Implicit Modelling of Geological Domains by Support Vector Machine: Tuning the Parameters Based on Consistency of the Indicator Variogram

I. Ongarbayev and N. Madani

Nazarbayev University, Kazakhstan

3D modelling of geological domains is an essential part of orebody modelling and resource classification. Conventionally, one applies a wireframing technique, which, in practice, is quite laborious. An automatic approach for building an implicit geological model is the application of the Support Vector Machine (SVM) algorithm. However, in this machine learning problem, the accuracy of models significantly depends on the selected parameters. Here, the authors introduce another criterion for selecting optimum parameters, in which the indicator variogram is considered as an aid. The results can be used as an instruction for implicit geomodelling, based on the SVM algorithm where one is dealing with two categories.

Keywords: SVM, Support Vector Machine, indicator variogram, geological modelling, implicit geomodelling.

INTRODUCTION

The building of three-dimensional (3D) geological models has proven an essential part of every mining project. A 3D geological model represents an interpretation of borehole data. To implement such a model, explicit approaches such as wireframing is a widely used technique. Despite the advantages of wireframing, this process is a complex and challenging task, which requires profound knowledge and skill. In addition, this method is subjective and time-consuming as it is done manually, and depends highly on the interpretation of a geologist or resource modeller. Due to the disadvantages of the above-mentioned wireframing technique, an implicit automatic approach for building such a model - Support Vector Machine (SVM) can be used instead. Recently, implicit geological modelling has become popular due to its straightforwardness, simplicity, and rapidity. The SVM algorithm is one of the optimum machine learning techniques that are usually used for classification (distinguishing between several groups or classes) and regression (obtaining a mathematical model to predict something) problems, especially when dealing with sparse data, which is a typical practice in real life (Smirnoff *et al*., 2008).

The SVM is an algorithm that has found copious applications in various fields for solving classification problems. For instance, it is used for the classification of unseen data, utilised in geotechnical engineering (Goh and Goh, 2007), in medicine (Veropoulos *et al*., 1999), in bioinformatics (Yang, 2004), and many other practical problems.

The SVM algorithm has already been applied to geological modelling (Smirnoff *et al.*, 2008). Nevertheless, in machine learning problems, where the corresponding variables are not regionalised, the accuracy of models significantly depends on the parameters, which should be proposed to the SVM algorithm. For instance, in the radial basis function— a popular kernel function used in SVM— one may decide about the underlying model based on opting two parameters: Cost function and gamma. However, in implicit modelling of geological domains, this practice is suboptimal, and may produce unrealistic geological maps due to ignoring the influence of spatial continuity of geological domains. In order to circumvent this problem, the authors introduced another criterion for the selection of optimum parameters, which considers the indicator variogram as the basis for tuning these parameters.

In this study, after briefly discussing the SVM algorithm and indicator variogram, the authors demonstrate through a synthetic case study how this spatial characteristic of resulting models can be used as an aid for better identifying the SVM parameters, which produce maps that comply with the original data and geological interpretations. This has been validated through several different cases of geological complexities. This study's results can be used as the instruction for implicit geological modelling based on SVM, and the selection of optimum parameters for the algorithm.

METHODOLOGY

Machine Learning Basics

Machine learning (ML) is programming computers so that the computer program accesses the data and learns from it. The ML algorithms are often categorised as supervised and unsupervised. The main difference is that a supervised algorithm works based on the labelled dataset, while an unsupervised algorithm works based on the unlabelled dataset. In this work, we used the SVM algorithm, which belongs to the supervised ML family. It is a widely used technique for classification and regression analyses. The classification is a function that separates the data into categories, i.e., it identifies the class of an unknown observation based on a training dataset. Regression is a function that predicts some variable based on the training dataset (e.g., the simplest regression model: $y = 2x$, predicts the variable ‘y’ based on variable ‘x’). In this respect, one has training data, which serves as a dataset to train the model. Apart from training data, one has testing data, a subset of the dataset, which tests the trained model. More details of the basics of ML can be found in various sources (Müller and Guido, 2017).

The SVM Algorithm

In this section, all the concepts of SVM algorithms, which are applicable in this study, are covered. This starts with a linear classification as it is the most straightforward problem, where one can ideally separate data. Nonetheless, this example is an excellent introduction of all relevant concepts and ideas underlying the SVM algorithm. Then, the Soft Margin Classifier (SVC – support vector classifier) and Non-linear Classifier are introduced as two enhanced SVM techniques.

Maximal Margin Classifier

To understand how the Maximal Margin Classifier works, it is necessary to understand the concept of a hyperplane. From a geometry perspective, a hyperplane is a subspace of one dimension less than its ambient space. To make it straightforward, if we consider a two-dimensional space, a hyperplane for such a case will be a line. Figure 1B presents different options of separating hyperplanes.

Let us consider a hyperplane in a practical example. There are some observation points required to be separated by using a hyperplane (Figure 1A). The objective is to find a hyperplane that ideally separates the data into two categories. In such a case, an infinite number of hyperplanes can be identified (Figure 1B). However, the question is: “which of these hyperplanes is the optimum one?” For this, the rational choice for an optimum hyperplane is selecting the one with hyperplane's biggest distance to the observation points (Figure 1C). The smallest perpendicular distance from hyperplane to observation is called the margin. Therefore, the optimum one is the hyperplane with the maximum value of margin (Figure 1D). This is known as the Maximal Margin Classifier (Vapnik, 2000). The observations which lie on margins are called support vectors. Support vectors play a crucial role in defining the classifier, and any movement of support vectors can dramatically change the classifier.

A B

C D

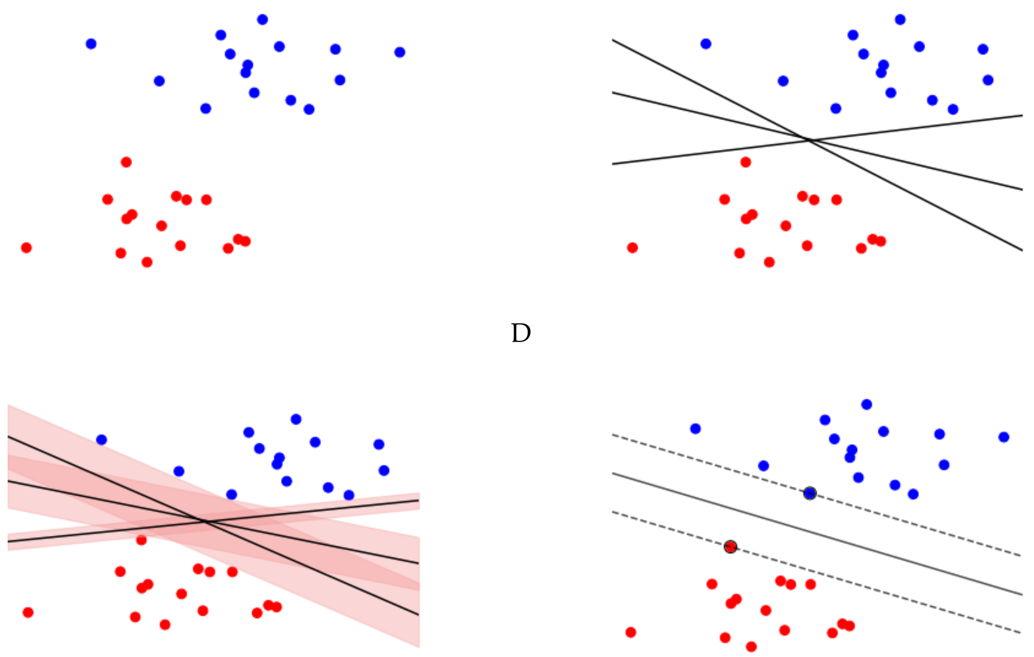


Figure 1. Maximal Margin Classifier: (A) random dataset, (B) different hyperplanes that separate the data, (C) hyperplanes with visualised margins, (D) Maximal Margin Classifier.

The Maximal Margin Classifier has two main limitations. Firstly, it is not always possible to separate observations perfectly along a straight line. Secondly, it is highly susceptible to support vectors. The first limitation as presented in Figure 2A is that we cannot perfectly separate observations by only using a flat hyperplane, which is not often a case in real-world problems. Hence, in such a situation, we cannot use a maximal margin classifier. The second limitation is the sensitivity of the classifier. If we supplement our dataset with only one additional observation, it can dramatically influence the hyperplane, completely changing the classifier. The new classifier (Figure 2B) is different from the previous one (Figure 1D). Hence, due to the Maximal Margin Classifier's disadvantages, one may use the Soft Margin Classifier.

A B

Figure 2. Limitations of Maximal Margin Classifier: (A) dataset, which we cannot separate by Maximal Margin Classifier, (B) sensitivity of Maximal Margin Classifier.

Soft Margin Classifier

In practice, the data often encompasses noises. Cortes and Vapnik (1995) introduced a modified version of the original SVM to solve these problems. In a new version of the SVM algorithm, it is allowed that some observations lie within the margin area (Figure 3), and some may even go on the wrong side of the hyperplane. This makes the Soft Margin Classifier less sensitive to individual observations.

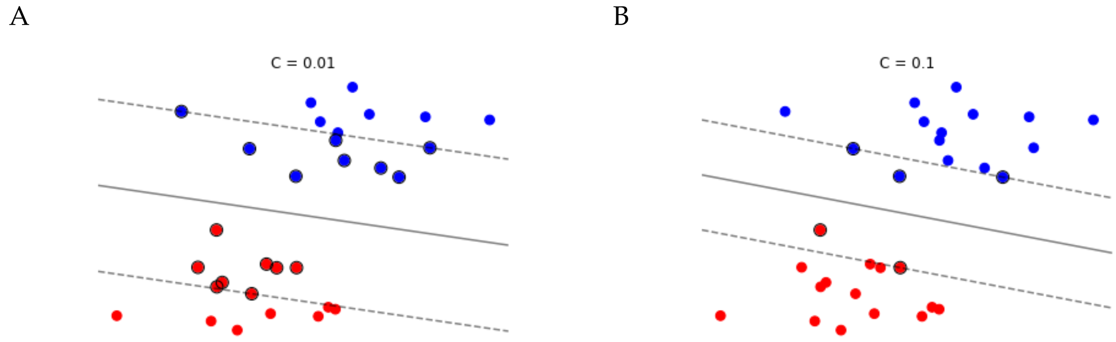


Figure 3. Soft Margin Classifier: (A) Soft Margin Classifier with Cost Parameter = 0.01, (B) Soft Margin Classifier with Cost Parameter = 0.1.

The Soft Margin Classifier concept is the same as the Maximal Margin Classifier, yet, trying to receive a hyperplane with a maximum margin that creates a misclassification budget ('B'). Hence, the algorithm will find a hyperplane with a maximum margin within the allowed budget of the authorised misclassification. All these misclassification errors are added, given that they are less than **B**. Thereby, we allow some misclassification to happen while trying to maximise the margins. This concept is represented by the 'C' parameter (the cost parameter). The definition of cost parameter is presented in formulas [1] and [2].

$$d_1 + d_2 + \ldots + d_n < B \qquad [1]$$

where d(n) is distance from point (n) to the margin.

$$B = 1/C \qquad [2]$$

The impact of the 'C' parameter is presented in Figure 3; if we decrease 'C', it means that there is a lesser cost of misclassification. In this case, more observations are allowed to lie within the margins, leading to misclassification of some more observations. On the contrary, if one increases the value of 'C', the cost of making mistakes in classification is high; so, there will be fewer observations within the margin and fewer misclassified observations, but this will increase the sensitivity of the model to individual observations, which may result in overfitting. Therefore, one should deal with cost parameters carefully. The high value of 'C' may lead to quite narrow margins, and there will be fewer support vectors, which may cause overfitting. In contrast, the low value of 'C' may widen the margins, and there will be many support vectors and many misclassified observations. Therefore, looking for an optimum value of 'C' to obtain the best SVM results is a requirement.

A significant limitation of the Soft Margin Classifier is that it works only with linearly separable data. At this moment, we covered only linear hyperplanes, but in practice, there may be scenarios where the classes are not linearly separable (Figure 4A). From this figure, we see that observations can be classified by drawing a circle. So, to handle this limitation of linearity, one may use a kernel method (SVM). Using these functions, the data can be separated as illustrated in Figure 4B by using nonlinear boundaries.

Non-linear Classifier
The SVM in the sense of non-linear classifier is an extension of the support vector classifier (soft margin classifier), which uses kernels to create non-linear boundaries. The kernel is functional relationship between two observations, where changing it leads to generating different shapes in boundaries. For instance, in Figure 4B, the implementation of the Radial Basis Function (RBF) kernel can be observed.

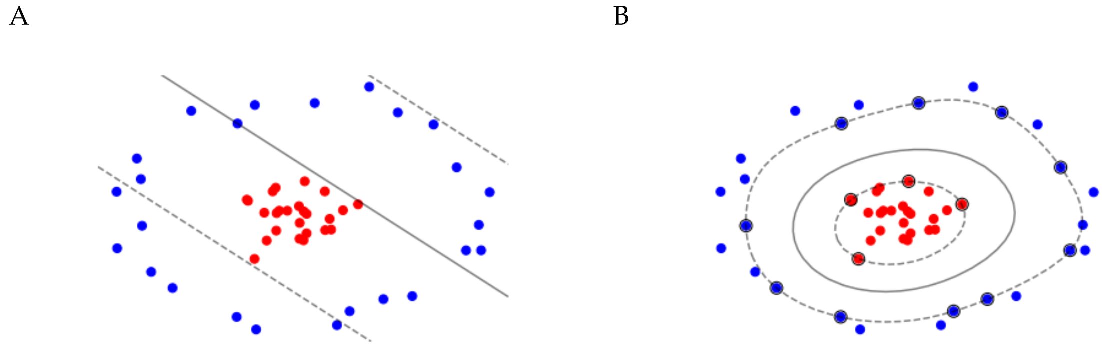


Figure 4. SVM kernels: (A) linear kernel, (B) RBF kernel.

Linear Kernel (Figure 4A) is represented by the following equation:

$$K(x_i, x_j) = x_i^T x_j \qquad [3]$$

where, x_i and x_j are feature vectors, K – kernel function.

The RBF kernel is the most popular algorithm among practitioners and is expressed as follows:

$$K(x_i, x_j) = exp\left(-\gamma\|x_i - x_j\|^2\right), \gamma > 0 \qquad [4]$$

Where 'γ' is an adjustable parameter, 'γ' defines how much influence a single training example has on any particular point. Therefore, a large gamma parameter value would mean that only a closer point impacts that particular point. Ultimately, using the RBF kernel of SVM necessitates tuning two hyperparameters, 'C' and 'γ'.

SVM Application to Geological Modelling
"*Support Vector Machine algorithm is a boundary classification method where the boundary (decision function) is built on the initial training set*" (Smirnoff *et al*. 2008); thus, the SVM algorithm does not take into account the spatial variability.

Spatial variability is when the measurement of a specific property differs over the distance; it is an intrinsic characteristic of geological data. There are a variety of options to measure spatial variability, but in this study, we use variogram analysis. By utilising the variogram analysis we supplement the SVM algorithm, therefore it can produce realistic geological maps which consider the spatial continuity in the region.

Machine Learning Library for Modelling
In this work, we used one of the most popular libraries for ML, namely, Scikit-learn, based on LIBSVM and LIBLINEAR, developed at the National Taiwan University (Fan *et al*., 2008; Chang and Lin, 2011).

Indicator Variogram
The solid variogram model is an essential part of 3D modelling as it represents spatial correlation. (Maleki *et al*. 2017) presented the usefulness of indicator variograms in modelling. An indicator variogram, in fact, can be calculated by traditional variogram formulae where the values at the head

and tails of the selected pairs in each lag are indicators. In general, a variogram is a function that shows the measure of dissimilarity over the specified distances (Goovaerts 1997).

$$\gamma(h) = \frac{1}{2N(h)}\sum_{\alpha=1}^{N(h)}(z(u_\alpha) - z(u_\alpha + h))^2 \quad [5]$$

where γ(h) is a measure of dissimilarity over distance. This function represents a spatial variance between two data points separated by the lag distance, h. N(h) is a number of all pairs separated by the distance, h; u_α is a data point at the location, α; u_α+h is a data point separated from u_α by the distance, h; $z(u_\alpha)$ is the indicator value of data point, u_α; $z(u_\alpha+h)$ is the indicator value of data point, u_α+h. Figure 5 shows a typical experimental variogram.

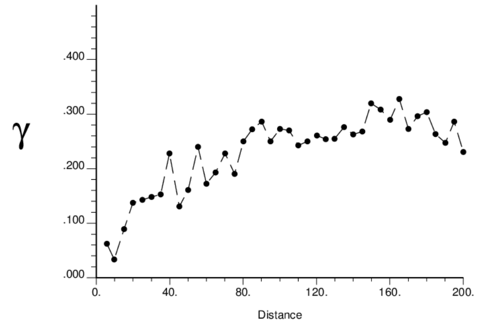


Figure 5. Experimental variogram.

There are several sources for a more detailed explanation of variogram interpretation and modelling (Gringarten and Deutsch, 2001).

Experimental Work

The objective of experimental work was to discover the effect of the indicator variogram on tuning the hyperparameters ('C' and 'γ'). For this, we used the following workflow.

Synthetic dataset used for modelling

For this study, we used an indicator synthetic dataset presented in Figure 6A The synthetic dataset is one realisation obtained by plurigaussian simulation (Armstrong *et al.*, 2011). The indicators are simulated in a 300 by 300 grid points.

General Approach

- A reference map is simulated by plurigaussian simulation in a grid of 300 by 300 showing two categories.
- 100 sample points are selected randomly from this reference map. This is denoted as the *original dataset.*
- Variogram of these 100 sample points is calculated. This is denoted as the *original variogram*.
- Split the original dataset into a training and a test subset. To do so, 80 and 20 data were selected for training and test dataset, respectively.
- Scaling the training dataset to z-score transformation as recommended by (Chih-Jen Lin Chih-Wei Hsu, 2008).
- Using Scikit-learn Library, build a prediction model based on the training dataset. A Grid Search technique was used to find possible hyperparameters of C and γ. In addition, an additional parameter to check the influence of hyperparameters was added.
- Based on the trained model, all 90,000 locations of the reference map are predicted for different hyperparameters.
- Calculation of indicator variograms for each map and compare with the original variogram. For this, mean relative errors (MRE) are calculated.
- Selection of the optimum hyperparameters C and γ that show the least MRE.

Validation

We compared the predicted classes with original classes to check how many points were correctly classified, thus evaluating the model's performance. For this purpose, we used the accuracy score technique.

RESULTS AND DISCUSSION

To build an SVM model, the parameters, which were described previously, are required. Conventionally, one uses the Grid Search technique to find possible parameters for the model. Table 1 (column 2) is a recommendation of parameters of the Grid Search technique for the SVM model. However, this technique does not consider spatial continuity of geological domains, and such a procedure may lead to the production of unrealistic geological maps. To show how significantly the algorithm depends on hyperparameters and how it can produce unrealistic maps, more parameters were added into the experiments. All parameters used are presented in Table 1.

Table 1. Parameters used for SVM algorithm and their accuracy scores

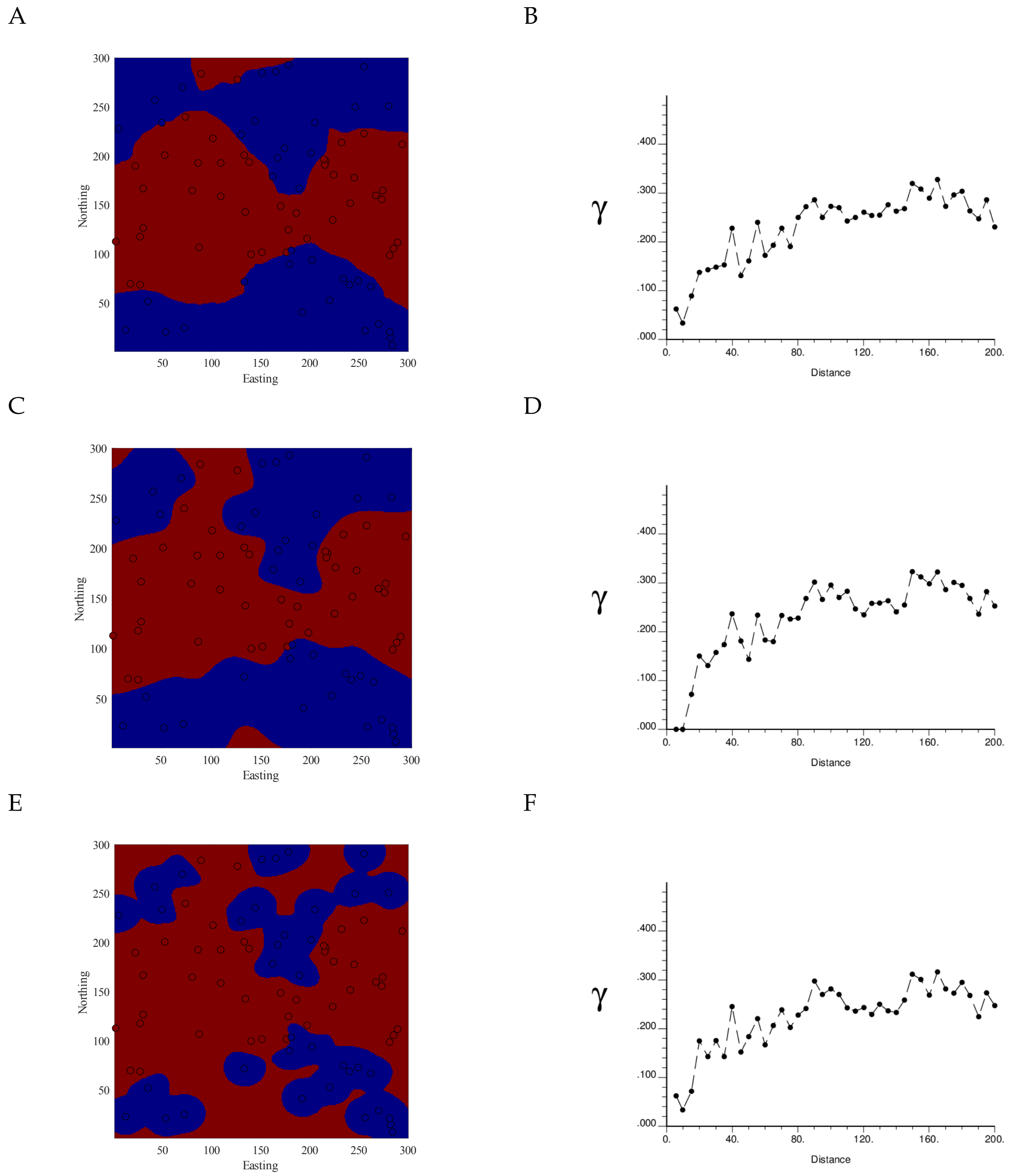
Parameter	Value			
C	10	10	500	500
Gamma	10	50	10	50
Accuracy Score	0.85	0.8	0.85	0.8

The last row in Table 1 describes the validation results. As shown in Table 1, the accuracy score of prediction is acceptable and shows no less than 80% accuracy. However, relying only on the accuracy score is tricky, as demonstrated further on.

The reference map and original indicator variogram are presented in Figures 6A and 6B. The reference map (Figure 6A) represents the reference dataset, which is a plurigaussian simulation on a grid of 300 by 300. The original indicator variogram (Figure 6B) represents the original dataset, which is a randomly selected 100 points. These 100 points are shown in Figure 6A (more details in section 2.5.1). The reference map (Figure 6A) and original indicator variogram (Figure 6B) are used for visual comparison with maps and indicator variograms produced by the SVM algorithm results.

Figure 6 (C, E, G, I) represent the produced maps of the SVM algorithm, and Figure 6 (D, F, H, G) represents the indicator variograms of the SVM model, corresponding to the maps. From the maps presented in Figure 6 (C, E, G, I), one can visually assess the hyperparameters' impact on the SVM model. Figure 6 (C, G) shows promising results, and similar to the reference map (Figure 6A), while maps from Figure 6 (E, I) show the signs of overfitting, despite the good results of accuracy score.

In real-world problems, we do not have a reference map as presented in Figure 6 (A); thus, we cannot fulfill visual comparison of produced maps with an original one. To manage this problem, we rely on the indicator variogram. A measure that considers the spatial continuity of geological domains, thereby using the indicator variograms, we can identify the model with optimum parameters, which produce the best results.


A
B
C
D
E
F
Northing
Easting
300
250
200
150
100
50
γ
.400
.300
.200
.100
.000
0.
40.
80.
120.
160.
200.
Distance

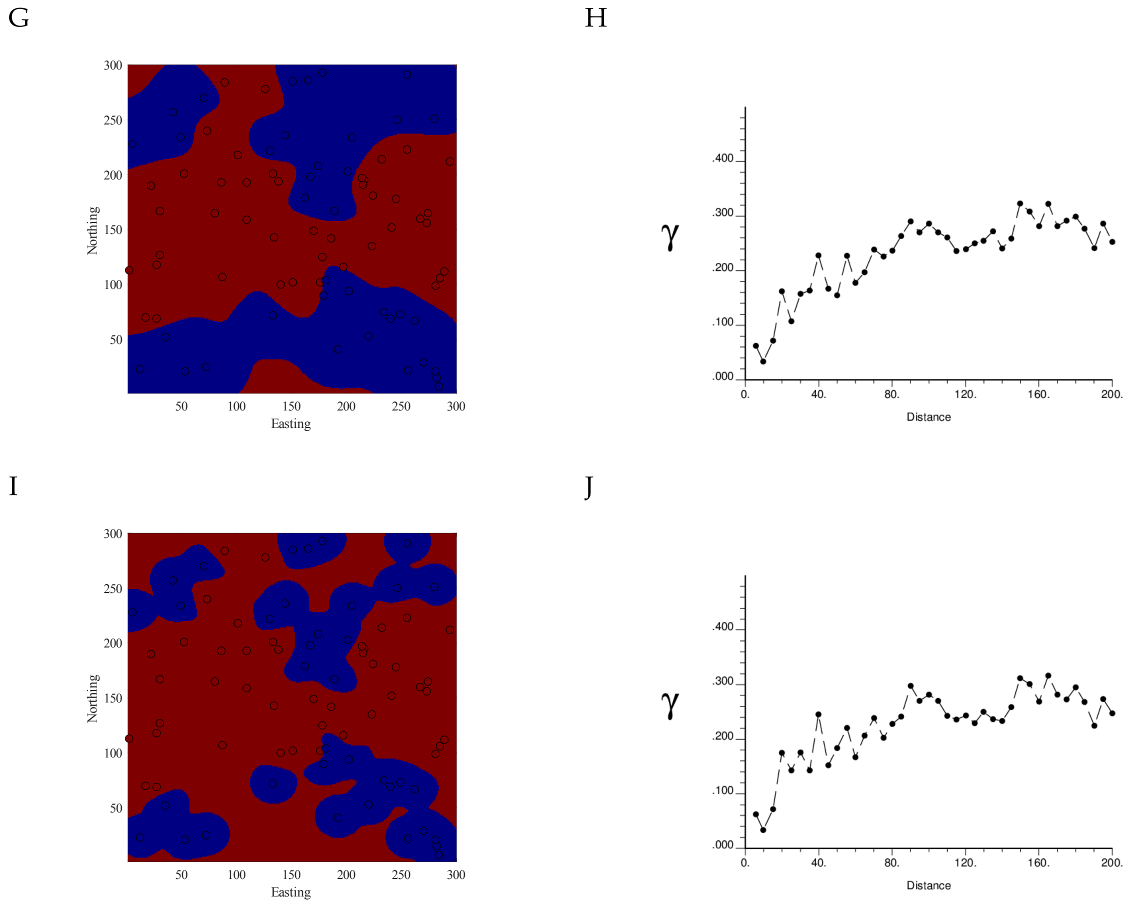


Figure 6. Categorical maps and indicator variograms: (A) Reference map, (B) original indicator variogram. (C, D) Categorical map and indicator variogram, where 'C' 10 and 'γ' 10, (E, F) Categorical map and indicator variogram, where 'C' 10 and 'γ' 50, (G, H) Categorical map and indicator variogram, where 'C' 500 and 'γ' 10, (I, J) Categorical map and indicator variogram, where 'C' 500 and 'γ' 50.

From Figure 6 (B, D, F, H, G), one intuitively notices the difference between the original variogram and variograms built from the SVM algorithm over the target locations. However, it is challenging to make any statements based on the visual interpretation of variograms. Therefore, the MRE for each variogram was calculated. The MRE is an average of relative error and represents how significant the error of predicted value is in comparison with the actual value. Results are presented in Table 2. These results show how different each variogram is in comparison with the original variogram. From Table 2, one can observe that the best results show the model with 'C' = 500, 'γ' = 10, its MRE is 14.6%. While the models with 'γ' = 50 show worse results because of the overfitting effect.

Table 2. Mean Relative Error of Variograms

Parameters		MRE
C	Gamma	
10	10	0.153102883
	50	0.186340367
500	10	0.14629639
	50	0.186340367

In addition to MRE, a sensitivity analysis over the models which took part in experiments was carried out. The objective was to check how significantly hyperparameters impact the SVM model's reconstruction of the reference dataset. Results are presented in Table 3; the best results highlight the model with parameters 'C' = 500 and 'γ' = 10, and the worst results show the models with a high 'γ' parameter. From Table 3, it is clear that the models with a high 'γ' parameter have a skewed proportion to the predominant class because when SVM deals with binary classification, and it cannot classify a point, it tends to attribute it to the prevailing class.

Table 3. Proportion of the class points

C Gamma	Training dataset	10 10	10 50	500 10	500 50
Class 0 Class 1	44 56	46568 43432	29870 60130	44321 45679	29870 60130
Proportion of class 0 (%) Proportion of class 1 (%)	0.44 0.56	0.517 0.483	0.331889 0.668111	0.492456 0.507544	0.331889 0.668111
Absolute error (0) Absolute error (1)	- -	0.077 0.077	0.108111 0.108111	0.052456 0.052456	0.108111 0.108111

CONCLUSION

Our experiments clearly showed successful application of indicator variograms for tuning the parameters for the SVM model. It was demonstrated that modelling of realistic geological maps by SVM algorithm depends highly on the parameters provided for the model. The parameters, conventionally determined by the suboptimal Grid Search technique, which does not consider the spatial continuity, but plays a crucial role in modelling of geological domains. Results of the experiments show that application of indicator variograms helps to circumvent this gap.

In addition, the proposed combination of an SVM algorithm, and indicator variograms can use only a small number of observations to build a credible classification model; thus, it is a promising method when dealing with a sparse dataset. The experiments intentionally used a scarce dataset, with only spatial coordinates as input data for the model, to demonstrate that the method is capable of producing good results in such conditions.

The study's objective was to introduce an indicator variogram as a new criterion for determining the optimum parameters for the SVM algorithm. The proposed method (SVM + indicator variogram) considers the spatial continuity of geological domains and is capable of building realistic geological maps. However, the proposed method is restricted to the isotropic geological domain. In the case of having anisotropy in the region, the current method is not applicable, and one needs to use other ML techniques.

ACKNOWLEDGEMENTS

We acknowledge Nazarbayev University's funding this work via Faculty Development Competitive Research Grants for 2021–2023, Grant No. 021220FD4951.

REFERENCES

Armstrong, M. *et al.* (2011) *Plurigaussian Simulations in Geosciences*, *Springer-Verlag Berlin Heidelberg*. Springer Berlin Heidelberg. doi: 10.1007/978-3-642-19607-2.

Chang, C. C. and Lin, C. J. (2011) 'LIBSVM: A Library for support vector machines', *ACM Transactions on Intelligent Systems and Technology*, 2(3), pp. 1–39. doi: 10.1145/1961189.1961199.

Chih-Jen Lin Chih-Wei Hsu, C.-C. C. (2008) 'A Practical Guide to Support Vector Classification', *BJU international*, 101(1), pp. 1396–1400.

Cortes, C. and Vapnik, V. (1995) 'Support-vector networks', *Machine Learning*, 20(3), pp. 273–297. doi: 10.1007/BF00994018.

Fan, R. *et al.* (2008) : 'A Library for Large Linear Classi cation', *Corpus*, 9, pp. 1871–1874.

Goh, A. T. C. and Goh, S. H. (2007) 'Support vector machines: Their use in geotechnical engineering as illustrated using seismic liquefaction data', *Computers and Geotechnics*, 34(5), pp. 410–421. doi: 10.1016/j.compgeo.2007.06.001.

Gringarten, E. and Deutsch, C. V (2001) 'Variogram interpretation and modeling', *Mathematical Geology*, 33(4), pp. 507–534.

Maleki, M., Emery, X. and Mery, N. (2017) 'Indicator variograms as an aid for geological interpretation and modeling of ore deposits', *Minerals*, 7(12). doi: 10.3390/min7120241.

Müller, A. C. and Guido, S. (2017) *Introduction to Machine Learning with Python*.

Smirnoff, A., Boisvert, E. and Paradis, S. J. (2008) 'Support vector machine for 3D modelling from sparse geological information of various origins', *Computers and Geosciences*, 34(2), pp. 127–143. doi: 10.1016/j.cageo.2006.12.008.

Vapnik, V. N. (2000) *The Nature of Statistical Learning Theory*, *Springer-Verlag New York*. Springer New York. doi: 10.1007/978-1-4757-3264-1.

Veropoulos, K., Cristianini, N. and Campbell, C. (1999) 'The Application of Support Vector Machines to Medical Decision Support : A Case Study', *Advanced Course in Artificial Intelligence*, (February), pp. 1–6.

Yang, Z. R. (2004) 'Biological applications of support vector machines.', *Briefings in bioinformatics*, 5(4), pp. 328–338. doi: 10.1093/bib/5.4.328.

Ilyas Ongarbayev

MSc Student in Mining Engineering
Nazarbayev University

Ilyas Ongarbayev is a Master of Sciences student at Nazarbayev University (Republic of Kazakhstan), specializing in the field of Geostatistics. Currently, he is working on the application of Machine Learning (ML) algorithms in the mining industry, particularly resource estimation. His Bachelor's degree is in Geology Engineering from Karaganda Technical University (Republic of Kazakhstan) with an interest to exploration of ore deposits