Comparing Geospatial Interpolation Methods for Modeling Snow Depths

David Larson

Dr. Silas Bergen

Abstract

Predicting values of unknown points at geographic locations using known points at other locations may seem simple, but many methods offer different solutions to try to predict such values. The goal of this research was to investigate how different geospatial interpolation methods work by comparing their accuracy of predicting snow depths in western Colorado. This data is publicly available and was collected by the Remote Sensing Center at the University of Alabama, who used an FMCW radar system to measure the snow depths. The interpolation methods explored in this project included ordinary kriging, inverse distance weighting, k nearest neighbor, thin plate splines, universal kriging, and co-kriging. Out of all these methods, it was shown that inverse distance weighting performed the best on this subset of snow depths data.

Introduction

Frequency Modulated Continuous Wave (FMCW) radar systems have made it much easier to measure the thickness of a layer of snow or ice, but only at point measurements. It would be extremely efficient to sample the thickness using the radar in a loose outline or grid and then use spatial interpolation methods that should be able to accurately approximate the depths for the rest of the surface. Many such interpolation methods exist, but which one should we use? Which one performs best for this type of data?

The data used for this project was gathered by the University of Alabama, and contains three million snow depth measurements (in meters) with attached latitude and longitude coordinates. The goal of this study was to use this snow depth data to compare different geospatial interpolation methods to see which performs the best.

Methods

Data Acquisition

The data was gathered by the Remote Sensing Center of the University of Alabama, where they used an FMCW radar mounted into an airplane that flew in designated flight paths in western Colorado (see Figure 1). Not all the flight paths were used to gather data, however, as the plot of points in Figure 3, explained later, shows the general shape of all the points given in the database.

Figure 1. Map of Flight Paths of Data Collection

Each file corresponded to the depth measurements along each flight line, with each row of the data being a single point location with a depth measurement of the snow and location coordinates. A typical flight path had about sixty thousand observations. All the points were arranged in a straight line because the radar system only measured the depth directly beneath the craft. Data collection happened on two different days, once on March 20, 2019, and the other on March 26, 2019.

Most of the methods only used the data mentioned above, but co-kriging used other variables that were retrieved through the United States Geological Survey. These variables included elevation, slope, aspect, and terrain ruggedness index along with longitude and latitude coordinates for all places across the United States (United States Geological Survey, 2022).

Using the USGS data, these four values were found for every point that had a measured snow depth value using raster object extraction (raster package in R).

Data Manipulation

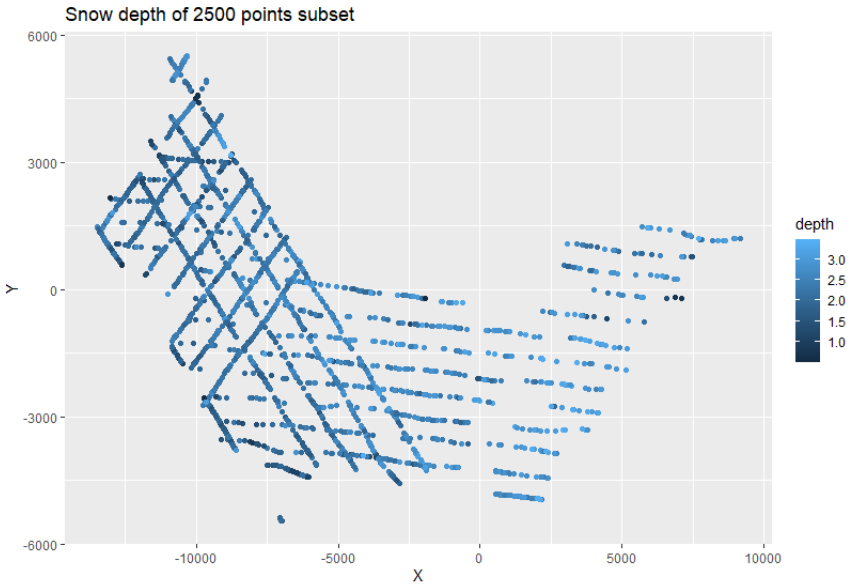
The data was originally formatted as a .mat file, so some data transformation in python (numpy, scipy, & pandas packages) was done to convert the files to a .csv format to be used for the duration of the study. Once the data was converted to a readable format, a small subset of about 2500 points with random points from all the flight paths from March 20 was used for the remainder of the work. Since each observation used longitude and latitude for their position, the values had to be converted to an (x, y) coordinate pair to be able to calculate distance. Using the Lambert Conformal Conic (LCC) projection, a function was written (shown below) to convert the location data to x and y coordinates.

Figure 2: Conversion to LCC function

```
LongLatToLCC<-function(x, y){
  require(sp)
  # converts latitude and longitude to x and y coordinates in the LCC projection system
  #           with its standard parallel = 39.04, center of longitude = -108.4, center of latitude = 39.04
  #           (This is in Western Colorado)
  # x - vector of x (longitude) coordinates to be converted
  # y - vector of y (latitude) coordinates to be converted
  # x and y must be same length
  # returns: data frame with two columns: x and y coordinates

  xy <- data.frame(ID = 1:length(x), X = x, Y = y)
  coordinates(xy) <- c("X", "Y")
  proj4string(xy) <- CRS("+proj=longlat")
  result <- spTransform(xy, CRS("+proj=lcc +lon_0=-108.1 +lat_1=39.04 +lat_0=39.04"))
  result.df <- as.data.frame(result)[,c(2,3)]
  return(result.df)
}
```

Figure 3: Plot of all points



Interpolation Methods: Kriging

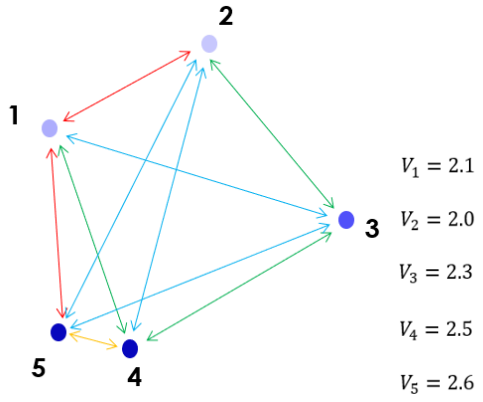
Once the data was in a usable format, it was time to begin interpolating. The first method that was explored was Kriging. Kriging is geostatistical, meaning that it investigates the relationship between the location of points and their values before deciding how much to weight known points for a given distance away when trying to predict at a location (ArcGIS Pro, 2022). The foundation of kriging is the semivariogram, which explains the change in semivariance of a spatial surface with respect to distance. Semivariance is calculated using all pairs of points within a surface. Semivariance, for a given distance h, is calculated as:

$$Semivariance(distance\ h) = \frac{1}{2} * average[(value_i - value_j)^2]$$

for all point pairs of (i, j) within a distance h of each other. In R, the semivariance is calculated for all pairs of points in a matrix, and then put into bins (bin size depends on the scale of the data: a surface that covers a large area has large distance values between some points, therefore wider distance bins generally) based off of the distance between the two points.. All the

semivariances within each distance bin are averaged and this results in the values shown on the semivariogram. Each dot represents the average semivariance for all points for the average distance of that bin apart. Shown below is a sample semivariogram. The yellow dot is the average semivariance for points that are a distance (0-5] units apart. The red dot is the average semivariance for all points that are a distance (5-10] units apart, and so on.

Figure 4: Example Surface of 5 points with depth values; Point Pair Table of all point pairs and their squared differences; and semivariogram of given surface



$V_1 = 2.1$

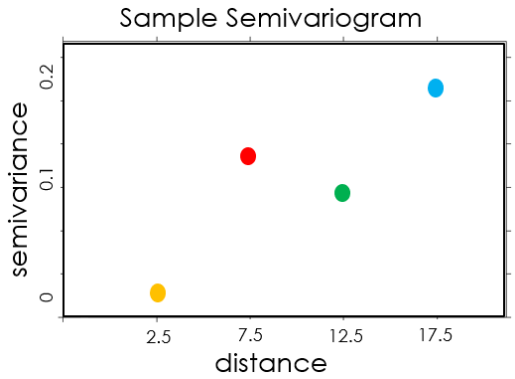
$V_2 = 2.0$

$V_3 = 2.3$

$V_4 = 2.5$

$V_5 = 2.6$

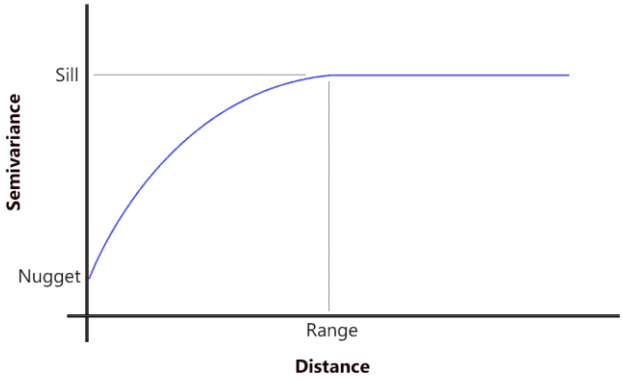
Point Pair	Distance	$(value_1 - value_2)^2$
(4,5)	3	$(2.5 - 2.6)^2 = .01$
(1,2)	9	$(2.1 - 2.0)^2 = .01$
(1,5)	10	$(2.1 - 2.6)^2 = .25$
(1,4)	12	$(2.1 - 2.5)^2 = .16$
(2,3)	12	$(2.0 - 2.3)^2 = .09$
(3,4)	15	$(2.3 - 2.5)^2 = .04$
(2,4)	18	$(2.0 - 2.5)^2 = .25$
(3,5)	18	$(2.3 - 2.6)^2 = .09$
(2,5)	20	$(2.0 - 2.6)^2 = .36$
(1,3)	20	$(2.1 - 2.3)^2 = .04$



The basic idea of a semivariogram visualizes and models the spatial autocorrelation of a surface. The general principle of spatial autocorrelation states that points that are close together geographically should have more similar values than ones that are farther apart (Pebesma, 2022). Once a semivariogram is built from a spatial surface, a model is fit to the points to give the

kriging model a way to measure how much weight points should be given for a given distance away from a point being interpolated. Each model has three parameters: nugget, sill, and range. The nugget is essentially the y-intercept, or the semivariance value of points that are nearly zero distance away. The sill is the semivariance value at which points that are beyond a certain distance away are independent of one another. Partial sill, which is often referred to instead of the full sill, is the sill minus the nugget. The range is the distance at which points are independent of one another based off their spatial autocorrelation. The visualization of these parameters is shown below.

Figure 5: Semivariogram Model Parameters: nugget, sill, range



This semivariogram model is optimized to just be the line of best fit given the specified models. Some of the types include Exponential, Matern, and Spherical (ArcGIS Pro, 2022), and the difference being that they are each designed to fit different patterns of curve shape and smoothness. Formulas for these different models are as follows:

Exponential: $\gamma(h) = nugget + (partial\ sill) * (1 - \exp(-\frac{h}{range}))$

Matern: $\gamma(h) = nugget + (partial\ sill) * \left[1 - \frac{1}{2^{\nu-1} * \Gamma(\nu)} * \left(\frac{h}{range}\right) * \mathrm{K}_{\nu}\left(\frac{h}{range}\right)\right]$

where $(Bessel\ function)\ \mathrm{K}_{\nu}(t) = \frac{\Gamma(\mathrm{range})}{2} * \left(\frac{t}{2}\right)^{-\nu}$

and $\Gamma(v) = (v-1)!$

Spherical: $\gamma(h) = \begin{Bmatrix} nugget + (partial\ sill) * (\frac{3}{2}\left(\frac{h}{range}\right) - \frac{1}{2}\left(\frac{h}{range}\right)^3) & 0 < h < range \\ sill & h > range \end{Bmatrix}$

For the snow data used in this project, the subset was split into a training set (75%) and a test set (25%) for cross validating purposes. Then, the semivariogram (Figure 4) of the training set was built using all pairs of points as explained earlier and then several semivariogram models were fit to the semivariogram. Matern resulted in the best predictive accuracy metrics (R^2), which indicates how much of the variance of a dependent variable is explained by the independent variable(s). R^2 is calculated as

$$R^2 = 1 - \frac{Unexplained\ Variance}{Total\ Variance}$$

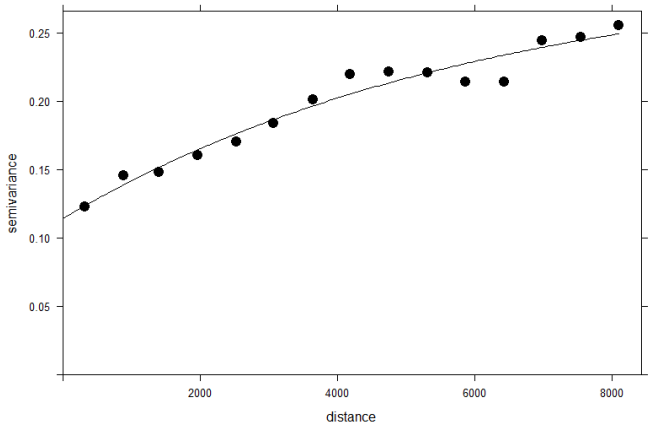
where $Unexplained\ variance = \sum_{i=1}^{n}(\hat{z}(x_i) - z(x_i))^2$

and $Total\ variance = \sum_{i=1}^{n}(z(x_i) - \bar{z}(x))^2$

where $n =$ number of observations, $\hat{z}(x_i) =$ fitted value of observation x_i, $z(x_i) =$ actual value at point x_i, and $\bar{z}(x) =$ mean of actual values. This R^2 value is calculated the same for all following models in this study and is used to show a model's effectiveness.

Returning to the kriging models, the parameters for the Matern model (shown below) were: nugget = 0.123, sill = 0.311, range = 8701.

Figure 6: Semivariogram of Snow Depth Training Set With Model Fit (Matern)



Once the optimal semivariogram model was generated, all that was left to do was to krige at the test locations using the model. The formula for an interpolated point using kriging is

$$\hat{z}(x_0) = \sum_{i=1}^{n} \gamma_i * z(x_i)$$

Where $n =$ number of points that are within the range away from x_0, $\gamma_i =$ is the weight given by the semivariogram for the point x_i based upon its distance from x_0, and $z(x_i)$ is the observed value at location x_i. The kriging predictions were compared with the test location's actual values, yielding an accuracy metric of how well the model was able to predict at those locations.

K Nearest Neighbor

The next examined method in this study was K Nearest Neighbor, which is a deterministic method. In a K Nearest Neighbor model, a certain k is chosen as a parameter and the model interpolates an unknown point as the average value of its k closest neighbors (R Spatial, 2022). For this investigation, a k value of 2 was used, as well as the same training and test sets from the kriging implementation. The formula for nearest neighbor interpolation uses Euclidean distances (D_E) represented as

$D_E(A,B) = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$ where some locations A, B exist such that

$$A = (x, y, \dots) \text{ and } B = (x, y, \dots)$$

Once all Euclidean distances are calculated between the interpolated point and all known points, the two points (because k = 2) that have the shortest distance are used to interpolate the value ($\hat{z}$) at the unobserved point x_0.

$$\hat{z}(x_0) = \frac{\sum_{i=1}^{k} z(x_i)}{k}$$

where $z(x)$ is a list of k number of closest neighbors of interpolated point x_0. For k = 2 used in this study, the following formula was used:

$$\hat{z}(x_0) = \frac{\sum_{i=1}^{2} z(x_2)}{2}$$

(Odgers, 2017) The test locations were predicted by using the training set's locations and values, and these predicted points are compared with the actual values at test locations to measure accuracy.

Inverse Distance Weighting

The third method explored in this study was Inverse Distance Weighting (IDW), which works similarly to K Nearest Neighbor. The difference between the methods is that IDW implements a weight system where points that are closer to the unknown point are given more weight than points farther away (ArcGIS Pro, 2022). The reason for this is because of spatial autocorrelation, meaning that we expect points that are small distances apart should have more similar values than the values of points that are large distances away. There are additional parameters available in IDW that can be adjusted as well, such as implementing a distance radius

where all points within a certain distance away are used to interpolate at a point instead of using a certain number of points. At a point x_0, IDW interpolates its value as

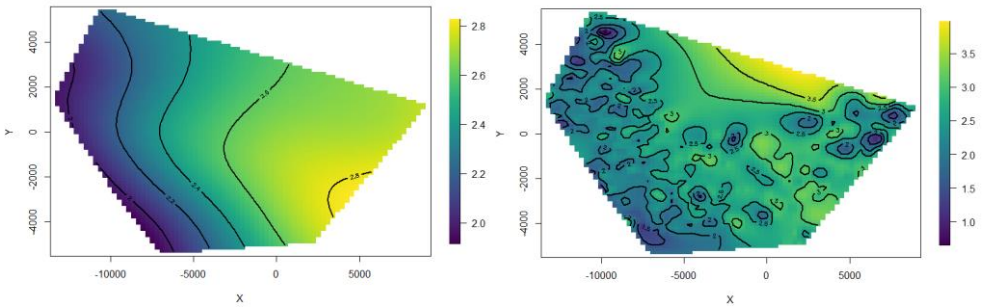
$$\hat{z}(x_0) = \frac{\sum_{i=1}^{n} \frac{z(x_i)}{d(x_i)^p}}{\sum_{i=1}^{n} \frac{1}{d(x_i)^p}}$$

where $n = (number\ of\ neighbors\ used)$, $p = (power\ setting)$, $z_i =$ actual value of point x_i, and $d(x_i) =$ distance between x_i and x_0 (GISGeography 2022). Typical power settings are small (1 or 2) and higher power values give less weight to points farther away in general. For its use in this study, the IDW function allowed for the use of an unknown points' 30 nearest neighbors to interpolate its value and used a power setting of 1.

Thin Plate Spline

A thin plate spline (TPS) is a bivariate fit of two continuous variables on a response variable by plotting each individual predictor on their own x-axis with the outcome being plotted on the y-axis, creating a three-dimensional surface. The smoothing of this surface is done by a generalized additive model (GAM), which uses a line of best fit along both predictors, with an error term as a smoothing function using a least squares method. Increasing the error term increases the smoothness of the spline (Columbia, 2022).

Figure 7: Thin Plate Spline fitted surfaces - left has lambda of 0.5; right has lambda of 0.00001



For the TPS model used in this study, a lambda of 0.00001 was used, as this resulted in the highest R^2 value when predicting to the test set.

Universal Kriging

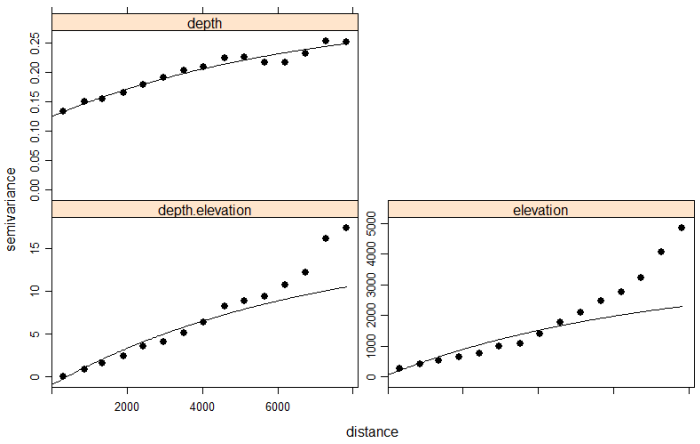
Universal kriging follows the same concepts as normal kriging, but now allows for a trend within the data by their locations. This is done is by fitting a model to the response using their position coordinates as predictors and then the variogram is built using the residuals from this model for each point. Then a variogram model is fit to this residual variogram, and the points are predicted using weights from this residual variogram model fit. The formula for universal kriging is the same as ordinary kriging, but keep in mind that the weights are now being generated using the residuals from the fitted model on the response.

Co-Kriging

Co-kriging is another form of kriging, now allowing the use of other variables in addition to location. Using the extra variables from the USGS, the variable that has the highest correlation coefficient with the response is used as an additional predictor. For the data in this study, only elevation had a significant correlation with depth (correlation = 0.48). Then, using the same training set, a semivariogram of the autocorrelation of elevation by the distance of the points is generated as well as a semivariogram of the autocorrelation of the depth. A model is fit to each of these semivariograms, then the semivariograms are crossed into one. Then the same semivariogram model for depth is put onto the crossed semivariogram, and then the partial sill and nugget are adjusted using least squares to get the best fit to this crossed semivariogram

(Ahmed, 2022). A plot of each semivariogram and the final model is shown below. This final model is what is used to generate weights for the co-kriging model.

Figure 8: Semivariograms for depth, elevation, and their cross variogram



The formula for co-kriging is the same as ordinary kriging but now the weights are generated using the cross variogram of both distance and the added variable used in co-kriging.

Results

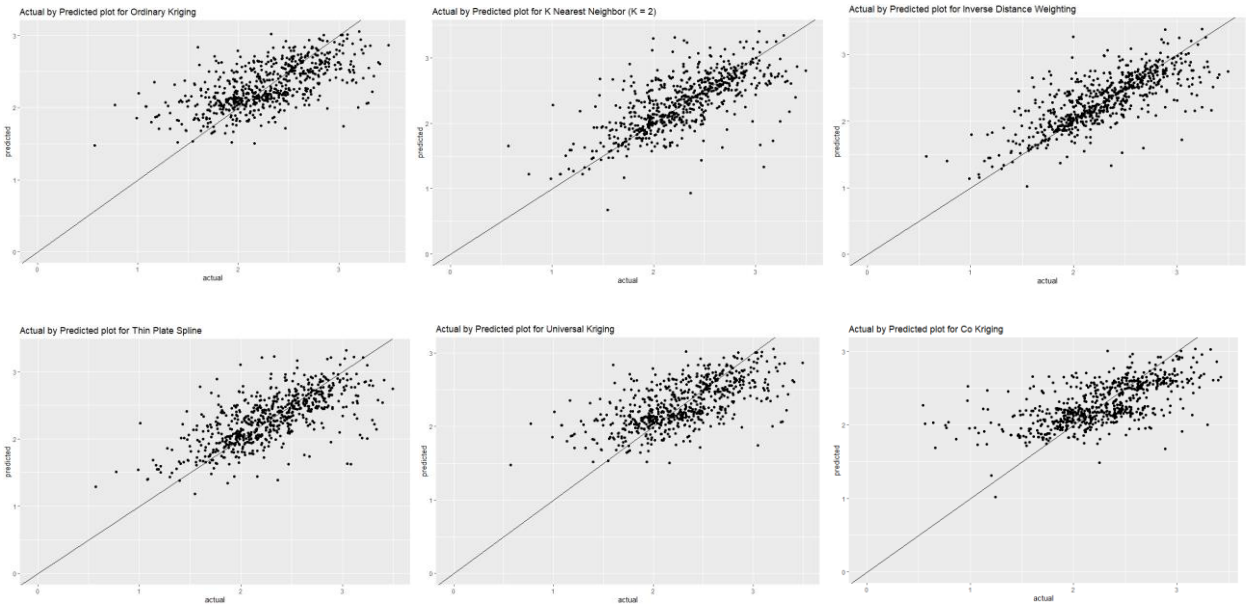
After fitting these six various models to the test set to compute accuracy metrics, it was found that inverse distance weighting was the best interpolation method for this snow data. The R^2 value is the highest as shown in the table below and its actual by predicted plot (see Table 1 and Figure 8) looks the best out of all these models.

Table 1: Table of R^2 values for all models

Model	R^2
IDW	0.543
TPS	0.442
K nearest neighbor	0.401
Universal kriging	0.396
Co-kriging	0.395
Kriging	0.392

IDW likely performed the best because it only took the nearest 30 points, and weighted them according to their distance from the interpolated point and gave the predicted value that way. This means it was only considering points nearby, and it was following spatial autocorrelation in a general sense.

Figure 9: Actual by Predicted Plots for all six models (kriging, k nearest neighbor, inverse distance weighting, thin plate spline, universal kriging, co-kriging); keep in mind that the depths are in meters.



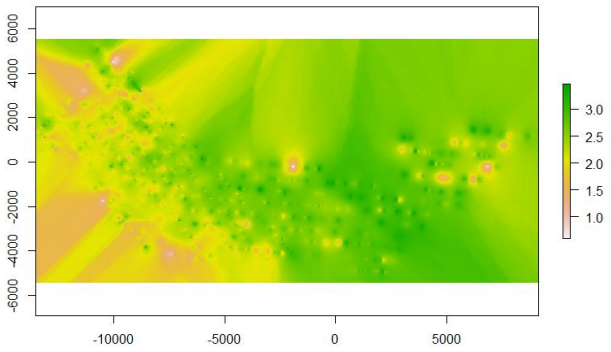
The actual by predicted plots (above) for all of the models seemed to show the same results as the R^2 table. The plot for IDW (upper right) had more points gathered tightly around the $y = x$ line than any of the other plots. The plots for the kriging models looked the worst, because their points were not following the $y = x$ line in a linear pattern, also indicated by their worse R^2 values.

The results are a little surprising, however, as kriging is often expected to perform the best with geographical data. However, it seems the spatial autocorrelation does not follow an obvious or drastic pattern, as the semivariogram model has a very high range value. Since the

range is so high (8700), the kriging model is considering points that are 8700 meters away into the equation to interpolate a value which is not reasonable. Intuitively, an expected range value would be no larger than 100 meters because it seems reasonable to be able to predict the depth of snow if you knew the depth of the snow at several points within 100 meters of it. Another noteworthy observation is that the kriging models hardly differ, and this is likely because of the poor spatial autocorrelation of the depth values as was explained earlier—the autocorrelation of the snow depth was not a convenient pattern for kriging to work well. The semivariogram never seemed to show a point where the depth measurements became independent of one another at a certain distance.

Given below is a final raster of the area in Colorado with all points interpolated using inverse distance weighting. This, according to this study, was the best model to predict the snow depth in this area with this data.

Figure 10: Raster interpolated by IDW model



Conclusion

Seeing that IDW performed the best at predicting points within the test set and it had a significantly larger R^2 value than the other models, this study provides confidence that IDW is

the best method for predicting snow depth measurements gathered by FMCW radars. Although, it must be mentioned that it only had an R^2 value of 0.54, which is not an outstanding value for prediction accuracy. This is evidenced in the ruggedness of the interpolated smooth raster in Figure 9 (see appendix). The surface on the edges looks bad because of the layout of the points used to train the model. But even within the bounds of the points used, the surface is doing a poor job of smoothing the gradient between the known points. Perhaps this would be better using different parameters to train the model on such as the number of points to aggregate from.

There were many takeaways from this research, but it would be wise to conduct additional studies with different geospatial data and in different locations to test the different strengths and weaknesses of the various interpolation models. Even though IDW worked the best for this research, it is not guaranteed that it will outperform the other models in every other spatial data set. An area of possible improvement in this study might be to try training the models with different amounts of the data. The same subset was used for all these models with the same training and test set. They might produce different results if a different amount of the available data was used. In addition to this, there are other geospatial interpolation methods such as block kriging, generalized additive model, triangular irregular surface that were not explored in this study. These methods might also be more accurate and should be considered as well.

References

Ahmed, Z. (n.d.). *Co-kriging*. Retrieved March 28, 2022, from https://zia207.github.io/geospatial-r-github.io/cokriging.html

ArcGIS Pro. (n.d.). *How kriging works*. Retrieved March 28, 2022, from https://pro.arcgis.com/en/pro-app/2.8/tool-reference/3d-analyst/how-kriging-works.htm#GUID-94A34A70-DBCF-4B23-A198-BB50FB955DC0

Columbia. (n.d.). *Thin Plate spline regression*. Columbia Public Health. Retrieved March 28, 2022, from https://www.publichealth.columbia.edu/research/populationhealth-methods/thin-plate-spline-regression

GISGeography. (2022, January 22). *Inverse Distance Weighting (IDW) Interpolation.* GIS Geography. https://gisgeography.com/inverse-distance-weighting-idw-interpolation/.

Mälicke, Mirko. (n.d.) *Variogram Models - SciKit GStat 1.0.1 Documentation*. Retrieved April 11, 2022. https://mmaelicke.github.io/scikit-gstat/reference/models.html.

Odgers, Nathan. (2017, September 28) *Simple Interpolation.* Github.io, Retrieved April 12, 2022, http://pierreroudier.github.io/teaching/20171014-DSM-MasterclassHamilton/20170928-simple-interpolation.html.

Pebesma, E. (2016, February 14). *Fitting variogram models in gstat*. R-spatial. Retrieved March 28, 2022, from https://r-spatial.org/r/2016/02/14/gstat-variogram-fitting.html

R Spatial. (n.d.). *Interpolation*¶. Spatial Data Science. Retrieved March 28, 2022, from https://rspatial.org/raster/analysis/4-interpolation.html

United States Geological Survey. (n.d.). *GIS Data Download.* The National Map – Data Delivery. Retrieved March 28, 2022, from https://www.usgs.gov/the-national-map-data-delivery/gis-data-download

University of Alabama College of Engineering. (n.d.). *Data distribution*. Remote Sensing Center. Retrieved March 28, 2022, from https://rsc.ua.edu/data-distribution/