Spatio-temporal Approximation: Interpolation and Extrapolation Queries on Climate Data

Undergraduate Group 8

Kayla Carter (kcarte26@kent.edu)

Nick Flowers (nflower5@kent.edu)

Vanessa Sanders (vsander2@kent.edu)

Jayden Stearns (jstearn2@kent.edu)

1. Introduction

An ever-increasing issue of modern climatology is the changing and degradation of Earth's atmosphere. This issue of the changing global climate poses many challenges for both science and mathematics. Among these challenges is the difficulty of measuring the vast amount of data that can be collected in a wide variety of ways. Climate data often contains a large number of dimensions, especially including spatial (location on Earth's surface) and temporal (time of measurement) elements. For example, a dataset may contain latitude and longitude coordinates of measurement and the date in which the measurement was recorded. Additional attributes of climate data may include: time of day (hour, minute, and/or second), city, continent/region, average temperature, normals, precipitation levels, atmospheric pollutant concentration, radiation levels, etc. In addition to the large variety of possible dimensions, many geologists/climatologists are limited to small scale, controlled experiments across the globe. However, the task of making these findings scalable and useful beyond their local scope is necessary for the shift in focus for the study of climatology, which is now treated as a predictive science on a more global scale. Creating an accurate model of global climate requires a vast amount of measurement technology placed all around the world. This technology is important for our understanding of climate trends over time and in different locations, but, as with all precision measurement technologies, requires potentially-prohibitive upfront cost, technologies, and upkeep.

Methods of measurement that could employ simpler, more ubiquitous technology, like computational calculations based on statistical models, could help this important field of study reduce both its complexity and cost. Specifically, estimation queries on these big datasets could be used to supplement the amount of (or distance between) associated measurement devices. Additionally, as is very useful in climate and weather information, model-based statistical estimation queries could provide alternate methods of predicting future trends. These predictions become extremely vital in the real world whenever we are analyzing issues such as the long term trends of air quality, for example. While we propose this solution with climate measurement and monitoring in mind, realistically, any spatio-temporal data set that is similarly-formed to climate data could benefit from the data approximation queries that we aim to implement.

2. Project Description

In this project, designed and implemented accurate and efficient estimation queries on a spatio-temporal climate data set. We aim to support the existing efforts of world-wide climate measurement by providing statistically-backed queries which can be used to approximate values using the surrounding spatio-temporal information. Our intention is to develop this set of queries as a new method to quantitatively analyze data using the statistical techniques of interpolation and extrapolation. One of the major challenges that exists for projects of this nature is data availability. Our group does not have the means to collect measurement data of the large scale necessary for this project, so we have gathered data from a collection of climatological surveys. Specifically, we are using a large dataset of over 300 cities around the world, measuring average daily temperature over the course of over 6 years. This dataset was provided by the University of Dayton, sourced from the dataset publication platform Kaggle. We parsed and sorted the original dataset, composed of the following dimensions. Each city of measurement was converted to its global latitude and longitude coordinates. The month, day, and year fields were converted to a more quantitatively-useful measurement: the number of days after the very start of the study's data, January 1, 1995 (this allows for more simple and intuitive correlation calculations on this variable of the data). Lastly, the measurement value of the average temperature at the given city on the given day, represented in degrees Fahrenheit with a significance to one decimal place.

More specifically, we have designed and implemented a small set of estimation queries, designed to support an existing dataset of spatio-temporal data. The existing dataset consists of measurement values across a variety of locations and times. These estimation queries can be used to approximate the value of the measured attribute at a time and/or location that is not explicitly measured within the dataset. To accomplish this, the statistical strategies of interpolation and extrapolation are used. Interpolation is used when approximating a value for which there exist other values surrounding it on multiple sides. For example, interpolation is used when estimating the temperature value in a city on a day for which there is not a known value; the surrounding values (earlier and later daily temperature measurements) will be used to approximate the unknown value. Extrapolation, in contrast, is used when approximating an unknown value that is outside of the range of known values. An example of extrapolation is approximating average rainfall in a region for a date in the future. Only the known (past) rainfall measurements can be used to approximate the value outside of the range of known values using extrapolation techniques.

Each member of the team was responsible for initial research on big climate data, including example attributes as described above. Individual tasks include implementing the functionality to parse through and sort the source dataset into a more usable form for our quantitative calculations. Additionally, the research behind the statistical concepts (explained below) that were integral to our proposed query set was conducted as well.

3. Background

The strategy of utilizing statistical techniques to analyze climate data is certainly not new. Extrapolation, in particular, is often used to mathematically approximate future trends. This is especially shown in a research paper titled “Identifying and Characterizing Extrapolation in Multivariate Response Data”, which gave our team insights on querying data that has multiple dimensions such as spatio-temporal data. This paper researched predicting lake nutrient and productivity variables by performing extrapolation on the LAGOS-NE database which is a multi-scaled geospatial and temporal database for thousands of inland lakes [1].

In order to perform these statistical analyses on our dataset, our team utilized the statistical tools of the programming language R. This suite of computational tools allows us to not only perform calculations upon our large dataset, but also to visually model the regression relationships that we create upon the data. In addition to this, C++ was used to parse through and sort the raw source data into more meaningful and quantitatively-useful forms. For example, the source data was formatted temporally with month, day, and year fields, and each individual location had an individual line entry for each measurement. The temporal dimension for each entry was converted using a C++ parsing program into a single integer representing the number of days elapsed from the start of the study (January 1, 1995). Then, this parsed data was run through a sorting program to restructure the data. Each location (regardless of the number of individual measurements,) only has a single line entry, followed by a sequence of (time, temperature) pairs for each date of measurement. Finally, this data is then further split into the individual time and temperature components by the sorting program.

Due to the intentionally-ubiquitous nature of the technology used, these algorithms and queries can be run on essentially any personal computer with reasonable computational power. This was a foundation of our design, as our primary goal was to support the world-wide science and research of climatology. The use of C++ and R (based partially in C) was chosen for this very purpose. Essentially all modern operating systems of any level of power are capable of compiling and running these languages without many additional dependencies. Naturally, due to the size of these large data sets, appropriate computation time must be given proportional to the size of the climate dataset used. Programming skills exemplified in these various implementations includes the following: functional programming, object-oriented programming, abstract data type utilization, searching algorithms, etc.

4. Problem Definition

Climate data forms an interesting subset of spatio-temporal data: one in which all data points are inter-related in some way. More generally speaking, if all climate information were possible to be measured across all points on Earth, the resulting field of data points would be differentiable (and therefore continuous) at each point in the field. There does not exist any data value within the dataset that has no possible correlation to the other points, both temporally and spatially. To provide a simple example, consider it was 32°F (0°C) at one measured location and 41°F (5°C) at another. These two data points obviously do not imply that there is some sudden boundary line between these two points where the temperature instantly changes between these two values; rather, as you traveled between these locations, the temperature would increase and/or decrease continuously and gradually as you approach the destination location. This, of course, is because temperature at any location depends largely upon the temperature of its surroundings. Essentially all measurable characteristics of climate information align with this fact as well, forming a spatio-temporal data set that is completely interconnected, as opposed to purely individual data points that could not be meaningfully related together.

The fact that climate data can be interconnected in this way allows for more unique predictive calculations to model values at points that are not explicitly measured (as explained below). Using statistical models to accurately approximate spatio-temporal information between two known, measured locations can reduce the required number of measurement sensors required to accurately measure a region. By supplementing the use of additional measurement devices, fewer measurement devices could be placed around the area of study, for example, which could help to reduce the cost and difficulty of climate data measurement.

The mathematical techniques that we intend to utilize include Interpolation and Extrapolation. Interpolation is the mathematical process of approximating a value that falls between other known values on one or more dimensions. An example of interpolation may include the following. An ecologist has placed one temperature sensor per nine square mile region of a large ecosystem they are monitoring. Later, they want to use the data they gathered to approximate the temperature of a small pond that is not directly nearby to any of their sensors. A simplistic approach to solving this issue could be to simply use the corresponding temperature value of the nearest sensor to the pond's location. However, using more advanced statistical interpolation techniques, the temperature at this pond's location could be more accurately approximated by performing a calculation using the temperature values of a variety of nearby temperature sensors. This was one primary goal of our project design and implementation.

The other mathematical technique, Extrapolation, serves a different purpose. While interpolation uses surrounding data points to approximate an unknown point, extrapolation makes an approximation using only data points on one side of the dimension in question. For this reason, Extrapolation techniques are often employed for predicting future trends. For example, if an ecologist wanted to predict the temperature of a region 2 years after the last measurement value, they would need to make an approximation of a data point that falls outside of the range of time for which there are known, measured values. Though there are only data points from the past that could be used for approximation of future data, statistical approaches can be used to accurately predict these values outside of the temporal range of the measurements of the dataset.

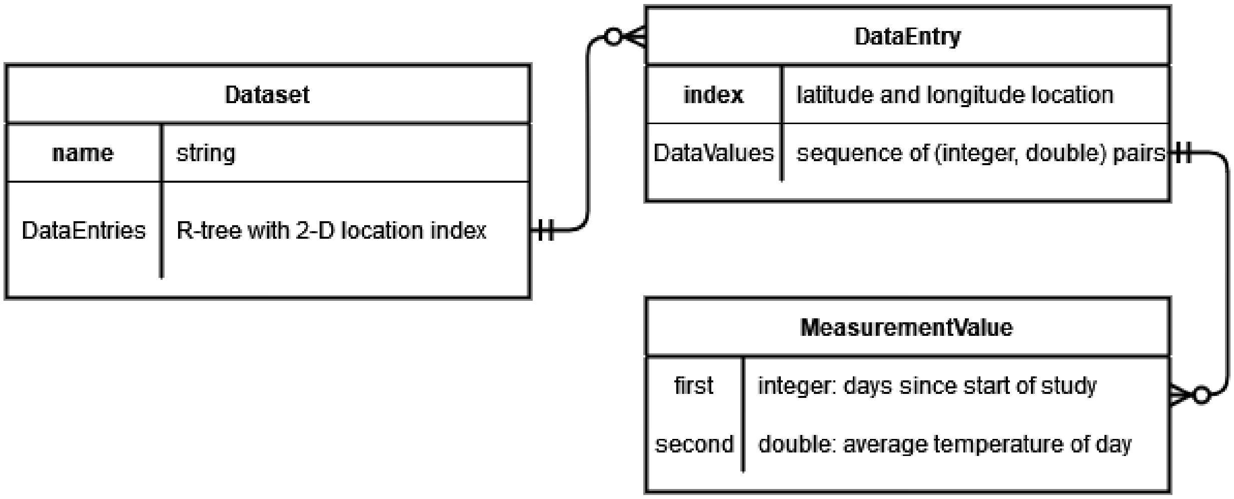
Additionally, Extrapolation can also find use in the spatial dimensions. If temperature data is desired from a region slightly outside of the boundary of our ecosystem measured, Extrapolation techniques could be used to approximate these values as well, as they fall outside of the bounds of the spatial dimensions within the dataset.

Naturally, these query approaches also introduce anticipated challenges as well. Due to the previously-mentioned interconnected nature of climate data, approximation techniques can become more accurate. However, with this increased accuracy comes complexity. For example, an overly simple approach to Interpolation could be to simply average all of the nearby surrounding temperature measurements together to produce an approximate result. However, this does not account for many factors, such as non-linear temperature trends across the landscape, or considering more highly the values of measurements closest to the desired query point. Our set of queries aims to combat these challenges, as detailed below. In addition to the need for accuracy of the regression model, another challenge that stems from these methods is the very nature of approximation itself. Because the queries that we are proposing are not simply correct or incorrect, additional care was taken in order to verify the correctness and margin-of-error of the algorithms that we have implemented. This requires an understanding of the statistical models used in this context. While the approximated results of the queries may not have correct or incorrect results, the results could still fall outside of a reasonable margin-of-error, thus resulting in an approximated result value that is not very useful.

Overall, we aimed to produce a meaningful consideration in the subject of approximating spatio-temporal data that is interconnected, especially including large climate datasets. Our goal was to produce a set of queries that can reasonably approximate unknown measurement values (on both temporal and spatial dimensions) from other values within the dataset. Whether a queried data point is within the dimensional boundaries of the data set (Interpolation), or the queried data point is at a time not yet measured (Extrapolation), we aim to properly utilize statistical models and approximation techniques to accomplish these goals. Above all else, we hoped to produce a set of queries that could meaningfully assist climatologists and other researchers in their efforts to better interpret global climate trends.

5. Proposed Techniques

To begin with a general overview of the structure of the spatio-temporal dataset as a whole, consider the following simplified Entity-Relationship Diagram:


Dataset
name
string
DataEntries
R-tree with 2-D location index
DataEntry
index
latitude and longitude location
DataValues
sequence of (integer, double) pairs
MeasurementValue
first
integer: days since start of study
second
double: average temperature of day

This spatio-temporal dataset can be represented as a collection of data entries sorted using a data structure in the R-tree family of data structures. This is very appropriate for this particular implementation, as the number of dimensions on which the R-tree is indexed is very small, consisting only of the latitude and longitude location of the individual data entry. It is upon a spatio-temporal dataset such as with this structure that we have designed and implemented our algorithms, described below.

The data entries themselves represent individual locations within the dataset. In our particular instance, these individual data entries are the locations of cities which have been measured in the source data. The data entry is indexed according to its location, and contains the attribute of a sequence of pairs. Each of these pairs is an individual measurement value.

The measurement values are the individual measurement readings at that location within the dataset. Again, in our particular example, this consists of the number of days elapsed since the beginning date of the study and the measured value of the average temperature for that particular day, at the location in which this measurement value is aggregated.

The following is an overview of the query set that we have designed and implemented for use with a dataset formed in such a way as described above, shown here in a more universal algorithmic notation. As previously explained, our particular algorithms were developed using a combination of C++ and R. However, as was the goal of our project, our algorithms are designed to be as general-purpose and ubiquitous as possible. This leads us to the query set that we have informally referred to as Regression Value Approximation.

```
# overarching function invocation handles dataset searching and querying
Algorithm   RegressionValueApproximation ( t , l , D ) :
            Input:  Desired time t and location l in spatio-temporal dataset D
            Output:  The estimated measurement value at time t and location l

    nearestNeighbors ← empty set of DataEntries within D
    minNumNeighbors ← min ( 3 , D.numEntries() )
    numNeighbors ← max ( minNumNeighbors , $\log_2$ (D.numEntries()) )
    nearestNeighbors ← kNearestNeighbor ( l , numNeighbors )
    # perform k-nearest-neighbor query with reasonable argument bounds,
    # proportional to the relative logarithmic size of the dataset as a whole

    return  RVA_WeightedAverage ( t , l , nearestNeighbors )
```

```
# this subroutine handles arithmetic operations on data entry regression models
Algorithm   RVA_WeightedAverage ( t , l , S ) :
            Input:  Desired time t and location l relative to set of DataEntries S
            Output:  The estimated measurement value at time t and location l

    regressionValues ← empty set of ( value , distance ) pairs

    for each  entry  in  S  do
        value ← regressionTrig ( entry , t )
        distance ← RVA_Distance ( t , l , entry )
```

```
            regressionValues.insert ( value ,  distance )
    sum ← 0
    totalDistance ← 0
    **for each** estimate **in** regressionValues **do**
            sum ← sum + ( estimate.value *  1 / estimate.distance )
            totalDistance ← totalDistance + (1 / estimate.distance)

    **return** sum / totalDistance
```

```
# this subroutine computes RVA distance from a data entry to a time & location
Algorithm  RVA_Distance ( t ,  l ,  d ) :
            Input:  Time t, location l, and DataEntry d
            Output:  RVA distance from d to time t and location l

    spatialDistance ← EuclideanDistance ( l ,  d.location )
    nearestDayNum ← nearestValueTo ( t ,  d.DataValues )
    temporalDistance ←  absVal( nearestDayNum - t )

    return  max ( spatialDistance + temporalDistance ,  1 )
```

To briefly overview the operation of these algorithms, the querying begins by providing a desired time and location to estimate from a given dataset. A k-nearest-neighbor query is performed on this dataset to obtain a set of the physically nearest locations within the dataset to the provided location. Then, the weighted average of these locations' measurement values is calculated and returned.

The weighted average subroutine computes a trigonometric regression for each data entry according to the trends in each data entry over time. This regression is used to provide an estimate for the value at that location in the dataset at the queried time. For each entry, the RVA distance to the desired location is calculated. First, the Euclidean distance between the given location and the data entry's location is calculated as expected, precisely like any Euclidean distance in metric space. Then, the temporal distance is calculated. This is calculated by determining the smallest distance in time from the desired queried time within the measurement values of the data entry. These values are added together, and returned. The smaller the RVA distance, the closer spatially and/or temporally two data points are located together. The minimum RVA distance is 1, signifying that there does exist a known, measured value for the time and location within the dataset.

Execution then resumes in the weighted average subroutine. The weighted average is computed using the value at each location within the set of nearest neighbors, weighted by the inverse of their RVA distance to the desired queried point. The average is then calculated by dividing the running summation of all neighbors' values by the total inverse RVA distance and returned from the queries. The motivation and explicit use cases for this set of algorithms are described below.

6. Visual Demonstration

As noted previously, our team utilized the R programming language to perform statistical analysis on our dataset. R allowed us to easily plot data points that represented the daily average temperature for a specific city over our time range. Figure 1.0 is one example of this visualization that R provided through its ggplot library. Figure 1.0 visualizes the Daily Average Temperature in Akron-Canton for approximately 2000 days.

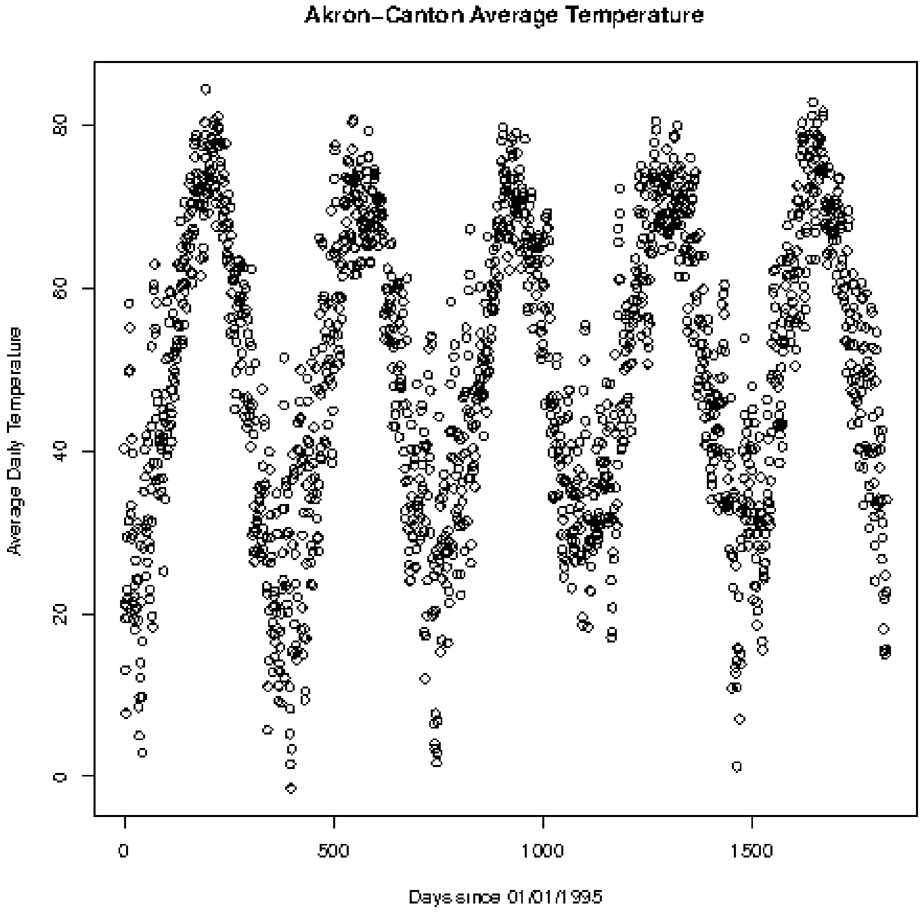

Akron-Canton Average Temperature
Average Daily Temperature
0
20
40
60
80
0
500
1000
1500
Days since 01/01/1995

Figure 1.0

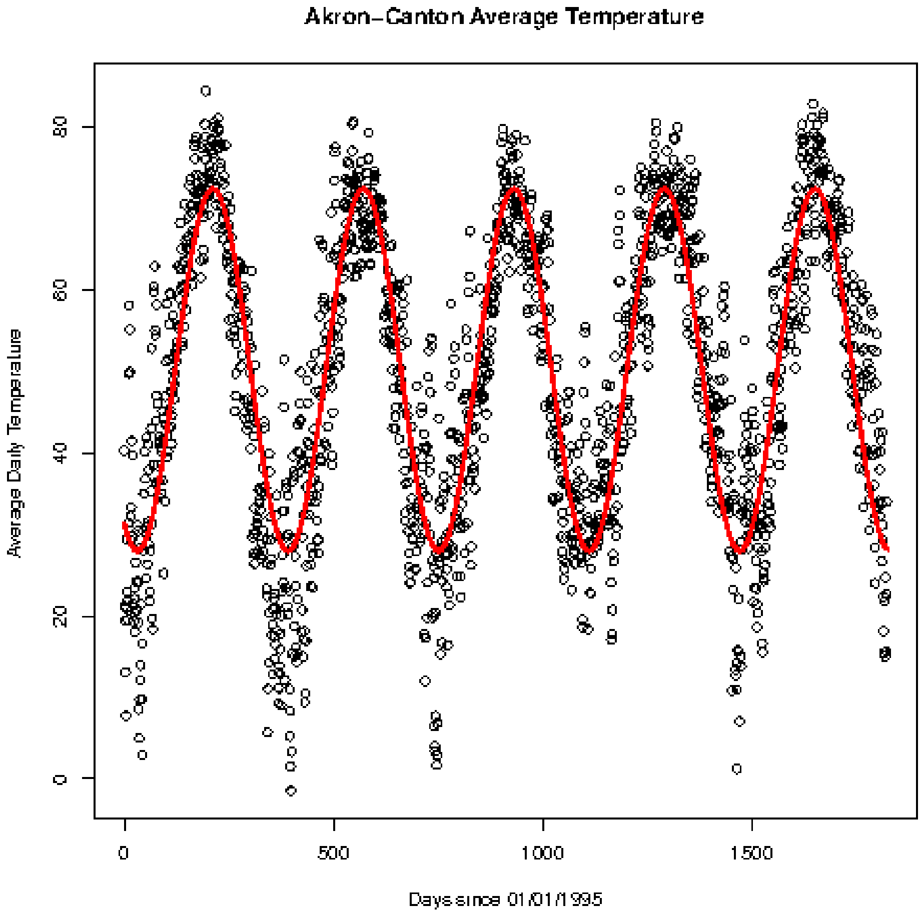


Figure 1.1

Figure 1.1 further displays the visualization provided by R in its ability to display the trigonometric regression line that is fit to our temperature data for the Akron-Canton area.

7. Experimental Evaluation

The dataset that we utilized had approximately 2.9 million data entries for approximately 326 unique locations across the globe. The original columns of the dataset utilized were Region(Continent), Country, City, Month, Day, Year, and Avg Temperature. The average temperature was recorded in degrees Fahrenheit. In order to perform our analysis on the data we modified the Month, Day, Year into days since the starting date of the study. We also turned the Cities into their latitude and longitude coordinates in order to have representations of the points.

Methods of interpolation on spatio-temporal data typically fall into two approaches: reduction approach and extension approach. The reduction approach, which is the approach that our technique would be categorized into, is where the interpolation cloud is calculated based on the results at each location. This treats time as an independent variable. Another example of interpolation that falls into this category is an IDW tension spline. The extension approach is known to be more difficult to implement and in many cases is known to not be cost effective. This approach is when the temporal part of the data is dealt with as another dimension in space. One example of this approach is the Kriging algorithm. Overall we found that our method would be relatively cost effective and also provide simplicity for the sake of our cause in closing the information gap in the climate data.

Our algorithms were designed with the intention that they would not be executed very regularly. As is the nature of this climatological research, data values are recorded very regularly. In our case, hundreds of new measurements were added every day. However, estimation upon these values is a less frequent and context-specific requirement. It is for that reason that we have internally treated the dataset as an R-tree, as the small number of indexed dimensions (only the latitude and longitude) leans into the strengths of that data structure. Additionally, the k-nearest-neighbor query type, employed in the overarching function invocation defined above, is conducive to this balance of data insert versus recall. The classification of this data in this way is deferred until execution of the k-nearest-neighbor query itself [2], not notably slowing the insertion or removal of measurement data from the larger dataset as a whole. The majority of the operations are performed largely in constant time, including the RVA distance calculation, assuming that the Euclidean distance and ordered sequence searching algorithms are implemented appropriately. For our dataset of nearly 3 million entries, this approximation can occur with notably impressive speed, including the calculation of the regression model for the data and the weighted average calculation.

8. Future Work

This project has presented a method by which to utilize statistical regression models for each location in a spatio-temporal dataset in order to approximate values for locations with non-existing data. Our experiment particularly showcases interpolation results on global climate data. A future aim for this research should involve further study in the selection of records that are utilized to make our regression models, as this could enhance the efficiency of the algorithms employed. For instance, certain cities may have significant or insignificant outliers based on natural disaster events for a certain period etc. In the future, a more researched selection of data could prove more beneficial for a specific approximation depending on the goal of the experiment.

Another extension of this project could be a meta-analysis showcasing the results of our method performing an extrapolation on our data and comparing its success against our interpolation results. Other useful extensions to our project that would allow us to have a better understanding of our margin of error involve applying our methods to other spatio-temporal datasets with variables of different units in order to understand its capabilities of approximating values that require high precision. In a similar effort, our experiment should be tested on both relatively sparse and dense data sets to measure its scalability and reliability with different sized datasets.

9. References

[1] Bartley M. L., Hanks E. M., Schliep E. M., Soranno P. A., & Wagner T. (2019). Identifying and characterizing extrapolation in multivariate response data. *PLOS ONE* 14(12): e0225715. https://doi.org/10.1371/journal.pone.0225715

[2] Cover R. M., & Hart, P. E. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory* 13(1). https://isl.stanford.edu/~cover/papers/transIT/0021cove.pdf

[3] Miller, J. R., Turner, M. G., Smithwick, E. A. H., Dent, C. L., & Stanley, E. H. (2004). Spatial extrapolation: The science of predicting ecological patterns and processes. *BioScience*, 54(4), 310-320. https://doi.org/10.1641/0006-3568(2004)054[0310:SETSOP]2.0.CO;2

[4] University of Dayton. Environmental protection agency average daily temperature archive. https://academic.udayton.edu/kissock/http/Weather/default.htm