Generating Prediction Map for Geostatistical Data Based on an Adaptive Neural Network Using only Nearest Neighbors

Sathit Prasomphan and Shigeru Mase

Abstract—**This paper proposes a new interpolation method for spatial data based on an adaptive neural networks using only the different of x-coordinate, y-coordinate between observed data and their nearest neighbors, and values of neighbors surrounding unobserved location for training network architecture. Unobserved data are interpolated by function of its absolute location and relative location in x-coordinate and y-coordinate and corresponding value at absolute location of k-nearest neighbors. We compared our new proposed method by using observed data to generate prediction map using simulation data set and real world data set. The experimental results show that, by using relationship between nearest neighbors of unobserved point can achieve the good accuracy compare to competitive method for various data set and at different rate of missing.**

Index Terms—**Interpolation method, adaptive neural networks, prediction map, geostatistics.**

I. Introduction

Some previous works in geostatistics for spatial interpolation were proposed to improve the accuracy of interpolation. The probabilistic framework of geostatistics is a continuous state random field and its main data type is a set of values observed at generally irregular locations. There are several interpolation algorithms which were proposed for this problem. The most well known method for spatial interpolation in geostatistics is Kriging [1]. The kriging predictor is the minimum variance unbiased predictor of the value at an unobserved location which is a weighted linear sum of data. The weights are chosen so that the prediction error variance for a given spatial covariance (or variogram) function of the underlying random field. Several researches used kriging method for spatial interpolation as appeared in [2] by Fernandez and Calceron. They studied the spatial regression analysis and proposed a new spatial estimation for a spatial econometric model using kriging estimator. Lokupitiya et al [3] presented a comparison of missing value imputation methods using a crop yield data. In their paper, a comparison of the performance of imputation algorithms was shown for four methods: multiple imputation, regression, kernel smoothing and kriging. As a result, multiple imputations were superior to other methods. Pollice & Lasinio [4] presented two approaches to the imputation of missing data and calibration of measurements coming from different monitoring networks. They analyzed an air quality data for the municipal area of Toronto using Bayesian kriging method. Sidler [5] proposed the kriging and the conditional simulation technique based on scale-invariant covariance models. He developed a matlab program called “vebyk” and used kriging algorithms as the basis for conditional simulations of the porosity distribution. The kriging method depends essentially on assumed second-order or intrinsic stationary of random fields and corresponding covariance or variogram function models. The use of a wrong model will effect on the correctness of the interpolation process. Also it is usual to use all the data for constructing predictors. This global approach may also cause a problem if the assumed global stationary is ambiguous and is only valid locally. An artificial neural network is another solution for solving the problem of prediction of unobserved locations. Neural networks are computational models inspired by the neural structure of human brain [6]. Their approaches make no assumption regarding the nature of the spatial data (stationary) and can handle linear relationships. It is not need to specify a particular variogram model as kriging.

Several researches used neural networks to apply for specific proposed with different input in networks architecture. Mervin et al [6] proposed a spatial interpolation based on only absolute location and spatial relationships associated with the relative location of a missing point. They used artificial neural networks for training data set by using five variables: absolute location, average distance, average sine, and average cosine between an interpolated point and its set of sample neighbor’s points. In this method, for training neural networks, whole observed dataset were used for training by using six point neighborhoods and 16 point neighborhoods of every observed point to train in neural networks. For each point to be interpolated, the nearest six point neighborhoods and 16 point neighborhoods surrounding these points were chosen with corresponding five variables as describes before, for using as inputs into the neural networks. Demyanov & Kanevsky [7] proposed the neural network residual kriging (DNRK) for a climate data. DNRK consists of two steps. The first step is estimations of large scale structures using artificial neural network (ANN) with a sum of squared errors. The second step is the analysis of residuals where geostatistical methodology is applied to model the local spatial correlation. Bellerby [8] proposed a neural network based approach to estimate the conditional distribution function of rainfall for multi-dimensional

Manuscript received October 10, 2012. ; revised December 27, 2012.

Sathit Prasomphan is with Department of Computer and Information Science Faculty of Applied Science, King Mongkut’s University of Technology North Bangkok, Thailand (e-mail: sathitp@ kmutnb.ac.th).

Shigeru Mase is with Department of Mathematical and Computing Sciences, Tokyo Institute of Technology,Japan(e-mail: mase@is.titech.ac.jp)

DOI: 10.7763/IJMLC.2013.V3.280

satellite-derived data. Sitharam [9] proposed a combination of three models, geostatistical, neural network and support vector machine, to predict a reduced level of rock. S.Wang [10] used neural networks for spatial interpolation with new input(x, y, xy, x^2, y^2) in training process. We found that, by using different type of input pattern in neural networks for spatial interpolation as presented in [11], [12], the interpolation results were different. On the other hand, if used more data to train since some input may not necessary for the training process it may causes time consuming and less accuracy. Moreover, if in each data set composed of several distributions, the global training and testing by using whole values of observed data to interpolate unobserved data could not give the correct interpolated values for some values which was valid only local. So, in this study, the new inputs in training process for neural networks were proposed. Also, localized nearest data was used for selecting suitable number of k by using cross-validation process to check the effect of using this k number. We used this number both the training process and testing process. Since most techniques used so far by researchers for interpolating geostatistical data are kriging, kernel smoothing and neural network, we compare our method with them.

The rest of this paper is organized as follows. Section 2 describes the background of our method and the proposed method is presented. Experimental setups and results are described in section 3 and section 4 respectively. Conclusions are presented in section 5.

II. New Artificial Neural Network Interpolation for Geostatistics Dataset

Typical geostatistical data consists of two parts: locations of data and corresponding values which may be multivariate. However, in this research, we consider only univariate data.

A. Notations and Definitions

Let X be a $n \times 3$ matrix of n geostatistical data and

$$X_i = (x_i, y_i, z_i) \in \Re^3$$

be the i-th row of X. (x_i, y_i) is the 2-dimensional coordinates of the location of i-th data and z_i is the corresponding value. We can consider them as the input and the output as

$$z_i = f(x_i, y_i), 1 \le i \le n \tag{1}$$

Unobserved data is denoted by (x_0, y_0, z_0) and is called the target vector. Thus we are interested in predicting the value

$$z_0 = f(x_0, y_0). \tag{2}$$

From basic theory of geostatistical data set, that is, spatial data values from location close to each other are more similar than data values from location far apart [13], [14]. So, the interpolation of unobserved values (z_0). in each points is $z_0 = f(x_0, y_0, \textit{other information relevance with}\ (x_0, y_0))$. In this formula, we use "*other information relevance with* (x_0, y_0) " which is the different between unobserved data and k nearest neighbors in x-coordinate, y-coordinate and the corresponding values in that position.

The processes for interpolating unobserved data are described as following subsection.

B. Calculation of the Number of Nearest Neighbors

The first step is to decide number of nearest neighbors of the target location (x_0, y_0) with respect to Euclidean distance. If the number of nearest neighbors is too large, it will increase the error in interpolation process since it will include irrelevant data. Number k of nearest neighbors is calculated by using a leave-one-out cross-validation for the complete dataset. At first, we let k=4. For each $i = 1,2,..., n,$ let v_i^* be the mean of k nearest neighbors of (x_i, y_i) calculate from

$$v_i^* = \frac{1}{k}\sum_{j=1}^{k} z_j \tag{3}$$

Next, compute

$$CV(k) = \sum_{i=1}^{n} (v_i - v_i^*)^2, k \ge 4\ . \tag{4}$$

Finally, let k be the number which gives the smallest $CV(k)$

$$k = \arg\min_{k=4}^{n} CV(k) \tag{5}$$

We use this k number for both network training and interpolating unobserved data. The range of k values are starting with four points, the maximum number of k for cross-validation process is 16 since this number come from number of point to generate surface of square lattice.

C. Learning Phase

The structure of our proposed neural network is shown in Fig. 1. We used feed-forward multilayer neural network with back propagation learning algorithms. A network consists of one input layer, two hidden layers and one output layer. In this network, number of input is equals to 2+3k, where 2 is number of absolute location variable: x-coordinate, y-coordinate of sampling point, k is number of nearest neighbors of sampling points which we have already calculated to find suitable this k values in the previous subsection, the number 3 is number of variable per one point which are relative location calculated from the different values in x-axis and y-axis between sampling point (x_i, y_i) and nearest neighbors point $(x_{a_j}, y_{a_j}), j = 1:k$. We denote with Δx_{a_j} for different between sampling point and nearest neighbors point in x-axis, Δy_{a_j} for different between sampling point and nearest neighbors point in y-axis.

So, if we have five nearest neighbors of unobserved point x_0 number of neural in input layer is 17, which is in each training set composed of $x_i, y_i, \Delta x_{ia_j}, \Delta y_{ia_j}, z_{a_j}$ where $j = 1:k, x_i, y_i$ is absolute location of sampling point i,

$\Delta x_{ia_j}, \Delta y_{ia_j}$ is $x_i - x_{a_j}$ and $y_i - y_{a_j}$, respectively, which is the different between sampling point i and the nearest neighbors a_j and z_{a_j} is value corresponding with x_{a_j}, y_{a_j} .

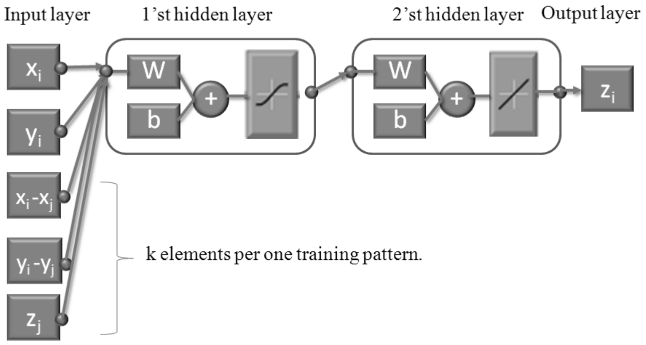


Fig. 1. Neural network architecture.

In our proposed method the training process is a supervised learning algorithms, that is, we must train neural network from set of input and desired output, feed this information into neural network to learn pattern of data. The process in hidden layer is to adjust weight connect to each node of input. In this model, back propagation algorithms are used. This algorithms works as follow: from the initial weight connect between input and each layer of hidden layer; they compare the RMS (root mean square) error between desired output and its calculated output. If the error not satisfied with the predefined values it will propagates error back to the former layer. This will be done from upper layer to input layer direction. It will adjust weight from initial weight until it gives satisfied RMS error. In our proposed algorithms number of node in each hidden layer is equal to number of node in input layer. From this neural network architecture, each unobserved data are calculated from the most nearest neighborhood. The order of each points in this k-nearest neighbors affect to the calculation of weighting in neural network. So, before sending each training points into neural network, data will be labeling and ordering by direction which is calculated from $\cos\theta$ of each nearest point(x_{a_j}, y_{a_j}) refers to the unobserved point(x_i, y_i) with this equation,

$$\cos\theta_i = \frac{(x_i, y_i).(x_{a_j}, y_{a_j})}{\|(x_i, y_i)\|.\|(x_{a_j}, y_{a_j})\|} \tag{6}$$

After that, the ordered points are fed into the network architecture. Each neuron performs a weighted summation of the inputs, after that passes this weighted summation of the inputs into a nonlinear activation function. We used sigmoid function to be activation function in each hidden layer. Finally, the network output, called output layer, is formed by weighted summation of the outputs from hidden layer. In this study, only one single output is used for the approximation of unobserved values.

D. Interpolating Unobserved Values

After the training process of neural network are finished, we can interpolate unobserved value z_0 by sending these parameters: $x_0, y_0, \Delta x_{0a_j}, \Delta y_{0a_j}, z_{a_j}$ as input pattern into trained neural network.

III. Experimental Setup

The following data, parameters, and comparison results are considered in our experiments. We compared the performance of our proposed algorithms with different experiments. The experiment is tested to interpolate every points of image by using observed points to generate interpolator for generating prediction map. In this part, we compare our proposed method with kriging algorithms, and neural networks with several input pattern and different k values. The following competitive methods were used for our experiment:

1) Our proposed method by sending $x_i, y_i, \Delta x_{ia_j}, \Delta y_{ia_j}, z_{a_j}$ of nearest neighbors of unobserved data as input pattern.

2) Neural network using the attribute of nearest neighbors of unobserved data itself as input data.

3) Neural network architecture proposed by Merwin which is sending $x_{a_j}, y_{a_j}, z_{a_j}, dist_{ia_j}, \cos_{ia_j} \sin_{ia_j}$ of nearest neighbors of unobserved data as input data.

4) Ordinary kriging with three variogram models: exponential model, Gaussian model and spherical model.

5) Kernel smoothing using Gaussian kernel function.

For easy to displays the results, we denote the following method: Our proposed method, Neural network using the attribute of nearest neighbors of unobserved data itself, Neural network architecture proposed by Merwin, Ordinary kriging, and Kernel smoothing with this symbol: ours, NN-1, NN-2, kriging and KS respectively.

A. Data Sets

We give two examples of interpolation's results by using our algorithms compare to kriging algorithms and neural network with various input pattern. The first example is simulation data set, we generated data random field using spherical covariance function which is defined as follows:

$$\mathrm{cov} = \frac{1}{\tau}(1 - 3/2(d/\theta) + \frac{1}{2}(d/\theta)^3), d <= \theta \tag{7}$$

We generated 500×500 simulation maps by using this covariance function. The number of data set is 250000 elements with *x*=1:500 and *y*=1:500. Next, 500 sampling elements for generating interpolator model was randomly sampling. Another example is real world data set, which is the Walker Lake data sets, used in Isaaks and Srivastava's book. In this dataset, composed of 780000 elements with *x*=1:300 and *y*=1:260. We used 300 randomly sampling of this dataset for experiments.

B. Accuracy evaluation

Our proposed method and competitive method were compared its performance by using mean absolute percentage error (MAPE) and root mean square error (RMSE). MAPE are calculate from following equation

$$MAPE = \frac{1}{2}\sum_{i=1}^{n} | \frac{y_i - \bar{y}_i}{y_i} | \times 100 \tag{8}$$

where *n* is number of unobserved data, $\bar{y}_i$ is an imputed incomplete data by using above five method, y_i is complete data. Another performance measure of the data set is measured by the RMSE which can be computed as

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y}_i)^2} \qquad (9)$$

where n is the number of unobserved data, y_i is the actual value and $\bar{y}_i$ is the predicted value.

IV. Results and Discussions

Results are shown in following sections.

MAPE's of kriging prediction for three variogram model (spherical, Gaussian, and exponential) are shown in Table I.

TABLE I: Mean Absolute Percentage Error (MAPE) of Simulated Gaussian Random Field Datasets for Various Variogram Models and Missing Rates

Missing rate	Exponential model	Gaussian model	Spherical model
10%	3.723	4.424	7.126
20%	4.154	7.746	12.281
30%	7.112	13.557	12.843
40%	11.609	14.113	23.215
50%	12.808	17.316	28.998

As a geostatistical prediction, kriging method is most well-known and has a long history. Kriging prediction is actually dependent on underlying models and one must estimate model parameters beforehand. By using our proposed method to generate prediction map the following results are shown. Fig. 2(a) shows original simulated prediction map, Fig. 2(c)-(f) shows prediction map which generated from five method. The rmse of ours proposed method and competitive method are shown in Table II. The results show that, the accuracy of prediction map generated by using our algorithms in these two dataset are higher than other neural networks which used different input pattern and give the similar accuracy as interpolated with kriging algorithms. The interest's point with using different neural network architecture is number and characteristics of input pattern for training neural network, they became a significant factor for measure performance of interpolation as we can see in Table II. That is, by training only absolute location values the accuracy of neural network are worst than sending more data into neural network such as the relative location in x-coordinate and y-coordinate, distance and angle between unobserved and nearest points. Moreover, the most importance factor is by sending this parameter into the training process and training data as global training instead of localized training as proposed by ours the accuracy of global training are worst than localized training as shown in Fig. 2.

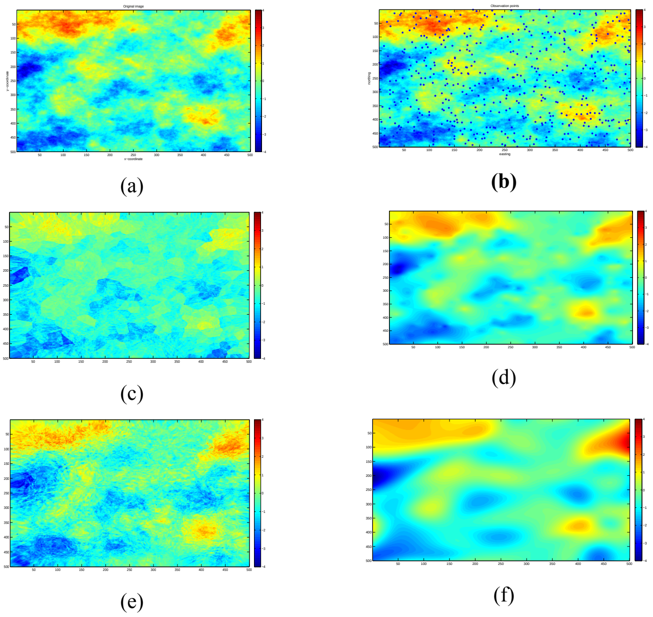
We can conclude that the local training based on different between unobserved data and k nearest neighbors in x-coordinate and y-coordinate and the corresponding value in that position is sufficient to interpolate the unobserved data.

The choice of number of nearest neighbors is important. If we use larger numbers of nearest neighbors, it will include irrelevant values and may give bias to the training process. On the other hand, if smaller numbers of nearest neighbors are used, it will not enough in the training process. In the prediction map numbers of nearest neighbors are 8 and 7 for simulated dataset and the Walker Lake dataset.

An apparent problem on our method as well as any other methods is the so-called the "edge effect". If a missing data locates in the edge of regions of available data, it may give a wrong imputed value because some of neighboring data locates actually far from the missing value location. This effect may also be present when we decide the number of neighboring data by leave-one-out cross validation and it may be better to exclude data which locate in the edge of data region from the first.

TABLE II: Root Mean Square Error of Prediction Map in Two Datasets at Various Method

Method	Simulated dataset		The Walker Lake data sets	
	k	rmse	k	rmse
1. Our proposed method	8	0.411	7	0.1756
2. Interpolation using only x,y attribute	500	1.6015	300	0.7055
3. Merwin's algorithms	16	0.4975	16	0.3958
4. S.Wang's algorithms	500	0.4922	300	0.2127
5. kriging method	500	0.3409	300	0.1548



(a)

(b)

(c)

(d)

(e)

(f)

Fig. 2. Prediction map by using 500 random sampling with five method (a) Original simulated image with 500x500 pixels (b) original map with 500 random sampling (c)the proposed method (d)kriging method (e) Merwin's algorithms (f) S.Wang's algorithms.

V. Conclusion

A neural network method for interpolating unobserved data in geostatistical data based on different of values in x-axis and y-axis between observed data and nearest neighbors is proposed. A leave-one-out cross validation is used to decide how many neighbors should be incorporated. Different in x-coordinate and y-coordinate between unobserved data versus nearest neighboring and data value of neighboring data of the location which is unobserved are used as input and output data to train the neural network.

Finally this data are fed into neural network to interpolate the missing value. We tested our method by generate prediction map by using ours, neural network with different input, and kriging algorithms. From the experiment, we can conclude that our method is most accurate in most cases. Also our method needs no further assumption on underlying models which is necessary for the kriging method. One problem of our method is that it is relatively time consuming in order to train the neural network if large number of nearest neighbors is used. Moreover, if missing data places in the edge of data region, our method as well as any methods may give wrong imputed values, a typical difficulty of spatial data analysis.

ACKNOWLEDGMENT

The authors would like to thank Prof.Dr.Chidchanok Lursinsap and Asst.Prof.Dr.Sirapat chiewchanwattana for kindly providing guidance throughout this research. Also, the authors would like to thank Prof.Dr.Shiogeru Mase, who gives a good suggestion in the field of Geostatistics and providing the facility during research at Mase's laboratory, Tokyo Institute of Technology, Japan.

REFERENCES

[1] A. Gebhardt, *PVM Kriging with R*, New York, Chichester, 2003.

[2] G. Fernandez and A. Calderon, "Spatial regression analysis vs. kriging methods for spatial estimation," *Int. Adv. Econ. Res.* vol. 15, no.1, pp. 44-58, 2009.

[3] R. S. Lokupitiya, E. Lokupitiya, and K. Paustian, "Comparison of missing value imputation methods for crop yield data," *Environmetric*, vol.17, pp. 339-349, 2005.

[4] A. Pollice and G. Lasinio, "Two approaches to imputation and adjustment of air quality data from a composite monitoring network," *Journal of Data Science.* pp. 43–59, 2009.

[5] R. Sidler, *Kriging and Conditional Geostatistical Simulation Based on Scale-Invariant Covariance Models,* 2nd ed., JohnWiley&Son's, New York, 2003, vol. 2.

[6] A. Merwin and D. Cromley, "Artificial neural networks as a method of spatial interpolation for digital elevation models", 2002.

[7] V. Demyanov and M. Kanevsky, "Neural network residual kriging application for climatic data," *Journal of Geographic Information.* 1998.

[8] T. J. Bellerby, "Satellite rainfall uncertainty estimation using an artificial neural network," *Journal of Hydrometeorology and Decision Analysis*," vol. 2, pp. 215-232, 2007.

[9] T. G. Sitharam, P. Samui, and P. Anbazhagan, "Spatial variability of rock depth in bangalore using geostatistical, neural network and support vector machine models," *Geotech Geol Eng,* vol. 26, no.5, pp. 503-517, 2008.

[10] S. Wang and F. Li, "Spatial interpolation of DEM using BP artificial neural networks," in *Proc. International conference on Earth observation Data Processing and Analysis*, 2008.

[11] K. Sen, M. U. Gumusay, and A. Kavas, "Programming an Artificial neural network tool for spatial interpolation in GIS-A case study for indoor radio wave propagation of WLAN," *Sensors*, vol. 28, pp.1091-1113, 2008.

[12] F. Liu, X. He, and L. Zhou, "Application of generalized regression neural networks residual kriging for terrain surface interpolation," in *Proceedings of SPIE*, Oct. 2009.

[13] E. Carrizosa and J. Gordillo, "Support vector regression for imprecise data," *Technical Report MOSI/35,* MOSI Department, Vrije Universiteit Brussel, 2007.

[14] A. M. Michalak and P. K. Kitanidis, "A method for the interpolation of nonnegative functions with an application to contaminant load estimation," *Stoch Environ Res Risk Assess*, vol. 17, pp. 519-533, 2005.

Sathit Prasomphan was born in Ubonratchatani, Thailand, on 21st May, 1979. He received B.Sc. in Computer Science from Khon Kaen University and M.Sc. in Software Engineering, from Chulalongkorn University, Thailand, in 2001 and 2003, respectively. His doctorate in Computer Science from Chulalongkorn University has been under the supervision of Professor Dr. Chidchanok Lursinsab and co-advisor is Assistance Professor Dr.Sirapat Chiewchanwattana. In 2010, he was granted a visiting Thailand Government scholarship to research as visiting researcher at Tokyo Institute of Technology, Japan under guidance of Prof.Dr.Shigeru Mase.

His field of interest includes various topics in neural network, time series, and geostatistics. He is also educated in the field of soft computation.