IDIAP RESEARCH REPORT

IDIAP

Martigny - Valais - Suisse

Evaluation of SVM Binary Classification with Nonparametric Conditional Stochastic Simulations

Mikhail Kanevski

IDIAP-RR-01-07

March 2001

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

[1] IDIAP – Dalle Molle Institute of Perceptual Artificial Intelligence, CP 592, 1920 Martigny, Switzerland
kanevski@idiap.ch

Evaluation of SVM Binary Spatial Classification with Nonparametric Conditional Stochastic Simulations

M. Kanevski[1]

Abstract.

The quality of Support Vector Machines (SVM) binary classification of spatial environmental data is evaluated with geostatistical nonparametric conditional stochastic simulations - a spatial Monte Carlo model based on sequential indicator simulation algorithm. Equally probable realizations are generated and compared with SVM classification. Uncertainty of predictions is described by conditional standard deviations. Case study is based on the classification of porosity data. Only binary problem is considered. Results obtained confirm the efficiency of the SVM binary classification of spatial data.

Contents

[1] Kanevski@idiap.ch

Introduction

The report present first results on evaluation of SVM by using geostatistical models of conditional spatial simulations. Recently SVM were successfully applied to the problem of binary and multi-class classification of environmental and pollution data [Gilardi et al 2000, Kanevski et al 2000]. It was shown that SVM are flexible data-adaptive classification models and can be adapted to environmental spatial data. Deterministic outputs of SVM classification were compared with geostatistical probabilistic model indicator kriging. Indicator kriging is a basic model for the estimation of a nonparametric local probability density functions [Goovaerts 1997; Deutsch and Journel 1998, Chiles and Delfiner 1999] and nonparametric method for the classification of spatial data. Indicator kriging gives unique the "best" estimate of the class probability. The main objective of this paper is to apply conditional stochastic simulation model for binary classification problem and to generate many equally probable realizations of the simulator. Analysis (post-processing) of similarity and dissimilarity between realizations describes spatial uncertainty and variability of the spatial data. Sequential indicator algorithms (SISIM) is used to generate realizations of categorical (discrete) variable. SISIM models reproduces one and two-points (described by variograms) statistics. For the completeness of work the basis of SVM binary classification and indicator conditional stochastic simulations are presented.

In general the problem is the following. There are spatially distributed categorical (2 classes) data (high and low level of porosity). The objective is to complete an original pattern (data measured on a nonhomogeneous monitoring network) and to prepare a map of classes (predictions on a dense regular grid). SVM binary model is used in a manner presented in [Gilardi et al 1999, Kanevski et al 2000]. Conditional stochastic model using sequential indicator simulation approach is applied to these categorical data to generate many equally probable realizations. Conditional simulations satisfy the following conditions: 1) simulated realizations reproduce representative histogram (after declustering if necessary) of the original data; 2) simulated realizations reproduce spatial variability described by semivariograms; 3) conditional simulations honor the data - at the sampling points simulated values equals to the original/measured data.

The realisations are determined by the conditional data, simulation model and random seed. From this point of view they are "equally probable".

Support Vector Machines Classification

Let us present short description of SVM application to the classification problems. Detailed theoretical presentation of the SVM can be found in Burgess (1998) and Vapnik (1998) on which the presentation below is based.

Traditional introduction to the SVM classification is the following: 1) binary (2 class) classification of linearly separable problem; 2) binary classification of linearly non-separable problem, 3) non-linear binary problem 4) generalisations to the multi-class classification problems. First results on application of support vector classifiers (binary classification of pollution data, multi-class classification of environmental soil types data) can be found in Gilardi et al. (2000) and Kanevski et al. (2000a).

The following problem is considered. A set S of points (x_i) is given in R^2 (we are working in a two dimensional $x_i = [x_1, x_2]$ space). Each point x_i belongs to either of two classes and is labeled by $y_i \in \{-1,+1\}$. The objective is to establish an equation of a hyper-plane that divides S leaving all the points of the same class on the same side while maximising the minimum distance between either of the two classes and the hyper-plane – maximum margin hyper-plane.

Optimal hyper-plane with the largest margins between classes is a solution of the constrained optimisation problem considered below.

Linearly separable case

Let us remind that data set S is linearly separable if there exist $W \in R^2, b \in R$, such that:

$$Y_i(W^T X_i + b) \geq +1, \quad i = 1,...N \tag{1}$$

The pair (W,b) defines a hyper-plane of equation $(W^T X + b) = 0$.

Linearly separable problem: Given the training sample $\{X_i, Y_i\}$ find the optimum values of the weight vector W and bias b such that they satisfy constraints:

$$Y_i(W^T X_i + b) \geq +1, \quad i = 1,...N \tag{2}$$

And the weight vector W minimises the cost function (maximisation of the margins):

$$F(W) = W^T W / 2 \tag{3}$$

The cost function is a convex function of W and the constraints are linear in W.

This constrained optimization problem can be solved by using Lagrange multipliers. Lagrange function is defined by:

$$L(W, b, \alpha) = W^T X / 2 - \sum_{i=1}^{N} \alpha_i \left[Y_i (W^T X_i + b) - 1 \right]$$

where Lagrange multipliers $\alpha_i \geq 0$.

The solution of the constrained optimisation problem is determined by the saddle point of the Lagrangian function $L(W, b, \alpha)$ which has to be minimised with respect to W and b and to be maximised with respect to α .

Application of optimality condition to the Lagrangian function yields:

$$W = \sum_{i=1}^{N} \alpha_i Y_i X_i \tag{4}$$

$$\sum_{i=1}^{N} \alpha_i Y_i = 0 \tag{5}$$

Thus, the solution vector W is defined in terms of an expansion that involves the N training data. Because of constrained optimisation problem deals with a convex cost function, it is possible to construct dual optimisation problem. The dual problem has the same optimal value as the primal problem, but with the Lagrange multipliers providing the optimal solution.

The dual problem is formulated as follows: maximise the objective function:

$$Q(\alpha) = \sum_{i=1}^{N} \alpha_i - (1/2) \sum_{i=1}^{N} \alpha_i \alpha_j Y_i Y_j X_i^T X_j \tag{6}$$

Subject to the constraints:

$$\sum_{i=1}^{N} \alpha_i Y_i = 0 \tag{7}$$

$$\alpha_i \geq 0, \mathrm{i} = 1,...\mathrm{N} \tag{8}$$

Note that the dual problem is presented only in terms of the training data. Moreover, the objective function $Q(\alpha)$ to be maximized depends only on the input patterns in the form of a set of dot products $\{X_i^T X_j\}_{i=1,2,..N}$.

After determining optimal Lagrange multipliers α_{i0} , the optimum weight vector is defined by (4) and the bias is calculated as follows:

$$b = 1 - W^T X_i^S, \text{ for } Y^{(s)} = +1$$

Note that from the Kuhn-Tucker conditions it follows that:

$$\alpha_i \left[Y_i (W^T X_i + b) - 1 \right] = 0 \tag{9}$$

Only α_i that can be nonzero in this equation are those for which constraints are satisfied with the equality sign. The corresponding points X_i , called *Support Vectors*, are the points of the set $\boldsymbol{S}$ closest to the optimal separating hyper-plane. In many applications number of support vectors is much less that original data points. The problem of classifying a new data point X is simply solved by computing:

$$F(X) = sign(W^T X_i + b) \qquad (10)$$

with the optimal weights *W* and bias *b*.

SVM classification of non-separable data: Soft margin classifier

In case of linearly non-separable set it is not possible to construct a separating hyper-plane without allowing classification error. The margin of separation between classes is said to be soft if training data points violate the condition of linear separability and the primal optimization problem is changed by using slack variables.
Problem is posed as follows: given the training sample $\{X_i, Y_i\}$ find the optimum values of the weight vector *W* and bias *b* such that they satisfy constraints:

$$Y_i(W^T X_i + b) \geq +1 - \xi_i, \ \xi_i \geq 0, \forall i \qquad (11)$$

The weight vector *W* and the slack variables ξ_i minimize the cost function:

$$F(W) = W^T W / 2 + C\sum_{i=1}^{N} \xi_i \qquad (12)$$

where C is a user specified parameter (regularisation parameter is proportional to 1/C).

The dual optimization problem is the following: given the training data maximize the objective function (find the Lagrange multipliers):

$$Q(\alpha) = \sum_{i=1}^{N} \alpha_i - (1/2)\sum_{i=1}^{N} \alpha_i \alpha_j Y_i Y_j X_i^T X_j \qquad (13)$$

subject to the constraints (7) and:

$$0 \leq \alpha_i \leq C, \mathrm{i} = 1,...\mathrm{N} \qquad (14)$$

Note that neither the slack variables nor their Lagrange multipliers appear in the dual optimization problem.

The parameter C controls the trade-off between complexity of the machine and the number of non-separable points. The parameter C has to be selected by the user. This can be done usually in one of two ways: 1) C is determined experimentally via the standard use of a training and testing data sets, which is a form of re-sampling; and 2) It is determined analytically by estimating VC dimension and then by using bounds on the generalization performance of the machine based on a VC dimension (Vapnik, 1998).

SVM non-linear classification

In most practical situations the classification problems are non-linear and the hypothesis of linear separation in the input space is too restrictive. The basic idea of SVM is: 1) to map the data into a high dimensional feature space (possibly of infinite dimension) via a non-linear mapping and 2) construction of an optimal hyper-plane (application of the linear algorithms described above) for separating features. The first item is in agreement of Cover's theorem on the separability of patterns which states that input multidimensional space may be transformed into a new feature space where the patterns are linearly separable with high probability, provided: 1) the transformation is non-linear; and 2) the dimensionality of the feature space is high enough (Haykin, 1999). Cover's theorem does not discuss the optimality of the separating hyper-plane. By using Vapnik's optimal separating hyper-plane VC dimension is minimised and generalisation is achieved. Let us remind that in the linear case the procedure requires only the evaluation of dot products.

Let $\{\varphi_j(x)\}_{j=1,...m}$ denote a set of non-linear transformation from the input space to the feature space; *m* – is a dimension of the feature space. Non-linear transformation is defined a priori.

In the non-linear case the optimization problem in the dual form is following: given the training data maximize the objective function (find the Lagrange multipliers):

$$Q(\alpha) = \sum_{i=1}^{N} \alpha_i - (1/2)\sum_{i=1}^{N} \alpha_i \alpha_j Y_i Y_j K(X_i^T X_j) \qquad (15)$$

Subject to the constraints (7) and (14), the kernel in (15) is:

$$K(X,Y) = \varphi^T(X)\varphi(Y) = \sum_{j=1}^{m} \varphi_j(X)\varphi_j(Y) \quad (16)$$

Thus, we may use inner-product kernel *K(X,Y)* to construct the optimal hyper-plane in the feature space without having to consider the feature space itself in explicit form.

The optimal hyper-plane is now defined as:

$$f(X) = \sum_{j=1}^{N} \alpha_j Y_j K(X, X_j) + b \quad (17)$$

Finally, the non-linear decision function is defined by the following relationship:

$$F(X) = sign\left[W^T K(X, X_j) + b\right] \quad (18)$$

The requirement on the kernel $K(X, X_j)$ is to satisfy Mercer's conditions (Vapnik 1998). Three common types of Support Vector Machines are widely used:

Polynomial kernel:

$$K(X, X_j) = (X^T X_i + 1)^p \quad (19)$$

where power *p* is specified a priori by the user. Mercer's conditions are always satisfied.

Radial basis function (RBF) kernel:

$$K(X, X_j) = \exp\left\{-\left\|X - X_j\right\|^2 / 2\sigma^2\right\} \quad (20)$$

where the kernel bandwidth σ (sigma value) is specified a priori by the user. In general, Mahalanobis distance can be used. Mercer's conditions are always satisfied.

Two-layer perceptron:

$$K(X, X_j) = \tanh\left\{\beta_0 X^T X_j + \beta_0\right\} \quad (21)$$

Mercer's conditions are satisfied only for some values of β_0 β_1.

For all three kernels (learning machines), the dimensionality of the feature space is determined by the number of support vectors extracted from the training data by the solution to the constrained optimization problem. In contrast to RBF neural networks, the number of radial basis functions and their centers are determined automatically by the number of support vectors and their values. In the present study only the results obtained with the RBF kernel are presented.

SVM usually are trained (tuning of hyperparameters, like kernel bandwidth in RN`BF kernel) by splitting data into training and testing data sets. The same technique was applied in this study. Details can be found in [Kanevski et al 2000]

Conditional Stochastic Simulations

Principles of Sequential Simulations

Sequential simulation is the only truly general simulation algorithm. The realizations are continuous functions (diffusive models), or piecewise continuous with fixed or random discontinuities (jump models). The idea of sequential simulations is well known and was introduced to geostatistical society by Alabert and Massonat in 1990. We present sequential simulation approach following presentation in [Chiles and Delfiner, 1999].

Let us consider a vector-valued random variable $\boldsymbol{Z=(Z_1, Z_2, \ldots, Z_N)^T}$ for which a realization of the subvector $\boldsymbol{(Z_1, Z_2, \ldots, Z_M)^T}$ is known and equal to $\boldsymbol{(z_1, z_2, \ldots, z_M)^T}$ ($0 \leq M < N$). The distribution of the vector $\boldsymbol{Z}$ conditional on $Z_i = z_i$ (I=1,2,…,M) can be factorized in the form

$$
\begin{aligned}
&\Pr\{z_{M+1} \le Z_{M+1} < z_{M+1} + dz_{M+1}, \ldots z_{N+1} \le Z_{N+1} < z_{N+1} + dz_{N+1} \,|\, z_1, \ldots z_M\} \\
&= \Pr\{z_{M+1} \le Z_{M+1} < z_{M+1} + dz_{M+1} \,|\, z_1, \ldots z_M\} \\
&* \Pr\{z_{M+2} \le Z_{M+2} < z_{M+2} + dz_{M+2} \,|\, z_1, \ldots z_M, z_{M+1}\} \\
&\bullet \\
&\bullet \\
&\bullet \\
&* \Pr\{z_N \le Z_N < z_N + dz_N \,|\, z_1, \ldots z_M, z_{M+1}, \ldots z_{N-1}\}
\end{aligned}
\qquad (22)
$$

Using this factorization random vector $\boldsymbol{Z}$ can be simulated sequentially by randomly selecting $\boldsymbol{Z_i}$ from the conditional distribution $\Pr\{Z_i < z_i \mid z_1, z_2, \ldots z_{i-1}\}$ for i= M+1,…, N. and including the outcome z_i in the conditioning data set for the next step.

This procedure of decomposition of joint pdf into product of conditional pdfs is very general and can be used for spatial random functions as well. Let us remind that spatial function is a collection of random variables. It makes possible the construction of both a nonconditional (M=0) and conditional (M>0) simulations. The same procedure can be applied to co-simulation of several nonindependent random functions. It produces simulations that match not only the covariance but also the spatial distribution. In general, it is not known where to take conditional distributions. But for a Gaussian random function with known mean, the conditional distribution is Gaussian with mean and variance obtained from simple kriging.

Sequential simulation is a theoretically simple and general simulation algorithm that is conditional by construction. Due to implementation problems some approximations are needed. The tests showed that these approximations do not have significant impact on the reproduction of the underlying Gaussian model [Gomez-Hernandez and Cassiraga 1994].

Sequential Indicator Simulations

The model is based on the principles of sequential simulation approach. But instead of working with continuous Gaussian random function we are working with indicator transformed data (or classes). Sequential indicator simulation (SIS) can be used both for the nonparametric simulations of continuous random fields, as well as for categorical variables. As in the case of indicator kriging, the indicator approach allows to account for class-specific patterns of spatial continuity through different indicator variogram models. Let us remind that in Gaussian simulation spatial variability is characterized by a single semivariogram function.

Indicator simulations are well suited for categorical variables simulations as well as for continuous variables.

The procedure of SIS of continuous variable is the following (see, e.g. Goovaerts 1997):

Discretize the range of variation into z into (K+1) classes using K threshold z_k. Transform each datum $z(\mathbf{u}_\alpha)$ into a vector of hard indicator data:

$$
i(u_\alpha; z_k) = \begin{cases} 1 & \text{if } z(u_\alpha) \le z_k \quad k = 1, \ldots K \\ 0 & \text{otherwise} \end{cases} \qquad (23)
$$

Define a random path visiting each node of the grid only once.

At each node:

Determine the K ccdf values $[F(\mathbf{u}'; z_k|(n)]^*$ using any of the indicator kriging algorithms: simple, ordinary, median, indicator co-kriging or probability kriging. The conditioning information consists of indicator transforms (and uniform transforms for probability kriging) of neighboring original z-data and previously simulated z-values.

Correct for any order relation deviations. Then build a complete ccdf model $[F(\mathbf{u}'; z|(n)]^*$, $\forall$ z, using the interpolation/extrapolation algorithms

Draw a simulated value from that ccdf

Add the simulated value to the conditioning data set

Proceed to the next node along the random path.

Repeat the entire procedure with a different random path to generate another realization.

At each node, the simulation can be considered as a two steps procedure: a simulated class-value (thresholds) is first assigned to the grid node; a simulated value is then drawn from that class. Consequently, indicator-based algorithms guarantee approximate reproduction of the K class proportions and corresponding indicator semivariograms and not reproduction of the cdf and semivariogram of the continuous z-values. Actually, approximation of one-point and two-point statistics by SIS depends on several factors: discretization level (number of thresholds), indicator kriging procedure; interpolation/extrapolation models used to increase resolution of modeled cdf.

Simulation of Categorical Variables

Let us consider simulation of categorical variables[Deutsch and Journel 1997]. By definition categorical spatial function consists of K mutually exclusive categories s_k $k= 1,\ldots,K$. At any location only one class can be detected. Let $i(u; s_k)$ be the indicator of category s_k set it to 1 if $u \in s_k$ and zero otherwise.

Direct kriging of the indicator variable $i(u; s_k)$ provides an estimate/model for the probability that s_k prevails at location u.

$$\Pr ob^{*}\{I(u;s_k)=1|(n)\}=p_k+\sum_{\alpha=1}^{n}\lambda_{\alpha}\left[I(u_{\alpha};s_k)-p_k\right] \qquad (25)$$

where $p_k = E\{I(u;s_k)\} \in [0,1]$ is the marginal frequency of category s_k inferred e.g. from the declustered proportion of data of type s_k . The weights are given by simple indicator kriging equations using indicator covariances of corresponding classes.

When the average proportions vary locally, one can explicitly provide the simple indicator kriging with smoothly varying local proportions.

The procedure of sequential simulation of categorical variables implemented in GSLIB is the following.

At each node u along the random path, indicator kriging followed by order relation correction provides K estimated probabilities $p_k(u|(ci)), k= 1,\ldots K$. The conditioning information (ci) consists of both the original data and the previously simulated indicator values for categories s_k

Next, define any ordering of the K categories, say 1,..K. This ordering defines a cdf-type scaling of the probability interval [0,1] with K intervals.

Draw a random number p uniformly in [0,1]. The interval in which p falls determines the simulated category at location u.

Update all K indicator data sets with this new simulated information, and proceed to the next location along with the random path.

The arbitrary ordering of the K probabilities does not affect which category is drawn nor the spatial distribution of categories, because of the uniform distribution of p.

Indicator Simulation of Continuous Variables

In accordance with [Goovaerts 1997] the sequential indicator simulations of continuous variable Z at N grid nodes can be carried out as follows:

Discretize the original data Z into (K+1) classes using K threshold values z_k – transformation into hard indicator data.

$$i(u_{\alpha};z_k)=\begin{cases}1, \text{ if } z(u_{\alpha})\le z_k\\ 0 \text{ otherwise}\end{cases}, \quad k=1,\ldots,K \qquad (26)$$

Define a random path visiting each node of the grid only once

At each node determine the K ccdf values using any kind of indicator kriging: simple, ordinary, median, indicator cokriging. The conditioning information consists of indicator transforms of neighboring original z-data and previously simulated z-values. Correct for any order relation deviations and then build a complete ccdf model using interpolation/extrapolation algorithms. Draw a simulated value z from that ccdf. Add the simulated value to the conditioning data set. Proceed to the next node and repeat corresponding steps.

Repeat the entire procedure with a different random path to generate another realization.

Thus, the indicator based simulation algorithm can be viewed as a two-step procedure: 1) simulation class-value; 2) draw a simulation value from that class using some within class distribution models (e.g., uniform, power, etc.). Consequently, indicator simulations guarantee approximate reproduction of only the K class proportions and corresponding indicator semivariograms and not reproduction of the cdf and semivariogram of the original continuous z-values. Therefore, actual approximation of one-point and two-point z-statistics by sequential indicator realization depends on several factors: number of thresholds, information accounted when performing indicator kriging, interpolation/extrapolation models used for increasing the resolution of ccdf.

Case Study

Case study is based on a porosity data set [Kanevski et al 2000]. The same problem of binary classification is considered. Original data were split into model development data set (200 samples wewre used to develop classification model) and validation data set (94 samples weree used only to estimate generalization abilities of the model). In case of SVM classification model development data set was split into training and testing (in order to tune hyperparameters) data sets.

Description of data

Model development and validation data sets (2 class problem) are presented in Figure 1. Random splitting procedure was used.

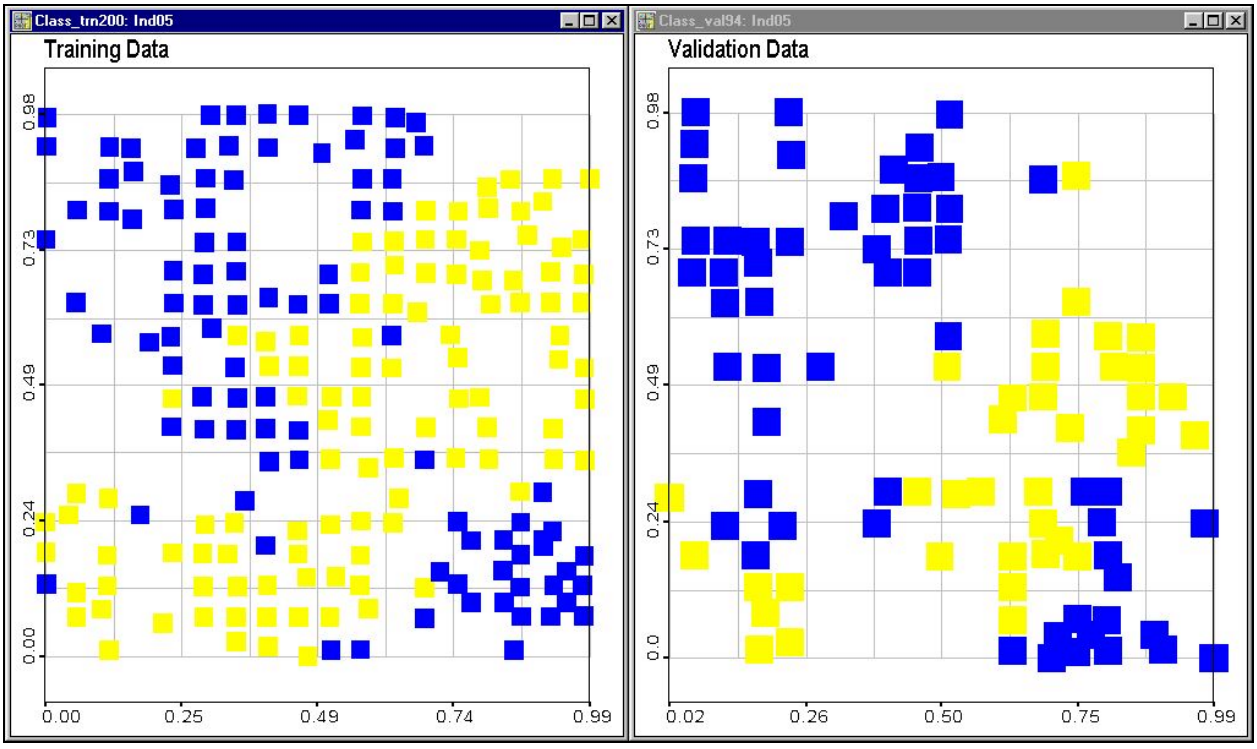

Figure 1. Model development and validation data sets potplots.

Results

The spatial data classification methodology consist of several steps, including exploratory data analysis, structural analysis (variography), model development, validation of model [see details in Kanevski 2000].

Variogram is widely used in geostatistics as a measure of spatial continuity of Random Function $Z(\mathbf{x})$ and is defined as:

$$\gamma(\mathbf{h}) = \frac{1}{2} Var \{Z(\mathbf{x}) - Z(\mathbf{x}+\mathbf{h})\} \qquad (27)$$

where **h** is a separation vector between two data points in space. In the case of the intrinsic hypotheses, the semivariogram is assumed to exist and depends only on the separation vector between pairs of values separated at that particular lag distance. The empirical estimate of the semivariogram is given by:

$$\gamma(\mathbf{h}) = \frac{1}{2N(\mathbf{h})} \sum_{i=1}^{N(\mathbf{h})} \left(Z_i(\mathbf{x}) - Z_i(\mathbf{x}+\mathbf{h})\right)^2 \tag{28}$$

where $N(\mathbf{h})$ is a number of pairs separated by the vector $\mathbf{h}$.

A common method to detect the possible presence of an anisotropy in the underlying data set is via the spatial correlation map, which is made up of semivariogram values computed for the different separation vectors Variogram roses (variograms computed in different directions and at different lag distances) for the categorical raw data set and SVM classification are presented in Figure 2. Details of the <SVM training can be found in [Kanevski 2000]. For the present comparison the following parameters of the SVM were sued: kernel bandwidth = 0.25 and C parameter = 1e6.

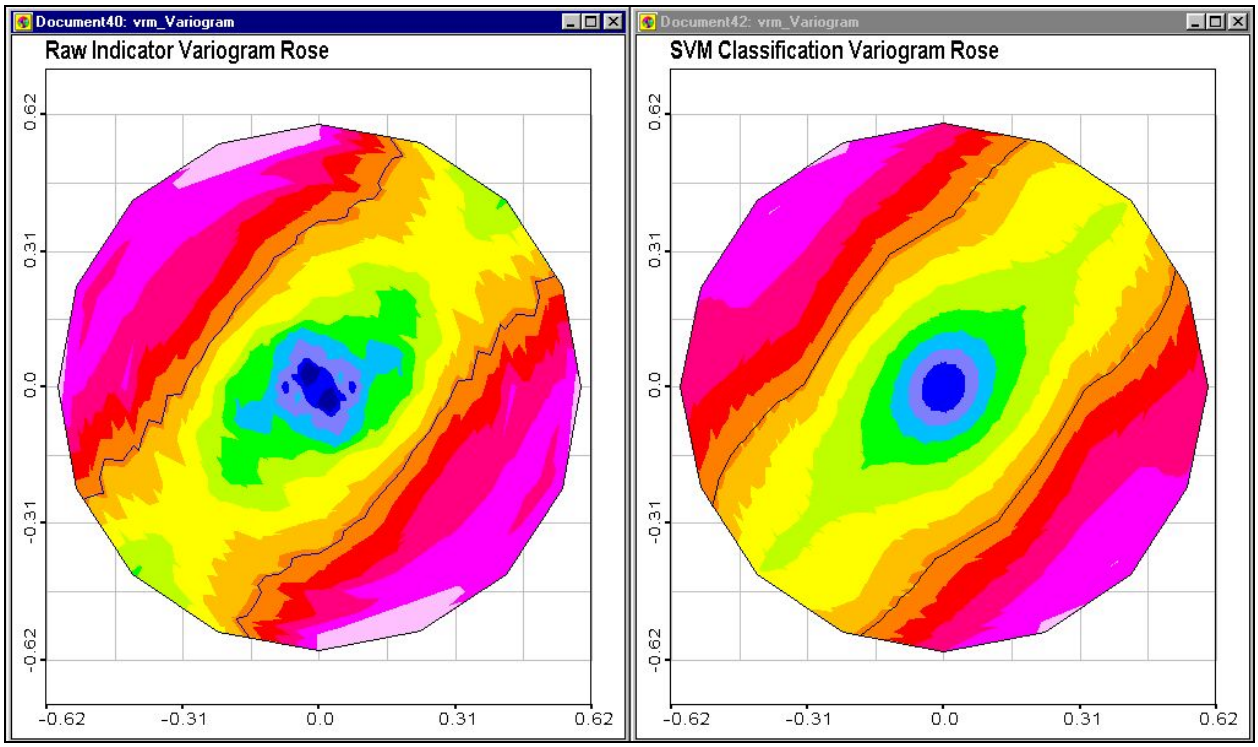

Figure 2. Variogram roses of model development and SVM classification data sets.

Theoretical anisotropic model was developed for the indicator data. This model was used in the sequential indicator kriging. Several realizations along with corresponding variogram roses are presented in Figures 3-8. SISIM realizations rather well reproduced variogram structures of the data. Fluctuations from realization to realization can be recognized.

In the Figure 10 SVM classification is presented along with conditional standard deviation of the SISIM model. In this case standard deviation describes uncertainty of the boundary between classes.

The results of SVM classification were compared with E-type estimates – averaged of 100 SISIM realizations (Figure 11).

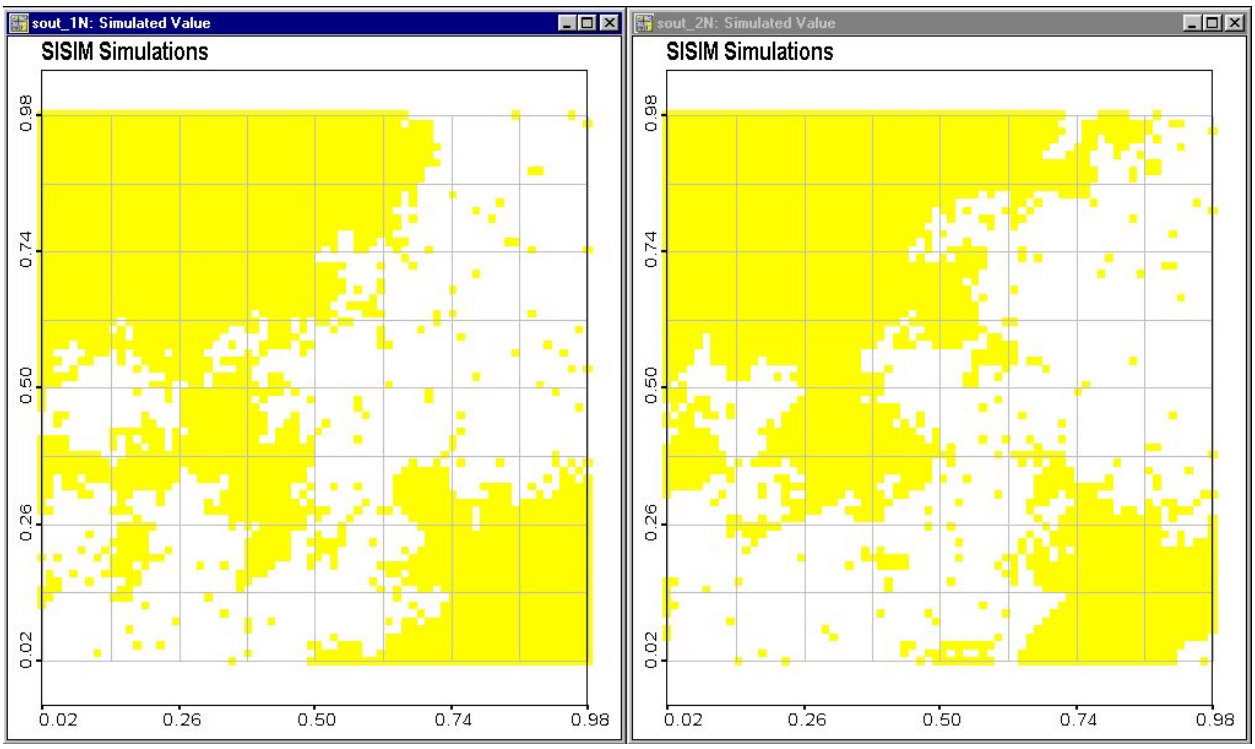

Figure 3. Realizations 1 & 2 of the SISIM model.

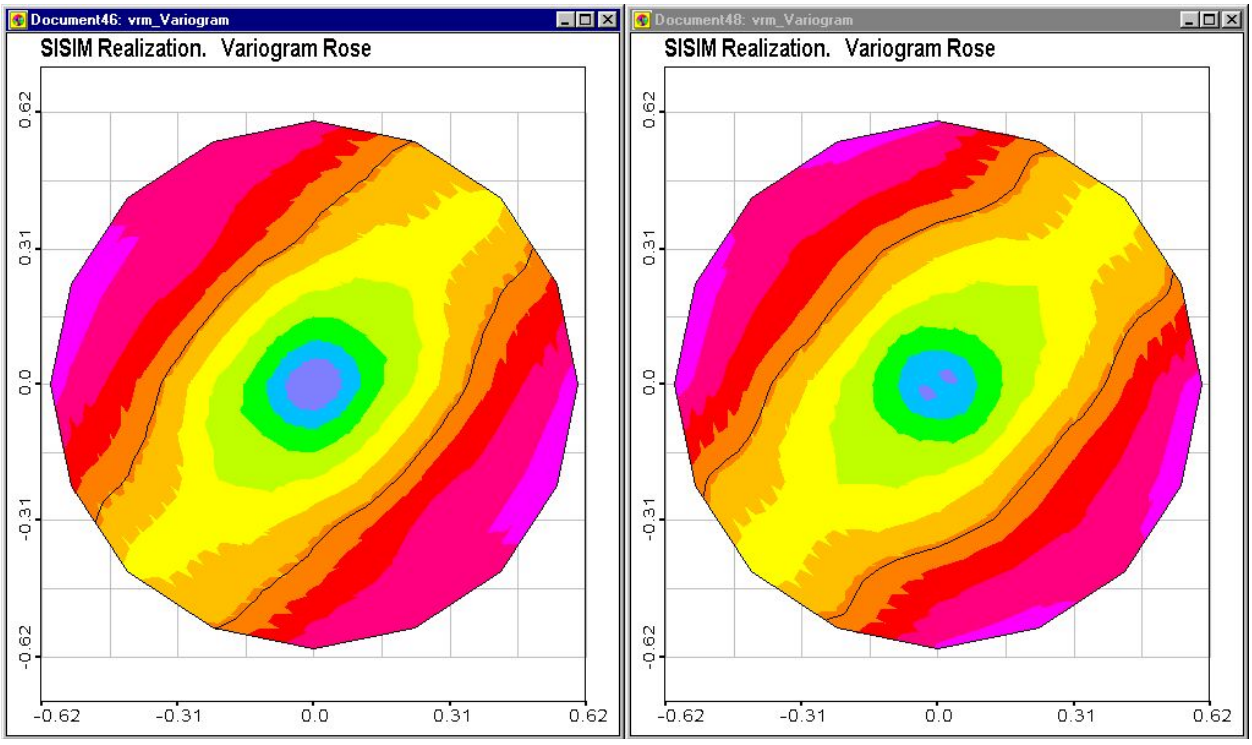

Figure 4. Variogram Roses of the realizations 1 & 2.

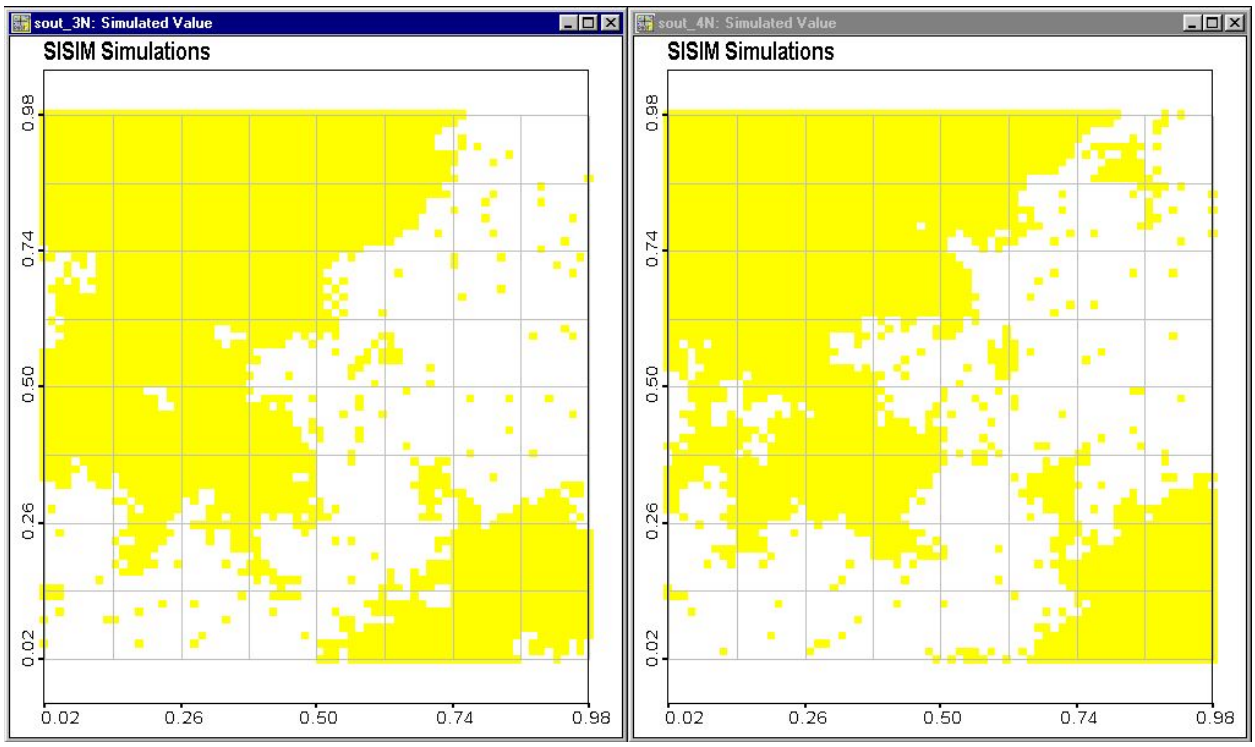

Figure 5. Realizations 3 & 4 of the SISIM model.

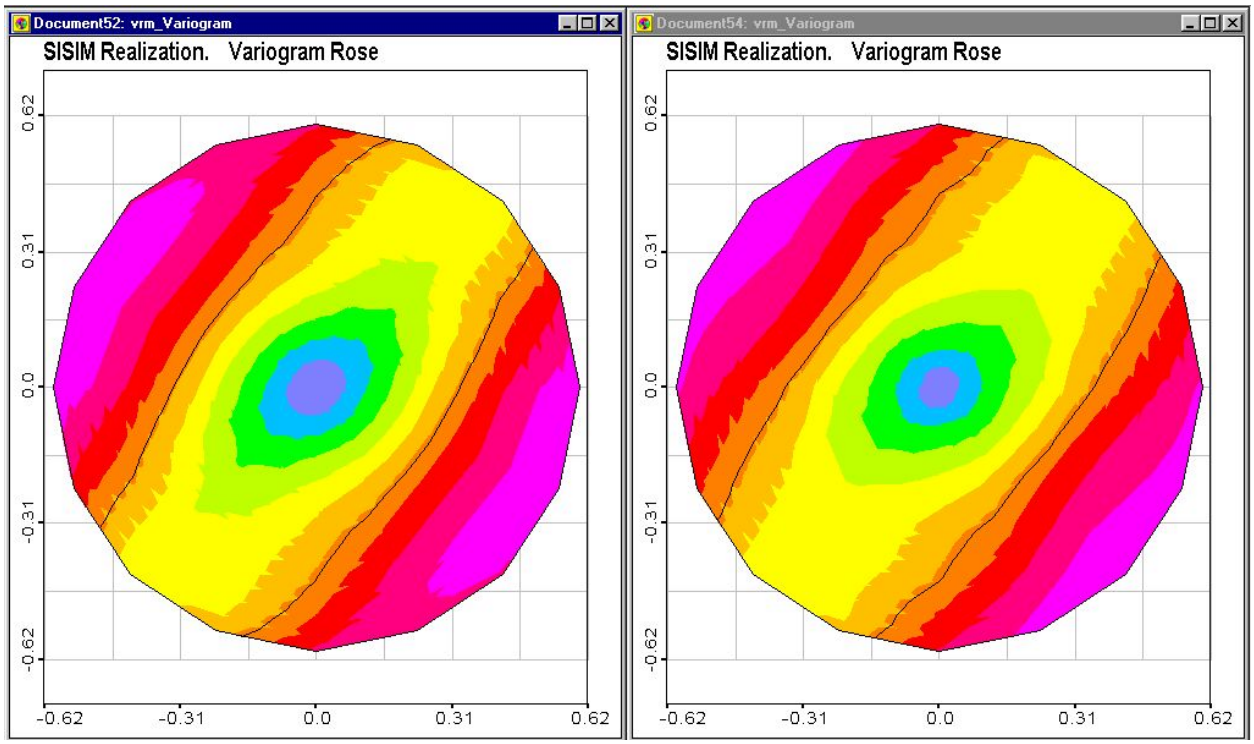

Figure 6. Variogram Roses of the realizations 3 & 4.

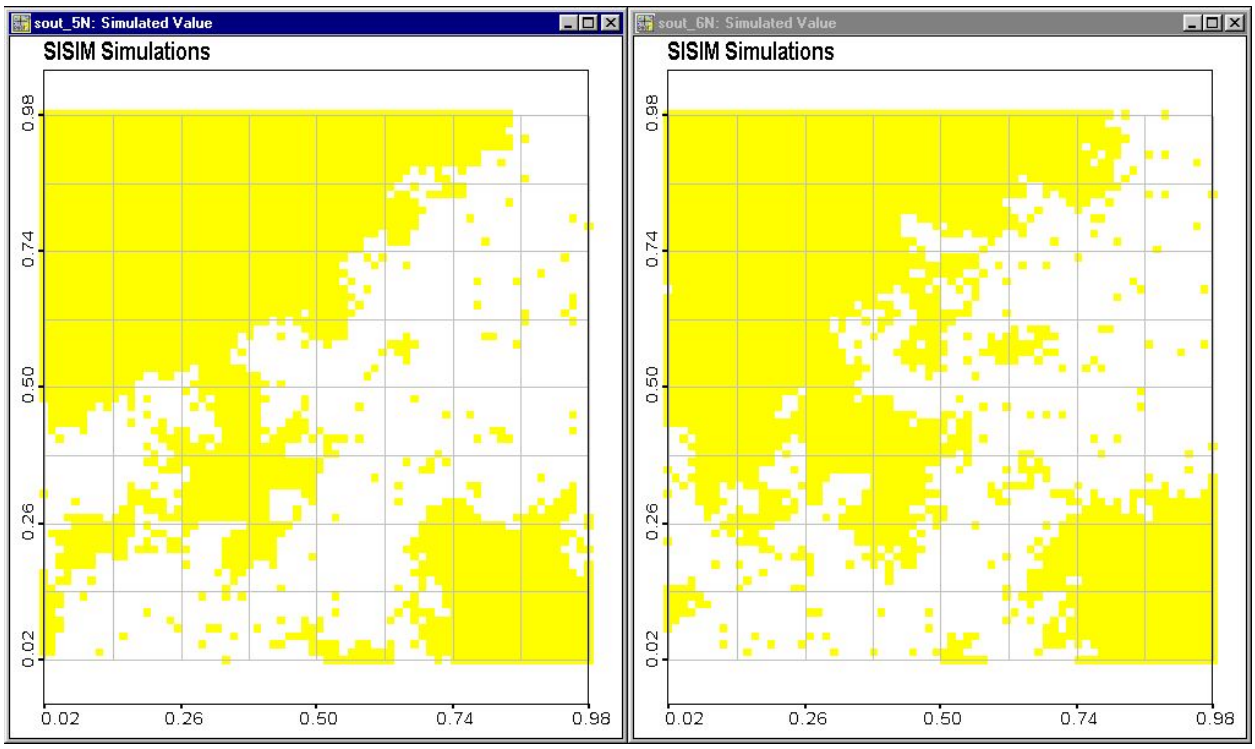

Figure 7. Realizations 5 & 6 of the SISIM model.

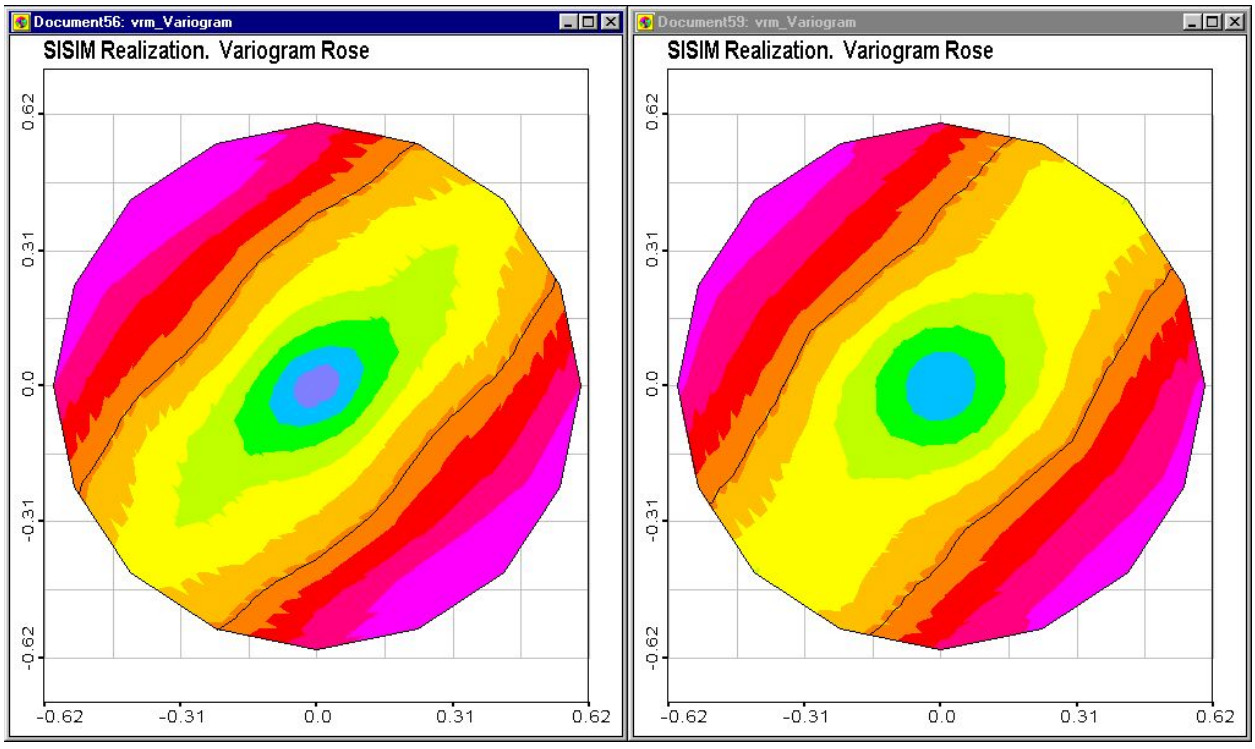

Figure 8. Variogram Roses of the realizations 5 & 6.

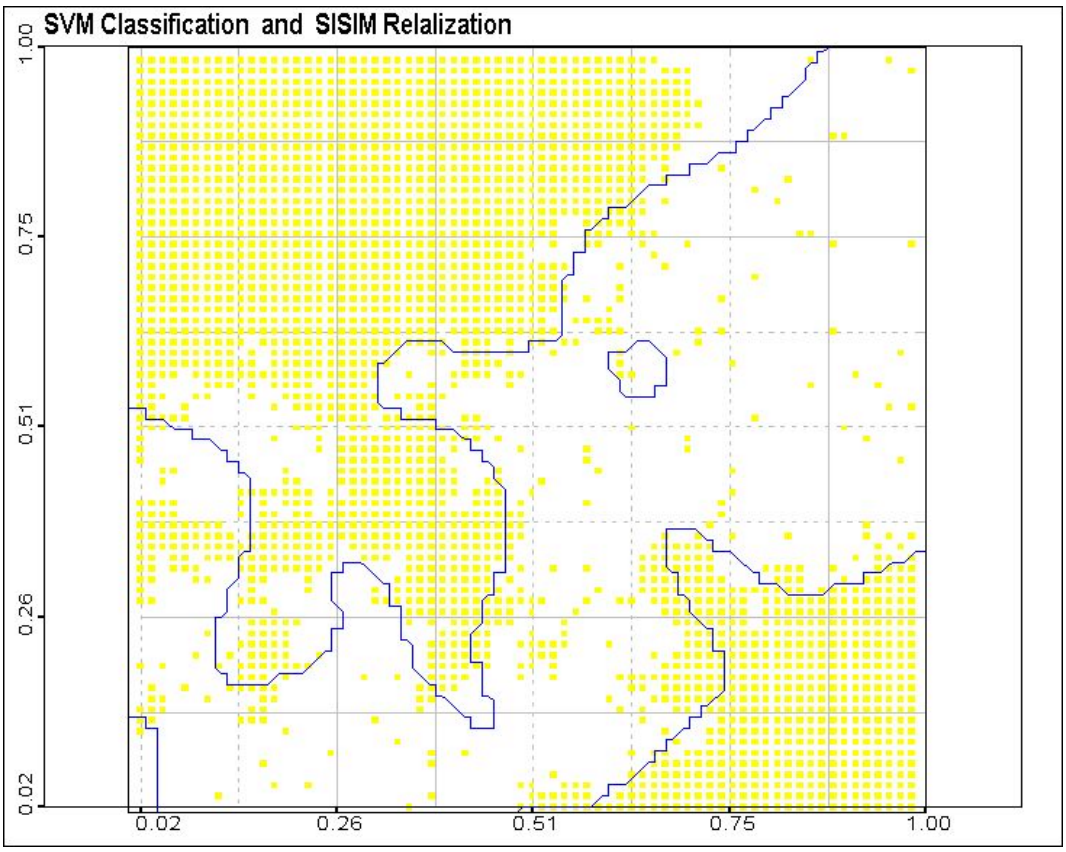

Figure 9. Example of the SVM classification and SISIM realization 1.

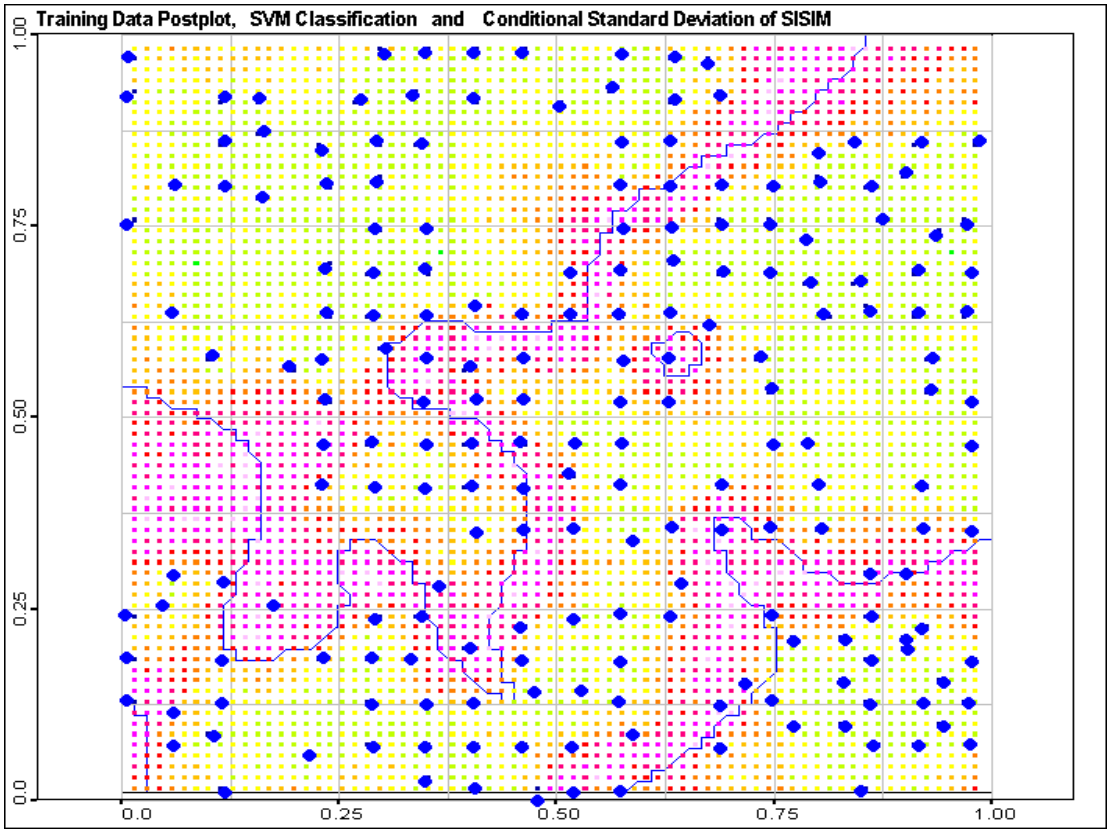

Figure 10. SVM classification, model development data postplot and conditional standard deviation of the SISIM model.

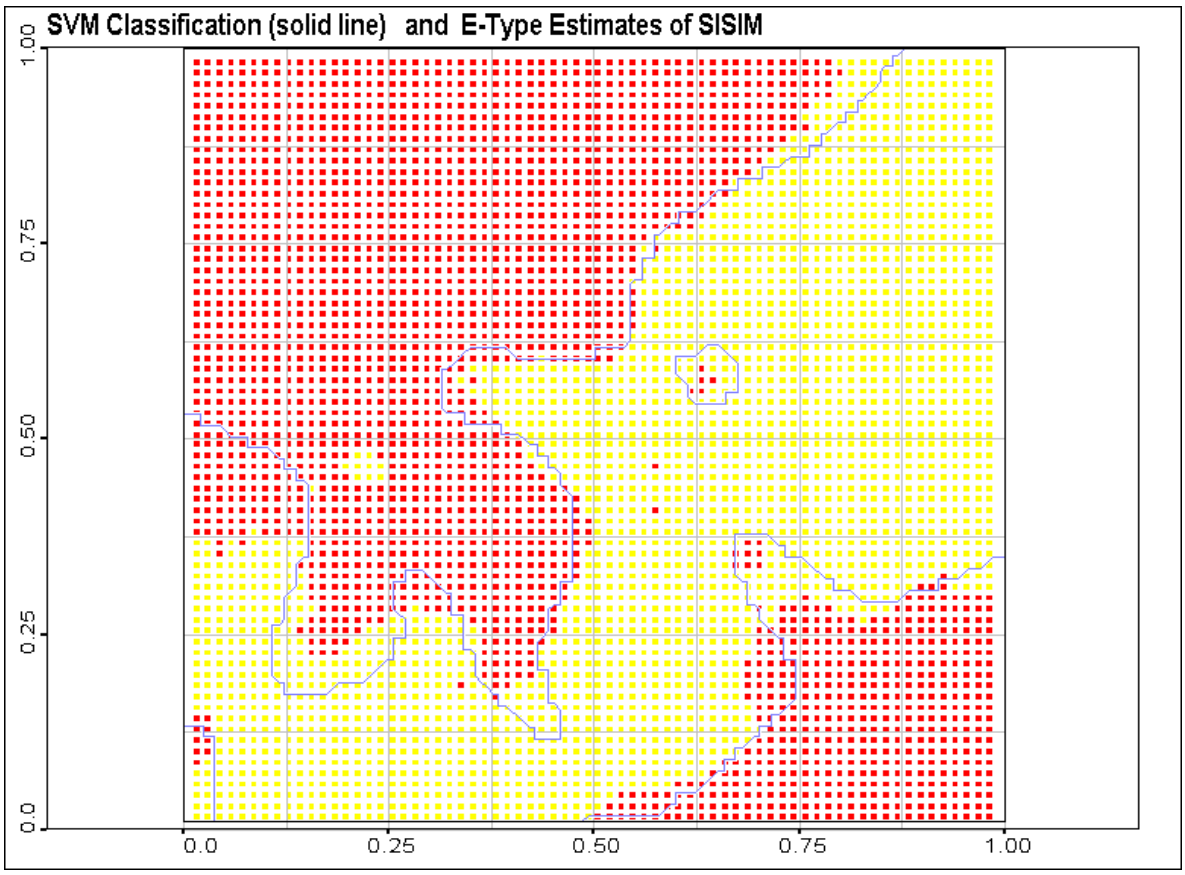

Figure 11. SVM Classification and E-type estimate of the SISIM model.

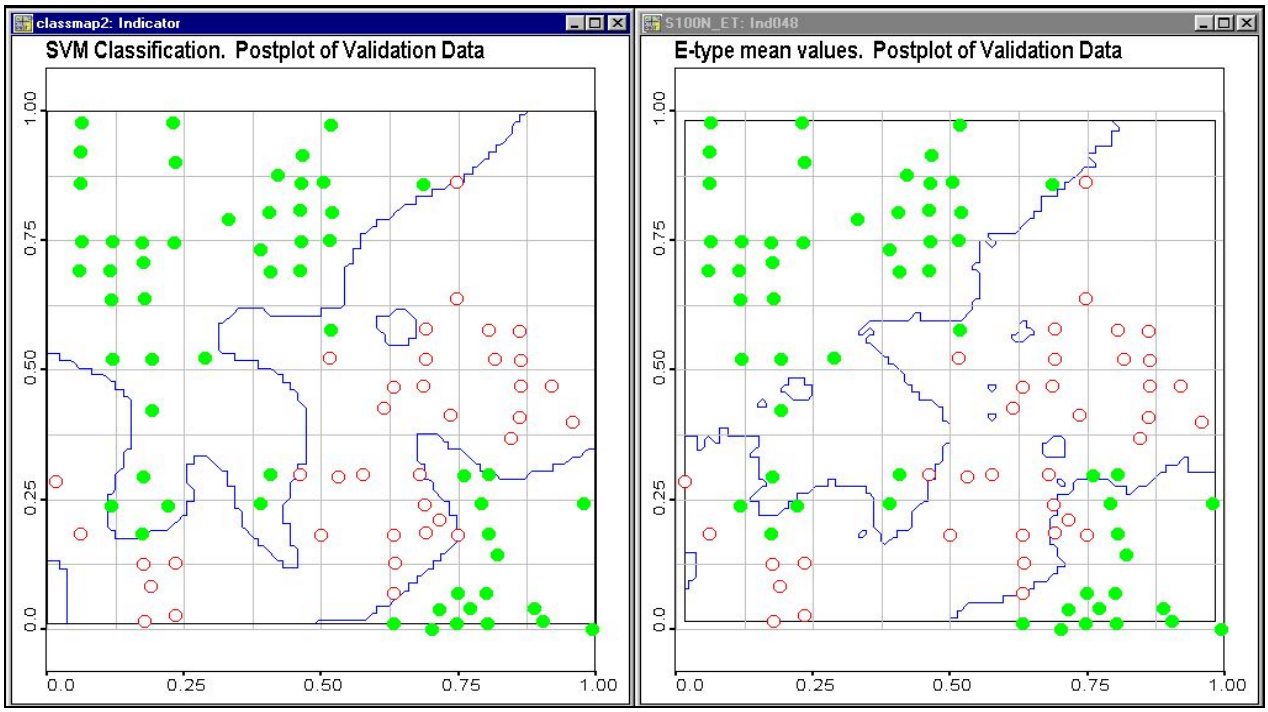

Figure 12. SVM Classification (right) and E-type estimates of SISIM (left) along with postplot of validation data set.

The results of SVM classification and EW-type estimates are presented in Figure 12 along with validation data set. From the point of view of validation set classification, SVM performs a bit better (let us remind that the objective of the simulations is not the best predictions, but reproducing of variability and uncertainty).

Conclusions

The first results on the comparison of SVM binary classification and sequential indicator simulations are presented. The results demonstrate that SVM model fits well within the framework of simulation uncertainties. It should be noted, that comparison is valid until two-points statistics used in the SISIM model.

Comparisons on multi-class classification problem using the same methodology seems to be interesting because SISIM model explicitly takes into account spatial variability of classes by modeling corresponding variograms of categorical variables. Comprehensive numerical experiments on binary and multiclass classification with SVM and stochastic simulations are in progress.

Acknowledgements.

The work was supported in part by INTAS grants 97-31726 and 99-00099. Geostat Office software[Kanevski et al 1999] was used for structural analysis and presentation of the results.

References

Burgess, C. A Tutorial on Support Vector Machines for Pattern Recognition. Data mining and knowledge discovery, 1998.

Chiles J.P. and Delfiner P. Geostatistics. Modeling Spatial Uncertainty. A Wiley-Interscience Publication. New York, 1999.

Deutsch C. and Journel A. GSLIB. Geostatistical software Library and User's Guide. Second Edition. Oxford University Press, Mew York, 369 p. 1998.

Goovaerts P. Geostatistics for Natural Resources Evaluation. Oxford University Press, 483 p, 1997.

Gilardi N., Kanevski M., Maignan. Spatial Data Classification with Support Vector Machines. Geostatistical Congress 2000, Paper T14, 2000.

Gomez-Hernandez J.J, and Cassiraga E.F. Theory and Practice of Sequential Simulations. In: Armstrong M. and Dowd P. 1994. pp. 111-124.

Kanevski, M, Demyanov, V., Chernov, S., Savelieva, E., Serov, A., Timonin, V. and Maignan, M. Geostat Office for Environmental and Pollution Spatial Data Analysis. Mathematische Geologie, N3, April, 73-83, 1999.

Kanevski, M., Pozdnukhov, A., Canu, S. and Maignan, M. Advanced Spatial Data Analysis and Modelling with Support Vector Machines. IDIAP Research Report, RR-00-31, 2000.

Vapnik, V. Statistical Learning Theory. John Wiley & Sons, 1998.