DATA INTERPOLATION USING KOHONEN NETWORKS

Olivier Sarzeaud
ECTIA
BP 92119
F-44321 Nantes cedex 3
FRANCE
Olivier.Sarzeaud@ectia.ec-nantes.fr

Yann Stéphan
EPSHOM / CMO
BP 426
F-29275 Brest cedex
FRANCE
stephan@shom.fr

1. Introduction

Physical data interpolation is a common issue in Geosciences. For many variable of interest, the measurements are often sparse and irregularly distributed in time and space. Analyzing the data usually requires a numerical model, which samples the data on a regular grid. Mapping irregular measurements on a regular grid is done by interpolation, which aims to generalize, but not to create, information. A popular method to map geophysical data is kriging [1]. This method, based on the hypothesis that the measurements are realizations of a random variable, has been proven to be optimal under certain conditions. It requires to solve a system of linear equations at each point where the interpolation must be done, which might be computationally heavy. This paper proposes an original interpolation method based on Kohonen networks. The method is applied on the problem of building a surface-temperature climatology in the Mediterranean Sea. The method performs very well, combining an accuracy comparable with usual kriging methods with a shorter computing time, and is especially efficient when a great amount of data is available.

The paper is organized as follows. Section 2 recalls the backgrounds of kriging techniques. Section 3 describes the adaptation of self-organizing maps to the spatial interpolation problem. The results of actual data interpolation in an oceanographic problem are presented and discussed. The last section draws conclusions and perspectives.

2. Interpolation

A model of a physical variable aims at predicting its value anywhere at any time. The simplest model is a numerical one, that is a discrete representation of the variable. To be efficient, this representation must be done under two constraints: on the one hand no information must be missed, on the other hand a reasonable amount of storage capacity is required. It must also be done on a regular grid, in order to be usable by most analyzes tools (plotting a map of the variable, computing Fourier Transform, ...).

2.1. Definition of interpolation

Considering n values (obtained by measurements) of a variable Z_i at locations x_i, $1 \le i \le n$, interpolation aims at building a numerical model of the variable on a regular pre-defined grid. A straightforward way to interpolate data on a specific location (a point of the grid) is to make a linear combination of the data:

$$Z^* = \sum_{i=1}^{n} \lambda_i Z_i \qquad (1)$$

where Z^* is the estimated value. The problem is to compute the weights λ_i in order to minimize the estimation error. Practically, this is not feasible, because the true values are not known. It is thus necessary to make assumptions on the behavior of the variable to define the optimality.

The simplest methods give higher weights to the nearest data. The weights are somehow inversely proportional to the distance. This corresponds to an implicit assumption of continuity of the variable, which seems reasonable for physical variables. Anyway, it is possible to do better, taking into account the spatial correlation of the data. In this case, the weights are the solutions of a system of linear equations, that can be obtained by writing the minimization of the estimation error. This is kriging.

2.2. Kriging

Kriging is based on a statistical interpretation of the measures. Indeed, it assumes that the data are realizations of a

0-7695-0619-4/00/$10.00 (C) 2000 IEEE

random variable, that is: $Z_i = Z(x_i)$. Some hypothesis are required on the behavior of this random variable, usually that the expectation of an increase is null, and its variance only depends on the distance (intrinsic random variable):

$$\begin{aligned} E[Z(x+h) - Z(x)] &= 0 \\ Var[Z(x+h) - Z(x)] &= C(h) \end{aligned} \tag{2}$$

Therefore, on each point x_0 where the interpolation is to be done, it is possible to write analytically the expectation and variance of the estimation error $Z^*(x_0) - Z(x_0)$. The nullification of the expectation (ensuring that the estimation is not biased) leads to a constraint on the weights λ_i:

$$\sum_{i=1}^{n} \lambda_i = 1 \tag{3}$$

The minimization of the variance (that is the optimality of the estimation) under the constraint of Eq. 3 leads to a system of linear equations, with coefficients depending on a model of the variance of the increment of the data [4]:

$$\begin{bmatrix} C_{11} & \dots & C_{1n} & 1 \\ \dots & \dots & \dots & \dots \\ C_{n1} & \dots & C_{nn} & 1 \\ 1 & \dots & 1 & 0 \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \dots \\ \lambda_n \\ \mu \end{bmatrix} = \begin{bmatrix} C_{10} \\ \dots \\ C_{n0} \\ 1 \end{bmatrix} \tag{4}$$

where $C_{ij} = Var[Z(x_i) - Z(x_j)] = E[(Z(x_i) - Z(x_j))^2]$

Once the weights are found, it is also possible to compute the (residual) variance of the estimation error on each point where an estimation is performed:

$$Var[Z^*(x_0) - Z(x_0)] = \frac{1}{2} \sum_{i=1}^{n} \lambda_i C_{i0} \tag{5}$$

This approach is also called objective analysis.

When a great amount of data is available, kriging at each point cannot be performed using all data, because it would lead to huge systems that may not be handled. Instead, it is necessary to choose a few data around the point where to interpolate. Furthermore, these data have to be chosen to avoid singularity of the system, which is usually done with the help of many parameters in kriging products. Anyway, it remains that a system of linear equations must be solved on each point of the final grid, which is computationally heavy.

The main advantage of kriging is that it relies on strong theoretical backgrounds, which demonstrate that the interpolation is unbiased and optimal. The main drawbacks are:

- The hypothesis done on the random variable are strong. It is possible to relax them (allowing a determinist drift on the data for example), but the kriging system is then more complex. In any case, a model of variance of the increment of the variable must be computed, which can be very long when a lot of data is available.
- It is difficult to ensure that a system built with some data, even carefully chosen, will be regular. Therefore, especially with big data set, it is possible to have completely wrong estimates, that can usually be detected. Anyway, it remains the possibility to have wrong estimates that are not far enough from the expected value to be detected.
- To make a numerical model on a grid, it is necessary to interpolate, that is to solve the system of equations, on each point of the grid. This again might be very long, depending on the desired model.

Kriging has been used during the european project MEDATLAS to make an atlas of the temperature climatology field over the Mediterranean sea [2]. The aim was to produce a series of maps of the temperature climatology state of the sea, for each month and at some 29 standard depths (that is 348 maps). The number of data available was related to the depth considered, from about 260,000 in surface to less than 1,000 at the bottom. The domain to model was the whole Mediterranean basin on a grid of about 25 km step, which makes more than 15,000 grid points. To achieve this goal in a reasonable computation time, the kriging approach was based on an adapted meshing of the data set, that allowed to select the points where it was worth computing. For the chosen points, a universal kriging method was used, based on a regional model of variance. The results were then re-gridded on a regular grid using a weight-distance linear combination of the 4 closest estimated values. An example of temperature map is given in figure 1.

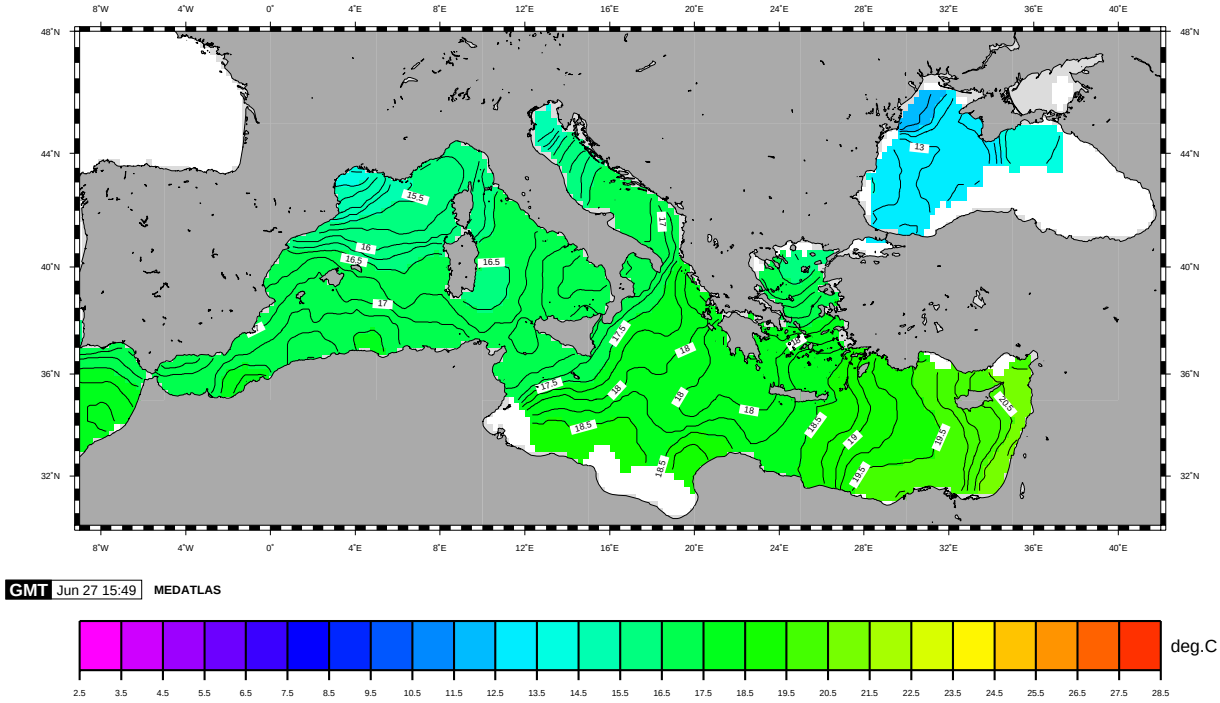

Figure 1: MEDATLAS map of May at 10 m immersion.

3. Neural interpolation

Kohonen networks need not be presented here, the reader can refer to [3] for details. Some work has already been done and is presented elsewhere [7][8] on the use of Kohonen networks for adaptive meshing. It was shown that a simple modification of the basic Kohonen self-organizing algorithm (to constrain the peripheral neurons of the network to stay on the border of the domain) allows to produce valid meshing, with some advantages over classical methods. The use of Kohonen networks for neural interpolation also relies on a slight modification of the basic self-organizing algorithm.

3.1. General algorithm

At each time step t, the input $x(t)$ of the network is a three dimensional vector, the first two dimensions giving the location of the measure, and the third one its value. We will note this decomposition: $x(t) = (x^{loc}(t), x^{val}(t))$. Of course, each neuron k also has a three dimensions weight vector $w_k(t)$, the first two being homogeneous to a geographical location, and the third one to the studied variable. The only modification of the basic self-organizing algorithm required is that the selection of the cluster $c(t)$, that is the search for the closest neuron to the input, be performed on the first two dimensions only:

$$c(t) = c(\{w_k^{loc}(t)\}, x^{loc}(t)) = k / \left\| w_k^{loc}(t) - x^{loc}(t) \right\| \leq \left\| w_l^{loc}(t) - x^{loc}(t) \right\| \qquad \forall l \tag{6}$$

This means that the cluster is chosen in the geographical space, according to the data location only, and is completely independent from the measures value. The idea is to trust the locations rather than the values, allowing thus very different values measured on close points to be combined. Once the cluster is found, the weight modification applies on all three weights of the cluster and its neighbors:

$$w_k(t+1) = w_k(t) - \alpha(t) h(k, c(t))(w_k(t) - x(t)) \tag{7}$$

where $\alpha(t)$ is a time-decreasing gain factor and $h(k, c(t))$ a neighboring function.

In this approach, the interpolation points cannot be chosen beforehand. Instead, they are determined during the learning process, and correspond to the final locations of the neurons. Therefore, at the end of the algorithm, we still do not have the values on a regular grid, and a post-processing is required. The question is thus: what is the advantage of the

final irregular distribution of the neurons over the initial irregular distribution of the data ? If the number of neurons is lower than the number of measures, complexity is reduced without loss of information. The neurons set is the best representation in space of the data set. If the associated values are optimal, it is then possible to re-grid the values with a simple interpolation method, as was made for MEDATLAS.

Like for neural meshing, numerous constraints can be applied to the neurons without changing the properties of the algorithm. Among the possibilities, we have successfully tried:

- Initializing the network as a regular grid in the whole space and at the mean value of the measures considerably reduces the computation time.
- Constraining the peripheral neurons of the network to slide on the border of the domain ensures an interpolation in the whole domain, even where data are very sparse.
- Constraining the neurons to move on an underlying regular grid can simplify the final reinterpolation.
- Constraining the neurons not to move at all allows to directly produce the result on a regular grid. Nevertheless, this constraint is too strong when data are irregularly distributed, because some neurons are then never updated.

3.2. Theoretical results

At any time step *t*, it can easily be shown that the weight vector of a neuron *k* is a combination of the input vectors presented until then and of its initial value:

$$w_k(t) = \sum_{i=1}^{t} a_k(i)\left(\prod_{j=i+1}^{t}(1-a_k(j))\right)x(i) + \left(\prod_{i=1}^{t}(1-a_k(i))\right)w_k(0) \tag{8}$$

where $a_k(i) = \alpha(i)h(k, c(i))$. The initial value $w_k^{val}(0)$ can always be set to 0 without loss of generality. The weights of the combination do not depend on the third dimension (value) of the input and neuron weight vectors, but rather on the gain factor and on the locations of the input and neuron. Therefore, the third weight of the neuron is at any time a true *linear* combination of the data presented until then:

$$w_k^{val}(t) = \sum_{i=1}^{t} \lambda_{kt}(i)x^{val}(i) \qquad \text{with} \qquad \lambda_{kt}(i) = a_k(i)\left(\prod_{j=i+1}^{t}(1-a_k(j))\right) \tag{9}$$

Is this linear combination optimal ?

Under the same hypothesis as for ordinary kriging, it can be theoretically shown that the estimation is not biased, that is: the expectation of the estimation error is null. Indeed, if the gain factor $\alpha(t)$ follows a decreasing law of the type $1/t^{\beta}$ with $0 \le \beta \le 1$, which is the usual convergence condition of Kohonen networks [5][6], then it can be shown that:

$$\sum_{i=1}^{t} \lambda_{kt}(i) = 1 \tag{10}$$

which is similar to Eq. 3 or the last line of the system of Eq. 4. To show the optimality, that is the minimization of the variance of the estimation error, is much more difficult. Indeed, the neural interpolation algorithm never uses an explicit knowledge of the variance of the increment of the random variable. Therefore, an assumption is needed on how this variance is taken into account by the algorithm. We suppose first that the variance depends only on the distance between data points *in the representation space of the map* instead of the input space. Furthermore, we suppose that the neighboring function is a good representation of this variance:

$$C_{ij} = C_0(1 - h(c(i), c(j))) \tag{11}$$

This assumption is intuitively true for very big data sets. Indeed, in this case, measurements are made where variations are expected rather than where the variable is known to be stable. Thus, the distribution is dense where variations are important, while it is sparse where the variable is stable. Under the hypothesis of a unitary neighborhood, the estimation can then be shown to be optimal, that is we can show that:

$$\sum_{i=1}^{t} \lambda_{kt}(i)C_{ij} = C_{j0} \qquad \forall j \tag{12}$$

which is similar to the first *n* lines of the system of Eq. 4.

If a model of variance of the increments is available, the variance of the estimation error (Eq. 5) can be iteratively computed during the learning process. An updating rule similar to the one of Eq. 7:

$$C_k(t+1) = C_k(t) - \alpha(t)h(k, c(t))(C_k(t) - C_{t0}) \quad (13)$$

would lead to the following result:

$$C_k(t) = \sum_{i=1}^{t} \lambda_{kt}(i) C_{i0} \quad (14)$$

which is simply twice the variance defined in Eq. 5.

3.3. Practical results and comparison

The neural interpolation method has been used on synthetic and real data sets. The results are available in [9]. The synthetic data sets aimed at controlling the optimality of the results, and were therefore not too big. Some criteria were defined to assess the mean bias and optimality of the interpolations for all the neurons of a network. It could be shown that the bias is nearly null if there are enough modifications of the weights, that is if the number of iterations and the gain are sufficiently high. Optimality is ensured with a very low relative error (less than 3%), and requires the same conditions as nullity of the bias. The residual relative error must be compared with the error between the experimental variance and its model, which is generally about 8%.

A real data set was taken from the MEDATLAS project, and aimed at showing the computing time gain. The data set contained about 25,000 measures. The interpolation was required on a 204x72 points grid. The result was checked using cross validation. The regular estimation and error grids were first reinterpolate to the data points using a classical bilinear interpolation. Then, the error between the value used to build the map and the value given by the map at each data point was computed. The mean error on all data points was found to be -0.02, which is close enough to 0 to say that the estimation is unbiased. Finally, the error on each point was compared with the predicted error. The mean value was found to be 0.82, which is close enough to 1 to say that the error map is coherent with the estimation. However, there can be cases where the estimation is very bad, although not biased, and coherent with the error computed. Therefore, it was necessary to make a graphical comparison between the maps. The map built by neural interpolation compared well with the MEDATLAS one (figure 2). The neural interpolation method required less than 1 minute of computing time, while the method used in MEDATLAS required more than four hours.

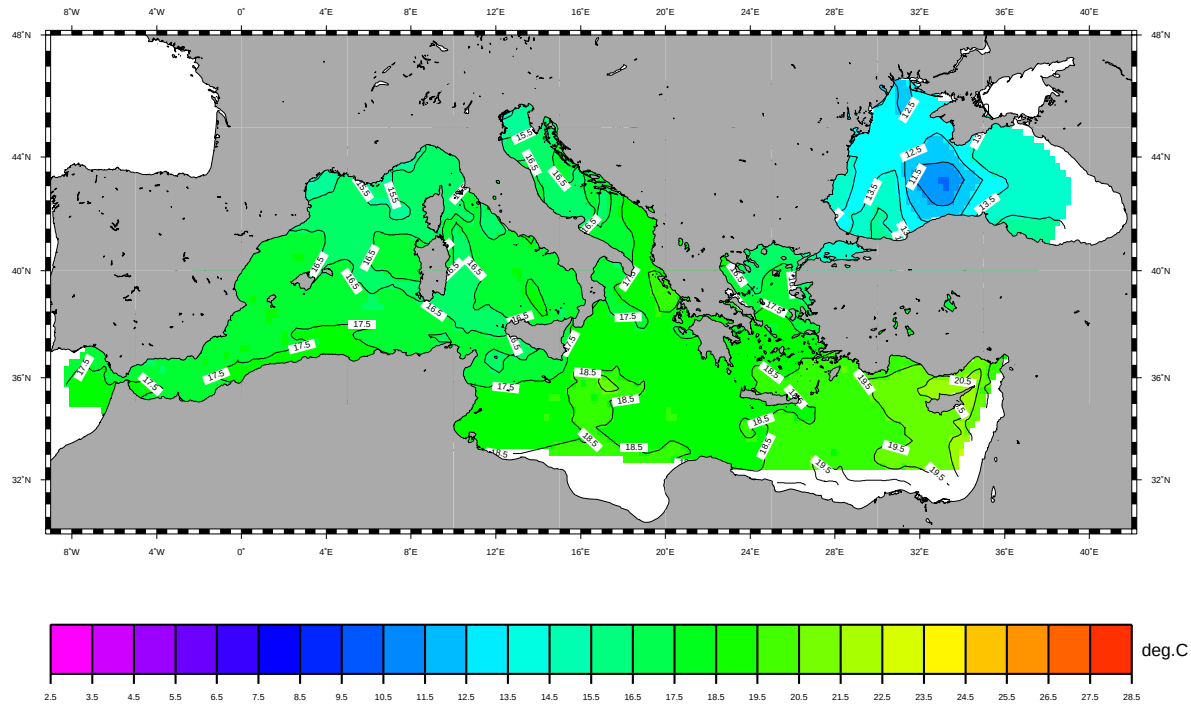

Figure 2: Map of the Med data set produced by neural interpolation.

4. Conclusion

An original method for optimal interpolation is presented in this paper. The method relies on some basic modification of the standard Kohonen algorithm. The method has been applied on several synthetic and actual data sets. In every cases, the results compare perfectly well with those obtained by kriging. However, the method is much faster than kriging when the data set is large, which is practically the case for actual problems. Future work will deal with taking into account an error on each data point, what kriging can do. Other studies will deal with the use of the Kohonen algorithm for data fusion and assimilation.

Acknowledgment: This work was funded by the Service Hydrographique et Océanographique de la Marine (SHOM) under contract 98.87.054. The authors wish to thank Didier Jourdan (CMO) for helpful comments and for making available the Med data set.

5. References

[1] M. David, D. Crozet & J.M. Robb, "Automated mapping of the ocean floor using the theory of intrinsic random functions of order k," *Marine Geophysical Researches*, 8, pp. 49-74, 1986.

[2] D. Jourdan, E. Balopoulos, M.-J. Garcia-Fernandez & C. Maillard, "Objective Analysis of Temperature and Salinity Historical Data Set over the Mediterranean Basin," IEEE, 1998.

[3] T. Kohonen, *Self-organizing maps*, Springer-Verlag, 1995.

[4] G. Matheron, "The intrinsic random functions and their applications," *Advances in Applied Probability*, vol. 5, pp. 439-468, 1973.

[5] H. Ritter & K. Schulten, "On the stationnary state of Kohonen's self-organizing sensory mapping," *Biological Cybernetics*, vol. 54, pp. 99-106, 1986.

[6] H. Ritter & K. Schulten, "Convergence properties of Kohonen's topology conserving maps: Fluctuations, stability, and dimension selection," *Biological Cybernetics*, vol. 60, pp. 59-71, 1988.

[7] O. Sarzeaud, *Les réseaux de neurones, contribution à une théorie*, Ouest Éditions, 1994.

[8] O. Sarzeaud, Y. Stéphan, F. Le Corre & L. Kerléguer, "Neural meshing of a geographical space in regard to oceanographic data location," OCEANS'94, Brest, France, 1994.

[9] O. Sarzeaud & Y. Stéphan, *Interpolation optimale et assimilation de données par réseaux de Kohonen*, Technical report OS/99003, 61 p., 1999.