PHYS 481 Assignment 4: Regression and Machine Learning

Due: Sunday Oct 13 (23:00)

AI policy for this assignment: no use of generative AI tools is allowed.

On D2L, there are 2 datasets: daily sunspot number since 1818 (SN_d_tot_V2.0.csv) and geomagnetic indices (Kp and Ap) since 1932 (Kp_def.zip). The sunspot number data are from the World Data Center SILSO, Royal Observatory of Belgium, Brussels (https://www.sidc.be/SILSO/datafiles), while the geomagnetic data are from the GFZ German Research Centre for Geosciences, Potsdam, Germany (https://datapub.gfz-potsdam.de/download/10.5880.Kp.0001/). There is also a PDF description for the geomagnetic data (kp_index_data_description_20210311.pdf). Download the data from D2L to a local directory and unzip the zip file.

Question 1: Look at the data before you use it

From the sunspot number CSV file, extract the date (first 3 columns) and the sunspot number (5th column). From the Kp_def files, extract the date (first 6 characters) and the daily Ap index (characters 56, 57 and 58, counting the first character as "1"). Trim the sunspot dataset to start at the same day as the Ap dataset (1932-01-01), check that the resulting datasets are the same length, and plot both datasets. You should see a clear 11-year period in the sunspot data and a less-clear 11-year period in Ap.

HINT: You may wish to store the date as a numpy datetime64 type. For example:

```
date = np.datetime64(f"{year:04d}-{month:02d}-{day:02d}")
```

This will make it easier to display the data without worrying about things like leap years.

Question 2: Linear regression

The sunspot number is a measure of the magnetic activity of the sun and the solar wind. The Ap index is a measure of electrical currents generated in Earth's upper atmosphere in response to the solar wind. (The same electrical currents are responsible for the aurora.) The solar wind takes a few days to get from the sun to the Earth, so there's no reason to suspect the Ap index varies on a day-to-day basis in response to the sunspot number, but the plot from Q1 already suggests there's some correlation over a longer period. Let's try averaging over the 27-day solar rotation period and performing a linear regression.

1. Filter (smooth) the Ap and sunspot data with a 27-day boxcar average.
2. Plot a heatmap of the joint probability distribution ("2d histogram") of the smoothed sunspot number (x) and Ap index (y). Normalize each vertical column in the 2d histogram (i.e. each slice of roughly-constant sunspot number) to sum to one. Include the marginal distributions ("1d histograms"). There is code in the template to assist you.
3. Using a package of your choice (numpy, scipy, scikit-learn, etc.), fit a linear least-squares fit to Ap versus sunspot number. Calculate R^2 and the root mean squared error (RMSE). Add the least-squares line to the plot of the joint probability function, with a label giving the equation of the line of best fit, R^2 and the RMSE.

Question 3: Polynomial Fits and Under- and Over-fitting

The linear fit in the previous question seems to miss a feature at low sunspot number, where the Ap data trends to zero. Let's see if we can pull that out with a nonlinear fit.

1. Split the data into a training set consisting of the first 70% of the data points and a validation set consisting of the remainder.
2. Fit the training set using polynomials from order one through order 20. (Use Polynomial.fit from numpy.polynomial.polynomial or a similar package.)
3. Plot the RMSE on the training set and the RMSE on the validation set as a function of polynomial order.
4. Choose the optimal order for the polynomial fit (i.e. the polynomial fit that best reduces the RMSE on the **validation** set)
5. Plot the joint probability distribution with:
   - the linear fit
   - the optimal polynomial fit
   - the polynomial fit with order 20

Question 4: Neural Networks

As an alternative to the polynomial fitting from the previous question, use a neural network to fit the Ap index as a function of sunspot number.

1. To reduce computation time and increase independence of the points, take every 27^{th} point from the smoothed sunspot number and smoothed Ap to use as the x and y values for the regression.
2. Using the Multi-Layer Perceptron regressor (MLPRegressor) from scikit-learn, train a neural network to predict Ap as a function of sunspot number. The problem is relatively small and simple, so you don't need many layers (1 or 2 should suffice) or a large numbers of neurons (100 is plenty). The relu activation function should work well.
3. Calculate the RMSE for the fit.
4. Re-plot the joint probability distribution with the linear fit and the neural network fit.

[**NOTE:** To check for under- or over-fitting, we could split the data into training and validation sets like in the last question, and search for an optimal set of hyperparameters for the neural network (number of neurons, activation function, regularization parameter alpha, etc.). That's not required for this assignment.]

[**HINT**: This is very similar to the fit performed in the textbook and in the lecture notes]

Data References:

Matzka, J., Bronkalla, O., Tornow, K., Elger, K., Stolle, C., 2021. "Geomagnetic Kp index". V. 1.0. GFZ Data Services. https://doi.org/10.5880/Kp.0001

SILSO World Data Center, 2024. "The International Sunspot Number", International Sunspot Number Monthly Bulletin and online catalogue, Royal Observatory of Belgium, avenue Circulaire 3, 1180 Brussels, Belgium, http://www.sidc.be/silso/

Rubric

All questions are worth 8 points each, assessed according to the following rubric. An example of a “minor error” in the 1-pt categories is if the code is commented, but not clearly, or the plot is missing a unit on one axis.

Code	**Commenting**: Clear and concise comments explaining the code.	1 pt	0: Missing or major error. 0.5: Minor error. 1: Correct.
	Logical Structure: Code is logically organized into functions and modules.	1 pt	
	Readability: Code is well-formatted with consistent and easily understood naming conventions.	1 pt	
Plot(s)	**Clarity**: Plot is clear and easy to understand.	1 pt	
	Labels and Units: Proper labels and units are included on all axes.	1 pt	
	Correctness: Plot shows the expected outcome of the question.	3 pts	0: Plot is missing or entirely incorrect. 1: Plot shows evidence of major conceptual errors. 2: Plot shows evidence of minor errors in the analysis. 3: Correct answer.