IDIAP RESEARCH REPORT

IDIAP

Martigny - Valais - Suisse

Confidence Evaluation for Risk Prediction

Nicolas Gilardi [a] [b] gilardi@idiap.ch
Tom Melluish [c] tom@dcs.rhbnc.ac.uk
Michel Maignan [a] michel.maignan@imp.unil.ch

IDIAP–RR 01-22

October 2001

published in
2001 Annual Conference of the International Association for Mathematical Geology

Dalle Molle Institute for Perceptual Artificial Intelligence • P.O.Box 592 • Martigny • Valais • Switzerland

phone +41 – 27 – 721 77 11
fax +41 – 27 – 721 77 12
e-mail secretariat@idiap.ch
internet http://www.idiap.ch

[a] Institut de Minéralogie et Géochimie, Université de Lausanne, LAUSANNE, SUISSE
[b] Institut Dalle Mole d'Intelligence Artificielle Perceptive, MARTIGNY, SUISSE
[c] Department of Computer Sciences, Royal Holloway, University of London, EGHAM, UNITED KINGDOMS

IDIAP Research Report 01-22

Confidence Evaluation for Risk Prediction

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

October 2001

published in
2001 Annual Conference of the International Association for Mathematical Geology

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 (iid). The method's confidence interval reliability is compared to conditionnal Sequential Gaussian Simulation. The robustness of this reliability against poor underlying regression models is also studied. The data set used is a digital elevation model of a part of the South-West part of Switzerland ("Valais"). Experiments to evaluate the robustness of RRCM against the iid assumption are conducted using the data set of cadmium concentration in Leman Lake sediments in 1983 ("Cd83").

1 Introduction

Maps drawn from a regression estimate of unknown data are often insufficiant 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 ([4][6][5], etc ...), but most of them need strong *a priori* knowlege 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) [7] 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 simultaneously a prediction and the confidence one has in it, either for classification [9] or regression [7].

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 $t : \mathcal{Z}^{l+1} \to [0, 1]$ which satisfy the condition

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

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 fulfiled by the following form of the typicalness function:

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

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

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

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 $f : \mathcal{X} \to \mathbb{R}$, 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 $t(z_1, \ldots, z_{l+1}) > r$. It has been shown in [7] 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 [8] as underlying regression model.
The Ridge Regression algorithm consists in calculating the value $\mathbf{w}$ that minimises the function

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

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

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

and

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

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

Using the "kernel trick", one can then replace the dot product matrix XX' in equations (5) and (6) by a *kernel*[1] *matrix* $K_{i,j} = exp\Big(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma_{kernel}{}^2}\Big)$, with σ_{kernel} a user defined positive constant, which allows to solve non-linear problems in a "linear" way [10], 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 $i = 1, \ldots, l$. It is thus possible to calculate the typicalness for $\hat{y}$ intervals rather than for unique $\hat{y}$ values, as follows:

1. for each example in $(z_1, \ldots, z_{l+1})$, 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. If necessary, rescale the inputs of the train and test sets, as well as the outputs of the train set. This rescaling can be done in order 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 0 mean and a 1 standard deviation.

[1] in this case, the gaussian radial basis function kernel

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 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.
   One Uses the hyper-parameters calculated by cross-validation, and process as described in section 2.1.2, the studied sequence being the full train set plus the test point.

4. Back transform the data.
   If data have been rescaled at step 1., it should then be transformed back to its original domain, as well as the confidence region's bounds calculated for each test point.

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 sometimes, 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 normaly distributed, and if not, to apply normal-score transformation on them [1]. Then, one calculates the variogram of the transformed data

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

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 $\gamma(\mathbf{h})$ with the continuous function $\hat{\gamma}(\mathbf{h})$. The next part is the simulation process itself.

1. Choose 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 ${\sigma_{krig}}^2$ 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. Randomly choose 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

Two procedures can be used to generate confidence intervals using Sequential Gaussian Simulations results: an non-parametric and a parametric.
The non-parametric one processes as follows: for a given simulated point, one can basically order the simulated values from the smallest to the highest. Then, if necessary, do the N-Score back-transformation on them. And finally, one finds the two values corresponding more or less to the 2.5 and 97.5 percentils of the all set of simulated values.
The parametric method uses the fact that *before* N-Score back-transformation the simulated data are normally distributed. It is thus possible to calculate the 2.5 and 97.5 percentils from the gaussian cumulative density function defined by $\mathcal{N}(\mu_{sim}, \sigma_{sim})$. The mean μ_{sim} and standard deviation σ_{sim} of the distribution of each point are calculated from the set of simulated values. When these bounds have been calculated, one is applying the N-Score back-transformation on them in order to get back to the real output space.

At the limit (i.e. for thousands of simulations), and for normally distributed data, both methods are similar. But the non-parametric procedure might be more robust against non-normally distributed data as the bounds are calculated in the "real" space, whereas the parametric one calculates the bounds in the transformed space. However, the main drawback of the non-parametric procedure is that it should use a large number of values to give reliable bounds, while the parametric one is less concerned by this aspect, thanks to the use of a theoretical cumulative density function.

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:

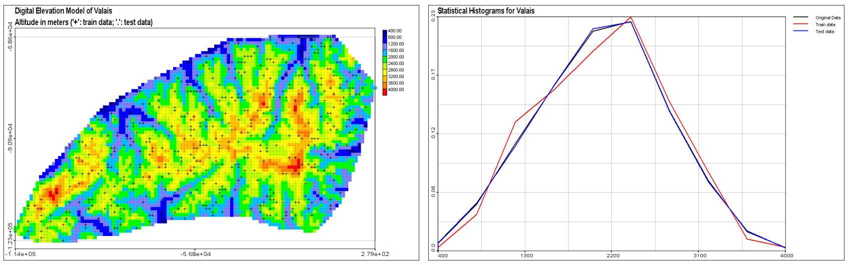
1. Normal score transformation of the train outputs if necessary.

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.

3 Experiments on digital elevation model of Valais



(a) data set (b) histograms

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

The digital elevation model of Valais used for these experiments is part of a larger digital elevation model of the whole Swiss region. It contains 4955 data points lying on a regular grid of cells of 1 kilometer aside. It covers high mountains and narrow valleys. The statisticals properties of the data set are the following:

mean	2307 m
standard deviation	668 m
minimum	466 m
median	2333 m
maximum	4469 m

In order not to spend weeks in experiments, the original data set has been splited into two data sets. The first one is a random extraction of 500 points from the full Valais set, and will be used as the train set, i.e. the known data. The second one is the rest, that is 4455 points, which constitute the test set, i.e. the data to predict.

3.1 Experimental protocols

3.1.1 SGS

As it is visible in figure 1b, the distribution of altitudes of the Valais data set is almost Gaussian. However, in order to respect SGS theory, data have been transformed using a Normal Score procedure.

The variogram model chosen is an exponential one, with no anisotropy and a range of 16.73 km. 100 simulations were performed using a maximum neighbourhood of 200 points. Sequential Gaussian Simulations were performed with GSLib [2].

3.1.2 RRCM

The first models developped using Kernel Ridge Regression on raw data gave similar results in term of mean absolute error compared to Ordinary Kriging. Thus it was not necessary to rescale data.

To speed up the choice of hyper-parameters, the training procedure was done on a series of five training/validation random splitting for each set of parameters. The hyper-parameter set was then chosen on the smallest mean absolute error on the five validation sets. For the Valais data set, the optimal hyper-parameter set was $a = 0.001$ and $\sigma_{kernel} = 201.0$.

3.2 Results analysis

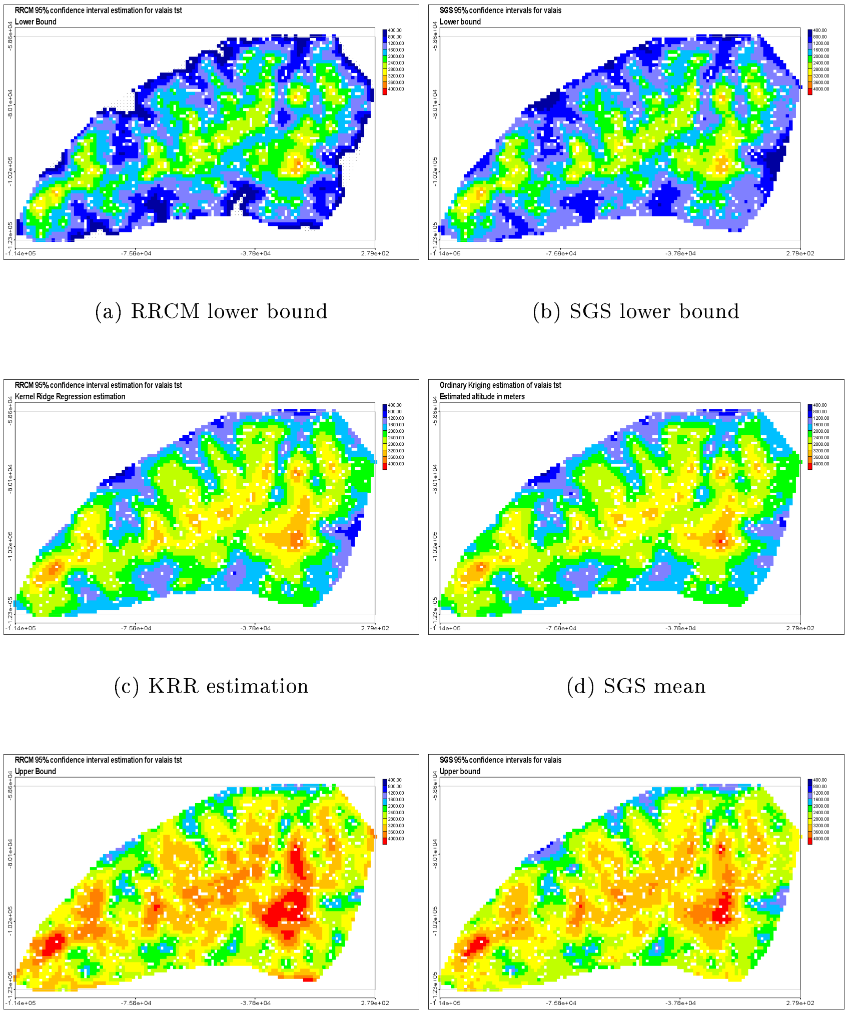
3.2.1 Confidence intervals reliability

	RRCM	SGS
Mean Absolute Error of Estimation	273 m	275 m
Points Over 95% Confidence Interval	2.1%	6.3%
Points Under 95% Confidence Interval	2.8%	6.9%
Total Outside 95% Confidence Interval	4.9%	13.2%
Minimum 95% Confidence Interval width	617 m	475 m
Median 95% Confidence Interval width	1260 m	1044 m
Maximum 95% Confidence Interval width	4188 m	2100 m

Table 1: Numerical results from 95% confidence intervals estimation on Valais data set, 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.

The evaluation of the reliability of a confidence interval is obtain as follows: for all the test set, one counts how many times the real value lies outside the 95% confidence interval. If this figure is significantly greater than 5% of the total number of test points, this means that the method is over-confident, giving too narrow confidence intervals. If it is significantly smaller than 5%, this means that the method is under-confident, giving too large confidence intervals. Finally, if the number of test points lying outside of the confidence interval is close to 5%, this means that the confidence interval given by the method is reliable.

In the present case, if both methods give similar performances in term of estimation error, the reliability of their 95% confidence intervals are very different. As shown in table 1, RRCM gives a very reliable confidence interval, even if sometimes, the size of the interval is very large, showing a high uncertainty in some regions. SGS' confidence intervals are more narrow than RRCM's. But the method is significantly over-confident as it makes more than 13% of error where it has predicted to do at most 5%.



(a) RRCM lower bound (b) SGS lower bound

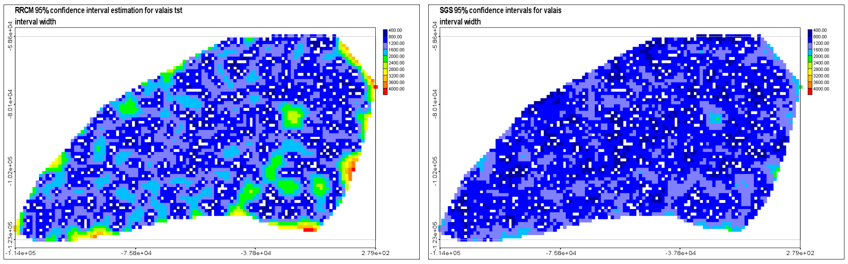
(c) KRR estimation (d) SGS mean

(e) RRCM upper bound (f) SGS upper bound

Figure 2: Comparison between RRCM and SGS bounds and estimation for Valais test set

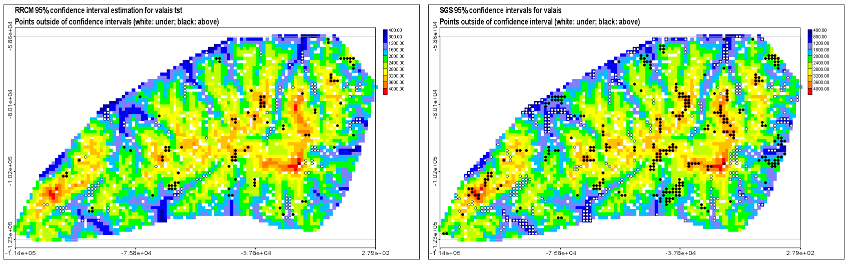
Figures 2 and 3 give another representation of the phenomenon. Figures 2c and 2d are almost identical, showing that both methods have the same regression prediction efficiency. But if the global patterns are similar, lower bounds (figures 2a and 2b) and upper bounds (figures 2e and 2f) don't reach the same levels. RRCM's bounds usually go lower and upper than SGS', and this is even more obvious in figures 3a and 3b. These pictures show also that RRCM's intervals are getting strongly wider near the borders of the data area, while SGS' are less influenced by this "border effect".

Figures 4a and 4b show the locations of the points lying outside of confidence intervals. One can see that these locations are not exactly similar for both methods. RRCM seems to make most of its errors on sharp altitude variation areas, while SGS has problems with extreme values.



(a) RRCM interval width (b) SGS interval width

Figure 3: Comparison between RRCM and SGS spatial distribution of interval width



(a) RRCM errors (b) SGS errors

Figure 4: Comparison between RRCM and SGS spatial repartition of points lying outside 95% confidence intervals

Given these results from the 95% confidence interval evaluation, RRCM seems to give very reliable confidence intervals, even if it appeared to be subject to border effects. In figure 5, one can see that the same performance are obtained for other confidence intervals.

In general, RRCM confidence interval reliability outperform the one of SGS. It is important to notice that such a difference of performances between the two methods wasn't expected. First, the distribution of values and the spatial correlation seemed to be quite simple. But also, both methods gave very similar results in term of regression estimation. However, one can suppose that the Gaussian hypothesis of SGS wasn't verified in this case. Thus, the normal-score transformation and back transformation might have disturbed the real data distribution. The fact that most of the SGS errors are extreme values is enforcing this idea: it means either that the simulation process failed to generate enough extreme values, due to a bad representation of the data distribution, or that the

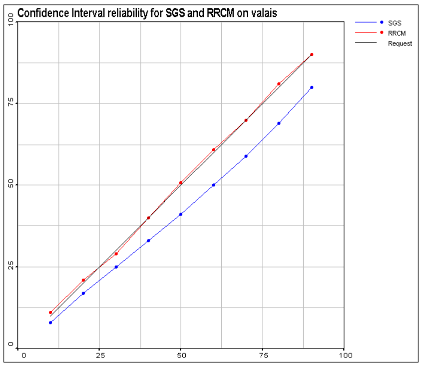


Figure 5: Comparison between RRCM and SGS spatial repartition of points lying outside 95% confidence intervals. X is requested confidence value, Y is actual confidence value of the methods. The plain line represents the optimal prediction result.

back-transformation destroyed the extreme values. In any case, the absence of asumption of RRCM is a great advantage to study this data set.

3.2.2 Robustness of RRCM to poor regression model

As it was shown in section 2.1, RRCM has a strong relationship with the underlying Kernel Ridge Regression model. This model is the “reference” that will be used to evaluate the confidence of a prediction.

It is thus important to check the consequences of using a non-optimal model. This can happen, for example, if the training procedure was not optimal. Please note that in the following experiments, the training set is a fairly good representation of the whole data set. This means that the models studied will have similar prediction performance on the training and the testing set. This is an important point: these experiments try to describe the robustness of RRCM in case of a poor *regression model*, not in case of a non-representative *training set*.

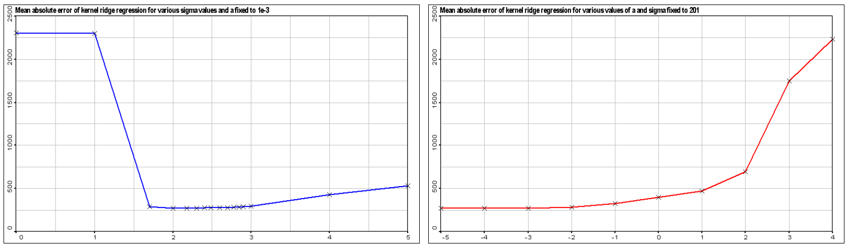
The robustness calculations were conducted using the same training set as for the other experiments, but with a smaller testing set (only 200 points) in order to speed-up the computations. Two confidence intervals were estimated: the 10% and the 90%. They were computed using various underlying models, built by fixing one of the hyperparameters to it's optimal value[2], σ_{kernel} or a, and varying the other one.

Varying kernel ridge regression hyper-parameters can lead to very different performances, as shown in figure 6. As a consequence, one can easily imagine an influence on reliability of RRCM's confidence intervals. Figures 7a and 7b Shows that this influence exist, but that it is quite small: the maximum distance from a real confidence interval and the request is around 6.5%. This is less than the error made by SGS on the 95% confidence interval.

This good performance is a direct consequence of the transductive approach of RRCM. In section 2.1, we have seen that the typicalness is a characteristic of a sequence and not of a single value. Though, if the underlying model is bad, one can expect that it will be “as bad” for every points, and then, even if their strangeness will be high, there will remain some structure inside the comparison of the strangenesses. And this structure is measured by the typicalness.

These good results in term of reliability do not mean that the quality of the underlying model has no importance for RRCM. Figures 8a and 8b demonstrate this: the size of the confidence intervals (and thus their interpretability) is directly correlated to the regression performance of the underlying model. Moreover, for bad models, all the intervals tend to have the same size, destroying the knowledge about information density in the data set.

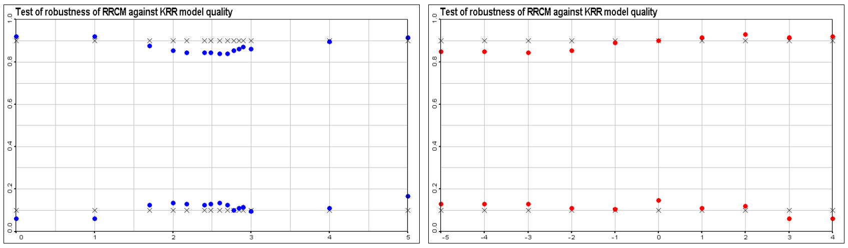
[2] “optimal” with respect to the training procedure performed in section 3.1.2



(a) KRR MAE for a fixed and various σ_{kernel}

(b) KRR MAE for σ_{kernel} fixed and various a

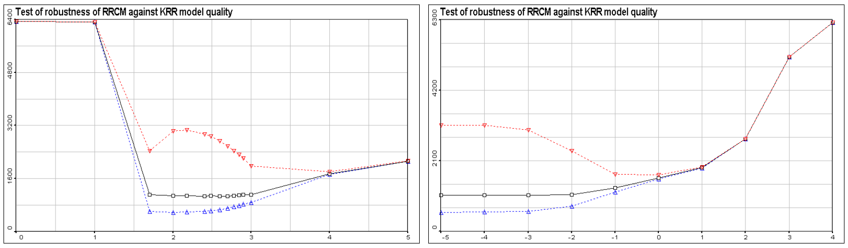
Figure 6: Mean absolute error on test set for various kernel ridge regression models. Black crosses' abscissa correspond to the decimal logarithm of the values of the non-fixed hyper-parameter



(a) Confidence intervals for various σ_{kernel}

(b) Confidence intervals for various a

Figure 7: RRCM's 10% and 90% confidence intervals predictions for various underlying models. X is the decimal logarithm of the varying hyper-parameter (a and σ_{kernel}), and Y is the confidence value of the interval. Crosses are the expected confidence (0.1 and 0.9) for the given hyper-parameters. Circles are the actual confidence of the estimated intervals.



(a) Confidence intervals width for various σ_{kernel}

(b) Confidence intervals width for various a

Figure 8: Minimum, median and maximum widths of RRCM 90% confidence intervals for various underlying models. Blue vertex-up triangles are the minimum widths, black squares are the median widths, and red vertex-down triangles are the maximum widths

The robustness of RRCM confidence interval reliability against poor underlying model is thus a reality. The transductive approach is insuring it, with the price of a lower interpretability of the intervals themselves.

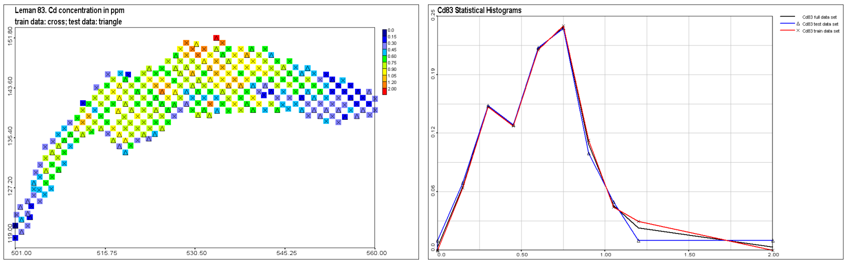
This is at least true for the data studied, which is a pretty nice one with a lot of training points, a short range correlation and good spatial distribution. Unfortunately, this is quite unusual for real geostatistical applications.

3.2.3 Robustness of RRCM to non-iid data

The RRCM assumption that data are independently distributed is often wrong when dealing with geostatistical data sets, either because sampling is made on a regular grid or because it is very dense in some areas and very sparse in other. It is anyway a widely accepted fact that sometimes, a methodology is based on hypothesis that are almost impossible to fullfil (from the mathematical point of view), and however it works quite well anyway. The aim of the following experiments is to test the efficiency of RRCM when it is used to study a non-iid data set.

The data set considered is part of a series of physico-chemical analysis of the Leman Lake sediments conducted by the CIPEL[3] in 1983. In the present case, only the Cadmium concentration is taken into account.

The data set (figure 9) contains 293 points and was separated into two subset: 150 points for the training set and 143 for the test set.



(a) Cd83 data set

(b) Cd83 histogram

Figure 9: Cd83 data set. Crosses indicate train data. Triangles indicate test data.

The same experimental protocol as described in section 3 was used for this study, and like in section 3.2.1, we have computed confidence intervals given by RRCM and SGS for various requests. For some extreme confidence values (20%, 30% and 90%), results of RRCM (figure 10) are worse than for the valais data set. However, they are still equal or better than SGS' ones.

In this case, it is quite difficult to say wether the lower performances are the consequence of the non-independence of input data or to the small and noisy data set. The second hypothesis might be the right one because a large part of the requested confidences are perfectly reproduced, except for some extreme values. One can suppose that if the problem was related to the spatial repartition of data, there will be a similar lack of quality for any confidence request. On the other hand, if the noise in the data has an influence on errors, it should be more obvious on the extreme requests, where there are fewer points respectively inside or outside of the confidence intervals. More experiments would be necessary to give a definitive answer to this question.

[3]Commission internationale pour le Protection des Eaux du Leman: http://www.cipel.org

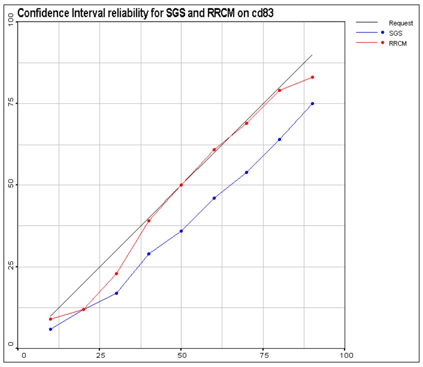


Figure 10: Cd83 confidence intervals reliability of RRCM and SGS for various confidence requests.

4 Conclusion

Based only on the assumption that input data are independently and identically distributed, ridge regression confidence machine proved to be a very efficient algorithm for confidence interval estimation. The reliability of the confidence appeared to be very good and robust to poor underlying regression models. And despite some border effects, with a good underlying model, the interval width gives a interesting advice about information repartition of a data set.

RRCM even outperformed the well known Sequential Gaussian Simulations on the two data sets studied in this paper. The most probable reason for the bad performances of SGS might be that the Gaussian assumption on data distribution was wrong on the Valais and Cd83 data sets, problem that does not affect RRCM.

However, RRCM has some drawbacks. First, it is a very new method. It thus needs a lot of experiments to test its behaviour facing complex data sets that can be found in the geostatistical field. The first test conducted on Cd83 data set was quite good, but the interpretation of the results is not easy. But the biggest problem of this method is that it needs as many matrix inversions as test data, the size of the matrix being the number of training points. As a consequence, it is limited to a few hundreds of training points, and testing a large test set can be very time consuming. For example, the computations for the 95% confidence intervals of the valais data set (500 training points and 4455 testing points) took about 21 hours on a Sun Blade 100.

Anyway, the performances given by this method are really amazing, and I think it is worth being used, especially as it is free from any distribution assumptions.

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 and GSLib software package. Normal-score transformations and back-transformations were done using the GSLib software package.

Thanks to Mikhail Kanevski for his help on simulation calculations, and 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] Pierre Goovaerts and Marc Van Meirvenne. Accounting for measurement and interpolation errors in soil contaminant mapping and decision-making. *Conference of the International Association for Mathematical Geology*, 2001.

[5] T. Graepel, R. Herbrich, and K. Obermayer. Bayesian transduction. *NIPS*, 1999.

[6] T. Heskes. Practical confidence and prediction intervals. *NIPS*, 1996.

[7] I. Nouretdinov, T. Melluish, and V. Vovk. Ridge regression confidence machine. *ICML*, 2001.

[8] C. Saunders, A. Gammerman, and V. Vovk. Ridge regression learning algortihm in dual variables. *15th International Conference on Machine Learning*, 1998.

[9] C. Saunders, A. Gammerman, and V. Vovk. Transduction with confidence and credibility. *IJCAI*, 1999.

[10] B. Scholkopf, C. Burges, and A. Smola. *Advances in Kernel Methods - Support Vector Learning*. MIT Press, Cambridge, 1999.