4/30/25

Least Squares Regression and Interaction Matrices for a Finite Dimensional Data Set

APPM 3310

CHIARA PESCE, DANNY ALEMAYEHU, CALEN MICHEL

Abstract

Understanding how variables in air quality impact one another is essential for assessing pollutant behavior and developing long-term solutions to air quality challenges. This project employs least-squares linear regression in combination with the construction of an interaction matrix to classify variables (such as carbon monoxide (CO), carbon dioxide (CO2), and volatile organic compounds (VOCs)) with respect to their systemic roles. Specifically, each variable is categorized as more or less interactive, and as dominant or subordinate within the system interaction. The dataset used in this analysis was intended to investigate carbon monoxide concentrations from tailpipe emissions with respect to distance from a highway in high alpine environments, however any data set can be analyzed using the processes outlined in this paper.

Attribution

Data was collected by Chiara for a different class using equipment from the Hannigan Air Quality and Technology Research Lab. Mathematical formulation is done by Chiara, Python least squares code is written by Danny, and writeup is done by all three team members.

Introduction

Interaction matrices have been used in environmental science since the development of the Leopold Matrix in 1971 (Leopold et al., (1971)) primarily as a tool to identify and effectively weight different activities to assess the potential environmental impact of a system. The interaction matrix utilized in this paper is a modification on the Leopold matrix originally proposed by Mavroulidou et al. (2004) to assess air quality in urban environments along with GIS mapping, the latter of which will not be investigated in this paper. A simple interaction matrix follows the format below:

$$\begin{bmatrix} Parameter\ 1 & \dots & I_{1,n} \\ \vdots & \ddots & \vdots \\ I_{n,1} & \dots & Parameter\ n \end{bmatrix}$$

Where the parameters of interest are placed along the diagonal and interaction coefficients are placed in each of the off-diagonal entries. A clockwise convention is adopted such that $I_{1,n}$ is the influence of parameter 1 on parameter n and $I_{n,1}$ is the influence of parameter n on parameter 1 for an n dimensional system. The challenge then comes with assigning appropriate weighing factors for each interaction. For this paper, we will employ linear least-squares regression to numerically calculate each interaction value.

Linear regression will be performed between each variable of interest and the value of the least-squares line will be taken as the coefficient for the variable interaction. For example, in the matrix described above, the slope from least-squares computation with parameter 1 as the independent variable and parameter n as the dependent variable will be taken as the value for $I_{1,n}$. In this way, we construct an interaction matrix representative of the system.

$$\begin{bmatrix} & j=1 & j=2 & j=3 & Cause \sum I_j \\ i=1 & Parameter\ 1 & I_{1,2} & I_{1,3} & I_{1,2}+I_{1,3} \\ i=2 & I_{2,1} & Parameter\ 2 & I_{2,3} & I_{2,1}+I_{2,3} \\ i=3 & I_{3,1} & I_{3,2} & Parameter\ 3 & I_{3,1}+I_{3,2} \\ Effect \sum I_i & I_{2,1}+I_{3,1} & I_{1,2}+I_{3,2} & I_{1,3}+I_{2,3} & \sum I \end{bmatrix}$$

From here, each row and column can be summed and expressed as a percentage of the whole such that each variable can be classified as more or less interactive and subordinate or dominant with respect to the system (Mavroulidou et al. (2007)). A representative chart can then be created to visually classify each variable in the categories listed above. Weighing factors for the data set can be calculated by averaging the cause and effect values for a given parameter and expressing it as a percentage of the total sum of the interaction matrix. We'll also demonstrate below that the solution to our least-squares regression for each pair of random variables has a unique solution indicating that our model is in fact unique for some input.

In this paper, the parameters of interest are chosen to be relative humidity (%), temperature (°C), light VOCs (ADU), heavy VOCs (ADU), ozone (ADU), CO (ADU), CO_2 (ADU), elevation (ft) and distance from the neatest highway (miles).

Mathematical Formulation

The bulk of the relevant computation will be performing least squares between each respective variable in the data set. Let the basic equation for a linear line be:

$$y = \beta_0 + \beta_1 \cdot x$$

For an nth dimensional system of equations, we can form the system

$$\begin{bmatrix} y_1 = \beta_0 + \beta_1 \cdot x_1 \\ y_2 = \beta_0 + \beta_1 \cdot x_2 \\ \vdots \\ y_n = \beta_0 + \beta_1 \cdot x_n \end{bmatrix}$$

Which can be represented in matrix form by the following equations.

$$\begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \\ \vdots & \vdots \\ 1 & x_n \end{bmatrix} \cdot \begin{bmatrix} \beta_0 \\ \beta_1 \end{bmatrix}$$

This system follows the standard form $\vec{b} = A \cdot \vec{x}$ where $\vec{b} \in R^n$, $A \in M^{nx2}$, and $\vec{x} \in R^2$. As you may note, this system is overdetermined and thus can be infered to be inconsistent due to the nature of the raw data not being precisely linear. Consequently, we need to utilize the least-sqares regression to obtain a best fit linear solution. A is rectangular and noninvertible, however its columns are linearly independent, assuming that $\forall x_1 \neq x_2 \cdots \neq x_n$. This is a valid assumption because each x represents an independent variable specific to a data point, and these will naturally be unique. This means that $\vec{b} \notin \text{img}(A)$, and thus no solution exists. However, ker(A) = $\{0\}$ since the columns are linearly independent meaning that our least squares regression will in fact have a unique solution which is a good indicator of the model's accuracy. For an inconsistent system, the best way to approximate a solution is to project $\vec{b}$ onto the img(A) which yields the

smallest possible deviation from the system solution (Olver, Chehrzad (2018)). Let the vector $\vec{b}$ be represented as

$$\vec{b} = \vec{b}^* + (\vec{b} - \vec{b}^*)$$

Where $\vec{b}^*$ represents the component of $\vec{b}^* \in img(A)$ and the quantity $(\vec{b} - \vec{b}^*)$ is component of the vector $\vec{b}$ perpendicular to the img(A). Since the img(A) and the ker (A^T) are complimentary subspaces, we can then conclude that the quantity $(\vec{b} - \vec{b}^*) \in$ ker (A^T). Using this idea, we can multiply the equation $\vec{b} = A \cdot \vec{x}$ by A^T to eliminate the component of $\vec{b} \in$ ker (A^T).

$$A^T \cdot \vec{b} = A^T \cdot A \cdot \vec{x}$$

Since the columns of A are linearly independent , the matrix formed by computing $A^T \cdot A$ is invertible (Olver, Chehrzad (2018)) and thus we can rearrange the above equation to get

$$A \cdot (A^T \cdot A)^{-1} \cdot A^T \cdot \vec{b} = A \cdot \vec{x} = \vec{b}^*$$

Where the sequence $A \cdot (A^T \cdot A)^{-1} \cdot A^T$ represents a projection matrix for the system. In this way, we can then solve the system for $\vec{x}$ and thus obtain our respective values of β_0 and β_1, taking β_1 as our interaction coefficient for the respective parameters in the interaction matrix.

Computationally, this method can induce large amounts of floating-point error especially since the condition number of $A^T \cdot A$ will be the condition number of A squared so that product will have a far higher condition number and in turn, more floating-point error (Cheney, Kincaid (2007)). Because of this computational limit, we may choose to create an orthogonal basis for A using QR factorization such that Q forms an orthonormal basis on M^{nx2}, and R is an upper triangular, invertible matrix on M^{2x2}.

$$(Q \cdot R)^T \cdot \vec{b} = (Q \cdot R)^T \cdot Q \cdot R \cdot \vec{x}$$

And thus

$$Q \cdot R \cdot \left((Q \cdot R)^T \cdot (Q \cdot R)\right)^{-1} \cdot (Q \cdot R)^T \cdot \vec{b} = (Q \cdot R) \cdot \vec{x}$$

Which will simplify down to

$$Q^T \cdot \vec{b} = R \cdot \vec{x}$$

$$Q \cdot Q^T \cdot \vec{b} = \vec{b}^*$$

Where the sequence $Q \cdot Q^T$ represents a projection matrix for the system. Similarly, we can then solve the system for $\vec{x}$ and thus obtain our respective values of β_0 and β_1, taking β_1 as our interaction coefficient for the respective parameters in the interaction matrix. This will be more numerically stable than using the normal equations as there are fewer operations and QR has factorization methods suitable for ill-conditioned matricies.

Examples and Numerical Results

As mentioned above, the QR factorization is critical to solving LSRL (least squares regression line) systems computationally. However, we must also consider the algorithm used in determining the QR implementation. The Gram-Schmidt process typically induces a large amount of floating-point error for ill-conditioned systems which is true for some of the matrices we generate. For example, when we consider CO(ADU) as our independent RV in our LSRL system, our coefficient matrix has a condition number $\kappa(A) \geq 19833$ which according to (Cheney, Kincaid (2007)) means we could lose up to 5 digits of accuracy not accounting for the additional floating-point loss from arithmetic operations which is not ideal.

Thankfully, matrix solutions are computed with NumPy's linear algebra library which uses householder matrices to calculate the QR factorization which is far more numerically stable for ill-conditioned matrices. Using the QR factorization to solve our system is computationally more expensive than traditional LU factorization but we obtain numerical precision for the tradeoff. In addition, using the QR factorization allows us to simplify our system, so we perform less matrix operations in comparison to our initial above definition which utilizes several

transposes, inverses, and matrix multiplications which introduces computational time and numerical imprecision in the case of inverses and multiplication.

There are also have a few minor advantages of the QR factorization. One is that the transpose for Q allows us to compute Q's inverse without introducing any additional floating-point error, so we perform the final formula $R^{-1} \cdot Q^T \cdot b = x$ to obtain our solution coefficients β_0 and β_1 for our LSRL. Additionally, we can obtain a space advantage by discarding the coefficient matrix after finding our QR matrix and performing an inplace transpose on Q and inplace inverse on R. This is opposed to our original system definition which requires us to save the coefficient matrix, the transpose, and inverse to fully solve our system at its simplest.

One interesting note that arose from the experimentation with our factorization and NumPy's linear system solving algorithm is that our QR implementation is on average, 300 ± 11 microseconds (Appendix GH reference) faster with 1000 samples of solving the LSRL line for independent RV, CO(ADU), and dependent RV, T(degC). We suspect that in our QR implementation we take advantage of our formulaic simplifications to our LSRL system definition to reduce the amount of matrix operations/factorizations to complete as opposed to utilizing NumPy's implementation which has less assumptions to simplify our problem. This showcases how powerful using QR to solve for our Interaction constants are as opposed to using more atypical methods like LU and the matrix inverse.

After computing the slope for each interaction between variables of interest, we get the following resultant interaction matrix.

										Cause
	Relative Humidity	-4.620E-01	2.606E+00	1.101E+01	-8.914E+00	-5.862E+00	4.340E+00	-3.098E+01	-1.688E-02	6.420E+01
	-1.924E+00	**Temperature**	-3.462E+00	-1.917E+01	2.194E+01	1.084E+01	-8.342E+00	3.647E+01	1.751E-02	1.137E+02
	8.941E-02	-2.852E-02	**LightVOC**	6.258E+00	1.046E+00	1.487E+00	4.497E-01	-5.851E+00	-3.799E-03	1.521E+01
	2.804E-03	-1.172E-03	4.644E-02	**HeavyVOC**	1.488E-01	3.211E-01	1.162E-02	2.490E-02	2.965E-05	5.568E-01
	-2.750E-02	1.625E-02	9.405E-02	1.802E+00	**Ozone**	3.454E-01	-1.358E-01	4.846E-01	3.172E-04	2.906E+00
	-2.803E-03	1.245E-03	2.072E-02	6.029E-01	5.354E-02	**CO**	-4.154E-03	4.007E-01	2.727E-04	1.086E+00
	1.912E-01	-8.825E-02	5.775E-01	2.012E+00	-1.940E+00	-3.828E-01	**CO2**	-7.585E+00	-4.813E-03	1.278E+01
	-8.254E-03	2.333E-03	-4.543E-02	2.606E-02	4.185E-02	2.232E-01	-4.586E-02	**Elevation**	6.973E-04	3.937E-01
	-8.170E+00	2.035E+00	-5.358E+01	5.635E+01	4.976E+01	2.760E+02	-5.286E+01	1.266E+03	**Distance From Road**	1.77E+03
Effect	1.042E+01	2.635E+00	6.043E+01	9.723E+01	8.385E+01	2.954E+02	6.619E+01	1.348E+03	4.433E-02	1.976E+03

Figure 1: Interaction matrix for system variables computed using least squares linear regression.

The sum of each parameter's interaction values is displayed in the last row and column of the matrix, and the sum of all interaction values is displayed in the lower right corner of the matrix. As described earlier, the matrix has a clockwise convention so all entries in a given parameter's row represent how it influences the system, and each entry in a given parameter's column represent how the system influences the parameter. Each variable is plotted with its "cause" interaction value against its "effect", classifying the variable based on its degree of interaction and the level which it is dominant or submissive in the system.

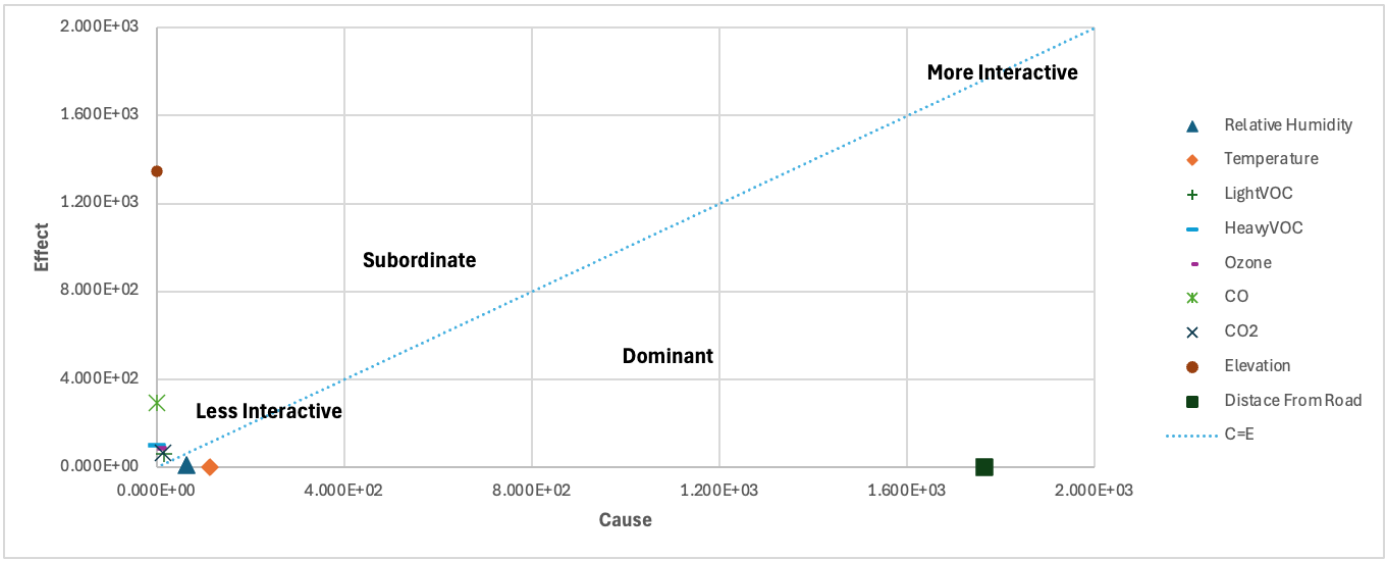

Figure 2: Interaction behavior / characteristics for each parameter in the system.

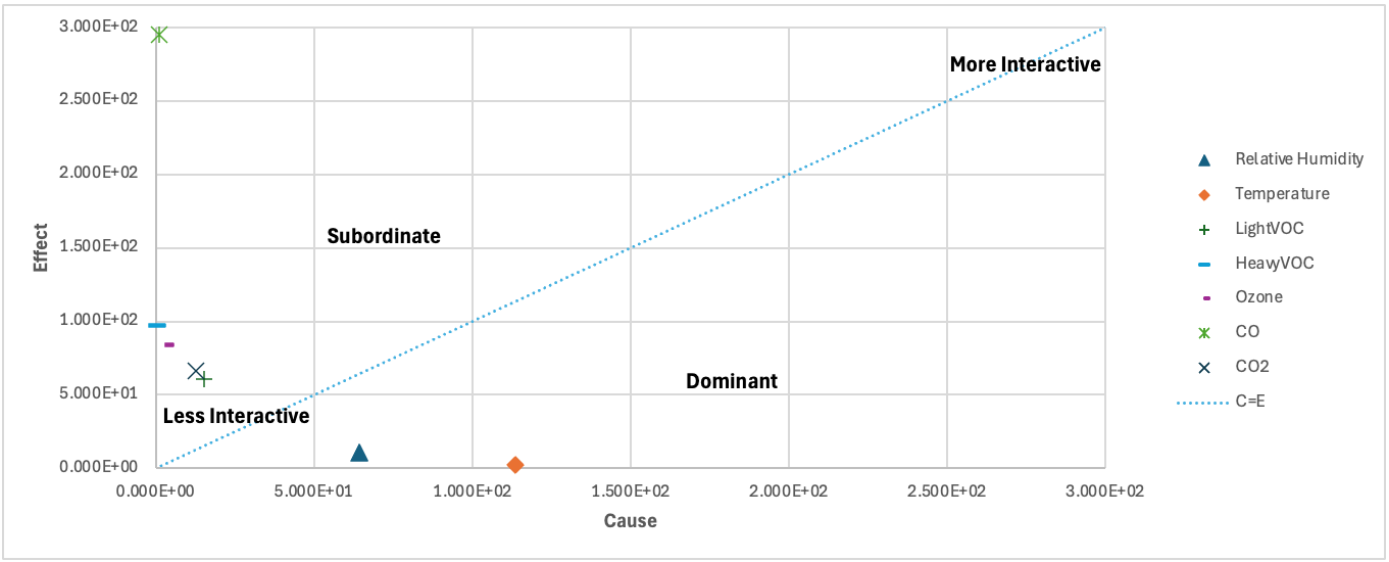

Figure 3: Interaction behavior / characteristics for parameters in the system excluding elevation and distance from road for better visualization.

Weighing factors are then computed by averaging each parameter's "cause" and "effect" value, normalized to the matrix total and expressed as a percentage.

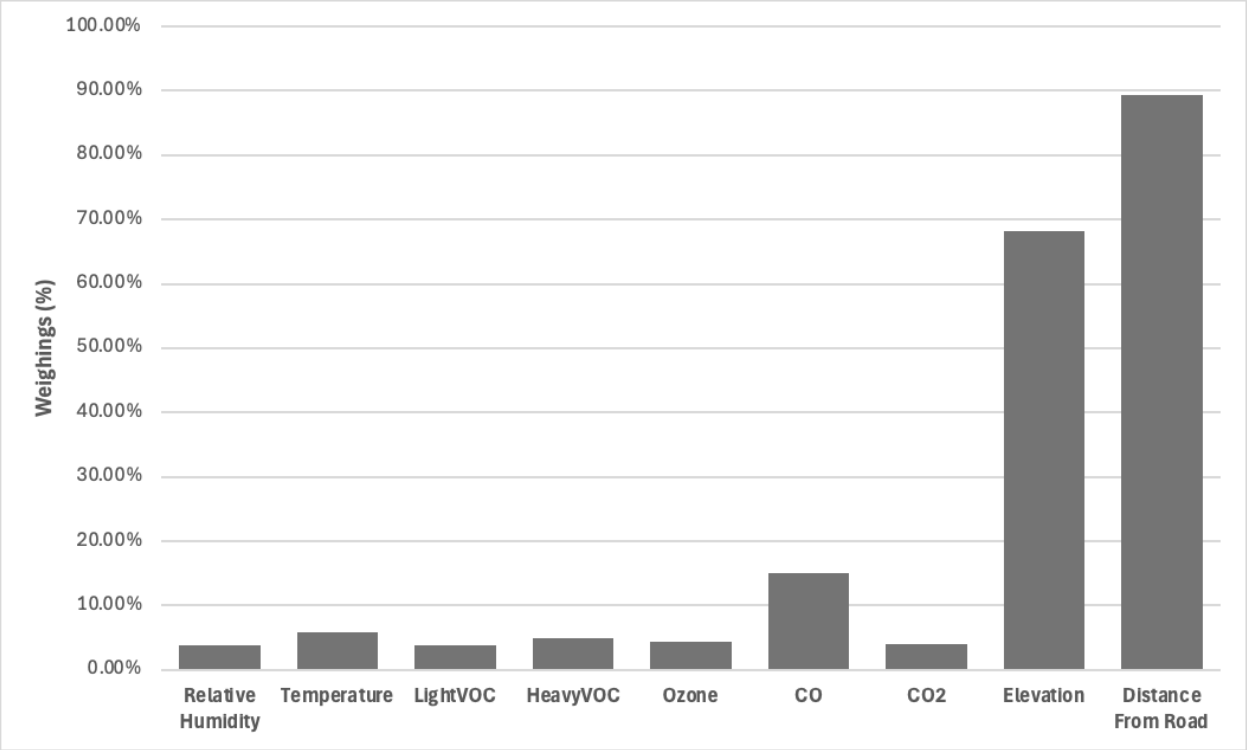

Figure 4: Weighing factors for each parameter in the system

Discussion and Conclusion

Our inquiry yielded results consistent with intuition surrounding the system. The objective of this process is to numerically quantify the interactions between variables in a system to better understand and predict concentrations of pollutants based on system parameters such as elevation, distance from a highway, temperature and relative humidity. Through least squares regression, we were able to construct an interaction matrix representative of the system. Within this interaction matrix we can either look at how a single variable impacts other system parameters or how the system behaves as a whole. For example, we can conclude that elevation has a greater impact on carbon monoxide concentration than any other variable for the system by looking at the interaction values displayed in row 8 of the matrix. As a system, *Figure 1* clearly displays the larger degree of interaction distance from a highway and elevation have on the system than other system parameters. However, elevation is system-subordinate and distance from road is system-dominant. Intuitively this is consistent with the area surveyed, since the road of interest is in a valley relative to the rest of the area where data was collected, and thus elevation is going to be heavily impacted by distance from the road as shown *in Figure 5*.

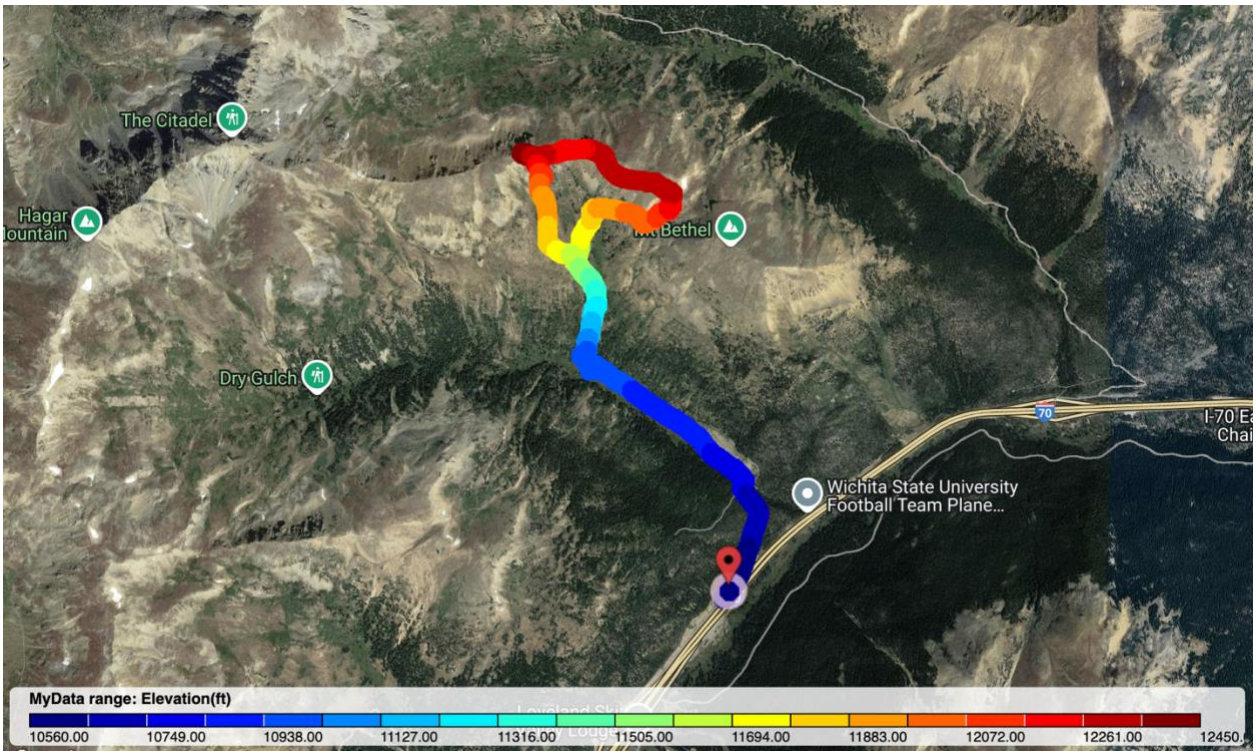

Figure 5: Geospatial plot of elevation for the area surveyed.

Furthermore, distance from road is classified as "dominant", meaning it has a very large influence on all other system variables. Again, this makes intuitive since combustion and tailpipe emissions are the primary source of pollutants in this area and are sourced at the highway. We can also conclude that temperature and relative humidity didn't have much of an impact on pollutant concentration relative to elevation or distance from the highway based on their relatively low weighing factors (*Figure 4)*.

The value of this study, however, can be seen in the differing mode of interaction for each pollutant in the system. Carbon monoxide is the most interactive pollutant in the system and very subordinate (*Figure 3).* It also has the largest weighing factor relative to all other pollutants by a factor of around 3. Consequently, we can infer that carbon monoxide is the pollutant most influenced by the system, and thus by distance from a highway and elevation (*Figure 6*). Since CO is a dangerous pollutant due to its toxicity and odorless nature especially at high elevations, it is critical to be able to predict its concentration in a given area and know how it is impacted by system conditions.

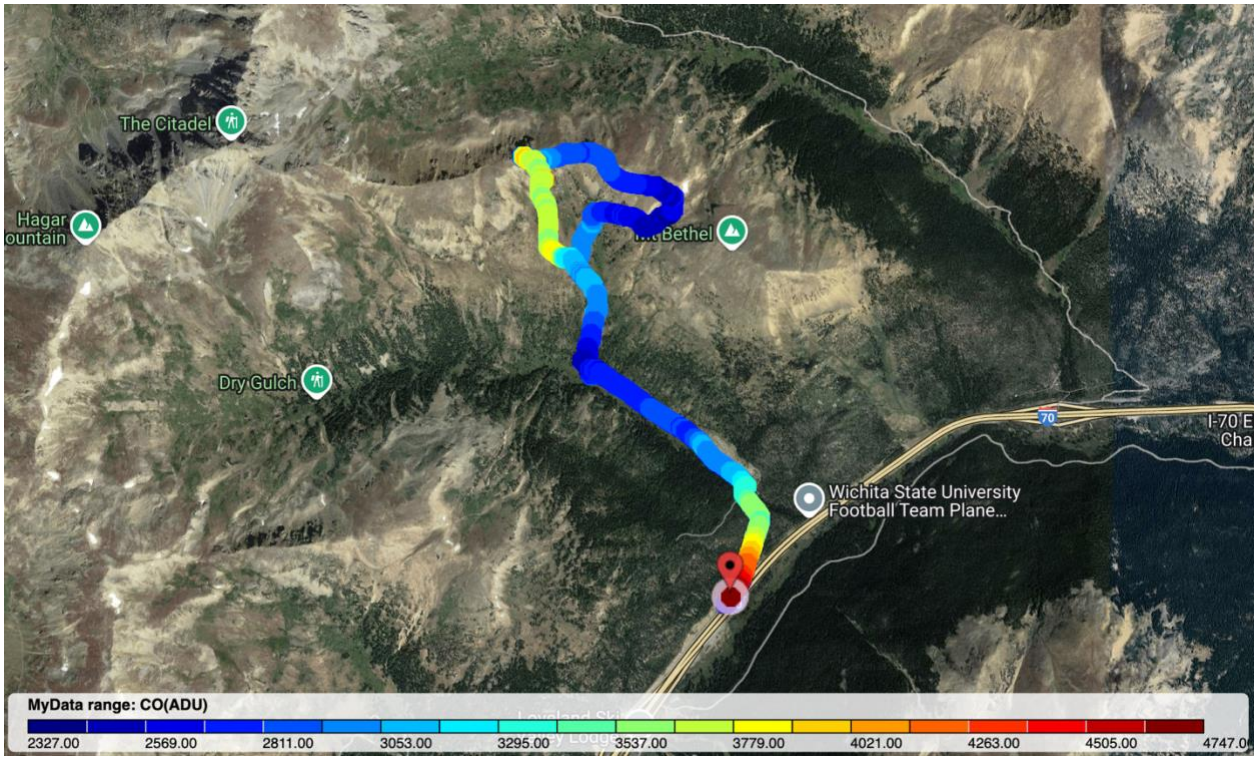

Figure 6: Geospatial plot of CO concentration for the area surveyed.

In the future, the weighing factors and interaction values of this system can be used to create a predictive model of the system as described by Mavroulidou et al. (2004), and help to better predict where carbon monoxide concentrations will reach dangerous levels.

While this model is a decent representation of the system, it is far from perfect. Error is introduced in the assumption that each variable's interaction is explicitly linear. A higher order least-squares regression or alternative level-regressions, such as log regression, could be performed to better approximate the system since no known empirical relationships exist between variables in this system. A weighted least-squares regression can also be computed and may be more representative of the system. There is also error that is introduced in assuming that there exists a value for each interaction. For example, pollutants cannot physically influence distance from the road or elevation, however least squares regression was performed on these variables and the corresponding values of β_1 were taken to be their interaction values. As such, a better representation of the system would include physical intuition on how each variable interacts. QR factorization solutions to alternative linear regression systems should be explored further since QR factorization allows for several simplifications to making solving for LSRL lines far simpler. Another matrix factorization to consider is the SVD and as those matricies

utilize alternative methodologies to express matrix inverses and may also possess interesting simplifications to LSRL problems. As described above, the least squares method typically has floating-point error even when utilizing QR factorization. This depends on the data set and the condition number of Q and R respectively but can nonetheless impact results.

References

Cheney, E. W., and David Kincaid. *Numerical Mathematics and Computing*. 6th ed, Brooks/Cole, 2007.

Maria Mavroulidou; Susan J. Hughes; Eemma E. Hellawell. (2004). “A qualititative tool combining an interaction matrix and a GIS to map vulnerability to traffic induced air pollution”. *Journal of Environmental Management*, Volume 70, Issue 4. p. 283-289. https://doi.org/10.1016/j.jenvman.2003.12.002.

Maria Mabroulidou; Susan J. Hughes; Emma E. Hellawell. (2007). “Developing the interaction atrix technique as a tool assessing the impact of traffic on air quality”. *Journal of Environmental Management*, Volume 84, Issue 4. p. 513-522. https://doi.org/10.1016/j.jenvman.2006.07.002.

Mazzoccola D. F; Hudson J. A. (1996). “A comprehensive method of rock mass characterization for indication natural slope instability”. *Quarterly Journal of Engineering Geology*, Volume 29, Issue 1. The Geological Society of London. doi:10.1144/GSL.QJEGH.1996.029.P1.03

Leopold, Luna Bergere; Clarke, Frank Eldridge; Hanshaw, Bruce B.; Balsley, James R. (1971). “A Procedure for Evaluating Environmental Impact”. *Circular.* Washington, D.C. p. 19. doi:10.3133/cir645 – via U.S. Geological Survey.

Olver, Peter J, and Chehrzad Shakiban. *Applied Linear Algebra*. 2nd ed., Springer International Publishing, 2018.

Watson, Geoffrey S. “Linear Least Squares Regression.” *The Annals of Mathematical Statistics*, vol. 38, no. 6, 1967, pp. 1679–99. *JSTOR*, http://www.jstor.org/stable/2238648.

Appendix

Data set used: https://o365coloradoedu-my.sharepoint.com/:x:/g/personal/chpe5809_colorado_edu/Ea6s5LFXd-BLqcmaAcQ0UIkBUFGun13I2MpWIhNEBjj3Ig?e=xTc8E1

RETIGO used for geospatial viewing: https://www.epa.gov/hesc/real-time-geospatial-data-viewer-retigo

Interaction Matrix Python Code: https://github.com/DTAlemayehu01/APPM-3310-Final-Project