Comparison of four machine learning algorithms for spatial data analysis

Nicolas Gilardi
UNIL, IDIAP
gilardi@idiap.ch

Samy Bengio
IDIAP
bengio@idiap.ch

Abstract:

This chapter proposes a clear methodology on how to use machine learning algorithms for spatial data analysis in order to avoid any bias and eventually obtain fair estimation of their performance on new data. Four different machine learning algorithms are presented, namely multilayer perceptrons (MLP), mixture of experts (ME), support vector regression (SVR) and a local version of the latter (local SVR). Evaluation criteria adapted to geostatistical problems are also presented in order to compare adequately different models on the same dataset. Finally, an experimental comparison is given on the SIC97 dataset as well as an analysis of the results.

Key Words:

Machine learning, evaluation methods, artificial neural networks, support vector machines, mixture of experts, local models, geostatistics, SIC97, non-stationarity.

1 Introduction

During the last decade, machine learning algorithms, such as artificial neural networks, have been extensively used for a wide range of applications. They have been applied for classification, regression, and density estimation tasks (see [Bishop, 1995] for a good overview). In fact, many fields of research needing feature extraction or data prediction have been trying some machine learning methods, with more or less success. Analysis of spatial data has been involving these methods as well, such as in [Kanevski, 1996] and [De Bollivier, 1997], but in general, these methods are not very well exploited in Geostatistical problems.

One reason to this might be the «black box» aspect of most of these algorithms. It is usually difficult to explain why a given model has produced a given answer, except in a statistical sense. Tuning them can also be very difficult and without a clear methodology based on statistical learning theory [Vapnik, 1995] and some prior information about data, it will often lead to bad performance. Therefore, it can appear unnecessary to use such complex methods when one has more simple but still efficient one.

However, when simple methods do not give acceptable performance on a given problem, or when there is a lack of knowledge about the studied phenomenon, machine learning algorithms can be considered in order to expect better results at the price of loosing some interpretability of the underlying models.

In this paper, we show how machine learning algorithms can be used on such spatial data problems and demonstrate their use on one specific dataset. We in fact compared four methods: multilayer perceptron (MLP), support vector regression (SVR), mixture of experts (ME), and a local adaptation of support vector regression (local SVR), which uses a modification of the training procedure of the original algorithm to adapt to local phenomena, in a similar way than it has been done by De Bollivier *et al* for local multilayer perceptron [De Bollivier, 1997].

In the following section, we briefly introduce the four methods compared in this paper. The next section is then devoted to the methodology we used in order to select the values of the hyper-parameters of all machine learning algorithms in order to avoid any bias. We then present a general methodology to evaluate the quality of a model with respect to Geostatistics requirements, showing which criteria are useful for what purpose, and which algorithm might be the best to optimise such criteria. Afterward we present a spatial data analysis problem, SIC97, on which the four machine learning algorithms will then be compare. This problem concerns the estimation of daily rainfall over 367 locations given 100 measurements points and a digital elevation model of the region.

2 Presentation of some machine learning algorithms

In this section, we introduce the reader to some of the most popular machine learning algorithms. We first present the most known method, namely the multilayer perceptron, as well as an extension of it to deal with non-stationarity, the mixture of experts. We continue with a regression extension of the recent but powerful classification method of support vector machines: the support vector regression, and finish our presentation with a local adaptation of the latter.

2.1 Multilayer Perceptrons

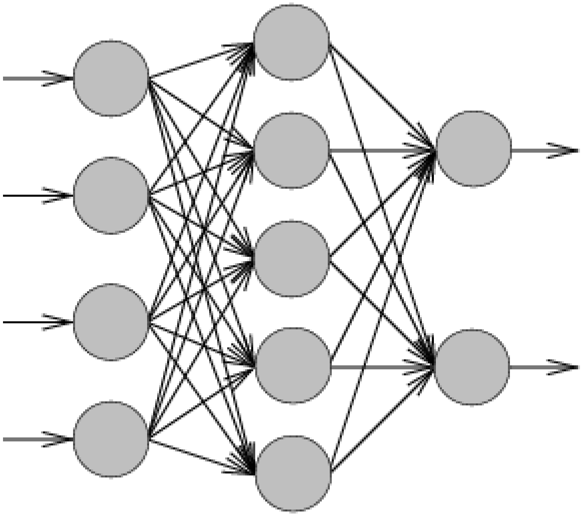


Figure 1: The architecture of an MLP

A multilayer perceptron (MLP) is a particular architecture of artificial neural networks, composed of layers of non-linear but differentiable parametric functions. For instance, Figure 1 shows an MLP with one input layer of size 4, one hidden layer of size 5 and one output layer of size 2. Alternatively, an MLP can be written mathematically as follows[1]:

$$f(\mathbf{x};\theta) = b + \sum_{n=1}^{N} w_n \cdot \tanh\left(b_n + \sum_{m=1}^{M} x_m \cdot w_{nm} \right)$$

where the estimated output $f(\mathbf{x};\theta)$ is a function of the input vector **x** (indexed by its M values x_m), and the parameters $\{\theta : w_n, w_{nm}, b_n, b \in \mathrm{R}; \text{with } n \in [1,N], m \in [1,M]\}$ to be found by a learning procedure.

[1] The equation is given here for only one output to simplify the notation

This MLP is thus a weighted combination of N hyperbolic tangents of weighted combinations of the input vector. Given a criterion Q to minimise, such as the mean squared error,

$$Q = \sum_{i=1}^{l} \left(y_i - f(\mathbf{x}_i;\theta)\right)^2$$

between the desired output y_i and the estimated output $f(\mathbf{x}_i;\theta)$, for a given training set of size l, one can search for parameters θ that minimise such criterion using a gradient descent algorithm [Rumelhart, 1986]. This algorithm is based on the computation of the partial derivative $\frac{\partial Q}{\partial \theta}$ of the criterion Q with respect to all the parameters θ of $f(\mathbf{x};\theta)$. The gradient descent can then be performed using

$$\theta = \theta - \lambda \cdot \frac{\partial Q}{\partial \theta}$$

for each parameter θ where λ is the *learning rate*. It has been shown that given a number of hyperbolic tangents N sufficiently large, one can approximate any continuous function using such MLPs [Hornik, 1989].

2.2 Mixture of Experts

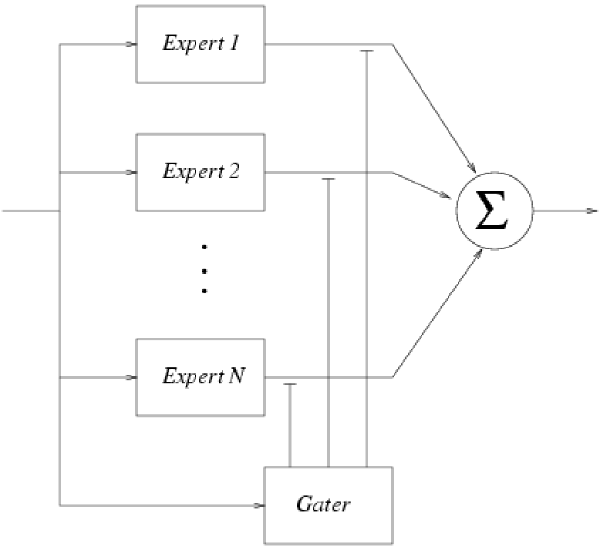


Figure 2: The architecture of a mixture of experts.

A mixture of experts [Jacobs, 1991] is a very simple model that embodies the divide-and-conquer principle: instead of trying to fit a unique model for a whole training set, one supposes that dividing the training set into many smaller training sets could simplify the problem. The idea of a mixture of experts is then to simultaneously (a) learn how to cut a training set into different parts[2] and (b) learn a different model on each part. As shown if Figure 2, in its simplest form, a mixture of experts is thus composed of N modules, each receiving the same inputs, and each trying to output the desired target. An additional module, the *gater*, also receives the same input but has N outputs which corresponds to the probability of each module to give the correct target. It thus computes a soft partition of the input space. More formally, for each input/output point $(\mathbf{x}_i, y_i)$, each model m_n is computing $E(y_i \mid \mathbf{x}_i, m_n)$ the expectation of the output y_i given the input $\mathbf{x}_i$, and the gater is

[2] In fact, as it will be seen with the equations, instead of attributing an example to one and only one model, each model will see every examples but with a different weight for each example.

computing $P(m_n \mid \mathbf{x}_i)$ the probability of model m_n given the input $\mathbf{x}_i$. The overall output of the mixture of experts is then

$$E(y_i \mid \mathbf{x}_i) = \sum_{n=1}^{N} P(m_n \mid \mathbf{x}_i) E(y_i \mid \mathbf{x}_i, m_n)$$

with the constraint that

$$\sum_{n=1}^{N} P(m_n \mid \mathbf{x}_i) = 1$$

In the particular case where the gater and the models are represented by differentiable parametric functions such as multilayer perceptrons[3], the whole system can be optimised jointly by minimising an overall criterion Q such as the mean squared error (cf. section 2.1) over the whole training set. For parameters θ of a given model m_n, the derivative of the criterion with respect to the parameters is as follows

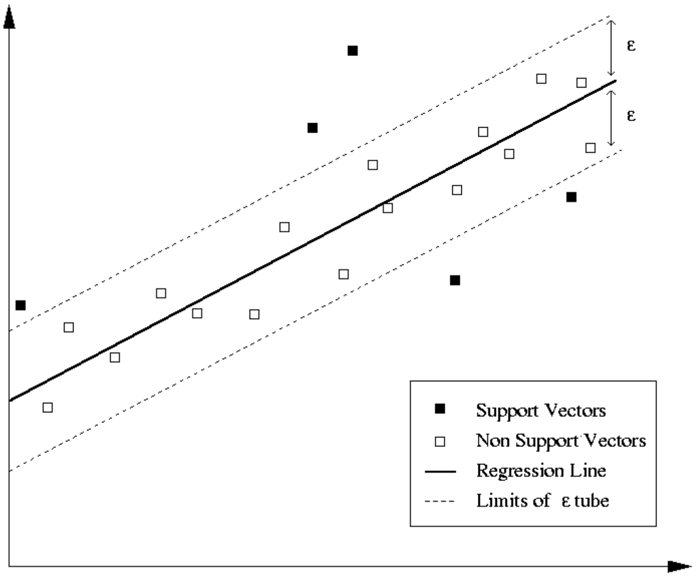
$$\frac{\partial Q}{\partial \theta} = \sum_{i=1}^{l} \frac{\partial Q}{\partial E(y_i \mid \mathbf{x}_i)} \frac{\partial E(y_i \mid \mathbf{x}_i)}{\partial E(y_i \mid \mathbf{x}_i, m_n; \theta)} \frac{\partial E(y_i \mid \mathbf{x}_i, m_n; \theta)}{\partial \theta}$$

and for parameters θ of the gater, the derivative is as follows

$$\frac{\partial Q}{\partial \theta} = \sum_{i=1}^{l} \frac{\partial Q}{\partial E(y_i \mid \mathbf{x}_i)} \sum_{n=1}^{N} \frac{\partial E(y_i \mid \mathbf{x}_i)}{\partial P(m_n \mid \mathbf{x}_i; \theta)} \frac{\partial P(m_n \mid \mathbf{x}_i; \theta)}{\partial \theta}$$

Finally, it is important to note that, if one does not have to decide the partition of the training set, one still has to decide the number of such partitions. This can be done using for instance a cross-validation technique, as described in section 3 on Model Selection.

2.3 Support Vector Regression



[3] Note that in order for the gater to output probabilities, some special output function should be used to ensure the necessary constraints, such as the well-known *softmax* function $y_i = \frac{e^{x_i}}{\sum_{j=1}^{N} e^{x_j}}$.

Figure 3: SVR linear regression with ε-insensitive loss function

Directly derived from Vapnik and Chervonenkis' *Statistical Learning Theory* [Vapnik, 1995], Support Vector Machines (SVM) for classification problems were developed during the beginning of the 90's (a good overview of SVMs can be found in [Burges, 1998]). Later, the algorithm was extended to deal with regression problems. This new algorithm was thus named Support Vector Regression (SVR) [Smola, 1998], and is presented briefly hereafter.

For a given set of data $(\mathbf{x}_i, y_i)_{1\le i\le l}$, $\mathbf{x} \in \mathrm{R}^n$ and $y \in \mathrm{R}$, the simplest linear SVR algorithm tries to find the function

$$f(\mathbf{x}) = w \cdot \mathbf{x} + b$$

by minimising the quadratic optimisation problem

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l} Q(y_i - f(\mathbf{x}_i))$$

where $Q(z) = \max\{0, |\,z| - \varepsilon\}$ is the ε-insensitive loss function proposed by Vapnik and does not penalise errors less than $\varepsilon \ge 0$ (cf. Figure 1). After some reformulation and taking into account the case of non-linear regression, the optimisation problem is then transformed into the minimisation of

$$\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i - \alpha_i^*)(\alpha_j - \alpha_j^*)k(\mathbf{x}_i, \mathbf{x}_j) + \varepsilon\sum_{i=1}^{l}(\alpha_i + \alpha_i^*) - \sum_{i=1}^{l} y_i(\alpha_i - \alpha_i^*)$$

subject to

$$\sum_{i=1}^{l}(\alpha_i - \alpha_i^*) = 0$$
$$0 \le \alpha_i, \alpha_i^* \le C, \text{for } 1 \le i \le l$$

where the α_i, α_i^* are Lagrange multipliers, solutions of the optimisation problem, *C* is the *soft margin* parameter, representing the amount of noise in the data, and $k(\mathbf{x}_i, \mathbf{x}_j)$ is a *kernel function*, defining the feature space in which the optimal solution of the problem will be computed in order to handle non-linear problems. An example of such kernel is the Gaussian Radial Basis Function (RBF) kernel

$$k(\mathbf{x}_1, \mathbf{x}_2) = \exp\left(-\frac{\|\mathbf{x}_1 - \mathbf{x}_2\|^2}{2\sigma^2}\right)$$

which has been used in this paper. Finally, to estimate a new point, we use the following function

$$f(\mathbf{x}) = \sum_{i=1}^{N}(\alpha_{s_i} - \alpha_{s_i}^*)k(\mathbf{x}, \mathbf{x}_{s_i}) + b$$

where b is a bias, computed as follows:

$$b = y_i - f(\mathbf{x}_i)_{b=0} - \varepsilon \quad \text{for } \alpha_i \in \left]0, C\right[$$
$$b = y_i - f(\mathbf{x}_i)_{b=0} + \varepsilon \quad \text{for } \alpha_i^* \in \left]0, C\right[$$

and the $s_i, 1 \le i \le N$ are the indices of the data points for which either α_{s_i} or $\alpha_{s_i}^*$ is non zero. Those points are called *support vectors* (black squares in Figure 3).

2.4 Local SVR

As proposed by Bottou and Vapnik in [Bottou, 1992], when the data set is clearly non-evenly distributed, one can significantly improve the performance of machine learning algorithms by building multiple local models instead of one global model. As environmental data are often influenced by various local phenomena, the idea of using local models applied to geostatistical problems was also developed for classical geostatistical interpolation methods, such as ordinary kriging [Haas, 1990].

In this section, we propose the use of a local method based on SVRs with an approach very similar to the one presented in [De Bollivier *et al.*, 1997] for spatial interpolation with MLPs.

The proposed algorithm builds one SVR model for each point to be estimated, taking into account only a subset of the training points. This subset is chosen on the basis of the Euclidean distance between the testing point and the training point in the input space. For each testing point, a new SVR model is thus learned using only the training points lying inside a user defined radius which centre is the current testing point.

The radius can be chosen using the *a priori* spatial correlation of the dataset, or like any other hyper-parameters, i.e. by cross-validation (see section 3 for an introduction to cross-validation). It is also possible to use an anisotropic neighbourhood, such as an ellipsoid, instead of a circle. Hence, one can «force» the local model to adapt itself to an anisotropic phenomenon.

A problem related to local SVR estimation (besides the computational time needed to create all these local models) concerns the number of training points for each local model. If the selected radius is too small, it might happen that some testing points will have a very small number of training points in their neighbourhood (or even none). Theoretically, the SVR algorithm can work even with only two training points (with input vectors in two dimensions) but, because of numerical instabilities during the optimisation, it is safer to estimate an SVR model with at least four training points. When this is not possible, one should use instead a simple mean value or inverse distance method.

On the contrary, if the research radius is too large, one might have a very large number of training points, which would lead to a long training time, considering the fact that one is computing one model for each testing point! Thus, in order to speed up the procedure, one might consider to limit the number of training points taken into consideration. It is also important to note that testing points that are located close together will probably have the same neighbours in the training set, and thus, only one model should be necessary to predict those points.

3 Model Selection

Most of the models proposed in the machine learning literature, such as those proposed in this paper, have some *hyper-parameters* that need to be selected prior to learning. *Hyper-parameters* are parameters of the algorithm that are defined by the user and which influence the training procedure. For instance, for an iterative algorithm, it could be the number of iterations; for a multilayer perceptron, it could be the number of hidden units; for a support vector machine, it could be a parameter related to the chosen kernel. In fact, most of the models usually have more than one hyper-parameter. In order to select them appropriately, some kind of *hyper-learning* method is needed.

The method depends on the size of the data set. When it is large enough (usually more than a few thousands examples), a simple method works as follows:

- Randomly divide the data set into two parts, a *training set* and a *validation set* (the validation set is usually smaller than the training set, depending on the total size of the data set).
- For each value of the hyper-parameter (if there is more than one hyper-parameter then, for each set of values of the hyper-parameters), train a model on the *training set* and compute the performance of the trained model on the *validation set*.
- Select the value of the hyper-parameter that produced the model that gave the best performance on the *validation set* and train the corresponding model with the whole data set.

The main idea behind this method is that the hyper-parameters have to be chosen with data that were not used for training in order to avoid any bias. However, when the size of the data set is too small, which is often the case when environmental data are not collected with the help of remote sensing techniques, this simple method becomes too noisy and depends strongly on the arbitrary division between the training and validation sets. An extension of this method, called *cross-validation*, and which has many variants, should then be used. For the current study, the *K-fold cross-validation* method has been used:

- For each value of the hyper-parameter (if there is more than one hyper-parameter then, for each set of values of the hyper-parameters), estimate the *validation* performance of the corresponding model as follows:
  - Randomly divide the data set into K partitions of approximately the same size.
  - For each partition, train a model using the data from the $K-1$ other partitions and compute the validation performance of all the examples of this partition.
  - Add all validation performances to compute the validation performance of the model with the current value of the hyper-parameter.
- Select the value of the hyper-parameter that produced the model that gave the best validation performance and train the corresponding model with the whole data set.

One should be aware that these methods do not give a good estimate of the performance of the selected model on new data since all the examples have been used to select the model. When one wants also to estimate the generalisation performance of the selected model, one needs to do two embedded *cross-validations*: one to select the right model and one to estimate its performance. In the current study, we did not estimate the generalisation performance since the goal was to select a model and then give predictions on a separate data set.

4 Model Evaluation

Given two models solving the same problem, it is important to be able to decide which model is best suited to the given task, and if its performances are satisfactory. In order to address these problems, one has to formally describe the goals to fulfil. For instance, in finance, between two models taking decisions in the financial markets, one could select the model that gives the highest returns by taking the lowest risks. For a pattern recognition task, one could select the model that gives the best classification performance.

In this section, we are interested in the goals underlying geostatistical problems. We will first try to give a mathematical formulation of each goal. The current methods used in geostatistics are then briefly presented. Finally, the methods used in the machine learning are also given.

4.1 Goals of Geostatistics

The goal of a geostatistical case study is to provide the full conditional distribution of a random variable at places where observations of the variable are not available. With such an information, it is possible to answer in a coherent way to many questions related to the statistical aspect of the given problem (value prediction, risk evaluation, spatial correlation reconstruction, etc…). Such a coherence might not be reached if one uses different methods to answer the different questions. The target conditional distribution of this random variable should depend on the exact location in the map, and eventually on extra information like a physical model or other data measurements supposedly correlated.

4.2 Possible methods to evaluate the conditional distribution

Without explaining how it could be possible to learn such conditional distribution, it is important to stress out how to evaluate a given conditional distribution in order to compare the results of many models. There are many ways one can think of in order to evaluate the quality of a conditional distribution.

Let us suppose we can create a parametric conditional distribution $P(Y\,|\,\mathbf{X};\theta)$, where Y is the desired target random variable, **X** is the input vector (usually a two dimensional vector describing the position in the map), and θ is a set of parameters defining the distribution. Let also $Z=\{(\mathbf{x}_i;y_i)\}_{1\le i\le l}$ be a set of test examples we can use to evaluate $P(Y\,|\,\mathbf{X};\theta)$. The obvious way to measure the quality of $P(Y\,|\,\mathbf{X};\theta)$ is to compute the likelihood L that all examples in Z have been drawn from $P(Y\,|\,\mathbf{X};\theta)$:

$$L=\prod_i P(y_i\,|\,\mathbf{x}_i;\theta)$$

and we would like L to be as high as possible. As long as **X** and Y are not transformed, two different models can then be compared using this measure.

On the other hand, if we cannot express the full conditional distribution but only some of its moments M, such as its conditional expectation $E(Y\,|\,\mathbf{X})$ and its conditional variance $Var(Y\,|\,\mathbf{X})$, then one can think of an evaluation function that would measure the quality of all of the available moments, such as the weighted sum

$$C=\sum_{i=1}^{l}\sum_j w_j\left(M_j^{1/j}(y_i\,|\,\mathbf{x}_i)-\hat{M}_j^{1/j}(y_i\,|\,\mathbf{x}_i)\right)^2$$

where $\hat{M}_j(y_i\,|\,\mathbf{x}_i)$ is the j^{th} estimated conditional moment, w_j is the factor weighting the relativeness of this moment, and C is a value to minimise. One important limitation of this model evaluation criterion concerns the "true" moment values that need to be available while the maximum likelihood method do not need any information about the true distribution except some test examples.

4.3 Current Methods Used in Geostatistics

Variography, which consists in measuring the spatial correlation of data by comparing examples positions and values, can be considered as an approximation of the local second order moment. Usually, one is using semi-variogram to fulfil this task. It is defined as:

$$\gamma(\mathbf{h})=\frac{1}{2}\,\mathrm{E}\left[(y(\mathbf{x})-y(\mathbf{x}+\mathbf{h}))^2\right]$$

where **h** is a directional vector defining the lag step of the semi-variogram calculation and $y(\mathbf{x})$ the value of the example lying at co-ordinates **x.** While studying a stationary data set, $\gamma(\mathbf{h})$ defines a local correlation zone and increases with **h** until it reaches the global variance of the data set. This zone is called the "range" of the semi-variogram.

With such a tool, one has a good representation of the information that can be extracted from the second moment. Geostatistical methods of prediction, like kriging or stochastic simulations, are using a model of this description of data as the main input information. Afterward, the optimisation criterion is different from one method to another: *kriging* is minimising the error variance, which is related to the optimisation of the mean squared error, while *stochastic simulations* are trying to reconstruct both the histogram and the variogram model, and thus the first and second moments of the data set. These two methods could then use the weighted sum of moments, presented in section 4.2, in order to be compared, as long as one have access to the true value of the moments or at least an approximation of them.

4.4 Methods Used in Machine Learning

In the machine learning community, most of the problems are classified into three categories: classification, regression, or density estimation. In classification, one is usually interested in minimising the number of classification errors; in regression, two evaluation methods are used, either the Mean Absolute Error

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - \hat{y}_i\right|$$

or the Mean Squared Error

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2$$

In density estimation, the most used method is the maximum likelihood, already described in section 4.2. *Hence, using the MAE or MSE criterion for geostatistical problems should give very good performance in reconstructing the first moment but will generally underestimate the second moment.*

5 Case Study

Presented in 1997 by the journal of Geographic Information and Decision Analysis (GIDA), the Spatial Interpolation Comparison 97 challenged interested people to use the method of their choice in order to solve a real problem: in case of the flyby of a nuclear cloud, consequence of a nuclear accident such as Tchernobyl's one, over a large region, estimating the simultaneous local rainfall is very important, as this information is highly correlated with the contamination of the surface by radioactive pollutants. In this case study, 100 measurements were given, as well as a digital elevation system of the region of Switzerland, and participants were asked to predict rainfall in 367 given locations.

5.1 Experimental Setup

In all experiments, only X and Y co-ordinates were used as input information. Experiments using also the altitude did not improve significantly the results of the first model tried (the SVR), and thus, for a better comparison, it has not been used for the other models.

The choice of the hyper-parameters was done by K-fold cross-validation on the training data, inside a user-defined set of hyper-parameters. Such an approach can become very time consuming when the number of hyper-parameters is high. It is therefore necessary to restrict the range in which these values should be selected.

5.1.1 MLP

The number of hidden units (N), the number of learning iterations and the value of the learning rate (λ) had to be selected for the multilayer perceptrons. All these hyper-parameters are related to the *capacity* of the learning system: the more examples one has, the higher the values of the hyper-parameters could be, but their optimal value are problem dependent and can only be chosen by cross-validation. Some simple rule of thumb still exists, such as the fact that the number of parameters (weights and bias), which is related to the number of hidden units, is usually smaller than the number of training examples. But these rules of thumb should be used carefully, only to give an idea on the range of the values to select with cross-validation. The number of hidden units was thus chosen in a range from 5 to 40 and the number of iterations was chosen in a range from 100 to 1000. Moreover, instead of using a simple gradient descent method, we used a conjugate gradient method, which takes into account second order information and does not need to select a learning rate λ.

5.1.2 Mixture of Experts

For the mixture of experts, we had to set the number of experts as well as to select a way to represent the experts and the gater. We decided to put most of the *capacity* into the gater so we represented it by an MLP, with various number of hidden units (from 5 to 40). The experts were then represented by simple linear models (weighted combinations of the inputs). The number of experts is not easy to select. It should reflect the non-stationarity of the data, but it should also take into account the total number of examples in the training set. Therefore, we chose it in a range from 2 to 12. Finally, the number of iterations was chosen in the same manner as for the MLP experiments.

5.1.3 SVR

The kernel parameter σ, which is the standard deviation of a Gaussian function, is directly related to the local variability of the data: the more the data are locally variable, the smaller it should be. In practice, this parameter should lie between half the smallest distance between two data points and half the highest. *Values outside these bounds are not reasonable* as it would mean that variability is either so high that no spatial correlation can be computed or so low that no improvement can be expected from an increase of σ.

The experimental results from [Kanevski, 2000] showed that the precision parameter ε is upper-bounded by the value of the local variability of the data, the so called «nugget level» of the semi-variogram[4]. In SIC97 data, this level is almost zero, so the optimal ε value should be very small with respect to data value, and a range from 0 to 50 was thus chosen.

The soft margin parameter C is much more difficult to limit. It is related to the confidence we have in our data: the highest it is, the more we believe in the training data. This hyper-parameter is unlimited, so usually, one gives it various powers of 10 in order to find the optimal one, but it might not be the most efficient method.

5.1.4 Local SVR

For the local SVR, one has to define the search neighbourhood, in addition to the other SVR hyper-parameters. This neighbourhood was chosen with respect to the range and the anisotropy given by the semi-variogram of the whole training set. The use of global parameters to compute

[4] The nugget level of a data set corresponds to the value the semi-variogram would have if one would interpolate it to a distance of 0. This is a quite subjective value as it is impossible to compute it precisely. It represents the local variability of the data or measurement noise, also called "nugget effect".

local models may surprise the reader but, since only a few data were available, the computation of local semivariograms would have been noisy and thus irrelevant. Moreover, a local adjustment of the anisotropic parameter would have been far too expensive in terms of computation time. Therefore, since the main goal is to extract the local correlation, only the first lags (cf. section 4.3) have been used.

5.2 Results

5.2.1 Estimation Maps

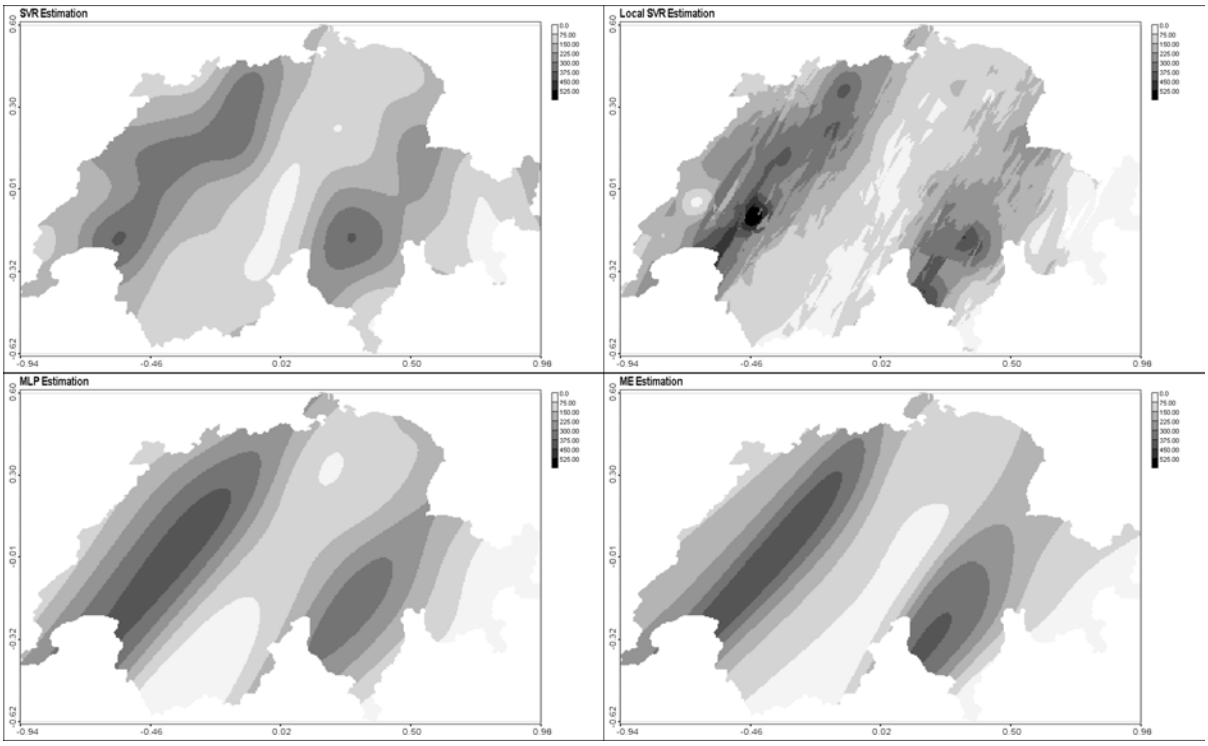


Figure 4: Estimations of rainfall over Switzerland on a dense grid using four machine learning models: SVR, Local SVR, MLP and ME. Models were trained on the SIC97 dataset. Colour scales are identical for all pictures above.

As shown in Figure 4, all the models did reproduce the large anisotropy of the SIC97 data set, and the general predictions are quite close to the original data, with a large band of low precipitation from south-west to north-east, surrounded by two medium-to-high precipitation areas.

The most noticeable differences between these pictures concern the very different "shapes" of the isolines. MLP and mixture of experts (ME) have a quite similar behaviour, with isolines defining some elliptic regions, while SVR's are more circular. This difference is related to the family of function used (hyperbolic tangent for MLP and ME and Gaussian radial basis function for SVR). But the most surprising is local SVR's map. While the other three are very smooth, this one is very sharp and presents a noisy pattern. The reason is that while MLP, ME and SVR are using a few tens of parameters, local SVR is using thousands (because of the density of the grid used here). Another aspect of this map concerns the discontinuities in isolines. This poor result is due to the strict limitation of the influence of the local models added to the small number of training data.

5.2.2 Error Maps

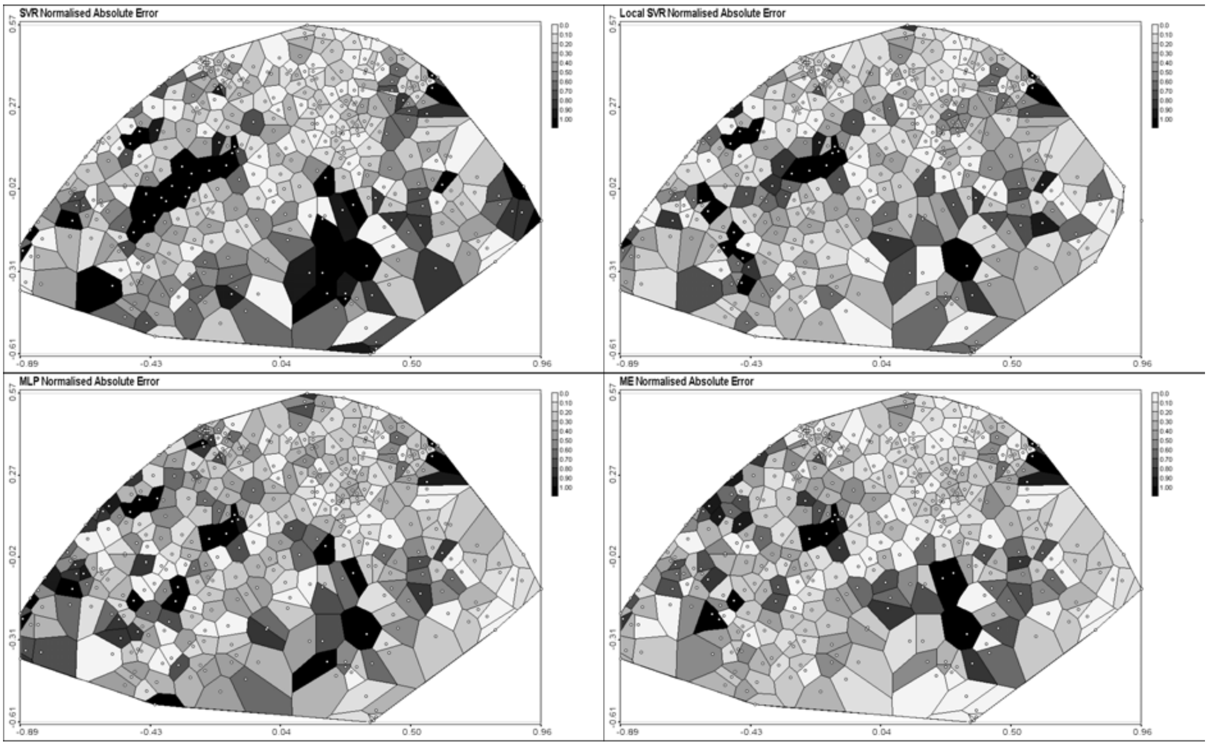

SVR Normalised Absolute Error
Local SVR Normalised Absolute Error
MLP Normalised Absolute Error
ME Normalised Absolute Error
0.57
0.27
-0.02
-0.31
-0.61
-0.89
-0.43
0.04
0.50
0.96
0.0
0.10
0.20
0.30
0.40
0.50
0.60
0.70
0.80
0.90
1.00

Figure 5: Voronoi Polygons of the normalised absolute error of SIC97 testing data predictions. The normalised absolute error corresponds to the absolute error made by the predicting model at each location, divided by the standard deviation of the whole SIC97 data set (train + test). Point locations are represented by the white circles.

The efficiency of each model on the SIC97 problem can be studied using the error maps of Figure 5. A first overview shows that all models failed to predict some specific regions, probably because of a lack of information in the training data. Also, as it has been shown with the prediction maps, the error maps of a model family have similar behaviour. Finally, one can say as a general remark that local SVR and ME are improving their "global" counterparts. More specifically, SVR are obviously unable to predict the high values. Large bands of high error can be seen on its error map. The other models are more efficient in this domain, but MLP is also showing some regional errors, especially along the West/North-West border. Regarding local SVR and ME, errors look less structured than for the other models.

5.2.3 Spatial Correlation

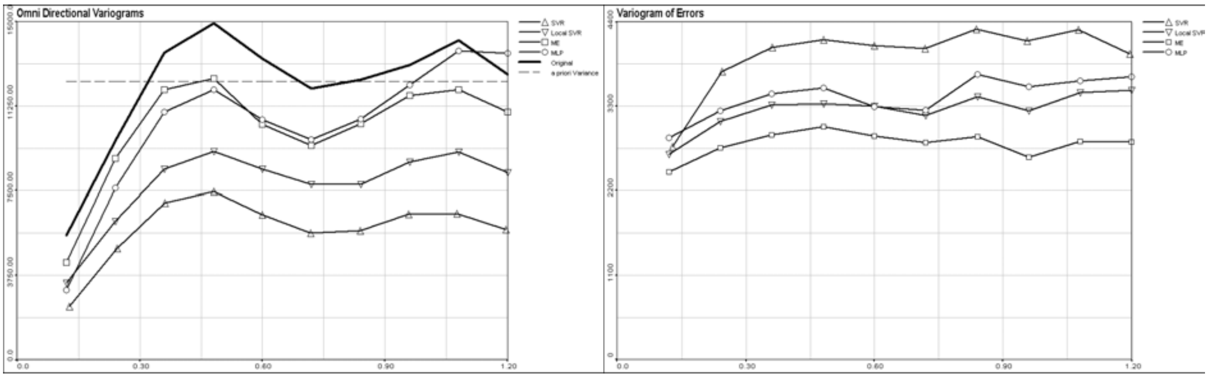


Figure 6: Omni-directional semi-variograms of SIC97 testing data estimations and residuals. The left figure shows the variograms from the four estimations, compared to the "true" one and to the *a priori* variance of the

testing data. The right figure shows the variograms of the residuals of the four in order to compare the remaining correlation inside error maps.

Figure 6 compares the omni-directional semi-variogram of the original testing data to the ones obtained by the proposed models. It is interesting to notice that all models managed to reproduce the general spatial correlation quite well over a large distance. However, SVR estimation is very smooth, as the reduction of variance attests. This aspect of reduction of general variability is also present for the local SVR, but the improvement with respect to the global SVR is quite significant. This result can be surprising with regard to the local SVR's prediction map which looks noisy. But one can also see in this picture that high values areas are common, and thus the mean variance is low. The noisy aspect of the picture appears in the semi-variogram as the nugget effect, which is higher for local SVR than for any other methods. Regarding MLP and the ME, they outperform both SVR approaches, reproducing almost exactly the rise of the short range correlation.

The variography of the residuals[5] gives also some information about the feature extraction quality of each models for the SIC97 data. Thus, one can notice that SVR did not manage to extract all information, as residuals appears to be strongly correlated, as it is also visible in Figure 7. But even if the remaining correlation is not as high, it exists also for the other methods, showing that some improvement is still theoretically possible. ME is again the best of the four models for feature extraction in this data set.

5.2.4 Numerical Results

Table 1 presents the numerical results of the four models' predictions of SIC97 testing data set. For each model, it shows the root mean square error (RMSE) and mean absolute error (MAE), as well as other characteristics of the obtained values, such as the minimum (MIN), maximum (MAX), median (MEDIAN), mean (MEAN) and standard deviation (STDEV). Comparisons to the real SIC97 testing data statistics is also presented.

In addition to the four models, we also present the results of a mixture of them. The aim of this mixture is to show how one can reduce the mean errors of the prediction of SIC97 testing data by mixing the predictions of the various models studied. Thus, for each point, the prediction of the mixture model was simply the mean of the predictions of the four models.

The last part of the table summarises the results of SIC97 as presented in [Dubois, 2000]. «SIC97 best» and «SIC97 worst» gives the corresponding results, in terms of absolute deviation to the real value, found by the submitted models for this specific table section (i.e. the best RMSE and the best MAE do not correspond to the same model). «SIC97 median» gives the interval inside which the statistics of the best 50 % of the submitted models are.

	RMSE	MAE	MIN	MEDIAN	MAX	MEAN	STDEV
SIC97 true	N.A.	N.A.	0	162	517	185	111
MLP	59	45.8	8.9	186.9	380.6	188.2	96.5
SVR	63.4	45.9	37.5	165.6	369.6	184	77.1
ME	53.2	38.6	0	165.3	453.8	182.5	101.7
Local SVR	57.1	41.9	0	163	472.7	182	88.8
Mixture of all	51.8	38.3	33.4	169.4	419.3	184.7	89.3

[5] A residual is the difference between the desired value and the obtained value

SIC97 best	53.1	32	0	162	514	185	111
SIC97 median	63	44	[-15.5;15.5]	[154;170]	[462.5;571.5]	[181;189]	[99;123]
SIC97 worst	99	70.6	-413	191	788	159	139.5

Table 1: Comparison of multiple models on SIC97 data set. The models compared were multilayer perceptrons (MLP), Support Vector Regression (SVR), mixture of experts (ME), local SVR, and the mixture of all these models. They were compared to the best, median and worst results for the SIC97 competition. Results are given in terms of root mean squared errors (RMSE), mean absolute error (MAE), as well as some statistics such as the minimum predicted value (MIN), the median (MEDIAN), the maximum (MAX), the mean (MEAN) and the standard deviation (STDEV).

Comparing the results of the machine learning models to the general results from SIC97 contributions, one can notice that except for SVR, all the other algorithms presented here are at least as good as 50 % of the methods published in 1998 in terms of mean error. In terms of standard deviation however, this is no longer true for SVR and local SVR. Regarding the distribution of values, machine learning algorithms appear to be unable to reach the best values of SIC97 data, but mean and median (except for MLP) are efficiently recovered. It is worthwhile to note that ME gives one of the best prediction results on the SIC97 data set.

Also interesting is the behaviour of the mixture of the four models. As it was foreseen, the results in terms of mean error are very good (it is even the best in terms of RMSE). But as a direct consequence of the optimisation of the first moment, the second one is not preserved at all: the standard deviation has not been improved.

5.2.5 Conclusion on SIC97 Experiments

SIC97 benchmark, due to its complexity, has raised some drawbacks of machine learning algorithms for Geostatistical data. While these methods were almost as good as other regression techniques, they were less efficient than some model based approaches, like ordinary kriging. Various explanations can be formulated to explain this, but we can summarise them into the problem of quantity of information. As they are model free, learning algorithms are very sensitive to «bad» data sets. If the data set is small and/or noisy, its probability distribution can be very far from the true probability distribution of the phenomenon. And without *a priori* knowledge, it becomes very difficult for a learning algorithm to solve such problem efficiently. When used by experimented users, model based methods are often less sensitive to data representativity because the model becomes user's prior information about the phenomenon.

To limit these problems, we have chosen to use models focusing on local phenomena. In the case of the local SVR, we used some specific knowledge we had on data (such as spatial correlation and anisotropy) to artificially specialise the algorithm. Results are good as the improvement on all criteria is significant, with respect to the standard SVR approach. The mixture of experts do not use more *a priori* knowledge than the underlying idea that one should focus on some local phenomena. And it is keeping some global information as it is building various global models locally weighted. As a consequence, it is able to extract phenomena to a larger scale than the local SVR (which is limited by its search neighbourhood), with thus a better prediction ratio, and finally one of the best proposed on SIC97 so far. Mixture of Experts are demonstrating here the great potential of adaptation of machine learning algorithms.

Machine learning algorithms are thus interesting for automatic learning as they can give fairly good results with a very limited human action (only the choice of hyper parameters range). But it is dangerous to generalise their efficiency to any spatial problem. As it was shown here, noisy or non representative data can absolutely destroy their efficiency. And as emergency data are more likely to be uncertain, the simplest machine learning algorithms might not be suited for automatic emergency mapping.

6 Conclusion

The results presented here attest that, when properly used, machine learning regression algorithms are at least as good as many more classical ones, even if support vector machines appear to be much more efficient for classification tasks than for regression ones. It has been shown also that machine learning algorithms are adaptable to the problem to deal with, such as non-stationary data sets.

However, machine learning algorithms, as they are model free, are usually more sensitive to non-representative data sets than model based approaches. For the same reason, they are also more adapted to medium to large data sets (1000 examples and more) than to small ones.

The problem of computation time was not presented here because it is not really relevant given the small size of the SIC97 data set. Machine learning algorithms are very different from one another, and can have many different implementations, more or less efficient. However, to give an idea, computation time to find optimal models for each method presented on SIC97, varied from a few hours to a few days on UNIX workstations. However, when the model has been computed, the time to predict a value at a given point is less than a few milliseconds.

Finally, it would be interesting to propose new machine learning algorithms dedicated to solve spatial statistics problems. Hence, such algorithms would have to optimise higher statistical moments of data and not only the first one in order to be closer to geostatistician's demands.

Acknowledgements

This work was supported by Swiss National Science Foundation (CARTANN project: FN 2100-054115.98), Dalle Molle Institute for Perceptual Artificial Intelligence (IDIAP) and University of Lausanne.

Post plot and variogram pictures were generated with Geostat Office software from Russian Nuclear Safety Institute (IBRAE). SVR calculation was done using Alex Smola's quadratic optimizer (http://kernel-machines.org/code/prloqo.tar.gz).

Special thanks to Grégoire Dubois, Mikhail Kanevski and Michel Maignan for their comments on this paper.

Bibliography

[Bishop, 1995] C. Bishop. *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford, 1995

[Bottou, 1992] L. Bottou and V. Vapnik. Local Learning Algorithms. *Neural Computation*, 4:888-900,1992

[Burges, 1998] C. J. C. Burges. A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, 2(2):1-47, 1998

[De Bollivier, 1997] M. de Bollivier, G. Dubois, M. Maignan, and M. Kanevski. Modified multilayer perceptron with local constraint: Artificial Neural Networks as an emerging method in spatial data analysis. *Nuclear Instruments and Methods in Physics Research*, A389:226-229, 1997

[Dubois, 1998] G. Dubois, J. Malczewski, and M. De Cort. Spatial Interpolation Comparison 97. *Journal of Geographic Information and Decision Analysis*, 2(2), 1998, special issue

[Dubois, 2000] G. Dubois. *Intégration de système d'information géographique et de méthodes géostatistiques*. PhD thesis, University of Lausanne, 2000

[Dubois, 2001] G. Dubois. *Spatial Interpolation Comparison 97. Description of the rainfall data set.* This Volume.

[Haas, 1990] T. C. Haas. Kriging and automated variogram modeling within a moving window. *Atmospheric Environment*, 24A:1759-1769, 1990

[Hornik, 1989] K. Hornik, M. Stinchcombe, and H. White. Multilayer Feedforward Networks are Universal Approximators. Neural Networks, 2:359-366, 1989

[Jacobs, 1991] R. A. Jacobs, M. I. Jordan, S. J. Nowlan, and G. E. Hinton. Adaptive Mixtures of Local Experts. *Neural Computation*, 3(1):79-87, 1991

[Kanevski, 1996] M. Kanevski, R. Arutyunyan, I. Bolshov, V. Demyanov and M. Maignan. Artificial Neural Networks and Spatial Estimations of Chernobyl Fallout. *Geoinformatics*, 7(1-2):5-11, 1996

[Kanevski, 2000] M. Kanevski and S. Canu. Environmental and Pollution Data Mapping with Support Vector Regression. Technical Report RR-00-09, IDIAP, 2000

[Rumelhart, 1986] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning Internal Representations by Error Propagation. In Rumelhart, D. E. and McClelland, James L., editors, *Parallel Distributed Processing*, volume 1. MIT Press, Cambridge, MA., 1986

[Smola, 1998] A. J. Smola and B. Schölkopf. A Tutorial on Support Vector Regression. Technical Report 30, NeuroCOLT2, October 1998

[Vapnik, 1995] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, 1995