GEOG 696c Spatiotemporal Data Analysis

Homework #4

Due: Thursday, October 19th, 2023

Write your own individual code to perform the following operations and calculations in a Jupyter notebook. If any answer requires you to respond to a query (e.g. 'What do you observe about these results?'), please place your answer behind a comment tag or in a Markdown box. When you have completed all the elements of the assignment, submit the Jupyter notebook to D2L.

1. In preparation for this assignment, please review *Deser and Blackmon* (1995).

2. We are going to mimic a different portion of the analysis in *Deser and Blackmon* (1995), again using a more recent vintage dataset (HadISST, *Rayner et al.* (2003)). We will once again use November through March sea surface temperature (SST) averages.

3. Drawing on your previous code *as well as any lessons learned in the interim or improvements in your programming approach*, calculate and map the eigen-modes (eigenvectors, eigenvalues, and signals) of variability of winter Pacific SSTs for the same domain as *Deser and Blackmon* (1995). This time, however, please calculate *all* the eigenmodes using the full `np.linalg.svd` and the Jupiter notebook on singular value decomposition on the centered data matrix. We'll use the complete SVD factorization in testing for significance.

New skill Before we move onto the correlation and compositing done by *Deser and Blackmon* (1995), we are going to calculate the 'significance' of the leading modes of Pacific winter SST variability using a *red noise* Monte Carlo procedure. In the previous homework, we generated 'white noise' (Gaussian) random matrices in order to evaluate which mode of the data might be 'significant', meaningful, or interpretable. Here, I will ask you to perform 2 different significance tests: the first as before, using Gaussian noise; the second, generating 'red' autocorrelated noise to more accurately mimic the structure of the data.

4. Perform a Rule N test of significance using Gaussian random noise conditioned (that is, with the same variance as the actual data imposed on the random data) on the data. Using this rule, how many modes of variability might be meaningful, significant, and/or interpretable? Create a scree plot (using `plt.stem`) to demonstrate the significance threshold vs. the actual data.

5. Now, perform a Rule N test of significance using '*red*' autocorrelated noise. You can calculate an estimate of the first-order autocorrelation of a *centered* (mean removed) vector (the simple correlation between consecutive points in a time series) using a procedure such as that shown in the notebook `eof_significance.ipynb`. How then to create random time series with the same autocorrelation as your *centered* data matrix? You can do this by starting with a random value and then calculating each subsequent value using the autocorrelation you determine for your real data plus an additional draw from a Gaussian distribution. If you have a data matrix and you want to find the lagged (auto)correlation for each column, you can use Panda's `.autocorr` in a lambda function to get these (because Pandas only uses `.autocorr` on individual series, not DataFrames, you need to `.apply` the method as a lambda or inline function):

```
acf1 = cereals.apply(lambda x: x.autocorr())
```

Note that because of the addition of Gaussian random noise at each time step when we create our red noise null series, especially for short series (<100 observations, or rows) you will find that the autocorrelation structure of your random data and your data sample are not exactly same, even though that was part of the calculation (*Ghil et al.*, 2002). For longer series and for more iterations of the noise model, however, this should provide a reasonable simulation of a population of random values with similar autocorrelation structure to your data. Also note that, in practice, a better estimate of the actual noise variance is desirable. More sophisticated algorithms can also be employed to develop more exact simulations of your data for significance testing (e.g. *Schneider and Neumaier*, 2001; *Neumaier and Schneider*, 2001; *Percival and Constantine*, 2006). In the `eof_significance.ipynb`, we modify the variance of the random draw as `sigma_e = np.sqrt((sigma ** 2) * (1 - acf1 ** 2))`:

```
c = mu * (1 - acf1)
sigma_e = np.sqrt((sigma ** 2) * (1 - acf1 ** 2))

signal[0,:] = np.array([c + np.random.normal(0, sigma_e, size=ncols)])

for i in range(1, nrows):
    signal[i,:] = np.array([c + (acf1 * signal[i-1,:]) \
    + np.random.normal(0, sigma_e,size=ncols)])
```

Based on your Gaussian and Red noise tests, how many modes of North Pacific winter SSTs are 'significant', meaningful, interpretable, and/or differentiated from noise? Support your assessment by using a scree plot to demonstrate the significance threshold vs. the actual data.

New skill We will now perform a 'field correlation' analysis (the correlation between a single time series and the time series at every single point in field) as described in lecture.

6. Please calculate and map the spatial field correlations between each of the the 2 leading time series (signals) from your EOF analysis and the original SST data field (similar to *Deser and Blackmon* (1995), see their Figure 1). There are several ways you might do this (you could even do it using just basic mathematics, since I am not yet asking for a significance test), but you may wish to look into how to best use `xarray.corr`. Your result for this step will therefore be two maps. How do your 2 maps (one each for the correlation between the time series and the original field) compare to those in *Deser and Blackmon* (1995)? And, how do your correlation maps compare to the loading (eigenvector) maps you created previously in Homework #2?

New skill We will now stratify the EOF signals and use these to form composite mean values for a separate field.

7. Looking only at EOF1 now (the leading mode of the data), identify the years corresponding to the highest/largest 10% of the signal values and the years corresponding to the lowest/smallest 10% of the signal values (hint: `.quantile`, `.where`, and `.sel` will be your friends for this task).

8. Having identified the years with the 10% largest and smallest values in EOF1, import the 20th Century Reanalysis 500mb height data (*Compo et al.*, 2011), available on D2L as `hgt.500mb.mon.mean.nc`, just as we've done previously for SST (this file is quite large). The netcdf files contains the height above the surface for the 500mb pressure level, which is an

indication of atmospheric circulation. The data are monthly resolution. More information about the 20th century Reanalysis is available here:

`https://climatedataguide.ucar.edu/climate-data/noaa-20th-century-reanalysis-version-2-and-2c`

One of the first things you might want to do is restrict your new 500mb pressure data to only the same time period covered by the SST data (hint: `.sel` and `.slice` using the `[''time'']` dimension of your original SST data might help).

9. Create a November-March average for each year, just as you did with STT. Remove the long-term mean from each gridpoint to give you seasonal anomalies for each year from the long-term mean average conditions.

10. Using the years you previously identified in step **7.** above, prepare and map a composite average for the Northern Hemisphere 500mb height anomaly field for each set of years. You may wish to use a stereographic projection when plotting these composites. **In answering this question you should end up with two maps** - one for the mean 500mb anomalies for years where the EOF1 signal has very positive values (probably El Nino years), and one map for the mean anomalies over those years where the EOF1 signal has very negative years (probably La Nina years). As I mentioned above, one challenge you may face will be that your SST data and your 500mb pressure data will span different time periods.

Briefly contrast these patterns associated with the positive vs. negative value years for EOF1. If you have the climate dynamics background, you may chose to interpret or make inferences based on these patterns.

References

Compo, G. P., J. S. Whitaker, P. D. Sardeshmukh, N. Matsui, R. Allan, X. Yin, B. Gleason, R. Vose, G. Rutledge, P. Bessemoulin, et al. (2011), The twentieth century reanalysis project, *Quarterly Journal of the Royal Meteorological Society*, *137*(654), 1–28.

Deser, C., and M. L. Blackmon (1995), On the relationship between tropical and North Pacific sea surface temperature variations, *Journal of Climate*, *8*(6), 1677–1680.

Ghil, M., M. R. Allen, M. D. Dettinger, K. Ide, D. Kondrashov, M. E. Mann, A. W. Robertson, A. Saunders, Y. Tian, F. Varadi, and P. Yiou (2002), Advanced spectral methods for climatic time series, *Review of Geophysics*, *40*(1), 1003, doi:10.1029/2000RG000092.

Neumaier, A., and T. Schneider (2001), Estimation of parameters and eigenmodes of multivariate autoregressive models, *ACM Transactions on Mathematical Software*, *27*(1), 27–57.

Percival, D. B., and W. L. Constantine (2006), Exact simulation of Gaussian time series from nonparametric spectral estimates with application to bootstrapping, *Statistics and Computing*, *16*(1), 25–35.

Rayner, N. A., D. E. Parker, E. B. Horton, C. K. Folland, L. V. Alexander, D. P. Rowell, E. C. Kent, and A. Kaplan (2003), Global analyses of sea surface temperature, sea ice, and

night marine air temperature since the late nineteenth century, *J. Geophys. Res.*, *108*(D14), doi:10.1029/2002jd002670.

Schneider, T., and A. Neumaier (2001), Algorithm 808: ARfit—A Matlab package for the estimation of parameters and eigenmodes of multivariate autoregressive models, *ACM Transactions on Mathematical Software*, *27*(1), 58–65.