Modeling Global CO_2 Growth Rate with Land Loss Indicators using Polynomial Regression Methods and Recurrent Neural Networks

Ishaq Kothari [1] **Nathan Englehart** [1] **Raul Segredo** [1]

Abstract

In this paper, we examine the relationship between the global carbon dioxide rate and different global land loss parameters using data from Yale University's School of Environmental Law and Public Policy, and the National Oceanic and Atmospheric Administration. We describe three different regression approaches used to model atmospheric carbon dioxide levels and their relationship to time, global grassland loss, global wetland loss, and global tree cover loss. This is accomplished by implementing methods for ridge polynomial regression, lasso polynomial regression, and recurrent neural networks using Python 3.10.2. Using cross-validation we compute optimal parameters for each method and using these optimal parameters evaluate each method's predictive performance. Furthermore, we evaluate the effect of each biome degradation on atmospheric carbon dioxide levels. We found that lasso polynomial regression produced the best fitting model, and found that carbon dioxide levels increased as time progressed while tree cover decreased. Additionally, both grassland loss and wetland loss caused a decrease in carbon dioxide levels. However, using background research and analyzing variance between our regression models we believe that polynomial models and recurrent neural networks may not be the optimal models for predicting carbon dioxide levels using land loss indicators, and a more complex model may be needed.

1. Introduction

The effects of global climate change on humans are serious and far reaching, hence, climate change is one of the most studied fields in the applied sciences. Among the most critical metrics in measuring climate change is the rate of buildup of CO_2 emissions in Earth's atmosphere, which has been shown to cause increased average temperatures globally (Clair et al., 2022). The consequences of this increase in average global temperatures includes but is not limited to: increased risk of natural disasters, large-scale disease outbreaks, and rising sea levels. To counter rising global temperatures and its negative effects, it has been shown that maintaining healthy ecological systems can significantly aid in offsetting global carbon emissions, while the destruction and removal of such ecosystems has the opposite effect (Malhi et al., 2020). This paper intends to model the effects of ecosystem loss and its impact on carbon emissions using various regression methods including lasso polynomial regression, ridge polynomial regression, and recurrent neural networks.

There have been many studies on the effects of different ecosystems mitigating carbon emissions. This paper will focus specifically on wetlands, grasslands, and forests. For one, wetlands act as an important sink for carbon emissions by efficiently sequestering carbon through tree and soil biomass. On the other hand, the loss of wetlands can cause the release of previously sequestered carbon, leading to large spikes in carbon dioxide levels in the atmosphere (Yahnke, 2022). Secondly, tree-cover plays an essential role in reducing carbon emissions by absorbing carbon emitted to the atmosphere through the process of photosynthesis (Janowiak et al., 2017). Similar to the loss of wetlands, deforestation releases carbon previously sequestered in the leaves of trees, causing an increase in atmospheric carbon dioxide. Finally, just as the aforementioned biomes, grassland ecosystems also reduce carbon emissions by sequestering carbon underground and undergoing photosynthesis (Chang et al., 2021).

Using data from the Environmental Performance Index (EPI) from Yale University's School of Environmental Law and Public Policy (Wendling et al., 2020) and NOAA's Mauna Loa Observatory (Tans & Keeling, 2022), we modeled the effects of three different indicators: wetland loss, grassland loss, and tree cover loss and their impact on atmospheric carbon levels. Through our regression analy-

[1] Oberlin College, Oberlin, OH, USA. Correspondence to: Ishaq Kothari <ikothari@oberlin.edu>, Nathan Englehart <nengleha@oberlin.edu>.

sis, we formulated a model which can help simplify trends in ecosystem degradation across the world. Furthermore, our model could be used to predict future carbon dioxide growth due to land loss, given parameters for wetland loss, grassland loss, and tree cover loss.

Before running regression on our data-sets, we first had to sanitize each data-set. To start, we needed to address the major discrepancy between the EPI data-set and the atmospheric carbon data-set. This is because the EPI data-set measured the percentage of ecosystem loss in each country, while the Atmospheric Carbon data-set measured the atmospheric carbon dioxide levels globally. We would not return significant findings if we ran regression comparing global carbon levels to national land loss, since land loss varies dramatically globally, while carbon levels are relatively constant globally. In an effort to remedy this discrepancy, we chose to average the percentage of land loss across countries to approximate the percentage change of each type of land loss globally. Furthermore, in averaging the data, we removed countries where insufficient amounts of data were collected for each land loss indicator. Thus our transformed data-set improved the correlation between atmospheric carbon dioxide levels and global land loss. After standardizing our data-set, we then used the data analysis library, pandas, to build our matrices for our regression methods. In our input matrix **X**, we added the data for each input parameter (year, average wetland loss, average grassland loss, average tree-cover loss). On the other hand, we populated our target matrix, **t**, as a single column consisting of the atmospheric carbon Levels from 1995 to 2020. At this point, we were able to use our data to run each regression method, starting with lasso polynomial regression.

2. Methodology

We wish to create the best model for the relationship between variables from year to year by finding the model that minimizes error when predicting the combined dataset's ground truth data for each year. To do so, first, we must create the ideal model for each method and compute its error rate.

2.1. Polynomial Regression

Polynomial Regression was the first method we used to model the relationship between our four indicators of ecosystem degradation and growth in carbon dioxide emissions. A polynomial regression lent itself well to our data, given that we saw a sharp increase in carbon emissions from 1995 to 2020 when first examining the combined dataset. As such, we concluded the rapid growth of a polynomial would effectively model our data. The equation we chose to initially model our data by minimizing error was

$$f(x_1, x_2, x_3, x_4) = c_1 + x_1^1 + x_1^2 + ... + x_1^n + c_2 + x_2^1 + x_2^2 + ... + x_2^n + c_3 + x_3^1 + x_3^2 + ... + x_3^n + c_1 + x_4^1 + x_4^2 + ... + x_4^n$$

This equation gave us weights for each of our parameters, allowing us to assess the impact of each individual ecosystem's degradation on carbon dioxide growth.

The generic version of the Polynomial regression uses the method of Ordinary Least Squares (OLS) to calculate the error between the actual value of growth in carbon dioxide emissions and the growth percentage given by our regression. We can see the equation for optimizing Ordinary Least Squares regression (Rogers & Girolami, 2016) below

$$\hat{\mathbf{w}} = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{t} \tag{1}$$

A potential problem with OLS is that our regression algorithm will always favor high degree polynomials, as they would arbitrarily fit our data better than low degree polynomials. This is because high degree polynomials tend to overfit data by dramatically changing their behavior to pass through as many points as possible. This often makes high degree polynomials ineffective in generalizing for continuous datasets. To prevent our model from only returning high order polynomials, we chose to implement regularization in our polynomial regression, which penalizes high order polynomials.

In order to run each of the polynomial regression methods, we first expanded our input matrix to include columns raised to the power specified by the polynomial order. We were then able to organize our input matrix, **X**, as shown below, where x_1 corresponds to years from 1996 to 2020, x_2 corresponds to wetland loss, x_3 corresponds to tree-cover loss, and x_4 corresponds to grassland loss (for code see lib/utils.py line 214).

$$\begin{bmatrix} x_1^0 & x_2^0 & x_3^0 & x_4^0 \\ x_1^1 & x_2^1 & x_3^1 & x_4^1 \\ ... & ... & ... & ... \\ x_1^n & x_2^n & x_3^n & x_4^n \end{bmatrix}$$

After building our prediction matrix, we then then standardized **X** using a Z-score to account for the different magnitudes of our parameters. After satisfying these two conditions, we were able to fit a multivariate linear regressions to our train data.

2.2. Lasso Polynomial Regression

We chose to implement the least absolute shrinkage and selection operator, also known as lasso regularization as method of regularization for our first model. To do so, we used the sci-kit learn library (Géron, 2019) for Python 3.10.2 (for code see lib/reg.py line 15). The lasso regression method encourages the selection of simpler models with fewer parameters by incorporating the OLS method of calculating the error between our regression's actual and predicted values, and including a second term that takes into account the magnitude of the coefficient, as seen in the equation below (Dangeti, 2017).

$$RSS(\beta) = \sum_{i=1}^{n}(y_i - \sum_{j=1}^{p}(x_{i_j}\beta_j))^2 + \lambda\sum_{j=1}^{p}|\beta_j| \tag{2}$$

The higher the magnitude of each coefficient, the greater the error generated; this penalizes high degree polynomials that have higher magnitude coefficients due to their erratic behavior around data points. As such, the lasso regression method reduces coefficients towards zero, generally returning a lower degree polynomial.

2.3. Ridge Polynomial Regression

Ridge polynomial regression is similar to lasso polynomial regression in that it uses coefficients in front of polynomial values to regularize the regression coefficients. The loss function for a ridge regression (Dangeti, 2017) is expressed with

$$RSS(\beta) = \sum_{i=1}^{n}(y_i - \sum_{j=1}^{p}x_{i_j}\beta_j)^2 + \lambda\sum_{j=1}^{p}\beta_j^2 \tag{3}$$

which can then be written in matrix form to represent an arbitrary size of data (Rogers & Girolami, 2016) as

$$\hat{\beta} = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1}\mathbf{X}^T\mathbf{t} \tag{4}$$

We can note that as opposed to lasso regression which penalizes large coefficients by adding a second term to the loss function and taking the absolute value of each coefficient, ridge regression squares each coefficient, but does not reduce the number of variables used, since coefficients of ridge regression never shrink down to zero (Rogers & Girolami, 2016).

In order to test our own implementation of a regression model (for code see lib/reg.py line 157 and lib/local_reg_class.py), we decided to write a program to calculate the weights of our polynomial. This can be done by first organizing our input matrix, **X**, as mentioned previously, so that each row corresponds with the desired degree. Then, we derive an expression to find the optimal regression for our model. We did this by taking the partial derivative of our ridge regression function with respect to our weights, setting the expression equal to 0, and then solving for our weights. The final result shown below, allows us to input a matrix, **X**, as well as a target vector, **t**, and return the coefficients of the weights required for our regression model.

$$L(\mathbf{w}, \lambda) = (\mathbf{X}^T\mathbf{X} + \lambda\mathbf{I})^{-1} \tag{5}$$

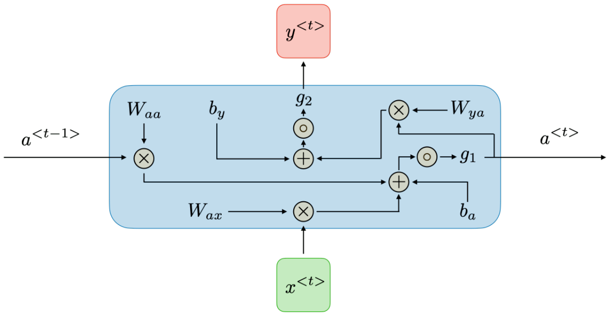
Our finished model is now able to fit an input matrix, **X** and target vector, **t**, and then predict a target matrix given another matrix of the same size as **X** (for code see main.py line 225).

2.4. Neural Network Regression

The second method we chose to formulate our model was a neural network regression model. To write our implementation, as with lasso regression, we again used the sci-kit learn library (Géron, 2019) for Python 3.10.2 (for code see lib/reg.py line 120). Unlike lasso regression, neural networks use nonlinear functions to model a possible function for the given dataset. Neural networks are unique in their ability to find hidden correlations between variables through simplifying our parameters using hidden layers. Furthermore, while neural networks have comparably longer runtimes than lasso regression, neural networks do not increase in size as rapidly as other regression methods allowing for the regression model to remain relatively concise, despite having large numbers of inputs (Géron, 2019).

Neural networks work by taking in our input, which in our case would be our parameter values for years, wetland loss, tree-cover loss, and grassland loss, and forwarding them to the next set of nodes, where they are influenced using a set of nonlinear functions as weights to transform the nodes to the next hidden layer. Through this transformation, we are able to produce a plausible model for a singular set of parameters.

The problem with generic neural networks is that they are unable to use past inputs to update the function continuously; as such, we are unable to make a model using time series data. We solved this problem by introducing recurrent neural networks which work by using feedback loops that allow us to feed our function back into our input layer in a process known as backtracking. This process allows us to continuously update our function as we iterate through our dataset for each year. The method of backtracking is illustrated in the diagram below.



Similar to the polynomial regression model, the recurrent neural network regression model uses a loss function (Amidi & Amidi, 2022) to optimize our function as seen below,

$$L(\hat{y}, y) = \sum_{t=1}^{\Gamma_y} L(\hat{y}^{<t>}, y^{<t>}) \tag{6}$$

which compares the loss at each time, t, between the projected and actual values. To incorporate backtracking into our loss function we take the partial derivative of the loss function with respect to the previous weights, and then optimize the derivative of the old loss function to update our parameters.

Additionally, to prevent overfitting, sci-kit allows one to use L2- Regression, also known as Ridge regression to apply regularization to our model, which takes into account the squared value of each of the coefficients in our regression model. The updated loss function for a deep neural network can be seen below.

$$Loss = Error(y, \hat{y}) + \lambda \sum_{i=1}^{n} w_i^2 \tag{7}$$

2.5. Evaluating Model Parameters

For each polynomial regression model, we wished to find the optimal parameters for polynomial order and regularization coefficient (α in lasso regression and λ in ridge regression). To do so, we ran both lasso polynomial regression and ridge polynomial regression through a grid search algorithm.

Specifically, in our grid search algorithm, we ran k-fold cross-validation with $k = 5$ (the number of folds can be change in main.py on line 256), for each combination of polynomial order and regularization coefficient. In k-fold cross-validation, we split the train data-set into k folds, then loop k times, and on each run set the one fold to act as the validation set and the other $k - 1$ folds to act as the train set. Plugging in $\mathbf{t}$ and $\mathbf{X}$ from the train set to fit the ridge function and the validation $\mathbf{t}$ for the ridge function to predict gave us our $\hat{\mathbf{t}}$. We then compared the sealed/ground-truth validation $\mathbf{t}$ to our predicted $\hat{\mathbf{t}}$ using mean squared error (MSE) function (Rogers & Girolami, 2016) given by

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (Y_i - \hat{Y}_i)^2 \tag{8}$$

Averaging the MSE across folds gave us the generalization error for each combination of regularization coefficient and polynomial order (for code see lib/kfcv.py lines 69, 146, and 223). In grid-search, we kept track of the combination that minimized generalization error in a global minimum tuple. When grid search function finished, we returned this tuple as our ideal parameters for lasso/ridge regression (for code see main.py line 27).

Our neural network implementation also required two parameters, an activation function and regularization constant, hence, grid-search worked in much the same way for neural networks as it did for polynomial regression. With grid-search and cross-validation, we were thus able to find the optimal parameters for both our neural network and polynomial regression implementations.

3. Discussion

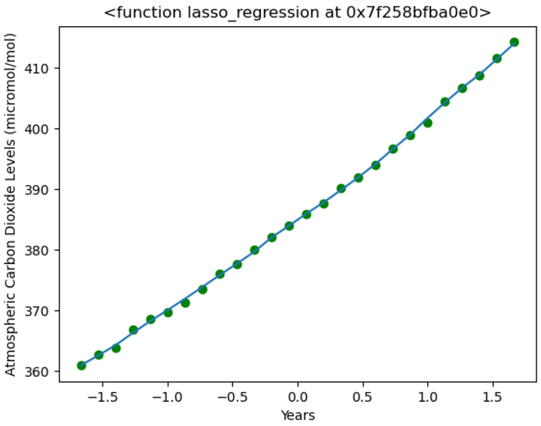
After creating ideal models that minimize error, we wish to discuss and reflect on the performance of each individual model. As in the previous section, we will first discus the two polynomial regression methods and then our recurrent neural network model.

3.1. Lasso Performance

Using our grid-search function, we then found the ideal polynomial order values and α constant combination to be $D = [3, 4, 2, 4]$ and $\alpha = 1 \cdot 10^{-6}$, which returned the minimized MSE of approximately 0.082827 and generalized MSE of approximately 51.353236. Given the regression coefficients coefficients given by our computation, where x_1 represents year, x_2 represents percentage wetland loss, x_3 represents percentage tree cover loss, x_4 represents grass land loss, the functional form of our model can be written as

$$f(x_1, x_2, x_3, x_4) = 0 + 15.8533572x_1^1 + 1.09659037x_1^2 + 0x_1^3 - 0.0829061786x_2^1 - 0.0497396655x_2^2 - 0.0214558427x_2^3 + 0x_2^4 + 0.201884654x_3^1 + 0.0182647057x_4^1 - 0.0133582034x_4^2 - 0.0295427239x_4^3$$

Graphing the predicted carbon dioxide levels, $\hat{\mathbf{t}}$, against years, x_1, gives us

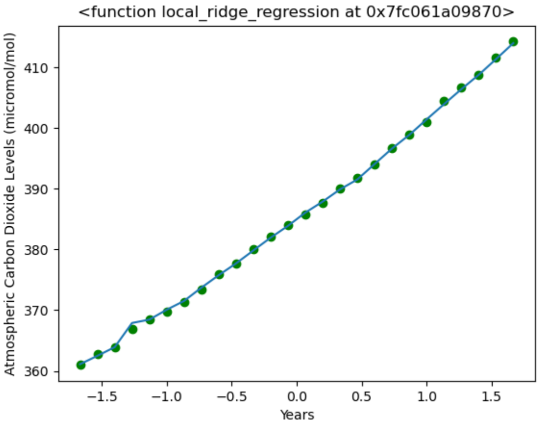


3.2. Ridge Performance

For ridge regression, we found that the ideal polynomial order values to be $D = [3, 4, 2, 4]$ and the ideal λ regularization coefficient to be approximately 0.215443469. We found the generalization mean squared error to be approximately 48.9417067, and the data-set mean squared error to be approximately 0.096699406. Again, given the regression coefficients coefficients given by our computation, where x_1 represents year, x_2 represents percentage wetland loss, x_3 represents percentage tree cover loss, x_4 represents grass land loss, the functional form of our model can be written as

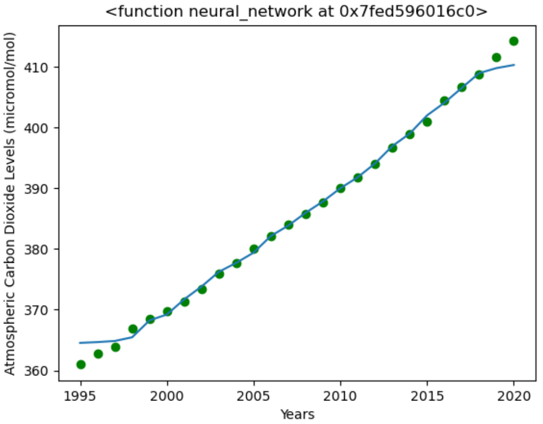
$$f(x_1, x_2, x_3, x_4) = 3.84670692 \cdot 10^2 + 1.50691665 \cdot 10^1 x_1^1 + 1.48282993 x_1^2 + 9.21453641 \cdot 10^{-1} x_2^1 - 8.61341048 \cdot 10^{-1} x_2^2 - 3.54295997 \cdot 10^{-1} x_2^3 + 0 x_2^4 + 1.10863923 \cdot 10^{-1} x_3^1 + 4.29048455 \cdot 10^{-2} x_4^1 - 3.33634363 \cdot 10^{-1} x_4^2 - 3.85946910 \cdot 10^{-1} x_4^3$$

Here, graphing the predicted carbon dioxide levels, $\hat{\mathbf{t}}$, against years, x_1, gives us



3.3. Recurrent Neural Network Performance

In our recurrent neural network implementation, we found the lowest generalized mean squared error to be approximately 36.33113 using tanh as our activation function and a λ value of 1.6681. In addition, we found the dataset squared error to be approximately 1.5616. We then graphed our regression function, as shown below, by predicting atmospheric carbon levels results using our standardized input data and plotted the actual atmospheric carbon dioxide levels on the same plot.



4. Conclusion

Our three regression models yielded varied results. We found that our neural network implementation had the lowest generalization error when running cross-validation. However, we found that lasso regression had the lowest mean squared error, when predicting atmospheric carbon levels from the Environmental Performance Index (EPI).

We can assess the generalization performance of our modeling techniques using our graphs for each model as well as our regularization coefficients. Due to the time-series organization of our data, we found that cross-validation did not perform well in creating a generalized model, as both ridge regression and lasso regression reported very low regularization coefficients. Additionally, all of our graphs show clear evidence of over-fitting due to the close fit between the atmospheric data and our regression models. We suspect this is due to our atmospheric carbon data being relatively constant in its increase from year to year. As such, despite choosing random data points for our validation sets, our cross-validation implementation still returned functions that fit the original data very well. If our EPI data-set had collected data more frequently, we could have added a wider range of values into our input matrix. Cross-validation is more likely to create more varied validation sets which would improve our generalization results.

Given our multiple parameters, we are able to identify trends in our data from our polynomial regression functions. All of our models accurately modeled the increase in atmospheric carbon levels over time with great success. However, our models had varying success in finding relationships between our three land loss parameters and atmospheric carbon levels. By examining the coefficients of our most accurate model: we found that our lasso regression model suggests a quadratic relationship between atmospheric carbon dioxide levels and time. We also found that there is a negative cubic relationship between wetland loss and atmospheric carbon dioxide levels, a linear relationship between tree cover loss and atmospheric carbon dioxide levels, and a negative cubic relationship between grassland loss and atmospheric carbon levels.

Our performance for ridge regression which had a slightly large mean square error than our lasso regression yielded somewhat similar results to lasso regression. Our ridge regression model predicted a quadratic relationship between year and atmospheric carbon levels. For wetland loss, we found that ridge regression predicts a negative cubic relationship between wetland loss and atmospheric carbon dioxide levels. For tree cover loss we found that ridge regression predicts a linear relationship between tree cover loss and atmospheric carbon dioxide levels. Finally, we found that our ridge regression model predicts a negative cubic relationship between grassland loss and atmospheric carbon levels.

In assessing our results for the recurrent neural network model, we found that our regression model did well to predict data values in the past, but the end-behavior of the tanh function led to the prediction curve leveling out as it predicted values for later years, this led to the recurrent neural network having the highest mean squared error of all our models. Furthermore, the inaccurate end behavior of the function led to our neural network not serving well as a predictive model. We believe that this inaccuracy was due to the use of a smaller data-set, which prevented the neural network from being able to create many hidden layers, thus resulting in a less accurate model.

Our results do not necessarily support our initial research which suggested that wetland, tree-cover, and grassland loss would cause an increase in atmospheric carbon dioxide levels. The discrepancies between our model and research may be due to potential problems with our methodology. Our parameters for land loss do not necessarily provide all the necessary information about global land utilization, and its relationship to mitigating carbon emissions. Furthermore, the size of our data-set is limited due to the lack of global land loss data. In addition, our practice of averaging land loss across many countries may not have yielded representative global land loss indicators. Finally, we assumed changes in atmospheric carbon dioxide due to human emissions, would be modeled using our time parameter, however, using additional parameters to model artificial carbon dioxide emissions could potentially better isolate the relationship between our land loss parameters and atmospheric carbon levels.

Inaccuracies in our model lead us to believe that the relationship between land loss and carbon dioxide levels is more complex than a polynomial model. Thus exploring different methods of regression could yield clearer results. For example, expressing our multivariate model as a system of differential equations would allow us to more accurately see the relationship between carbon emissions growth and different land loss parameters. These models are more commonly used in climate science because of their ability to model quantitative data and physical processes which cause fluctuation in atmospheric carbon dioxide levels.

References

Amidi, A. and Amidi, S. Recurrent neural networks. Stanford University, 2022.

Chang, J., Ciais, P., and Gasser, T. Climate warming from managed grasslands cancels the cooling effect of carbon sinks in sparsely grazed and natural grasslands. pp. 118. Nat Commun 12, 2021.

Clair, A. L. S., Zhang, G., Dolezal, A. G., O'Neal, M. E., and Toth, A. L. Agroecosystem landscape diversity shapes wild bee communities independent of managed honey bee presence. Agriculture, Ecosystems & Environment 327, 2022.

Dangeti, P. In *Statistics for Machine Learning*, pp. 75–76. Packt Birmingham - Mumbai, 2017.

Géron, A. In *Hands on Machine Learning with Scikit-Learn Keras & TensorFlow*, pp. 137–139, 139–141, 293. O'Reilly, 2019.

Janowiak, M., Swanston, C., and Ontl, T. Importance of forest cover. Washington, DC, 2017. U.S. Department of Agriculture, Forest Service, Climate Change Resource Center.

Malhi, Y., Franklin, J., Seddon, N., Solan, M., Turner, M., Field, C., and Knowlton, N. Climate change and ecosystems: Threats, opportunities and solutions. Philosophical Transactions of the Royal Society B: Biological Sciences 375, 1794, 2020.

Rogers, S. and Girolami, M. In *A First Course in Machine Learning*, pp. 25. Nat Commun 12, 2016.

Tans, P. and Keeling, R. gml.noaa.gov/ccgg/trends/, 2022. NOAA's Mauna Loa Observatory Hawaii.

Wendling, Emerson, de Sherbinin, and Esty. Environmental performance index. https://epi.yale.edu/, 2020. Yale University Center for Environmental Law & Policy.

Yahnke, A. Wetlands & climate change. Pullman, WA, 2022. Washington State Department of Ecology.