

Confidence Evaluation for Risk Prediction

Nicolas Gilardi Tom Melluish Michel Maignan
UNIL, IDIAP RHUL UNIL
gilardi@idiap.ch tom@dcs.rhbnc.ac.uk michel.maignan@imp.unil.ch

August 8, 2001

Abstract

This paper describes an application of a transductive method for risk mapping which allows one to compute confidence intervals for estimates, without any assumption on data distribution, except that inputs should be independently and identically distributed. The method is compared to conditional Sequential Gaussian Simulation. The data set used is a digital elevation model of Switzerland¹.

1 Introduction

Maps drawn from a regression estimate of unknown data are often insufficient for making important decisions. Knowing also the confidence one has in the estimate is crucial. Many methods exist in statistics and machine learning for solving such problems ([5][4], etc. . .), but most of them need strong *a priori* knowledge of the nature of the distribution of data. For example, conditionnal Sequential Gaussian Simulations (SGS) [1] assume that data outputs are normally distributed. If it is not the case, normal-score transformation and back-transformation [2] are necessary, which can create some artifacts in the final results.

The Ridge Regression Confidence Machine (RRCM) [6] appears to be a good alternative to these methods when studying data far from “normality”, as the only assumption made by RRCM is that data should be independently and identically distributed (iid).

In this paper, we first give a short theoretical description of the RRCM method, as well as of SGS. Then, we describe experiments conducted, among others, on the digital elevation model of Switzerland (Swiss DEM).

2 Theory

More detailed theoretical descriptions of the methods presented here can be found in the references, especially the mathematical demonstrations.

2.1 Ridge Regression Confidence Machine

This transductive method was introduced in [3]. Its goal is to produce a machine learning algorithm which is able to estimate the confidence one has in a prediction, either for classification [8] or regression [6].

¹please refer to <ftp://ftp.idiap.ch/pub/reports/2001/IDIAP-RR-01-22.ps.gz> for a more up-to-date version of this paper

2.1.1 General theory

Suppose that one has a set of labeled data $(z_1, \dots, z_l) \in \mathcal{Z} = ((\mathbf{x}_1, y_1), \dots, (\mathbf{x}_l, y_l)) \in \mathcal{X} \times \mathcal{Y}$, where $\mathbf{x} \in \mathcal{X}$ is a vector of coordinates and $y \in \mathcal{Y}$ is a scalar, corresponding to the output of these coordinates. Given a new vector of coordinates \mathbf{x}_{l+1} , the goal is to find the output y_{l+1} and the confidence region of the estimation.

To solve this task, the idea consists in scanning all the possible sequences $((\mathbf{x}_1, y_1), \dots, (\mathbf{x}_l, y_l), (\mathbf{x}_{l+1}, \hat{y}))$, and evaluating how *typical* this sequence is. This is possible using a typicalness function $t : \mathcal{Z}^{l+1} \rightarrow [0, 1]$ which satisfy the condition

$$P((z_1, \dots, z_{l+1}) : t(z_1, \dots, z_{l+1}) \leq r) \leq r \quad (1)$$

where $0 \leq r \leq 1$. This means that the probability (under i.i.d assumption) to find a sequence (z_1, \dots, z_{l+1}) such that $t(z_1, \dots, z_{l+1}) \leq 1\%$ will never exceed 1%.

This property is fulfilled by the following form of the typicalness function:

$$t(z_1, \dots, z_{l+1}) = \frac{\#\{i = 1, \dots, l+1 : \alpha_i \geq \alpha_{l+1}\}}{l+1}. \quad (2)$$

In this function, α_i represents the *strangeness* of the element z_i . The condition 1 is verified if these parameters are defined by a function of the form

$$\alpha_i = F(\{z_1, \dots, z_{l+1}\}, z_i), i = 1, \dots, l+1 \quad (3)$$

When working in a regression framework, one can use the function $\alpha_i = \|y_i - f(\mathbf{x}_i)\|, i = 1, \dots, l+1$, residuals of $f : \mathcal{X} \rightarrow \mathbb{R}$, which is an underlying regression model approximating the data z_1, \dots, z_{l+1} . Then, given a significance level r (such as 5%), the predictive region $R(\mathbf{x}_1, y_1, \dots, \mathbf{x}_l, y_l, \mathbf{x}_{l+1})$ is the set of all $\hat{y} \in \mathcal{Y}$ for which $t(z_1, \dots, z_{l+1}) > r$. It has been shown in [6] that this predictive region is *at least* a $100(1-r)\%$ probability tolerance region.

Dealing with the calculation of all possible \hat{y} can become almost impossible. The Ridge Regression Confidence Machine algorithm allows one to speed-up such task.

2.1.2 The Ridge Regression approach

In order to speedup the calculation of the typicalness function, one solution is to use the Kernel Ridge Regression [7] as underlying regression model.

This algorithm consists in calculating the value \mathbf{w} that minimises the function

$$a\|\mathbf{w}\|^2 + \sum_{i=1}^{l+1} (y_i - \mathbf{w} \cdot \mathbf{x}_i)^2 \quad (4)$$

where a is a user defined positive constant. The Ridge Regression approximation is then $f(\mathbf{x}) = \mathbf{w} \cdot \mathbf{x}$.

An interesting aspect of this method is that one can calculate directly the vector of the strangeness parameters α_i , and moreover, its expression varies piecewise linearly with \hat{y} and can be written as $A + B\hat{y}$, where

$$A = (y_1, \dots, y_l, 0)(I - (XX' + aI)^{-1}XX') \quad (5)$$

and

$$B = (0, \dots, 0, 1)(I - (XX' + aI)^{-1}XX') \quad (6)$$

with $X = (\mathbf{x}_1, \dots, \mathbf{x}_{l+1})'$.

In addition, using the “kernel trick”, one can replace the dot product matrix XX' in equations (5) and (6) by a *kernel matrix* $K_{i,j} = \exp\left(\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma_{kernel}^2}\right)$, with σ_{kernel} a user defined positive constant, which

²in this case, the gaussian radial basis function kernel

allows to solve non-linear problems in a “linear” way [9], and thus improve the prediction abilities of the algorithm.

As the α_i vary with \hat{y} , so does the typicalness function t . But it can change only for \hat{y} where $\alpha_i = \alpha_{l+1}$ for some $i = 1, \dots, l$. It is thus possible to calculate the typicalness for \hat{y} intervals rather than for unique \hat{y} values. The way of calculating the typicalness with this approach is the following:

1. for each example in (z_1, \dots, z_{l+1}) , define $S_i = \{\hat{y} : \alpha_i \geq \alpha_{l+1}\}$ as being the set of all possible \hat{y} for which z_i 's residuals is greater than z_{l+1} 's.
2. for each interval of the space of \hat{y} defined by $\alpha_i = \alpha_{l+1}$, count how many S_j include it. To get the typicalness, one has to divide this number by the total number of examples, i.e. $l + 1$.

Finally, the confidence region one is looking for is the union of all the intervals of the space of \hat{y} for which the typicalness is greater than the significance level r .

2.1.3 Procedure of RRCM experiments

To calculate the 95% confidence intervals of a *test* data set (known inputs, but unknown outputs), given a *train* data set (known inputs and outputs), the algorithm goes as follows:

1. Rescale the inputs of the train and test sets, as well as the outputs of the train set.
This rescaling might be done to improve numerical stability of the program and thus to expect better results. A typical rescaling for data is to evaluate mean μ_i and standard deviation σ_i of each variable x_i of the training set, and transform it like this: $normX_i = \frac{x_i - \mu_i}{\sigma_i}$. The new variable will have a near 0 mean and a near 1 standard deviation.
2. Build a regression model using the train set.
This consists mainly in defining the optimal values of the hyper-parameters of the kernel ridge regression algorithm. The most classical way of doing it is to use a leave-one-out cross-validation procedure on the train set, trying different values of these hyper-parameters (a and σ_{kernel}).
3. For each point of the test set, calculate the confidence region of the output.
Using the hyper-parameters calculated by cross-validation, one operates as it is described in the section 2.1.2, the studied sequence being the full train set plus the test point.
4. Back transform the data.
When all the test points have been studied, one just has to back transform coordinates and confidence region's bounds.

2.2 Sequential Gaussian Simulations

Widely used in geostatistics, this family of algorithms [1] has proven its efficiency despite its strong *a priori* assumptions about the data's distribution. It was thus logical to use it as a reference to test the efficiency of the RRCM on spatially distributed data, but also to show that sometime, the results obtained are to be considered with care.

2.2.1 General description

The main goal of simulations in geostatistics is to produce “realistic” maps of a phenomenon rather than minimising the prediction error, which usually leads to smooth maps, not really representative of the real world.

To solve this task when data are normally distributed, one estimates the parameters of the local probability density function at each location of the test set, and randomly generates a value from this distribution.

The first part of a sequential gaussian simulation is to check if the known data are normally distributed, and if not, to apply normal-score transformation on them [1]. Then, one calculates the variogram of the transformed data

$$\gamma(\mathbf{h}) = \frac{1}{2N(\mathbf{h})} \sum_{i=1}^{N(\mathbf{h})} (y(\mathbf{x}_i) - y(\mathbf{x}_i + \mathbf{h}))^2 \quad (7)$$

where \mathbf{h} is a directional vector, and $N(\mathbf{h})$ is the number of pair of points separated by \mathbf{h} . One then modelises it with the function $\hat{\gamma}(\mathbf{h})$. The next part is the simulation process itself.

1. One chooses at random a point to estimate inside the unknown data set.
2. Apply the kriging procedure, using the known data set and the modelised variogram, to estimate the mean μ_{krig} and variance σ_{krig}^2 of this value. These two parameters are then considered respectively as the mean and variance of the local Gaussian probability density function of the point.
3. One randomly chooses a value for the unknown point, following the law $\mathcal{N}(\mu_{krig}, \sigma_{krig})$.
4. The simulated point is then considered as a known point and will be used “as is” to simulate the next randomly chosen unknown point.
5. Repeat from step 1. until there are no more unknown points.

The result of one simulation is thus a “noisy” version of a kriging procedure, which reproduces the statistical histogram *and* the variogram of the known data, giving a more realistic aspect to the output but a lower prediction performance. However, when performing multiple sequences of simulation, one is able to draw reliable probability maps.

2.2.2 Confidence interval calculation

As the simulated data are normally distributed, it is also easy to estimate the confidence interval of a prediction for each data point. The procedure used in this paper is the following. First, one computes the mean μ_{sim} and standard deviation σ_{sim} of the simulation of each data point. Then, one defines the 95% confidence interval bounds as the 2.5% and 97.5% quantils of the cumulative density function of the normal law defined by $\mathcal{N}(\mu_{sim}, \sigma_{sim})$. When these bounds have been calculated, one is applying the N-Score back-transformation both on them and on the mean, in order to get back to the real output space.

2.2.3 Procedure of SGS experiments

As for RRCM, one has to simulate a test set using the information given by a train set. The SGS 95% confidence interval calculation has been done like this:

1. Normal score transformation of the train outputs if necessary.
2. Variogram modelisation using the transformed outputs of the train set.
3. 100 or more sequential Gaussian simulation procedures of the test set.
4. Evaluation of the confidence intervals as described in the section 2.2.2.
5. Back-transformation of the confidence intervals bounds if necessary.

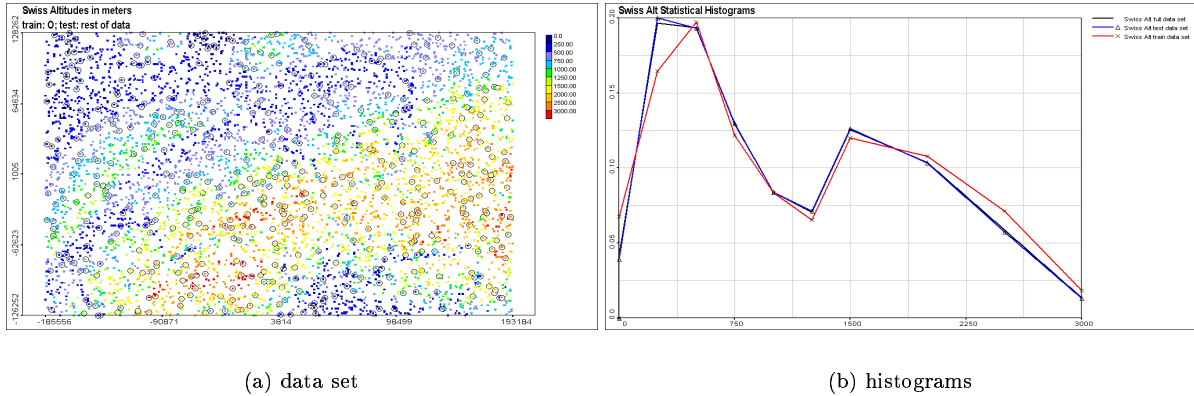


Figure 1: Swiss DEM data set and statistical histograms. Repartition between train and test sets

3 Experiments on digital elevation model of Switzerland

The digital elevation model of Switzerland (Swiss DEM) used for these experiments is a regular grid of 376x253 points, defining square cells of 1 kilometer aside. The grid covers a large part of Alpine region, Jura, and the Swiss Plateau inbetween, producing highly anisotropic variation of altitude, important local variability and a bi-modal behavior of data histogram. For these reasons and the high number of points, this data set was considered a good case study to evaluate the efficiency of RRCM and compare it to SGS.

In order not to spend weeks in experiments, the original data set has been split into two data sets. The first one is a random extraction of 500 points from the full digital elevation model, and will be used as the train set, i.e. the known data. The second one is a random extraction of 5000 points from the rest of the data set. This constitute the test set, i.e. the data to predict.

3.1 Experimental protocols

3.1.1 SGS

As it is visible on figure 1b, the distribution of the Swiss DEM has two populations geographically distinguishable as the Alpine and Jura regions and the Swiss plateau. However, we have decided to build a unique variogram model in order to get a coherent map, avoiding the side effects of separated analysis and modelisations.

Due to this bi-modal distribution, we had no choice but to do a normal-score transformation of the train set. This was done (as well as the back-transformation) using the GSLib [2] software package.

The omni-directional variogram of transformed data has been modelised by a spherical function with a range of 165 km and 26% of nugget effect.

The 100 simulations were performed using the GStat software (<http://www.gstat.org>), with a maximum neighbourhood of 200 points and no *a priori* mean.

3.1.2 RRCM

As for SGS experiments, only one Kernel Ridge Regression (KRR) model has been used for the whole Swiss DEM data set.

The input data have been normalised using the mean and standard deviation of the X and Y coordinates of the train set, whereas the outputs have just been divided by 4500, which is a bit more than the highest value of the data set, in order to get them lying between 0 and 1.

For performance reasons, the training procedure was performed by a series of 5 train/validation random splitting for each set of hyper-parameters. The set of hyper-parameters getting the best mean absolute error over the series was chosen for the RRCM model. In the case of Swiss DEM, we got $a = 0.01$ and $\sigma_{kernel} = 1.0$.

3.2 Results analysis

	RRCM	SGS
Mean Absolute Error of Estimation	262 m	286 m
Points Over 95% Confidence Interval	3.24%	1.6%
Points Under 95% Confidence Interval	1.68%	1.86%
Total Outside 95% Confidence Interval	4.92%	3.46%
Minimum 95% Confidence Interval width	771 m	275 m
Median 95% Confidence Interval width	1806 m	1943 m
Maximum 95% Confidence Interval width	3582 m	3037 m

Table 1: Numerical results from 95% confidence intervals estimation on Swiss digital elevation model, using Ridge Regression Confidence Machine (RRCM) and Sequential Gaussian Simulations (SGS). The “Mean Absolute Error of Estimation” is calculated by comparing the real test outputs with Kernel Ridge Regression estimations for RRCM, and with the mean of 100 simulations for SGS.

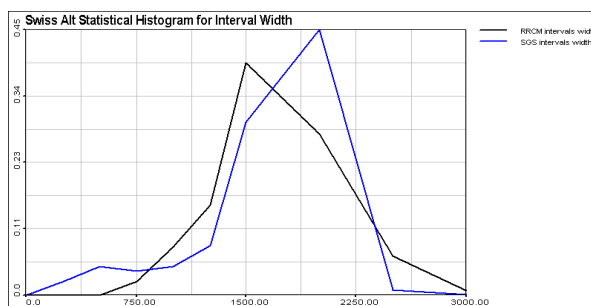


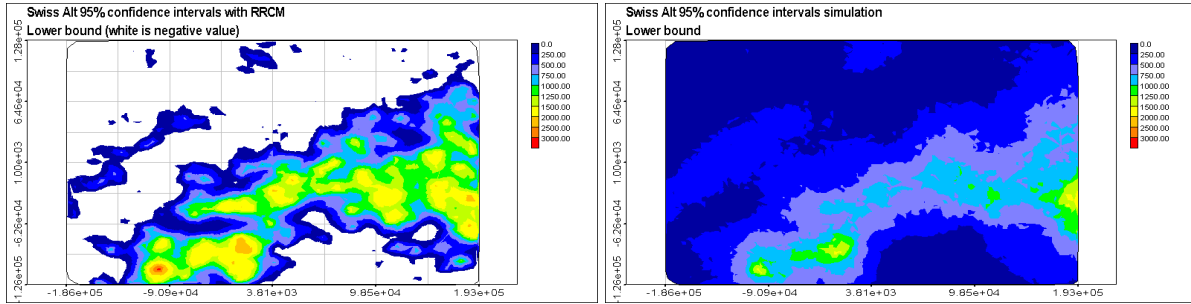
Figure 2: Statistical histograms of RRCM and SGS 95% confidence interval width

Both approaches give quite similar numerical results (table 1). First of all, in term of data prediction, mean absolute errors from the “underlying models” are very close, allowing to compare the methods.

The number of points lying outside of confidence interval is smaller for SGS than for RRCM. The consequence is that the effective confidence of the intervals found by SGS is a bit higher than required (96.5%). RRCM is very close to the demand (95.1%), but one can notice that, by opposition to SGS, the number of errors over and under the confidence interval is not symmetric.

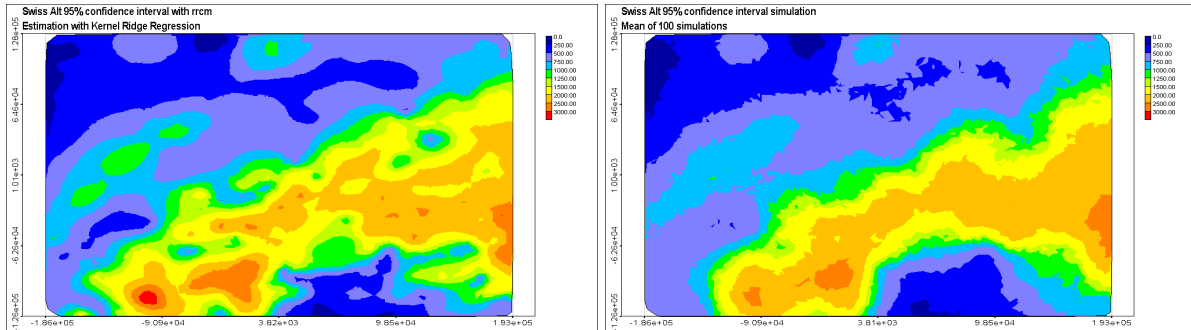
Even if confidence intervals of SGS are sometimes thinner than those of RRCM (figure 2), they are mostly similar in size for both methods, which is consistent with the similarity of their confidence values.

Comparison of visual results (figure 3) is showing more differences between SGS and RRCM confidence intervals. First, and despite a locally noisy behavior, SGS appears smoother than RRCM. The structure



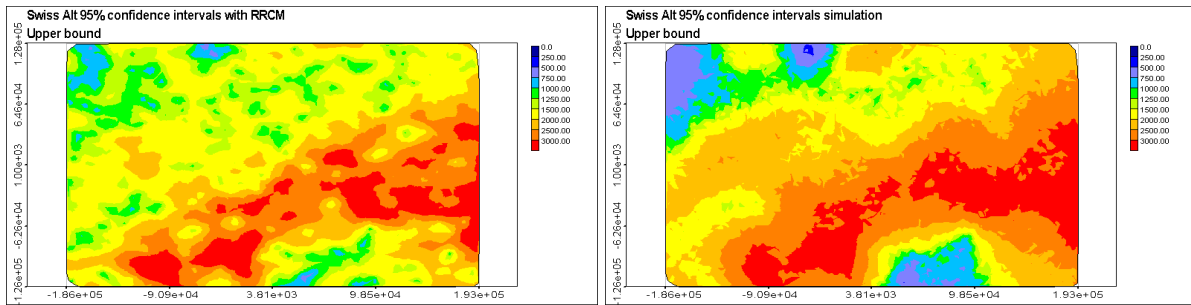
(a) RRCM lower bound

(b) SGS lower bound



(c) KRR estimation

(d) SGS mean



(e) RRCM upper bound

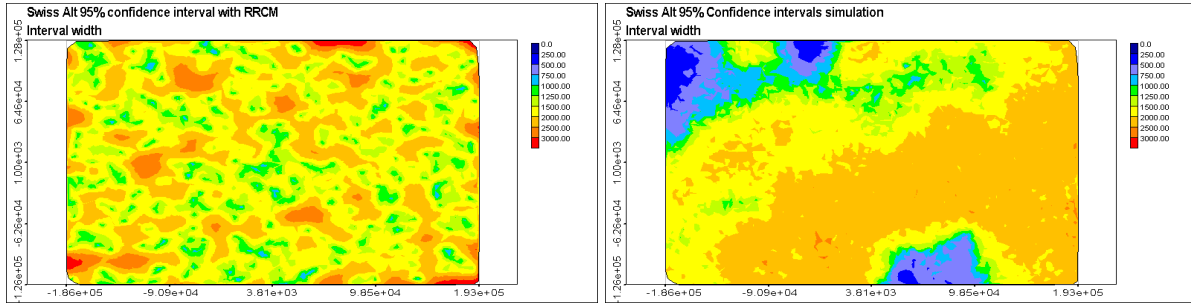
(f) SGS upper bound

Figure 3: Comparison between RRCM and SGS bounds and estimation for Swiss DEM test set

of the latter is representative of a radial basis function regression algorithm, while the smoothness of the former is a consequence of the large range of the variogram model.

One can notice also that RRCM lower bounds are sometime negative (figure 2a). It would be interesting to check if better performance can be obtained by constraining the lower bounds to positive values when data are strictly positive.

A strange result is obtain by comparing figures 4a and 4b. These images represent the spatial distri-



(a) RRCM interval width

(b) SGS interval width

Figure 4: Comparison between RRCM and SGS spatial distribution of interval width

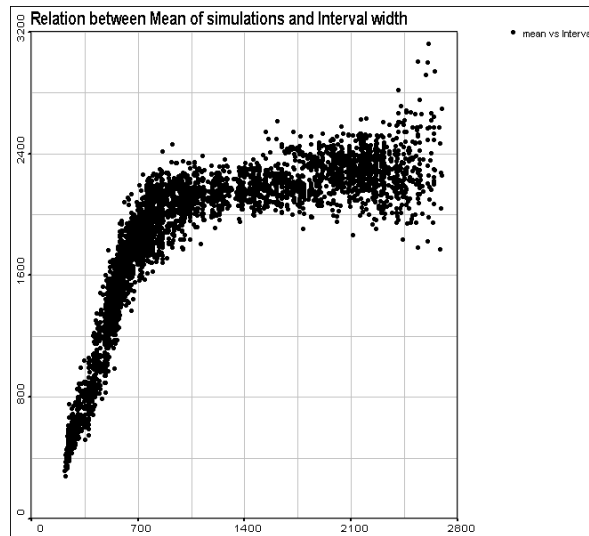
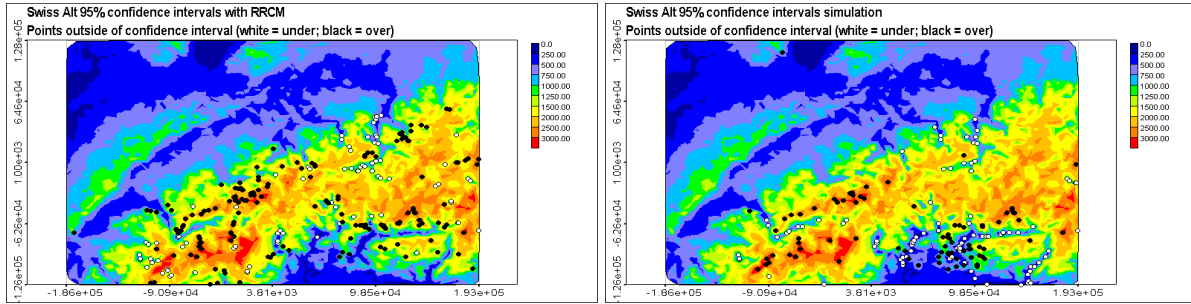


Figure 5: Scatterplot of SGS mean versus confidence interval width, showing a strong correlation for the low values

tribution of interval width. RRCM's appears to be related to the spatial repartition of the training points, and gives a useful information about the spatial representativity of the train set. But the one of SGS is strongly correlated with data output while the calculation of the bounds should only be related to the simulation variance. Figure 5 shows an obvious correlation between small simulation means and interval widths. A potential explanation is that it is due to the normal score back-transformation applied on a complex data set. Thus, one has to be careful when doing such a transformation, because the results might not be reliable.

One can also look at error locations on the map. Figure 6a and 3b shows that RRCM and SGS are making errors mainly on extreme values in high variability areas. SGS seems to have more difficulties to predict low values than RRCM, which, on the other hand, seems to be less efficient than SGS to predict high values.



(a) RRCM errors

(b) SGS errors

Figure 6: Comparison between RRCM and SGS spatial repartition of points lying outside 95% confidence intervals

4 Conclusion

When used to solve similar problems, SGS and RRCM gave very similar and good numerical performances but significantly different risk maps. One part of these differences comes from the underlying regression models used (kriging for SGS and kernel ridge regression for RRCM). But for SGS, the Gaussian hypothesis imposes a normal-score transformation and back-transformation of data if they are not yet normal. Such a transformation seems to cause artifacts in the solutions, especially because of the bimodal form of data distribution. RRCM does not need other constraint than “iidness” of data coordinates. The data set used in this paper was fulfilling this point, and thus, results were very good. But one has to remember that geostatistical data are usually not iid at all. As a consequence, further experiments are needed in order to evaluate the robustness of RRCM to a non-iid data set, and see if, like most of the geostatistical methods, it can be used even if all theoretical hypothesis are not fulfilled. Another problem that rose from the experiments conducted for this paper is the very long computation time of RRCM. This has to be mentioned as it is a big drawback when dealing with large data sets like the one of Swiss DEM.

Acknowledgments

This work was supported by Swiss National Science Foundation (CARTANN project: FN 20-63859.00), Dalle Molle Institute for Perceptual Artificial Intelligence (IDIAP), Royal Holloway University of London and University of Lausanne.

Pictures were generated with Geostat Office software. Simulations were performed with GNU GSTAT software. Normal-score transformations and back-transformations were done using the GSLIB software package.

Thanks to Samy Bengio for his review of this article.

References

- [1] C. Deutsch and A. Journel. Chapter 5: Simulation. *GSLIB - Geostatistical Software Library and User's Guide*, pages 117–195, 1992.
- [2] C. Deutsch and A. Journel. *GSLIB, Geostatistical Software Library and User's Guide*. Oxford University Press, 1992.

- [3] A. Gammerman, V. Vovk, and V. Vapnik. Learning by transduction. *Uncertainty in Artificial Intelligence*, pages 148–156, 1998.
- [4] T. Graepel, R. Herbrich, and K. Obermayer. Bayesian transduction. *NIPS*, 1999.
- [5] T. Heskes. Practical confidence and prediction intervals. *NIPS*, 1996.
- [6] I. Nouretdinov, T. Melliush, and V. Vovk. Ridge regression confidence machine. *ICML*, 2001.
- [7] C. Saunders, A. Gammerman, and V. Vovk. Ridge regression learning algorithm in dual variables. *15th International Conference on Machine Learning*, 1998.
- [8] C. Saunders, A. Gammerman, and V. Vovk. Transduction with confidence and credibility. *IJCAI*, 1999.
- [9] B. Scholkopf, C. Burges, and A. Smola. *Advances in Kernel Methods - Support Vector Learning*. MIT Press, Cambridge, 1999.