                  c = param[st_i,0,ind_coeffs:ind_dmin]
                  dmin = param[st_i,0,ind_dmin]
                  dmax = param[st_i,0,ind_dmax]
                  tck = (t, c, k) #tuple contains the input parameters to the
                  #spline method
                  tmin = t[k]
                  tmax = t[-k]
                  ssr_threshold = param[st_i,0,ind_ssr_threshold]

                  n_ts=len(d) # Number of time steps
```

```
            for ts in range(0,n_ts):
                #Below or about the spline limits, a constant bias-
                adjustment value is assumed (dmin or dmax)
                under_range = d[ts] < tmin
                over_range = d[ts] > tmax
                if under_range:
                    d[ts] = d[ts] + dmin
                elif over_range:
                    d[ts] = d[ts] + dmax
                else:
                    d[ts] = splev(d[ts], tck, ext=2)

#After the loops, d contains bias-adjusted data and can be written to
#a netcdf-file.
```

4) The daily maximum temperature is reconstructed from bias-adjusted daily minimum temperature and diurnal temperature range. This is done by adding the diurnal temperature range to the daily minimum temperature. In CDO, this could be done by

   ```
   cdo add #TASMINADJUSTEDFILE# #TNGADJUSTEDFILE#
   #TASMAXADJUSTEDFILE#
   ```

5) The daily mean temperature is derived as the mean of daily minimum and maximum temperature. In CDO, this would be

   ```
   cdo mean #TASMINADJUSTEDFILE# #TASMAXADJUSTEDFILE#
   #TASADJUSTEDFILE#
   ```