Locally weighted support vector regression for spatial predictions

Camilla Z. da Silva and Jeff Boisvert

Machine learning algorithms have been increasingly applied to geostatistical framework, benefitting from the fact that these techniques are capable of capturing complex features directly from the data without being explicitly programed to so. Machine learning often achieves this goal by minimizing a global risk function. However, when the input data is unevenly distributed over the input space, a global criterion may not be adequate to sub regions of the input space, compromising the generalization capability of the algorithm. A different paradigm lies in locally weighted learning, that aims to fit models on patches of the input space based on the premise that nearby information is a better indicator of the system than the training set as whole. This paper aims at evaluating the applicability of a locally weighted learning to environmental data

Introduction

Evaluation of mineral resources is linked to geostatistical methods which describe spatially continuous phenomena using samples collected over an area of interest. The goal is to obtain the best estimate possible at an unsampled location. This is done by quantifying and modeling the spatial patterns observed on the data that arise from the numerous geological process during the deposition process (Rossi and Deutsch, 2014). Geostatistics provides the tools to model and describe these patterns, nonetheless, a common assumption is that the variables under analysis are within a stationary domain. That is, that the locations inside the domain and the variables belong to the same statistical population (Rossi and Deutsch, 2014). Stationarity is critical for the appropriateness of geostatistical methods, pooling different domains together can mask important features of the variable of interest. However, defining stationary domains is not straightforward due to the complexity of the geological process.

On the other hand, machine learning (ML) is a vast set of techniques based on the premise that if the data set is representative of the targeting problem, it can be used to exploit meaningful relationships directly from the data without being explicitly programmed to so, and without the assumption of stationarity. The use of ML has been increasingly applied on the geostatistical framework (Dowd and Saraç, 1994; Kapagerdis, 1999; Tahmasebi and Herzakahni, 2012; Dai et al., 2014; Gangappa et al., 2017, Maniar et al., 2018, Tomislav et al., 2018, Samson and Deutsch, 2018; Samson and Deutsch, 2019 Walch et al., 2019), however it does not guarantee data reproduction as does ordinary kriging and does not explicitly account for spatial correlations, an important feature observed on geological data sets. Samson (2019) has shown the ML models can be incorporated into a hybrid geostatistical framework, where the relationships captured through ML are implemented as an auxiliary information on the estimation process.

Most of ML models are built on the whole training set available and then, the fit model is used to predict new instances. But, when data is not evenly distributed in the input space, the function fit globally, with the entire training set, could have the generalization capability compromised. That is because, in the machine learning context, models are built to extract general properties of the data and not specificities of individual training points. This characteristic arises from the fact ML models are built to minimize the global error, which may not be appropriate for certain regions of the data set. Hence, global fitting sometimes affects negatively the generalization capability of the model (Galvan et.al, 2011). An alternative lies in local learning (Atkeson et al.,1997) or *lazy learning* as it is also referred in the literature.

Local learning defers the process of training until an instance needs to be predicted, and it is done by considering solely relevant data to that particular instance. Relevance usually is defined by a specific metric. In geological data the spatial patterns are important features that demonstrate that samples close in space are more similar than samples further away. Moreover, rarely geological data is evenly distributed in the input space. Due to these common features of environmental attributes, global ML models may end up to be suboptimal to the problem at hand. Thus, the approach of local learning will be explored.

Background theory

Local learning

Local learning algorithms attempt to adjust the training system to the specific properties of a region of the input space (Bottou and Vapnik, 1992). The procedure mounts to breaking the global complex problem into several smaller simpler problems throughout the space. Yet, it is counter intuitive. Instead of finding a sole fit, the process is performed several times. Instead of using the maximum data available, it only considers the training data in the specific subregion. This may seem slow and inefficient. Be that as it may, Bottou and Vapnik, (1922) have shown that often global models can benefit from the locality component.

Locally weighted learning

A locally weighted algorithm not only considers a sub region of the input space and the sub set of training points inside the vicinity of the region, but it also treats the data samples differently, according to a metric of relevance. A common way to measure similarity is through the Euclidean distance. One can emphasize relevance either by assigning different weights to the sample or by weighting the training criteria (Atkenson, 1997). For example, a nonlinear global model generally aims at minimizing the following cost function:

$$\sum_i L(f(\mathbf{x}_i,\theta),y_i)$$

Where y_i is the target value, $\mathbf{x}_i$ is the features input vector, θ is the model parameter vector and L is a general loss function. Weighting the cost function leads to forcing the model to fit well nearby data while being less concerned about the fit of distant points. The above cost function becomes:

$$\sum_i L(f(\mathbf{x}_i,\theta),y_i)W(d)$$

Where $W(d)$ is a weight function dependent on the distance of the training data to the unsampled location. This approach mounts to a local model that is adjusted to each subregion.

Weighting functions

According to Atkenson et al. (1997) and Ellatar et al., (2010), the weighting function should be one that has its maximum value when the distance between the unsampled location and the data point is zero and decay as distance increases. One common weight function is the inverse distance function.

$$W(d) = \frac{1}{d^p}$$

Where p the power that defines how rapidly the weight function decays. Another weight function is the gaussian given by

$$W(d) = e^{-\frac{d^2}{2\sigma}}$$

Where σ is a smoothing parameter which controls the range of the generalization. It can be set globally through an optimization process or based on the local information (Ellatar et al.,2010). Also, there is the tricube weighting function, that has a finite extent of 1.

$$W(d) = \begin{cases}(1-|d^3|)^3 \; if \; |d| < 1 \\ 0 \; otherwise\end{cases}$$

Other weighting functions can be tailored specifically to the problem.

Support vector regression

Support vector machines (Vapnik, 1995;1998) is a ML algorithm based on statistical learning theory, in which the goal is to minimize the structural risk function, in contrast to neural networks that aim in minimizing the empirical risk function (Ellatar et al., 2010). This particular concept mounts to minimizing the upper bound of the generalization error other than minimizing the training error. Generally, support vector machines are applied to classification problems, nonetheless, the formalism can be extended to regression problems through the introduction of Vapnik's ε-intensive region around the fitted function (Awad and Khanna, 2015).

The goal is to obtain $f(x)$ that has a maximum of ε deviation from the actual observed target values while retaining the regression coefficients, w, as small as possible. Considering a training dataset x_n with N observations and response values y_n, the aim is to find the linear function:

$$f(x) = \langle w, x\rangle + b$$

In this context, small w values are obtained by means of minimizing the Euclidean norm $\frac{1}{2}\|w\|^2$. Formally, it can be stated as a convex optimization problem:

$$\min \frac{1}{2}\|w\|^2$$

Subject to the constraints:

$$\begin{cases} y_i - \langle w, x\rangle - b \leq \varepsilon \\ \langle w, x\rangle + b - y_i \leq \varepsilon \end{cases}$$

The optimization presented is feasible when exists a function f that approximated the pair (x, y) with ε precision. However, there is not always such function. When the solution is not feasible some errors are allowed by means of slack variables, ξ_n and ξ_n^*, while ensuring that it is as flat as possible. For that, the following expression must be minimized:

$$\frac{1}{2}\|w\|^2 + C\sum_{n=1}^{N}(\xi_n + \xi_n^*)$$

Subject to the constraints:

$$\begin{cases} \forall n\colon y_n - \langle w, x\rangle \leq \varepsilon + \xi_n \\ \forall n\colon \langle w, x\rangle - y_n \leq \varepsilon + \xi_n^* \\ \qquad \forall n\colon \xi_n^* \geq 0 \\ \qquad \forall n\colon \xi_n \geq 0 \end{cases}$$

The constant C is a positive constant that controls the penalty imposed on values that lie outside the ε-intensive region, that is, a regularization factor. ξ_n and ξ_n^* allow error to exist, analogue to the soft margin concept on the support vector machine formulation. To obtain the solution for nonlinear problems Lagrange multipliers are introduced, which leads to the following optimization problem:

$$L(\alpha) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i - \alpha_i^*)\,\langle x_{i,}x_j\rangle + \varepsilon\sum_{i=1}^{N}(\alpha_i + \alpha_i^*) + \sum_{i=1}^{N} y_i(\alpha_i^* - \alpha_i)$$

Subject to the constraints:

$$\begin{cases} \displaystyle\sum_{n=1}^{N}(\alpha_n - \alpha_n^*) = 0 \\ \forall n\colon 0 \leq \alpha_n \leq C \\ \quad 0 \geq \alpha_n^* \geq C \end{cases}$$

The function to predict new values is described as:

$$f(x) = \sum_{n=1}^{N}(\alpha_n - \alpha_n^*)\langle x_n, x\rangle + b$$

And the conditions Karush-Khun-Tucker are necessary to obtain optimal solutions.

$$\begin{cases} \forall\, n\colon \alpha_n(\varepsilon + \xi_n^* + y_n - \langle w, x_n\rangle - b) = 0 \\ \forall\, n\colon \alpha_n(\varepsilon + \xi_n - y_n + \langle w, x_n\rangle + b) = 0 \\ \forall\, n\colon \xi_n(C - \alpha_n) = 0 \\ \forall\, n\colon \xi_n^*(C - \alpha_n^*) = 0 \end{cases}$$

However, some solutions are not obtained with a linear function. This is overcome by replacing the dot product $\langle x_{i,} x_j\rangle$ with a non-linear kernel function $G(x_i, x_j) = \langle \varphi(x_i) | \varphi(x_j)\rangle$ which maps the data into a high dimensional space. Hence:

$$L(\alpha) = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)G(x_i, x_j) + \varepsilon\sum_{i=1}^{N}(\alpha_i - \alpha_i^*) - \sum_{i=1}^{N} y_i(\alpha_i + \alpha_i^*)$$

And the function to predict new values is:

$$f(x) = \sum_{n=1}^{N}(\alpha_n - \alpha_n^*)G(x_{n,} x) + b$$

Commonly used kernels on support vector regression

An important aspect of SVR is the mapping of the data into high dimensional spaces, which is efficiently achieved through kernel functions (Smola and Scholkpf, 2003). Three kernel types are widely applied on support vector regression, the linear kernel, the polynomial kernel and the radial basis function kernel. The radial basis function (RBF) kernel is very popular due to its good learning ability (Smola and Scholkpf, 2003) and is described as follows:

$$e^{-\gamma\|x-x\|^2}$$

Most machine learning packages for support vector regression come with built in different kernels such as linear, polynomial and gaussian.

Influence of the support vector regression parameters on the final model

The key to SVR is the tuning of the parameters that immediately affect model performance. The process involves particularly the choice of the kernel function, which must be adequate to the data characteristics (Wang and Xu, 2017), selecting the relevant parameter for the kernel function, as well as setting the parameter C and ε from SVR cost function.

Parameter ε

As mentioned in the previous sections in the SVR algorithm the aim is to minimize a structural risk function, adopting an ε-intensive loss function. The process penalizes predictions that fall outside the ε-region. The value of ε defines the width of the region, therefore, smaller values for ε mean narrower regions, with less tolerance for errors, while larger values of ε mean higher tolerance for errors. Several loss functions can be adopted on the support vector regression, including linear, quadratic and Huber (Awad and Kahnna, 2015). Such functions are convex to ensure that the problem has a unique solution. The choice of loss function affects the final model, for example, the Huber function applies higher penalties than the linear loss function as error increases. The choice of loss function depend on noise affecting the data set. Figure 1 shows the linear, quadratic and Huber loss functions.

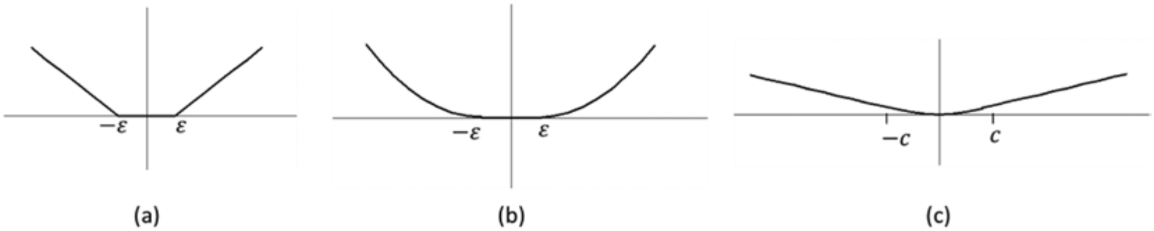

Figure 1: Loss function types: (a) linear, (b) quadratic, (c) Huber (Awad and Kahnna, 2015)

Parameter C

The performance of the model is directly affected by the parameter C which represents the penalty imposed on data that lie outside the range of error defined. That is, if the C value is large a higher penalty will be applied on prediction errors. If C is lower, than the penalty imposed is lower. This results in models that are more or less complex according to the value set for C. If the value is too small, the model is too simple and does not capture data complex features adequately. On the other hand, if C is too large, then the model is too complex and will not generalize well on new predictions. Figure 2 shows different interpolations obtained with different C values.

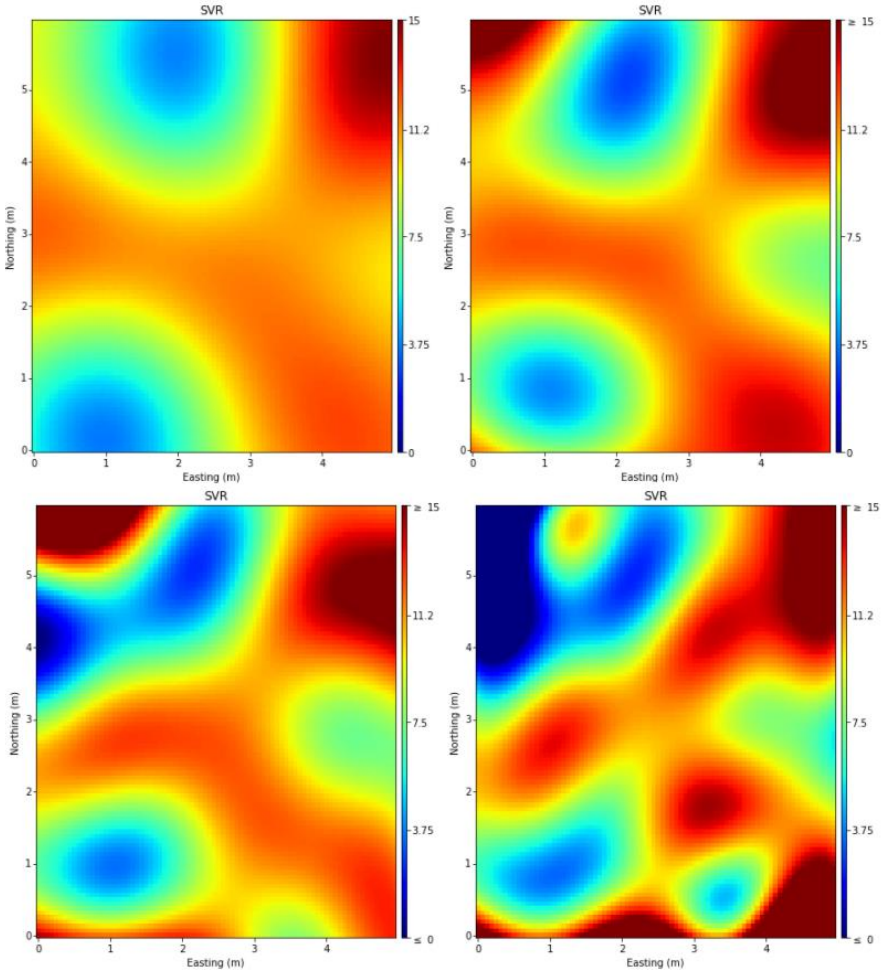

Figure 2: Influence of C on the SVR model. Top right: $C = 1$; top left:$C = 10$; bottom right $C = 100$; bottom left: $C = 1000$

It is seen from Figure 2 that the patterns captured from the data are more complex as the value of C increases.

RBF Kernel parameter: gamma

The parameter gamma is discussed on the present paper, given that the proposed algorithm uses the RBF kernel. Intuitively, the gamma controls how rapidly the RBF decays. It can be interpreted as how far a training examples acts, being that low values of gamma mean that the training example reaches far, and large values of gamma mean that the training example acts only close to the example itself. The model is sensitive to the choice of gamma. At limit, If gamma is too large, the radius of influence will only include the training data itself. On the other hand, if gamma is too small, then the model will present linear behavior. Figure 3 presents the interpolated models obtained with different values of the gamma parameter.

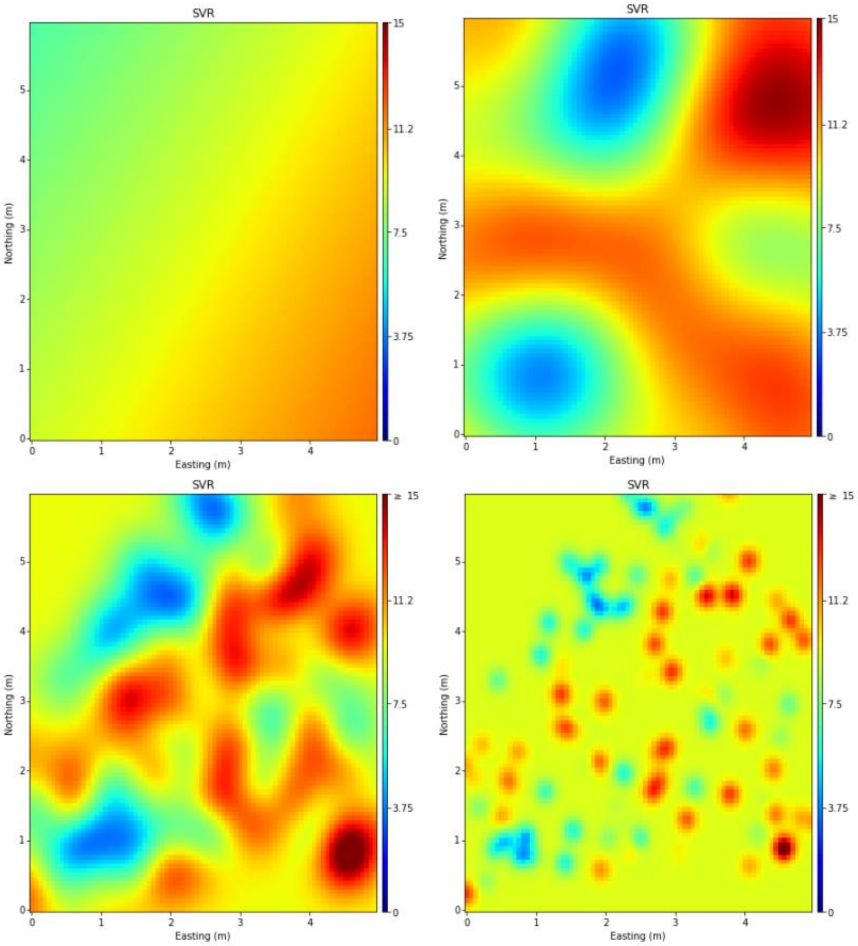

Figure 3: Influence of gamma on the SVR model. Top left: gamma=0.01; Top right: gamma=1; bottom left: gamma=10; bottom right: gamma = 100

It is clear from Figure 3 that if the gamma parameter is to high the generalization capability of the algorithm is compromised. For gamma = 100 it seen that between samples, the interpolation is simply the average from the data set, therefore, the prediction performance on new instances is very poor.

Generally, the choice of kernel function and parameters on SVR are obtained through grid search cross validation and evolutionary methods. According to Wang and Xu (2017), these optimization methods are not efficient due to the exhaustive search for optimal parameter and the possibility of the evolutionary algorithm fall into a local optimization, leading to suboptimal solutions.

Locally weighted Support Vector Regression (LWSVR)

Based on the principals of local models and specific weighting for each training point, Ellatar et al. (2010) proposed an algorithm in which the SVR risk function is modified to account for data relevance. As demonstrated in the previous section C is a fixed regularization parameter, defined *a priori* by the user, commonly chosen through grid search. If C is considered a constant value, it means that every data point in the training set contributes to the function to the same extent. However, the algorithm proposed by Ellatar et al., (2010) considers C as a function of the distance between the training point and the estimation location, so that when training data is close to the prediction location, more accurate must be the model, in other words, the algorithm uses the concept of locally weighted learning. So, the modified risk function is formulated as follows:

$$\frac{1}{2}\|w\|^2 + C_i \sum_{i=1}^{N}(\xi_i + \xi_i^*)$$

and

$$C_i = W_i C$$

W_i is the weight calculated for each training data. Replacing the constant C in the previous formulation the constraints become:

$$\begin{cases} \sum_{i=1}^{N}(\alpha_i - \alpha_i^*) \\ 0 \leq \alpha_i, \alpha_i^* \leq C_i \end{cases}$$

From locally weighted learning literature, there are a number of weighting functions that can be applied. The algorithm proposes the gaussian weighting function as:

$$W(d) = e^{-\left(\frac{d_E}{2\sigma}\right)^2}$$

Where d_E is the Euclidean distance between the training data and the estimation point, σ is the smoothing parameter, that controls the range over the generalization is performed. As mentioned in the previous section, the smoothing can be defined as a fixed value prior to the estimation process, or it could be also defined as local parameter. Ellatar et al. (2010) define the smoothing parameter as a local function, based on the Mahalanobis distance between the data points and the estimation location, since the Mahalanobis distance anchored on correlation between variables it avoids any problems related scale (Ellatar et al.,2010). In this paper the bandwidth is defined as a parameter of the neighborhood and is defined as follows:

$$\sigma = \left(\frac{d_{E_{min}}(d_{E_{max}} - d_{E_{mean}})}{d_{E_{mean}}(d_{E_{max}} - d_{E_{min}})}\right)^2 + 1$$

$d_{E_{min}}$ is the minimum Euclidean distance inside the neighborhood, $d_{E_{max}}$ is the maximum value for the Euclidean distance in the neighborhood, $d_{E_{mean}}$ is the average Euclidean distance inside the neighborhood. As the Euclidean distance of the nearest neighbor inside the region increases the larger will be the smoothing parameter leading larger radius of generalization. The unit value on the smoothing parameter equation avoids that weights become too small causing instability on the SVR algorithm.
New predictions obtained through LWSVR are generated considering the sample data that fall inside a search radius defined by the user as starting point. Given the size of the neighborhood, the algorithm will follow 4 steps:

- Considering all the points that fall inside the search radius the smoothing parameter is calculated for the neighborhood;
- For each point inside the neighborhood, the algorithm calculates the C value applied on the support vector regression model;
- With the SVR models, predict the value for the estimation location;
- Finally, the predictions are combined using the previously defined weighting function.

The computational time is directly linked to the number of data samples that fall inside the neighborhood. So, it is possible to limit the number of neighbors used inside each neighborhood, however, this feature must be considered with care given that it limits the amount of information used. The minimum number of samples is also defined by the user, although it is recommended to be no less than two samples per neighborhood in a two-dimensional case study. Also, the gamma parameter at this point in this research is kept as a fixed value of 0.1.

Locally weighted support vector regression and support vector regression: a comparison

The proposed algorithm is tested on the Jura data set (Goovaerts, 1997), which consists of 259 samples collected by the Swiss Federal Institute of technology at Lausanne. It consists of concentrations of seven heavy metals, from which the variable Cobalt is selected. The sample spacing is approximately 250m. The statistics for the Cobalt are presented in Table 1:

Table 1: Cobalt statistical summary

Count	Mean	Std	Minimum	Maximum	Median
259	9.30	3.57	1.55	17.72	9.76

Statistically the data set is fairly well behaved. The distribution is nearly symmetric, given that the median value is close to the population mean. Also, outliers are not detected on the data set. The location map presented in Figure 4 illustrates the distribution over the area of interest.

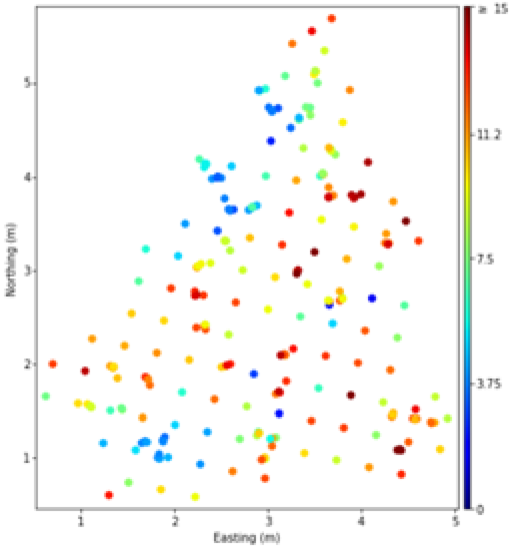

Figure 4: Location map of Cobalt samples throughout the area of interest

It is seen from Figure 4 that data is collected irregularly, however, it does not show significant data clustering. Also, on central portion of the map it is seen a majority of medium to high concentrations of Cobalt while on the south-west and north-west portions there are a number of samples with significantly lower concentrations (below the mean value).

Both algorithms, LWSVR and SVR will be applied. The parameter setting for SVR is the defined by grid search, which pointed the optimal kernel as the radial basis function, with $C = 10$ and gamma = 0.9. For the LWSVR the parameter set is search radius, defined as 750m; and the type of kernel, which is set as a radial basis function.

Model performance

The model performance is assed by a 5-fold cross validation, visual validation and local mean trend reproduction on the validation sets.

Table 2: Mean squared error and R^2 value obtained on the validation set of the 5-fold cross validation

Fold	**SVR MSER**	**LWSVR MSER**	**SVR ρ with validation samples**	**LWSVR ρ with data samples**
#1	13.55	8.18	0.28	0.63
#2	18.72	7.42	0.14	0.69
#3	11.65	4.99	0.4	0.8
#4	13.31	6.35	0.37	0.69
#5	11.75	8.92	0.43	0.67
Average	**13.79**	**7.12**	-----	------

It is seen on Table 2 that LWSVR systematically performed better on the data set than the SVR algorithm. Also, LWSVR obtained higher correlation to the validation set samples than SVR. Nonetheless, it is clear that the use of grid search to determine the parameters of SVR led to a suboptimal model, with less generalization capability than the locally weighted model. Even with a fairly simple data set the sample sparsity and irregularity affect the SVR prediction. It is clear from Figure 5 that the SVR captured general patterns of the data, however it failed to reproduce local specificities of the data. The LWSVR reduced the MSER on the final model in 51% relative to the 5-fold average value.

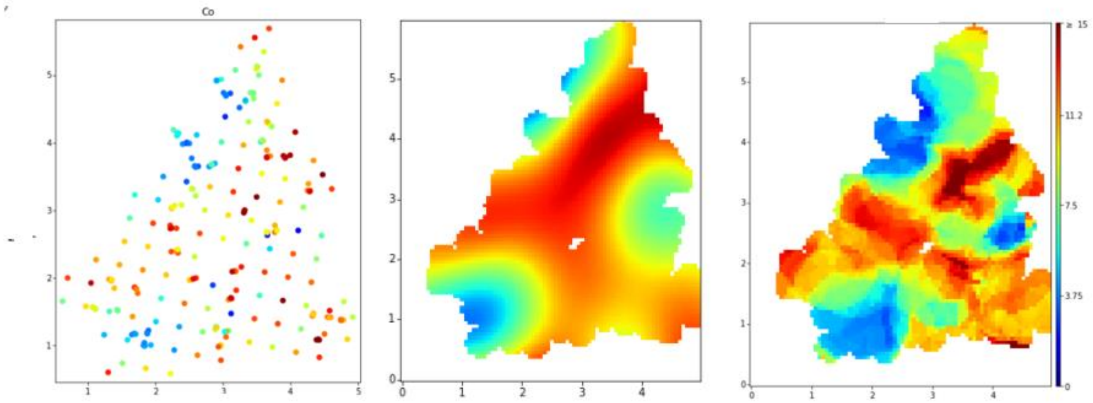

Figure 5: Left- sample data location; Center - interpolated model from training set number 1 through SVR; Right – interpolated model obtained through LWSVR

On figure 5 it shown the sample locations and the interpolated models obtained with use of the training set of the first fold. It is seen, that general behaviors of Cobalt concentrations throughout the area are captured by the SVR model. On the south-west and north-west portion, it seen a decrease of the interpolated values as observed on the sample location map. However, regions on the center, where samples vary from medium to high concentrations is over estimated by the SVR, where the model is overly smoothed. On the other hand, the LWSVR reproduces the patterns observed on the spatial data distribution with higher accuracy. The regions of lower values are more defined in LWSVR, as are the regions with medium concentration values. Figure 6 presents the local trend of the mean on the validation set from the first fold, along with the local mean of the LWSVR and SVR predictions on the validation set.

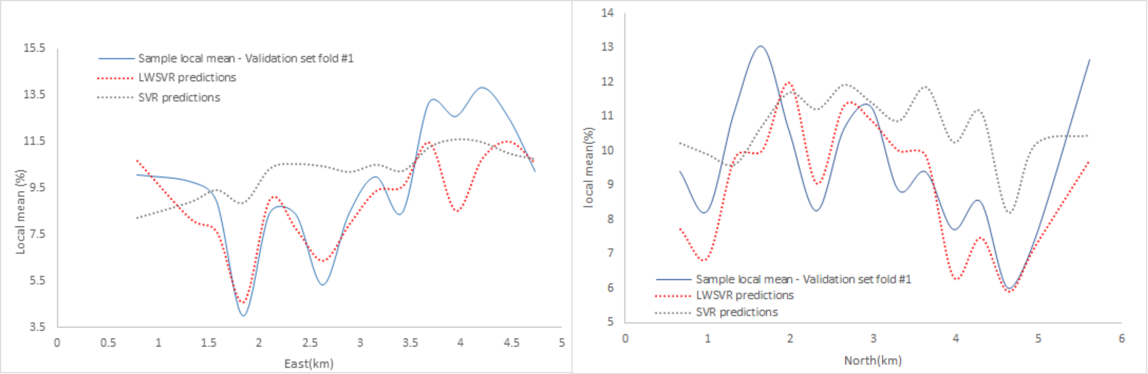

Figure 6: Swath plot on the X direction of the area of interest, performed over the sample data on the Validation set from the first fold.

Figure 6 corroborates the smoothing effect observed on the SVR interpolated map, on the slice from 1.5km to 2km on the East direction, the SVR follows the general behavior presented by the data samples, but still presents high degree of over estimation. This effect becomes evident on the central region of the curve. As seen on the interpolated map, due to a suboptimal global fit of SVR the lower values on this region are not reproduced and the area is overestimated. In contrast LWSVR captures the data behavior closely reproducing subregions trends. Nonetheless, the LWSVR does not perform well on the boundaries of the data set, underestimating the values. It is important to highlight that the parameter definition accordingly to a relevance metric has not only significantly improved performance, but the algorithm practical use is simpler. The user does not have to previously optimize the parameters, avoiding suboptimal models. Also, since the LWSVR algorithm fits as many SVR models as there are samples inside the neighborhood, the workflow increases the cost of computational time.

Conclusions

The particular features that permeate geological data sets have shown, in this research that machine learning algorithms benefit from local learning. This is due to sparsity and irregularity on the data collected, which turn the global model fit challenging, often leading only to suboptimal models. Such situations affect greatly the model generalization capability. Also, the algorithm herein proposed does not require parameter setting prior to the training stage. The parameter is defined dynamically according to the neighborhood information to which the model is fit. The local component on the model leads to a higher capacity in capturing sub regions specificity that mounts to a more accurate final model. It is important to highlight that the parameter gamma is not fine-tuned on the LWSVR, and is used as a fixed value over modeling. However, gamma translates on how far a training example influences on the model, and it is reasonable that it should also be defined according to the training data inside the neighborhood.

References

Atkeson, C. G., Moore, A. W., Schaal, S. (1997). Locally weighted learning. Artificial Intelligence Review. 11. https://doi.org/10.1023/A:1006559212014

Awad M., Khanna R. (2015) Support Vector Regression. In: Efficient Learning Machines. Apress, Berkeley, CA. https://doi.org/10.1007/978-1-4302-5990-9_4

Bottou, L., Vapnik, V. (1922). Local Learning Algorithms. Neural Computation, 4(6). doi: https://doi.org/10.1162/neco.1992.4.6.888

Dai, F., Zhou, Q., Lv, Z., Wang, X. and Liu, G. (2014) Spatial prediction of soil organic matter content integrating artificial NN and OK in Tibetan Plateau. Ecological Indicators 45. https://doi.org/10.1016/j.ecolind.2014.04.003

Dowd, P. A. and Saraç, C. (1994). A NN Approach to Geostatistical Simulation. Mathematical Geology, 26(4).

Ellatar, E. E., Goulermas, J., Wu, Q.H.(2010) Electric load forecasting based on locally weighted support vector regression. IEEE Transactions on systems, man, and cybernetics—part c: applications and reviews. 40(4). https://doi.org/ 10.1109/TSMCC.2010.2040176

Galvan. I.M., Valls, J.M., Garcia, M., Isasi, P. (2011). A lazy learning approach for building classification models. International Journal of Intelligent systems 26(8). https://doi.org/10.1002/int.20493

Gangappa, M., Mai, C. K., Sammulal, P. (2017) Techniques for machine learning based spatial data analysis: research directions. International Journal of Computer Applications, 170(1). https://doi.org/10.5120/ijca2017914643

Kapageridis, I. K. (1999). Application of NNs systems to grade estimation from exploration data. (PhD). University of Nottingham.

Maniar, H., Srikanth, R., Kulkarmi, M.S., Schlumberger, A. A. (2018) Machine Learning methods in geoscience. Society of Exploration Geophysicists. International Exposition and 88th Annual Meeting. https://doi.org/10.1190/segam/2018-2997218.1

Samson, M. and Deutsch, C. (2018). Estimation with ML. CCG annual report 20. Paper 118.

Samson, M. and Deutsch, C. (2019). Elliptical Radial Basis Function vs. Radial Basis Functions. CCG annual report 21. Paper 110.

Samson, M. (2019). Mineral Resource Estimates with ML and Geostatistics. Master Thesis. Edmonton, Canada.

Smola, A. J., Scholkopf, B. (2003). A tutorial on support vector regression. Statistics and Computing. 14. https://doi.org/10.1023/B:STCO.0000035301.19549.88

Tahmasebi, P. and Hezarkhani, A. (2012). A fast and independent architecture of artificial NN for permeability prediction. Journal of Petroleum Science and Engineering 86. doi: http://doi.org/10.1016/j.petrol.2012.03.019

Tomislav, H., Nussbaum, M., Wright, M. N., Heuvellink, G. B. M., Graler, B. (2018) Random forest as a generic framework for predictive modeling of spatial and spatio-temporal variables, PeerJ 6: e5518. https://doi.org/10.7717/perrj.5518

Vapnik, V. N. (1995) The nature of statistical learning theory. New York. Springer.

Vapnik, V.N. (1998) Statistical Learning Theory. New York. Wiley.

Walch, A. Castello, R., Mohajeri, N., Guinard, F., Kanevski, M. Scartezzini, JL., (2019) Spatio-temporal modelling and uncertainty estimation of hourly global solar irradiance using extreme learning machines. Energy Procedia. 158. https://doi.org/10.1016/j.egypro.2019.01.219

Wang, H., Xu, D. (2017). Parameter selection Method for Support Vector Regression Based on adaptive fusion of the mixed kernel function. Journal of Control Science and Engineering. 17. https://doi.org/10.1155/201/3614790