An Introduction to Support Vector Classification for Geostatistical Applications

Enrique Gallardo and Oy Leuangthong

Support vector algorithms are fairly new in machine learning. It has been successfully applied to solve a wide variety of problems, such as speech and handwriting recognition, natural language processing, medical diagnosis, stock market analysis, classifying DNA sequences, bioinformatics, etc (Kecman, 2001). Despite its success in these areas, relatively few examples of applying support vector machines to the analysis of spatially distributed data can be found in the literature. This paper provides an introduction to the Support Vector Classification (SVC) algorithm and explores simple examples of its application to the following geostatistical problems: classification of categorical data, post processing of categorical realizations (image cleaning), rapid generation of local refined boundaries, comparison of spaces of uncertainty, and classification of realizations. This paper is the first of a series whose goal is to explore the application of Support Vector Machines to geostatistical problems.

Introduction

The support vector algorithm (Boser, Guyon and Vapnik, 1992) was initially developed for solving classification problems. Soon after, it was extended to deal with regression problems (Muller et al. 1997). In the context of spatially distributed data, the classification problem is similar to the problem of assigning a single category (e.g. facies or rock types) to unsampled locations based on observed data. The goal of SVC is to find a boundary that separates the observed data with a maximum margin as shown in Figure 1. This boundary is then used to classify unsampled locations. According to the structural risk minimization principle developed in statistical learning theory, it is expected that a boundary with a maximum margin will better classify unsampled locations (or better generalizes) compared to a boundary with a narrow margin. The introduction to the support vector classification (SVC) algorithm presented below follows Kecman (2001). To illustrate the algorithm, consider the problem of estimating the category *s* at any location **u** over the domain *A*, based on the following set of *n* observed data:

$$\{s_k(\mathbf{u}_\alpha); \alpha = 1,\ldots,n; k = 1,2\} \tag{1}$$

where k = 1,2 indexes the number of mutually exclusive categories s_1, s_2. The SVC algorithm seeks the weight parameter **w** and bias term *b* that characterizes a decision boundary (or a hyperplane) of the form:

$$\mathbf{w}^{\mathrm{T}}\mathbf{u} + b = 0 \tag{2}$$

The boundary will separate the categories given on the observed data with a maximum margin (Figure 1) and it will assign a single category s_1 or s_2 to the unsampled locations **u** according to the rule:

$$s(\mathbf{u}) = \begin{cases} s_1 & if \quad \mathbf{w}^{\mathrm{T}}\mathbf{u} + b > 0 \\ s_2 & if \quad \mathbf{w}^{\mathrm{T}}\mathbf{u} + b < 0 \end{cases} \tag{3}$$

The implementation of SVC has the following steps: (1) preprocessing of data, (2) SVC training and (3) SVC testing. Each of these steps is described below.

Preprocessing of data. The SVC algorithm requires coding the observed data as:

$$i(\mathbf{u}_\alpha) = \begin{cases} 1 & if \ s(\mathbf{u}_\alpha) = s_1 \\ -1 & \text{if } s(\mathbf{u}_\alpha) = s_2 \end{cases} \tag{4}$$

Note that this coding will help to classify the unsampled locations based on the sign (positive or negative) of the SVC response rather than on its actual absolute value.

SVC training. Finding the weighting parameters **w** and the bias term *b* of the decision boundary (2) using the observed data is referred to as training the SVC. In machine learning jargon the observed data is called the *training set*. The percentage of observed data misclassified by the decision boundary is called *training error or empirical error*. In this paper, the complement of the empirical error to add to unity is called *empirical accuracy.*

The boundary (2) is determined to maximize the margin of separation between the categories s_1 and s_2 (Figure 1). If the data (1) is linearly separable, the optimization problem is expressed as:

$$\begin{aligned} &\underset{\mathbf{w},b}{minimize} \quad \frac{1}{2}\|\mathbf{w}\|^2 \\ &subject\ to \quad i(\mathbf{u}_\alpha)[\mathbf{w}^{\mathrm{T}}\mathbf{u}_\alpha + b] \geq 1 \quad ; \quad \alpha = 1,...,n \end{aligned} \tag{5}$$

where ||**w**|| represents the Euclidean norm of the vector **w**. This nonlinear optimization problem with inequality constrains is solved using the Lagrange formalism and leads to the following results for **w** and *b*:

$$\mathbf{w} = \sum_{\alpha=1}^{n} \eta_\alpha i(\mathbf{u}_\alpha)\mathbf{u}_\alpha \tag{6}$$

$$b = \frac{1}{Nsv}\left(\sum_{\alpha=1}^{Nsv}\left(\frac{1}{i(\mathbf{u}_\alpha)} - \mathbf{u}_\alpha^{\mathrm{T}}\mathbf{w}\right)\right) \quad ; \quad \alpha = 1,...,Nsv \tag{7}$$

where η_α are Lagrange multipliers and *Nsv* is the numbers of support vectors; that is, training data whose η_α are not zero. Substituting (6) into (2) the boundary becomes:

$$\sum_{\alpha=1}^{n} i(\mathbf{u}_\alpha)\eta_\alpha \mathbf{u}^{\mathrm{T}}\mathbf{u}_\alpha + b = 0 \tag{8}$$

An overlap of the categories may indicate that a plane that separates them does not exist. To deal with this case, the linear SVC was adapted (Cortes, 1995; Cortes and Vapnik, 1995) by the introduction of slack variables ξ_α (α = 1,…,*n*) in the optimization problem. The slack variables ξ_α relax the constraints in (5), so, some classification errors are permitted but at a certain cost. Now, the optimization problem is:

$$\begin{aligned} &\underset{\mathbf{w},b,\xi}{minimize} \quad \frac{1}{2}\|\mathbf{w}\|^2 + P\sum_{\alpha=1}^{n}\xi_\alpha \\ &subject\ to \quad i(\mathbf{u}_\alpha)[\mathbf{w}^{\mathrm{T}}\mathbf{u}_\alpha + b] \geq 1 - \xi_\alpha \\ &\qquad\qquad \xi_\alpha \geq 0 \quad ; \quad \alpha = 1,...,n \end{aligned} \tag{9}$$

Here, *P* is a user-defined penalty parameter. The optimization problem has the same solution shown in (6) , (7) and (8), the only difference is the bounds of the multipliers η_α that appear in the Lagrange formalism. To cope with data that is not linearly separable, the vectors **u** are mapped into a higher-dimensional space $\mathcal{F}$ by a function Φ. In the space $\mathcal{F}$, the linear SVC algorithm is applied. The linear classifier in the space $\mathcal{F}$ will create a non-linear decision boundary in the original input space (Figure 2).

The implementation of the SVC algorithm in the space $\mathcal{F}$ is done by using a kernel; this consists of replacing the scalar product between training data with a kernel function in the formulation of the SVC algorithm. The kernel is a function in the input space of the vectors **u**, which returns the dot products of the images in some space $\mathcal{F}$, without even knowing the form of the map Φ:

$$k(\mathbf{u}_\alpha, \mathbf{u}_\beta) = \left\langle \Phi(\mathbf{u}_\alpha), \Phi(\mathbf{u}_\beta) \right\rangle \tag{10}$$

SVC testing. SVC testing means to use the decision boundary found in the training step to allocate a single category *s* to the unsampled location **u** according to the rule:

$$s(\mathbf{u}) = \begin{cases} s_1 & if \quad \left(\sum_{\alpha=1}^{n} i(\mathbf{u}_\alpha)\eta_\alpha K(\mathbf{u},\mathbf{u}_\alpha) + b\right) > 0 \\ s_2 & if \quad \left(\sum_{\alpha=1}^{n} i(\mathbf{u}_\alpha)\eta_\alpha K(\mathbf{u},\mathbf{u}_\alpha) + b\right) < 0 \end{cases} \tag{11}$$

where $K(\mathbf{u}, \mathbf{u}_\alpha)$ is a symmetric positive-definite matrix with the values of the kernel function.

In the machine learning jargon the unsampled locations (or data not used in the training procedure) taken together are called the *testing set*. The percentage of unsampled locations misclassified by the decision boundary is called *testing error or generalization error*. The complement of the generalization error to add to unity is called *generalization accuracy.*

After training and testing the algorithm the final result is the assignment of a single category s_1 or s_2 to every unsampled location in the domain *A*.

Implementations of SVC

Implementing SVC requires choosing a kernel function with its parameters and the penalty parameter *P.* These decisions, also called model selection, are now discussed.

Kernel selection (or kernel design). The design of kernel functions is a very active area of research in support vector machines; the goal is to generate the kernels tailored to the problem at hand to obtain the best possible performance. However, for many practical applications, good results are obtained selecting the function from a pool of basic licit kernels, such as the linear, the polynomial and the Gaussian radial basis function (Grbf). For an extensive and in depth description of these and other more complex kernels see Cristianini and Shawe-Taylor (2004).

The Grbf kernel is of special interest due to its versatility and previous good performance in geostatistical applications (Wohlberg, Tartakovsky and Guadagnini, 2006 ; Pozdnoukhov and Kanesky , 2006; and, Kanesky et al. , 2001). It has the form:

$$k(\mathbf{u},\mathbf{u}') = \exp\left(-\gamma\left\|\mathbf{u}-\mathbf{u}'\right\|^2\right);\ \gamma > 0 \qquad (12)$$

where **u** and **u'** represent any two different locations and γ is a kernel parameter that must be selected.

Parameter selection. Training the SVC algorithm using the Grbf kernel implies simultaneous selection of the pair of parameters (*P*, γ), so that the boundary classifier can predict unsampled locations with the maximum generalization accuracy. These two parameters are often selected using *k*-fold cross-validation. In *k*-fold cross-validation, the observed data is randomly divided into *k* equal sized subsets. Then, the SVC algorithm is sequentially trained using the *k*-1 subsets and tested in the remaining subset. Training is repeated *k* times and the percentage of data correctly classified for all the *k* subsets that are not included in the training data is recorded as the cross-validation accuracy (Abe, 2005, p. 73). The cross-validation accuracy, as a proxy of the generalization accuracy, is used to select the pair of parameters (*P,* γ). The typical approach calculates the *k*-fold cross-validation accuracy for every pair (*P,* γ) on a predefined grid-search and it chooses the one with the maximum value. To explore a wide range of parameter combinations, the grid is designed as an exponentially growing sequence of *P* and *γ* values (Hsu, Chang and Lin, 2008), for instance: P={2^{-3} , 2^{-2} , …. , 2^{8} , 2^{9}} and γ ={2^{-10} , 2^{-9} , …. , 2^{11} , 2^{12}}. The selected pair of parameters (*P,* γ) is used to train the SVC algorithm with the complete set of observed data. More information about cross validation and for model selection can be found in Anguita, Boni, Ridella, Rivieccio and Sterpi (2005).

Some examples

The following examples were created to illustrate the type of problems that can be solved by SVC. There is no intention of (1) determining the advantages or disadvantages of SVC compared to other techniques, and (2) offering an in depth discussion of the results obtained.GSLIB (Deutsch and Journel, 1998), LIBSVM (Chang, C.-C. and C.-J.Lin, 2001) and MATLAB software were used in the construction of these examples. The Spatial Interpolation Contest (SIC) data set of 1997 was used in this paper. It consists of x, y locations in km and 467 rainfall measurements in 1/10th of mm. The rainfall measurements were transformed to a binary variable of low and high values using an arbitrary threshold of 100/10th of mm (Figure 3). The proportions of high and low values are 74.0% and 26.0%, respectively. However, the spatially biased collection of the data suggests the application of a declustering technique to determine representative statistics. The cell declustered proportions are to 65.0% and 35.0%, respectively. The domain was gridded with a resolution of 5 km x 5 km spacing (3500 nodes in total) that spans an area of 350 km x 250 km.

Classification (Estimation): Assigning a single category s_1 or s_2 to unsampled locations based on observed data is the primary application of SVC. This application has been implemented by many authors with different degrees of success (Wohlberg, Tartakovky and Guadagnini , 2006; Kanevski et al. , 2001). Figure 4 shows the classification results obtained from the indicator kriging (IK) and the SVC algorithm.

Post-processing (Cleaning): Figure 5 shows the result of cleaning a realization generated by sequential indicator simulation (SIS) with MAPS (Deutsch. C.V., 1998) and SVC. Applying MAPS with the default smoothing window results in a 5.8% change of the simulated values. In the SVC cleaning, the number of cells to change can be strictly controlled using the γ versus accuracy plot and γ versus number of support vectors plot (See Figure 6).

Local Refinement: Figure 7 shows that SVC can be used to locally refine areas of interest. The idea is to take advantage of the continuity of the boundary. A grid 5000x5000x1 can be generated in fractions of a second if the boundary had already been determined.

Comparison of spaces of uncertainty: Every set of realizations has its own "cloud" of γ vs accuracy curves and γ vs number of support vectors curves. These clouds can be used to characterize the spaces of uncertainty of different simulation algorithms, for instance: SIS and TGS (Figure 8).

Classification of realizations: This example shows that SVC can be used to classify realizations generated using different algorithms. This may be practically useful to differentiate the modelling methodology used to construct legacy models, should this information be lacking.

For this exercise, a label of interest (e.g. after processing the realization through a transfer function) is assigned to a small subset of realizations from two approaches. This subset is used to train the SVC algorithm. The output hyperplane is used to classify the remaining unlabeled realizations. Figure 9 sketches an example where 60 realizations are taken out from a set of 200 realizations that consist of 100 realizations generated by SIS and 100 realizations TGS. 30 realizations were labeled as coming from SIS and 30 realizations were labeled as coming from TGS. These 60 realizations were used to train the SVC algorithm and the boundary hyperplane output was used to classify the 140 unlabeled realizations. The generalization accuracy for this example was over 90%, that is, over 90% of models were correctly classified based on the results of the modeling approach.

Conclusions

This paper introduces a relatively new algorithm in machine learning. The examples show that support vector machines for the analysis of spatially distributed data is a wide open area of research. Exploring its applicability to old and new geostatistical problems is a worthwhile and promising endeavour. SVM also may open the window to consider the study of other kernel methods for geostatistical applications; these may include Kernel Principal Component Analysis, Least square SVM, and Kernel Ridge Regression.

References

Abe, S. (2005) Support vector machine for patter classification. Springer, USA.

Anguita, D., Boni, A., Ridella, S., Rivieccio, F., and Sterpi, D. (2005) Theoretical and practical model selection methods for support vector classifiers, StudFuzz 177, pp. 159–179. Springer-Verlag.

Boser, B.E., Guyon, I., and Vapnik, V. (1992) A training algorithm for optimal margin classifers. In Proceedings of the Fifth Annual Workshop on Computational Learning Theory, pp. 144-152. ACM Press.

Chang, C.-C. and C.-J.Lin. (2001) LIBSVM: a library for support vector machines. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm. Retrieved 22 December 2008.

Cortes, C. (1995) Prediction of generalization ability in learning machines. PhD Thesis, Department of Computer Science, University of Rochester, Rochester NY 14627.

Cortes, C., and Vapnik, V. (1995) Support vectors networks. Machine Learning, 20, pp. 273-297.

Cristianini, N., and Shawe-Taylor, J. (2004) Kernel methods for pattern analysis. Cambridge University Press.

Deutsch, C.V. (2002) Geostatistical reservoir modeling. Oxford University Press, New York.

Deutsch, C.V., and Journel, A.G. (1998) GSLIB: geostatistical software library and user's guide. Oxford University Press, New York.

Hsu C-W., Chang Ch-Ch., and Lin C-J. (2008) A practical guide to support vector classification. Department of Computer Science. National Taiwan University, Taipei 106, Taiwan. http://www.csie.ntu.edu.tw/~cjlin. Last updated: May 21, 2008.

Kanevski, M., Canu, S., Maignan, M., Wong, P., Pozdnukhov, A., Shibli, S. (2001) Support vector machines for classification and mapping of reservoir data. IDIAP Research Report. IDIAP-RR-01-04.

Kecman, V. (2001) Learning and softcomputing. The MIT Press.

Pozdnoukhov A., Kanevski M. (2006) Monitoring network optimisation for spatial data classification using support vector machines. Int. Journal of Environment and Pollution. Vol. 28. 20 p.

Rosenblatt, F. (1962) Principles of neurodinamics: Perceptron and theory of brain mechanism. Spartan Books, Washington D.C.

Vapnik, V. (1998) Statistical learning theory. Wiley, New York

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

Wohlberg, B., Tartakovsky, D.M., and Guadagnini, A. (2006) Subsurface characterization with support vector machines. IEEE transactions on geoscience and remote sensing, Vol. 44, No. 1, Jan. 2006.

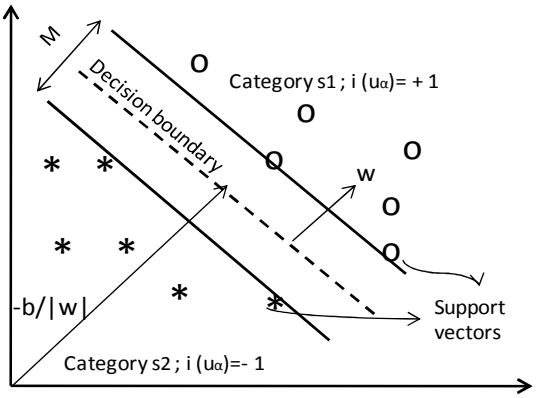


Figure 1: Linear separable case. Basic SVC concepts: codification (±1), margin (*M*), weights (**w**) and support vectors.

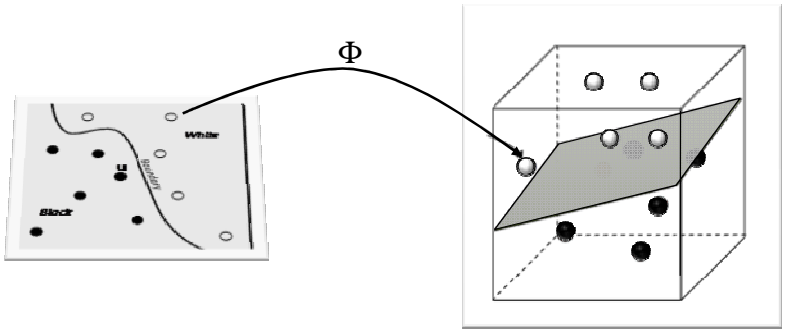


Figure 2: The SVC algorithm applied in a high dimensional space will produce a non-linear classifier in the original input space.

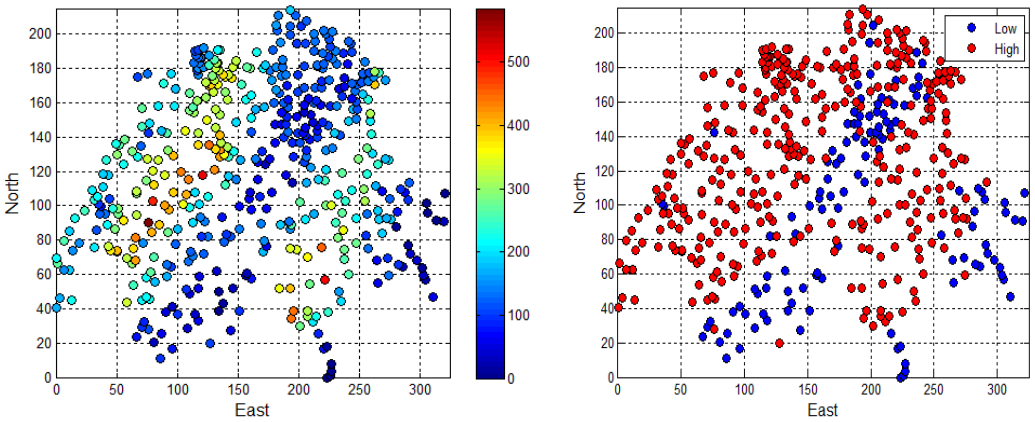


Figure 3: Location map for sic97. (a) Original data and (b) transformed binary data.

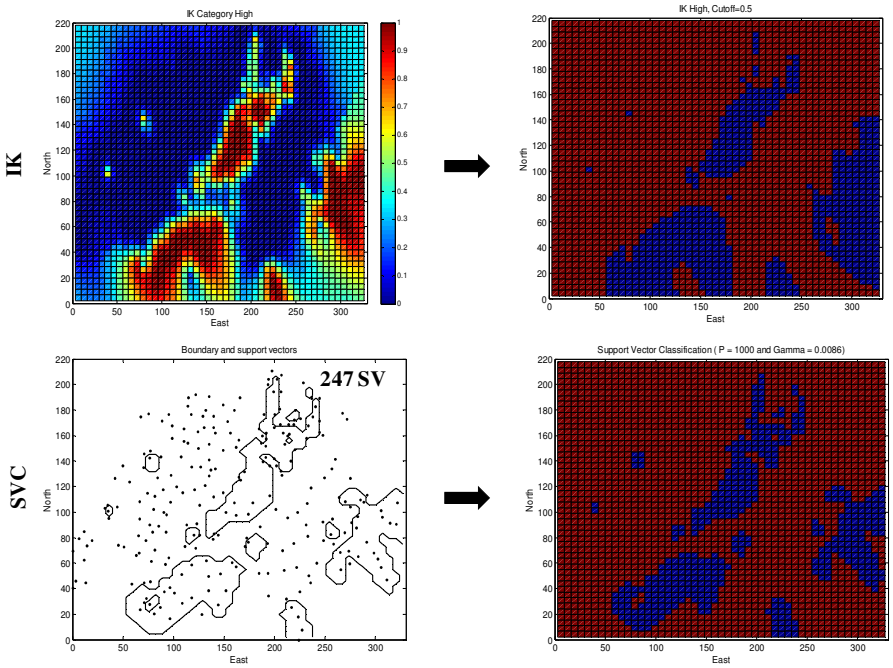


Figure 4: IK map (top left), estimated values with cut-off of 50% probability (top right), classifier boundary and associated support vectors (bottom left), and SVC classified locations (bottom right).

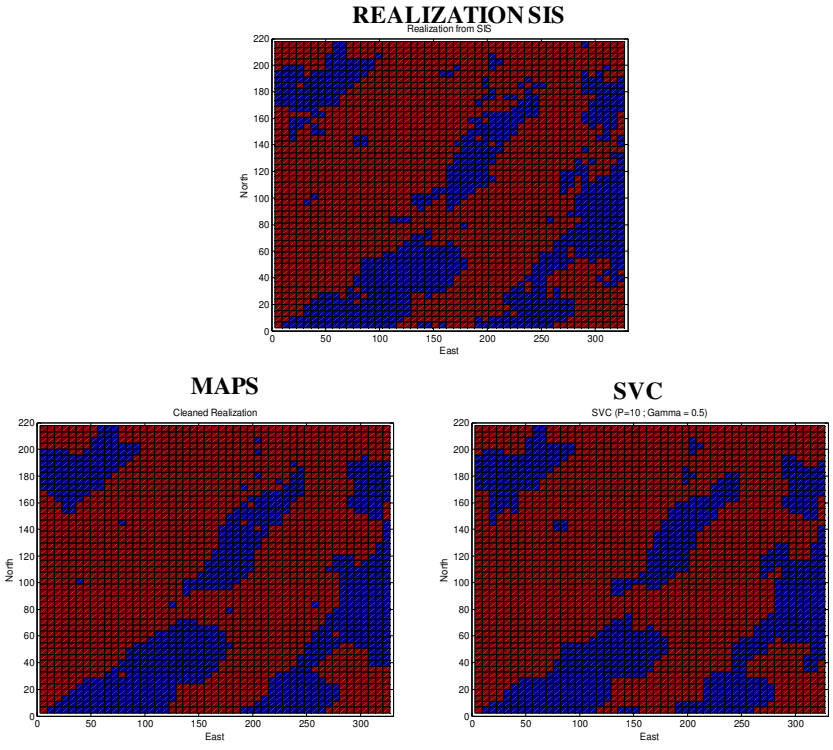


Figure 5: Top, original SIS realization. Bottom left, realization cleaned by MAPS. Bottom right, realization cleaned by SVC.

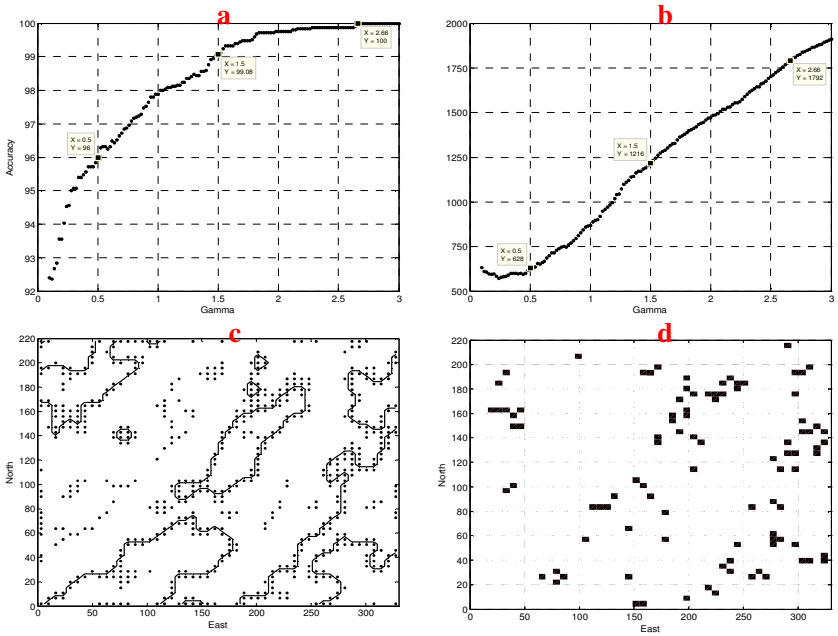


Figure 6: a) γ versus accuracy, b) γ versus number of support vectors, c) Boundary and support vectors for ($\log_2\gamma$,$\log_2 P$) = (0.5 , 10), d) Misclassified locations for ($\log_2\gamma$,$\log_2 P$) = (0.5,10).

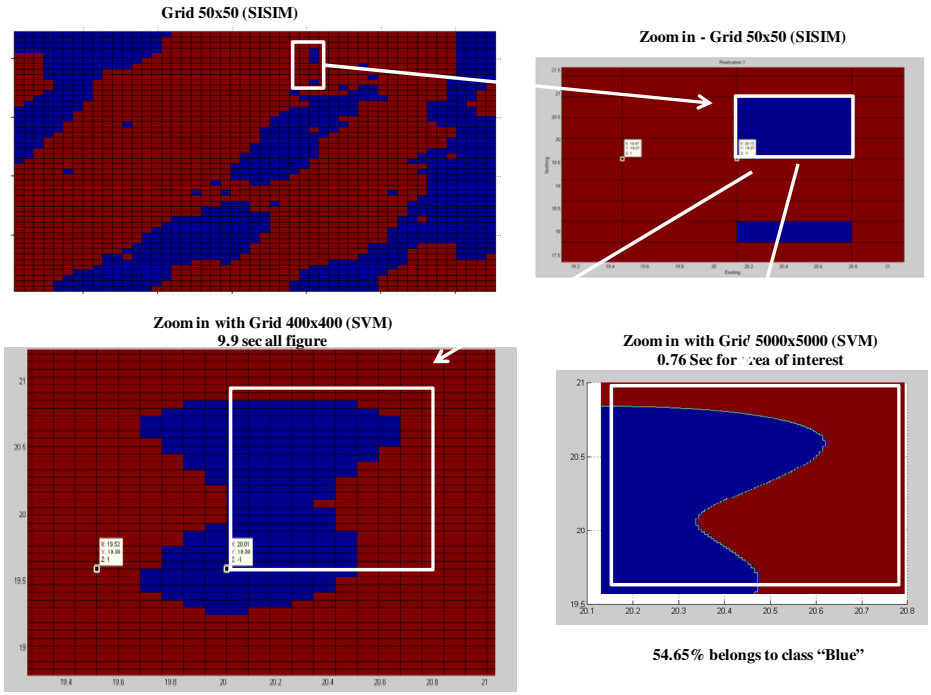
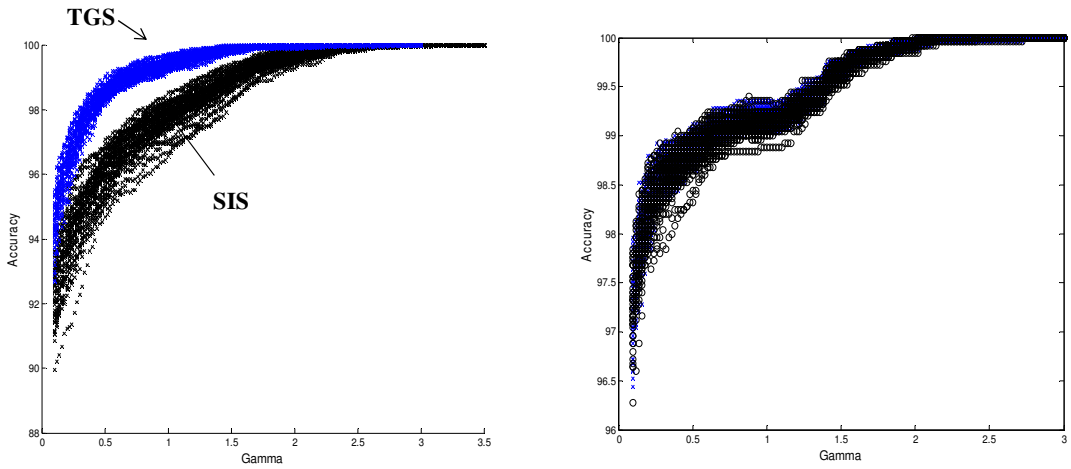


Figure 7: Example of local refinement. A grid 5000x5000x1 can be generated in a very short time.



Different space of uncertainty
Short scale variability

Same space of uncertainty
Without short scale variability

Figure 8: Comparison of spaces of uncertainty SIS vs TGS (With and without cleaning).

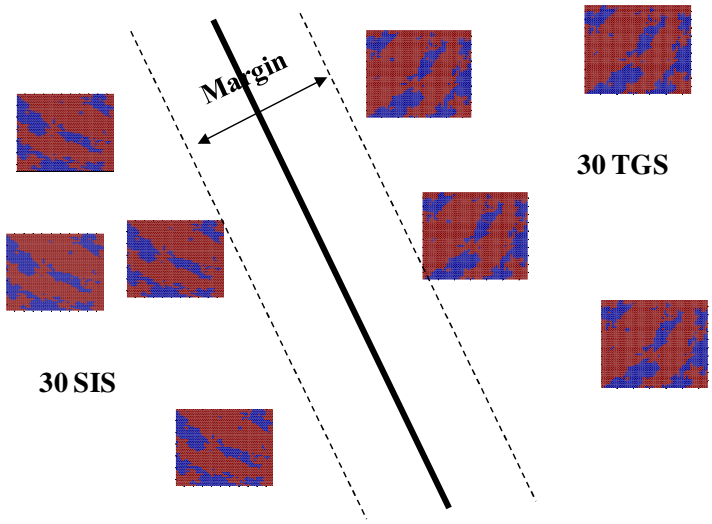


Figure 9: After training a SVC with 60 realizations (30 from SIS and 30 from TGS) the classification rates for 140 unlabeled realizations were: SIS = 64/70 (91.42%) and TGS = 65/70 (92.85%)