

Sérgio Aurélio Ferreira Soares

**SPATIAL INTERPOLATION AND GEOSTATISTIC SIMULATION
WITH THE INCREMENTAL GAUSSIAN MIXTURE NETWORK**

Dissertação submetida ao Programa de Pós-
Graduação em Ciência da Computação para
a obtenção do Grau de Mestre em Ciência
da Computação.

Orientador: Prof. Dr. Mauro Roisenberg

Florianópolis

2016

Ficha de identificação da obra elaborada pelo autor
através do Programa de Geração Automática da Biblioteca
Universitária da UFSC.

Soares, Sérgio Aurélio Ferreira
Spatial Interpolation and Geostatistic
Simulation with the Incremental Gaussian
Mixture Network / Sérgio Aurélio Ferreira
Soares ; orientador, Mauro Roisenberg -
Florianópolis, SC, 2016.
77 p.

Dissertação (mestrado) - Universidade
Federal de Santa Catarina, Centro
Tecnológico. Programa de Pós-Graduação em
Ciência da Computação.
Inclui referências

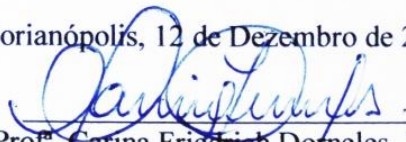
1. Ciência da Computação. 2.
Geostatística. 3. Modelo de Mistura de
Gaussianas. 4. Rede Neural Artificial. I.
Roisenberg, Mauro. II. Universidade Federal
de Santa Catarina. Programa de Pós-Graduação
em Ciência da Computação. III. Título.

Sérgio Aurélio Ferreira Soares

**SPATIAL INTERPOLATION AND GEOSTATISTIC
SIMULATION WITH THE INCREMENTAL GAUSSIAN
MIXTURE NETWORK**

Esta dissertação foi julgada adequada para obtenção do título de mestre e aprovada em sua forma final pelo Programa de Pós-Graduação em Ciência da Computação.

Florianópolis, 12 de Dezembro de 2016.




Prof.^a Carina Friedrich Dorneles, Dr.^a
Coordenadora do Programa

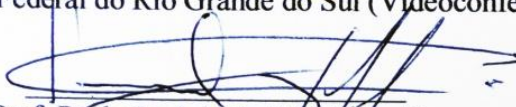
Banca Examinadora:



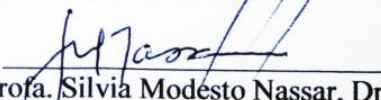
Prof. Mauro Roisenberg, Dr.
Universidade Federal de Santa Catarina
Orientador



Prof. Paulo Martins Engel, Dr.
Universidade Federal do Rio Grande do Sul (Videoconferência)



Prof. Paulo José de Freitas Filho, Dr.
Universidade Federal de Santa Catarina



Profa. Silvia Modesto Nassar, Dra.
Universidade Federal de Santa Catarina

I dedicate this work to my wife, Cecília.

ACKNOWLEDGEMENTS

First, I would like to thank God for guiding me and blessing me every day in this journey. My wife, Cecília, whose dedication, support and encouragement helped me to stay strong and to overcome the hard times. My mother, Lucilene, who never forget to include me in her prayers. My father, José Sérgio, whose curiosity and ability with electronics inspired my career choice. My advisor, Mauro Roisenberg, who was always generous and positive, besides giving great contributions to this work. My friends Alexandre, Clarissa, Gui and Lu for your friendship and the fun moments on weekends, which were essential to recharge my energy. My colleagues from L3C: Isaac, Tcharlies, and Felipe, for the companionship and contributions. Last, I would like to thank CNPq for my scholarship and Petrobras for the financial support of the projects I took part.

First, there is a mountain, then there is no mountain, then there is.

(Buddhist saying, via Donovan, 1967)

ABSTRACT

Geostatistics aggregates a set of tools designed to deal with spatially correlated data. Two significant problems that Geostatistics tackles are the spatial interpolation and geostatistical simulation. Kriging and Sequential Gaussian Simulation (SGS) are two examples of traditional geostatistical tools used for these kinds of problems. These methods perform well when the provided Variogram is well modeled. The problem is that modeling the Variogram requires expert knowledge and a certain familiarity with the dataset. This complexity might make Geostatistics tools the last choice of a non-expert. On the other hand, an important feature present in neural networks is their ability to learn from data, even when the user does not have much information about the particular dataset. However, traditional models, such as Multilayer Perceptron (MLP), do not perform well in spatial interpolation problems due to their difficulty in accurately modeling the spatial correlation between samples. With this motivation in mind, we adapted the Incremental Gaussian Mixture Network (IGMN) model for spatial interpolation and geostatistical simulation applications. The three most important contributions of this work are: 1. An improvement in the IGMN estimation process for spatial interpolation problems with sparse datasets; 2. An algorithm to perform Sequential Gaussian Simulation using IGMN instead of Kriging; 3. An algorithm that mixes the Direct Sampling (DS) method and IGMN for cluster-based Multiple Point Simulation (MPS) with training images. Results show that our approach outperforms MLP and the original IGMN in spatial interpolation problems, especially in anisotropic and sparse datasets (in terms of RMSE and CC). Also, our algorithm for sequential simulation using IGMN instead of Kriging can generate equally probable realizations of the defined simulation grid for unconditioned simulations. Finally, our algorithm that mixes the DS method and IGMN can produce better quality simulations and runs much faster than the original DS. To the best of our knowledge, this is the first time a Neural Network model is specialized for spatial interpolation applications and can perform a geostatistical simulation.

Keywords: Geostatistics. Gaussian Mixture Models. Artificial Neural Networks.

RESUMO

A Geoestatística agrega um conjunto de ferramentas especializadas em dados espacialmente correlacionados. Dois problemas importantes na Geoestatística são a interpolação espacial e a simulação. A Krigagem e a Simulação Sequencial Gaussiana (SGS) são dois exemplos de ferramentas geoestatísticas utilizadas para esses tipos de problemas, respectivamente. A Krigagem e a SGS possuem bom desempenho quando o Variograma fornecido pelo usuário representa bem as correlações espaciais. O problema é que a modelagem do Variograma requer um conhecimento especializado e certa familiaridade com o conjunto de dados em estudo. Essa complexidade pode tornar difícil a popularização dessas técnicas entre não-especialistas. Por outro lado, uma característica importante presente em Redes Neurais Artificiais é a capacidade de aprender a partir dos dados, mesmo quando o usuário não possui familiaridade com os dados. No entanto, os modelos tradicionais, como o *Multilayer Perceptron* (MLP), têm dificuldade em identificar a correlação espacial entre amostras e não apresentam um bom desempenho em problemas de interpolação espacial. Com essa motivação, nós adaptamos e aplicamos a *Incremental Gaussian Mixture Network* (IGMN) em problemas de interpolação espacial e simulação geoestatística. As três principais contribuições deste trabalho são: 1. Melhoria no processo de estimação da IGMN para problemas de interpolação espacial; 2. Um algoritmo para realizar simulação sequencial gaussiana utilizando a IGMN como interpolador; 3. Um algoritmo que mistura o método *Direct Sampling* (DS) e a IGMN para realizar simulação multiponto (MPS) a partir de imagens de treinamento. Os resultados mostram que a nossa abordagem é mais precisa que o MLP e a IGMN original em problemas de interpolação espacial, especialmente em conjuntos de dados esparsos e com anisotropia (em termos de RMSE e CC). Nosso algoritmo de simulação sequencial que utiliza a IGMN como interpolador é capaz de gerar simulações não condicionadas que respeitam características do conjunto original de dados. Finalmente, nosso algoritmo de simulação multiponto, que mistura o método DS e a IGMN, é capaz de realizar simulações condicionadas e produz realizações com qualidade superior num tempo de execução inferior ao do DS. Até onde sabemos, esta a primeira vez que um modelo de rede neural é especializado para aplicações de interpolação espacial e é capaz de realizar simulação geoestatística.

Palavras-chave: Geoestatística. Modelo de Mistura de Gaussianas. Rede Neural Artificial.

LIST OF FIGURES

Figure 1	a) Estimation of the unknown permeability z_0 based on a set of n known values; b) Estimation of the unknown z_0 given 7 known values. Numbers in parenthesis are weights assigned based on inverse distance. Adapted from (ZHANG, 2011).	24
Figure 2	Geostatistical estimation workflow using Kriging. Adapted from (ZHANG, 2011).	25
Figure 3	Example of a variogram plot. Source: (PNNL, 2016).	26
Figure 4	Plots of Gaussian, Spherical and Exponential theoretical variogram models. Adapted from: (BOHLING, 2005).	28
Figure 5	Geostatistical simulation workflow. Source: (ZHANG, 2011).	30
Figure 6	Illustration of SGS Operation. Adapted from (Doyen, P., 2007).	31
Figure 7	The variogram as a poor descriptor of geological heterogeneity. Three different geological heterogeneities result in three similar variograms. Adapted from: (CAERS; ZHANG, 2004).	31
Figure 8	Comparison of distribution modeling. a) histogram of an audio data; b) maximum likelihood uni-modal gaussian; c) GMM and its underlying 10 components. Adapted from (REYNOLDS, 2015).	34
Figure 9	General architecture of IGMN. Source: (HEINEN; ENGEL; PINTO, 2012).	38
Figure 10	Illustration of the DS method. (a) Define the data event in the simulation grid. The question mark is the node to be simulated. The white and black pixels represent nodes that have been previously simulated. (b) Define a search window in the TI grid by using the dimensions a, b, c, d of the data event. (c) scan the search window starting from a random location until (d) the simulation data event is satisfactorily matched. (e) Assign the value of the central node of the first matching data event to the simulated node. Source: (MARIETHOZ; RENARD; STRAUBHAAR, 2010).	43
Figure 11	Ordinary Kriging, IGMN and Modified-IGMN interpolating a simple 1-dimensional data set.	46
Figure 12	Example of placing the moving Gaussian trend component in three different locations	47
Figure 13	Top view illustration of a 2-d gaussian function with: a) covariance matrix aligned to the reference axis; b) covariance matrix rotated by θ	48

Figura 14 a) Training Image (TI); b) A realization of Direct Sampling based on the provided TI.....	51
Figura 15 a) Training Image (TI); b) IGMN multi-variate gaussian components after training	53
Figura 16 Meuse River Data Set - Concentration of Zinc (ppm).	58
Figura 17 Directional variograms (0° , 45° , 90° and 135°) of the Meuse River dataset. Source: (Pebesma, Edzer, 2015).....	59
Figura 18 Correlation Coefficient (Left) and RMSE (Right) between 55 real samples and interpolated values by Kriging, IDW, MLP, IGMN and IGMN-MOD. The known samples were divided into training set sizes of 25, 50, 75 and 100 samples.	60
Figura 19 Left: Ordinary Kriging interpolation over the simulation grid. Right: IGMN interpolation over the simulation grid.	62
Figura 20 Nine realizations of the IGMN sequential simulation algorithm.	63
Figura 21 Mean of 30 realizations performed with IGMN over the simulation grid.....	64
Figura 22 Training images used to evaluate DS+IGMN: a) 100x100 TI used to evaluate noise; b) 150x150 TI used to evaluate speed	65
Figura 23 a) DS+IGMN Realizations with <i>acceptance_threshold</i> = 0.1; b) DS Realizations with <i>acceptance_threshold</i> = 0.1; c) DS+IGMN Realizations with <i>acceptance_threshold</i> = 0; d) DS Realizations with <i>acceptance_threshold</i> = 0.	67
Figura 24 Example of a bad DS+IGMN realization with <i>radius_range</i> set too high [20,25]	68

LIST OF ABBREVIATIONS AND ACRONYMS

ANN	Artificial Neural Network
ART	Adaptive Resonance Theory
DS	Direct Sampling
DSS	Direct Sequential Simulation
EM	Expectation-Maximization
GMM	Gaussian Mixture Model
IDW	Inverse Distance Weighting
IGMN	Incremental Gaussian Mixture Network
ML	Maximum Likelihood
MLP	Multi-Layer Perceptron
MPS	Multiple-Points Statistics
OK	Ordinary Kriging
PDF	Probability Density Function
RBF	Radial Basis Function
SG	Simulation Grid
SGS	Sequential Gaussian Simulation
SIS	Sequential Indicator Simulation
SK	Simple Kriging
SOM	Self-Organizing Maps
TI	Training Image
UK	Universal Kriging
CC	Correlation Coefficient
RMSE	Root Mean Squared Error

TABLE OF CONTENTS

1	INTRODUCTION	19
1.1	MOTIVATION	19
1.2	OBJECTIVES	21
1.2.1	General Objectives	21
1.2.2	Specific Objectives	21
1.3	OUTLINE	21
2	BACKGROUND	23
2.1	GEOSTATISTICS	23
2.1.1	Geostatistical Estimation	24
2.1.1.1	Variograms	25
2.1.1.2	Ordinary Kriging	27
2.1.2	Geostatistical Simulation	29
2.1.2.1	Sequential Gaussian Simulation	29
2.1.3	Multiple-Point Statistics	30
2.2	ARTIFICIAL NEURAL NETWORKS	32
2.2.1	Multi-Layer Perceptron	32
2.3	GAUSSIAN MIXTURE MODELS	33
2.3.0.1	Maximum Likelihood Parameter Estimation	35
3	RELATED WORK	37
3.1	SPATIAL INTERPOLATION	37
3.2	GEOSTATISTICAL SIMULATION AND NEURAL NETWORKS	37
3.3	THE INCREMENTAL GAUSSIAN MIXTURE NETWORK	38
3.3.1	Learning Process	39
3.3.2	Adding New Gaussian Components to the Model	39
3.3.3	Inference on IGMN	39
3.4	MULTIPLE-POINT SIMULATION	40
3.4.1	The Direct Sampling Multiple-Point Simulation Algorithm	41
4	PROBLEM FORMULATION AND PROPOSED SOLUTION	45
4.1	SPATIAL INTERPOLATION WITH IGMN	45
4.1.1	Problem Description: Lack of Trend Component in the IGMN	45
4.1.2	Proposed Solution	46
4.1.3	Problem Description: Diagonal Covariance Matrix for New Gaussian Components	48
4.1.4	Proposed Solution	48

4.2	SEQUENTIAL GAUSSIAN MIXTURE SIMULATION WITH IGMN	49
4.3	MULTIPLE-POINT SIMULATION WITH IGMN AND DS ..	50
4.3.1	Problem: Noise and Speed of the Direct Sampling Method .	50
4.3.2	Proposed Solution	52
4.3.2.1	The Multiple-Point Simulation Algorithm with IGMN and DS .	52
5	EXPERIMENTS AND RESULTS	57
5.1	SPATIAL INTERPOLATION COMPARISON EXPERIMENT	57
5.1.1	Methodology	57
5.1.2	Results	60
5.2	SEQUENTIAL GAUSSIAN MIXTURE SIMULATION EXPERIMENT	61
5.2.1	Methodology	61
5.3	MULTIPLE-POINT SIMULATION EXPERIMENT	62
5.3.1	Methodology	62
5.3.2	Results	65
6	CONCLUSION AND FUTURE WORK	69
	References	71

1 INTRODUCTION

1.1 MOTIVATION

Georeferenced data are an essential component of studies in environmental sciences, such as natural resources evaluation, conservation biology, and oil exploration. Often, only a small dataset of samples are available because this kind of study might involve obtaining data in harsh environments, such as mountainous regions or deep water. Interpolation techniques are commonly applied to estimate mean values in unsampled areas. Some of these studies also aim at analyzing risk, uncertainty and obtain equally probable events in certain areas. Geostatistics provides a set of tools for assessing these kinds of problems. For interpolation of mean values, one of its most popular techniques is Kriging. For analyzing uncertainty, risk and generating equally probable scenarios, Sequential Gaussian Simulation (SGS) is a well accepted geostatistical tool in the petroleum industry (PetroWiki, c2015).

To use Kriging or SGS, it is necessary to model a function describing the degree of spatial correlation between samples, which can be represented as a variogram or correlogram. The variogram modeling procedure determines the main directions of anisotropy and range of influence of each data point. However, modeling the variogram requires expert knowledge of geostatistics procedures and familiarity with the dataset. Therefore, this modeling process might be an obstacle for non-experts in geostatistics.

On the other hand, Artificial Neural Networks (ANNs) models have an interesting feature: the ability to learn from data, even when the user does not know much information about the particular dataset. They also do not require any special model as input, like the variogram. Popular ANNs models, like the Multi-Layer Perceptron (MLP), are well suited for function approximation (regression) and classification problems. However, they are not usually the best choice in problems involving a spatial correlation between samples, as showed in Chapter 5 of this dissertation and in these comparative studies (Nevtipilova, V., et al, 2014; Gumus, K.; Sen, A., 2013).

The main reason to perform simulation rather than interpolation of mean values is that it allows analyzing uncertainty, heterogeneity and visualizing different possible scenarios (realizations). After generating many realizations, one can model the local Probability Density Function (PDF) in every simulated location. Then, this PDF can be used to create risk maps or analyze the probability of occurrence of a particular event. For instance, one may be interested in analyzing the probability of Zinc concentration being over 1700

ppm in a particular area. This kind of analysis cannot be performed with only mean values (interpolation).

In a typical study involving SGS, a realization is generated sequentially. Each new simulated point is incorporated into the current model, which will be used to simulate the next points (Doyen, P., 2007). Consequently, the model has to be updated after the incorporation of every new point. Then, using a neural network model that does not allow online changes, such as MLP with backpropagation learning, would cause an enormous overhead due to the need for retraining the complete model for each new simulated point. Also, MLP networks do not provide the variance of their estimates, which is an essential requirement to perform SGS.

Thereupon, an ideal solution would be to combine the good performance of geostatistical methods with the simplicity of use found in neural networks. With this motivation in mind, we found the Incremental Gaussian Mixture Network (IGMN) (HEINEN, 2011; HEINEN; ENGEL, 2011), a neural network based on parametric probabilistic models. IGMN meets most of the requirements we are interested in, i.e., allows online learning, has good performance in function approximation, does not require variogram modeling and provides variance of its estimates. However, when solving spatial interpolation problems, IGMN may have some issues, especially when the data set is clearly anisotropic and sparse. This is one of the problems we address in this work.

Besides the classical geostatistical methods, when dealing with connectivity patterns, a different class of simulation techniques that look promising is the Multiple-Point Simulation (MPS) (GUARDIANO; SRIVASTAVA, 1993). MPS introduced the idea of using a Training Image (TI) as input data to define the spatial correlation, instead of a two-point Variogram model. From this class, the SNESIN (STREBELLE, 2002) has gained a lot of attention by the oil industry (AITOKHUEHI; DURLOFSKY, 2005; HOFFMAN; CAERS, 2007) but has difficulties in simulating continuous variables and performing co-simulation.

Recently, (MARIETHOZ; RENARD; STRAUBHAAR, 2010) proposed a powerful technique, called Direct Sampling (DS), that can deal either with categorical or continuous variables and also perform co-simulation. However, DS has the disadvantage of generating a lot of noise in its simulations. As stated in (MEERSCHMAN et al., 2013), DS also provides a post-processing step that aims at removing this noise. This post-processing step consists in re-simulating every point multiple times with an informed neighborhood. Although this might help to improve the quality of the image, the computational cost involved in post processing every realization may be impractical.

Motivated by these two problems found in DS, we propose to use

IGMN and DS together to reduce the noise and speed up the simulation. Our proposal consists in pre-processing the training image with IGMN to identify clusters representing regions of spatial continuity. These clusters are later used to allow simulating multiple points at each step, instead of copying only one point at a time (like the DS). The incremental aspect of IGMN allows us not to specify the number of clusters beforehand. Also, the spatial dimensions of the clusters identified by IGMN help to determine the amount of points that might be copied at once.

1.2 OBJECTIVES

1.2.1 General Objectives

The goal of this work is to adapt an Artificial Neural Network model, based on Gaussian Mixture Models with incremental learning, to solve spatial interpolation, sequential simulation, and multiple-point simulation problems, without requiring expert knowledge in Geostatistics or a Variogram model as input.

1.2.2 Specific Objectives

The specific objectives of this work are:

- Improve the IGMN estimation process for sparse and anisotropic spatially correlated datasets;
- Develop a method to perform unconditional Sequential Gaussian Simulation using IGMN, without requiring a Variogram model;
- Develop a method that mixes Direct Sampling and IGMN to perform Multiple-Point Simulations faster and less noisy than DS.

1.3 OUTLINE

The remaining of this dissertation is organized as follows. Chapter II presents basic concepts and some background about Geostatistics, Neural Networks and Gaussian Mixture Models. Chapter III discusses the related work. Chapter IV describes the specific problems we are trying to solving and the proposed solutions. In that chapter, we firstly show the ideas for im-

proving the estimation process of IGMN. Then, we describe our Incremental Gaussian Mixture Sequential Simulation algorithm. Finally, we present our MPS algorithm that mixes IGMN and DS. Chapter V details the experiments performed to evaluate our proposal and the obtained results. Chapter VI presents our conclusions and future work.

2 BACKGROUND

2.1 GEOSTATISTICS

Geostatistics has its origins in the petroleum and mining fields, starting with Krige (1951) and further developed by Matheron (1962). It is concerned with solving interpolation and estimation problems that arise when scientists have sparse data to analyze. In the petroleum industry, for example, Geostatistics tools are usually applied to help taking decisions concerning expensive operations based on interpretations over a sparse spatially correlated dataset available. Nowadays, Geostatistics is applied to many other fields, e.g., oceanography, meteorology, soil sciences, forestry, hydrology, landscape ecology.

Different from traditional statistics, Geostatistics do not assume, for example, all samples of a population are normally distributed and independent from one another. This assumption is often not satisfied in earth science datasets, which may contain high spatial correlation between nearby samples. Instead, Geostatistics incorporates both the statistical distribution of the samples and the spatial correlation between them. That is why many earth science problems are better addressed with geostatistical methods (ZHANG, 2011).

One may think, why not just use simple interpolation? We will illustrate the answer to this question with an example. But before that, we can formalize the notation for a random process that has become conventional in Geostatistics as follows (OLIVER; WEBSTER, 2014):

1. A *realization* of the process is the value of a property $z(x)$ at any location x (equivalent to x_1, x_2 in two dimensions) is one of an infinity of values of a random variable $Z(x)$ at that place.
2. The set of random values at all such locations, in a region, is a random process, also denoted $Z(x)$.
3. The random variable is spatially correlated at some scale.

Now, back to our example, suppose we want to estimate the permeability at the unsampled location z_0 (Figure 1). Using the basic Inverse Distance Weighting (IDW) method, z_0 can be calculated as:

$$z_0 = \sum_{i=1}^n w_i z_i \quad (2.1)$$

$$w_i = \frac{1/d_i}{\sum_{i=1}^n (1/d_i)} \quad (2.2)$$

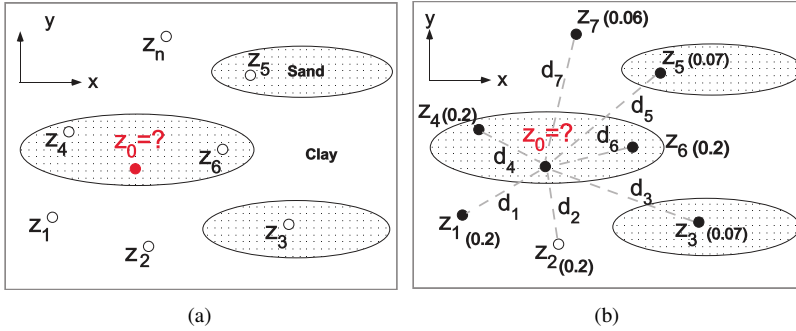


Figure 1: a) Estimation of the unknown permeability z_0 based on a set of n known values; b) Estimation of the unknown z_0 given 7 known values. Numbers in parenthesis are weights assigned based on inverse distance. Adapted from (ZHANG, 2011).

From the above equations we can see that z_0 is calculated as a weighted sum of the n known values. Each weight w_i is determined solely by the distances d_i between z_0 and each known point z_i . For $n = 7$, the weights calculated to points 1, 2, 4 and 6 are all equal to 0.2. On the other hand, from the point of view of geology, we would expect points 4 and 6 to be more similar to z_0 (have greater weight) because they are located in the horizontal direction inside the sand body. So, this method does not incorporate spatial correlation information. On the other hand, Geostatistics methods take into account both spatial correlation and distance information. This information is encoded into a model called Variogram.

2.1.1 Geostatistical Estimation

A typical geostatistical estimation consists of 3 steps (ZHANG, 2011): (1) examining spatial correlation of the dataset via an experimental variogram analysis; (2) modeling the Variogram, which consists in fitting a permissible mathematical function to the experimental Variogram; (3) performing Kriging interpolation based on the Variogram model and the dataset. Figure 2 illustrates a workflow for performing a geostatistical estimation study using Kriging, one of the most well known geostatistical interpolation method.

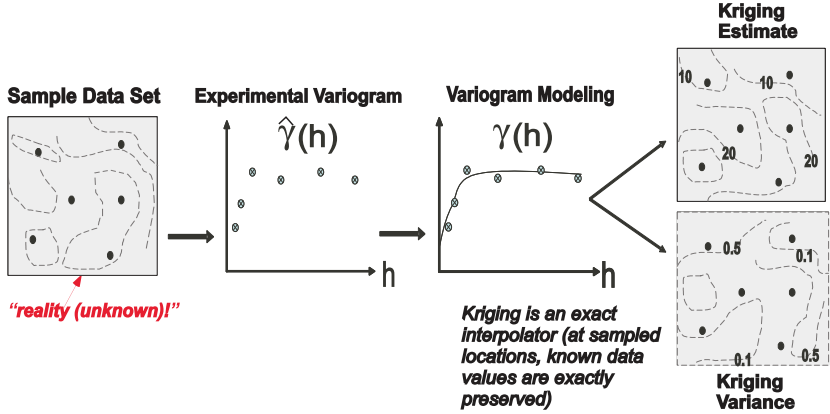


Figure 2: Geostatistical estimation workflow using Kriging. Adapted from (ZHANG, 2011).

2.1.1.1 Variograms

The variogram is defined by Cressie (1993) as the variance of the difference between field values at two locations (x and y):

$$2\gamma(x, y) = \text{var}(Z(x) - Z(y)) = E[(Z(x) - \mu(x)) - (Z(y) - \mu(y))]^2] \quad (2.3)$$

We can rewrite this as the expectation for the squared increment of the values between locations x and y assuming the spatial random field has a constant mean μ (WACKERNAGEL, 2003):

$$2\gamma(x, y) = \text{var}(Z(x) - Z(y)) = E[(Z(x) - Z(y))^2] \quad (2.4)$$

where $\gamma(x, y)$ is called the *semivariogram*, or simply variogram. The assumption made by Matheron (1965) of intrinsic stationarity, allows us to represent the variogram as a function $\gamma(h)$ of the difference $h = y - x$, where $\gamma(h)$ is equivalent to $\gamma(0, y - x)$. We can understand h as the separation distance between the locations and it is commonly known as lag distance. When $h = \|y - x\|$, the variogram is said to be isotropic, i.e., it depends only on the separation distance h and does not take into account the direction between the samples. When it takes into account the direction, it is said to be anisotropic and we can specify a main direction of anisotropy.

When we speak about variogram it is important to separate two main

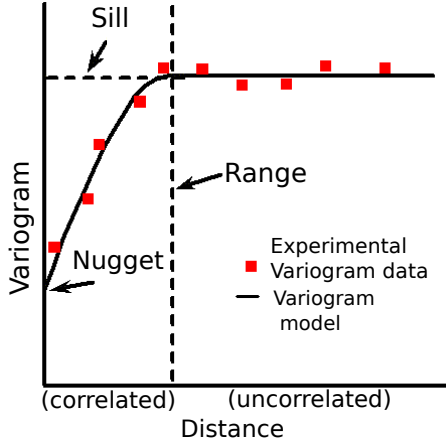


Figure 3: Example of a variogram plot. Source: (PNNL, 2016).

concepts: the experimental (or empirical) variogram and the theoretical variogram (or variogram model). The experimental variogram can be calculated over the field data using the method of moments (MATHERON, 1965):

$$\hat{\gamma}(h) = \frac{1}{2m(h)} \sum_{j=1}^{m(h)} \{z(x_j) - z(x_j + h)\}^2 \quad (2.5)$$

where $m(h)$ is the number of paired comparisons at lag distance h . By incrementing h in steps we obtain an ordered set of values, as shown in Figure 3. Calculating experimental variograms to examine the dataset is the first step in an estimation process. The basic idea is to plot multiple directional variograms to find out the main direction of anisotropy and other important parameters:

- *sill* s : Limit of the variogram tending to infinity lag distances.
- *range* r : Represents maximum distance at which two samples are correlated. Conventionally, the distance when the variogram reaches 95% of the sill.
- *nugget* n : The height of the discontinuity at the origin.

The second step in the estimation workflow is to replace the empirical variogram with a variogram model, i.e., fit a model to the sample data. This is necessary because when using Kriging, it may be required to access the variogram values for lag distances different from those in the empirical variogram.

More importantly, the variogram needs to be a positive definite function, in order the Kriging system of equations to be non-singular (BOHLING, 2005). Therefore, there is a list of *licit* variogram models where geostatisticians can choose from and combine them. Three of the most frequently used models are (CRESSIE, 1993; CHILES; DELFINER, 2009):

- The spherical variogram model:

$$\gamma(h) = (s-n) \left(\left(\frac{3h}{2r} - \frac{h^3}{2r^3} \right) 1_{(0,r)}(h) + 1_{[r,\infty)}(h) \right) + n1_{(0,\infty)}(h) \quad (2.6)$$

- The exponential variogram model:

$$\gamma(h) = (s-n) \left(1 - \exp\left(\frac{-h}{r}\right) \right) + n1_{(0,\infty)}(h) \quad (2.7)$$

- The gaussian variogram model:

$$\gamma(h) = (s-n) \left(1 - \exp\left(-\frac{h^2}{r^2}\right) \right) + n1_{(0,\infty)}(h) \quad (2.8)$$

Figure 4 illustrates these three variogram models. The gaussian model has a parabolic behavior at the origin and represents properties that vary very smoothly. The spherical and exponential models have linear behavior at the origin and represents well properties with a higher level of short-range variability (BOHLING, 2005).

2.1.1.2 Ordinary Kriging

Given spatial data $Z(x_i)$ and assuming an intrinsically stationary process, i.e. having constant unknown mean μ and known variogram function $\gamma(h)$, the value at an unobserved location x_0 can be calculated as a linear combination of Ordinary Kriging weights ω_j^{OK} and the known samples $Z(x_i)$:

$$\hat{Z}(x_0) = \sum_{j=1}^n \omega_j^{OK} Z(x_i) \quad (2.9)$$

The weights ω_j^{OK} are calculated by Ordinary Kriging and they have two important goals in the estimation process: 1. reflect the proximity of samples to the estimation location; 2. avoid bias caused by groups of samples clustered (screening effect). The Ordinary Kriging system of equations can be written as (LINCHTENSTERN, 2013):

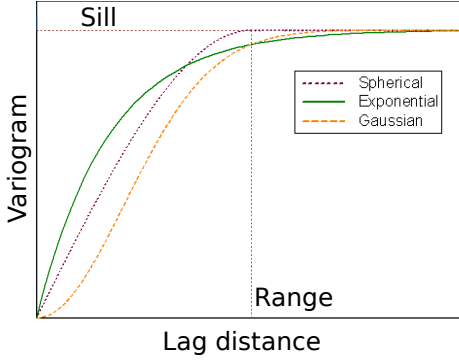


Figure 4: Plots of Gaussian, Spherical and Exponential theoretical variogram models. Adapted from: (BOHLING, 2005).

$$\sum_{j=1}^n \omega_j^{OK} \gamma(x_i - x_j) + \lambda_{OK} = \gamma(x_i - x_0) \text{ for } i = 1, \dots, n \quad (2.10)$$

In the geostatistical formalism, the calculation of the weights ω_j^{OK} has two objectives: minimize the variance of estimation and ensure that the model is unbiased. We can also re-write the above equation in a matrix form:

$$(2.11)$$

$$\begin{pmatrix} \gamma(\mathbf{x}_1 - \mathbf{x}_1) & \gamma(\mathbf{x}_1 - \mathbf{x}_2) & \cdots & \gamma(\mathbf{x}_1 - \mathbf{x}_n) & 1 \\ \gamma(\mathbf{x}_2 - \mathbf{x}_1) & \gamma(\mathbf{x}_2 - \mathbf{x}_2) & \cdots & \gamma(\mathbf{x}_2 - \mathbf{x}_n) & 1 \\ \vdots & \vdots & \cdots & \vdots & \vdots \\ \gamma(\mathbf{x}_n - \mathbf{x}_1) & \gamma(\mathbf{x}_n - \mathbf{x}_2) & \cdots & \gamma(\mathbf{x}_n - \mathbf{x}_n) & 1 \\ 1 & 1 & \cdots & 1 & 0 \end{pmatrix} \begin{pmatrix} \omega_1^{OK} \\ \omega_2^{OK} \\ \vdots \\ \omega_n^{OK} \\ \lambda_{OK} \end{pmatrix} = \begin{pmatrix} \gamma(\mathbf{x}_1 - \mathbf{x}_0) \\ \gamma(\mathbf{x}_2 - \mathbf{x}_0) \\ \vdots \\ \gamma(\mathbf{x}_n - \mathbf{x}_0) \\ 1 \end{pmatrix}$$

where $\omega_{OK} := ((\omega_1)^{OK}, \dots, (\omega_n)^{OK})^T \in \mathbb{R}^n$ denotes the vector providing the optimal weights ω_i^{OK} . $\lambda_{OK} \in \mathbb{R}$ is the Lagrange multiplier of ordinary kriging. $\gamma(x_i - x_0)$ are the variogram values between the known samples and the estimation location. The Ordinary Kriging variance can be calculated as (LINTCHENSTERN, 2013):

$$\sigma_{OK}^2 = \lambda_{OK} + \sum_{i=1}^n \omega_i^{OK} \gamma(x_i - x_0) \quad (2.12)$$

The weights can be calculated by solving the system for ω using Gauss elimination, for example. Thus, inverting the γ matrix will be necessary. This matrix inversion operation can be computationally expensive when the number of known points n is large.

There are other kinds of Kriging, such as the Simple Kriging (SK) and Universal Kriging (UK). In the SK, the mean is assumed constant and known over the entire domain, not only in the search neighborhood as in the Ordinary Kriging. In the UK, it is possible to specify the trend using a general polynomial model.

2.1.2 Geostatistical Simulation

Geostatistical simulation is a stochastic approach that provides means for calculating different equally probable solutions (realizations), for an interpolation problem. Interpolation methods, such as Kriging and IDW, do not aim at representing the actual variability of the studied variable. In fact, they aim at providing a good average value (PetroWiki, c2015). However, some applications, such as oil reservoir modeling and risk mapping, are interested in studying different scenarios, capturing heterogeneity and assessing uncertainty.

Therefore, in a typical Geostatistical simulation study, we are interested in generating multiple estimations for the same locations and use these estimations to build, for example, local probability density functions (PDF). Then, it is possible to analyze uncertainty, heterogeneity, build risk maps and other kinds of analysis. It is important to notice that this uncertainty is due to our lack of knowledge about the problem (sparse samples). Figure 5 illustrates a common Geostatistical simulation workflow.

2.1.2.1 Sequential Gaussian Simulation

The Sequential Gaussian Simulation is a well accepted method for simulating continuous variables in the petroleum industry (PetroWiki, c2015). Figure 6 illustrates how this method works. Before running the algorithm, it is defined a simulation grid and performed a normal score transformation over the raw data (if data is not normally distributed). Then, the algorithm works as follows (Doyen, P., 2007):

1. Randomly picks a non-simulated cell ($i = 7$);
2. Calculate Kriging mean and variance;

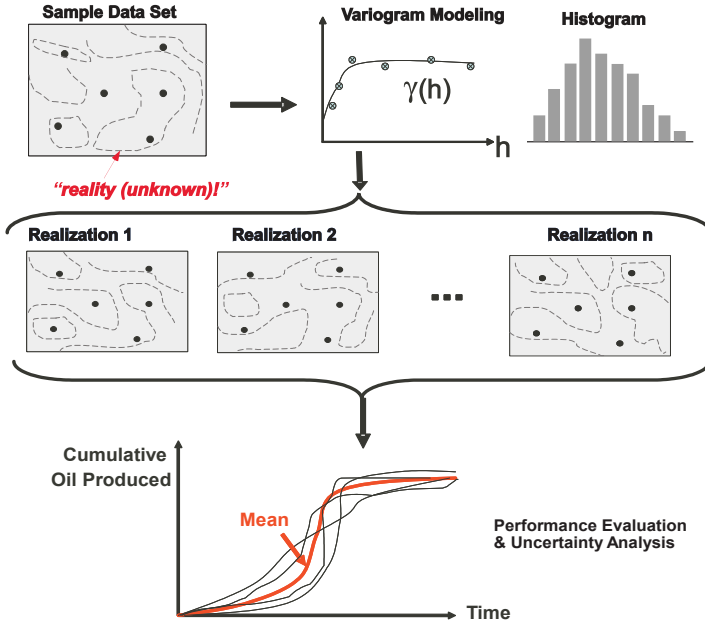


Figure 5: Geostatistical simulation workflow. Source: (ZHANG, 2011).

3. Assigns a random value x_i taken from a Gaussian distribution with mean and variance calculated in step 2;
4. Incorporates x_i as a known data point;
5. Repeat steps 1-4 until the whole grid is simulated.

In section 4.2 we present a modified version of this algorithm to perform simulation using IGMN as interpolator instead of Kriging.

There are, also, other sequential simulation methods such as the Sequential Indicator Simulation (SIS), for discrete variables and the Direct Sequential Simulation (DSS), which is applied direct to the available data, not performing any transformation.

2.1.3 Multiple-Point Statistics

Traditional geostatistics uses variogram models to characterize the spatial correlation between samples. This is known as two-point geostatistics, since a variogram value depends only on the separation distance and

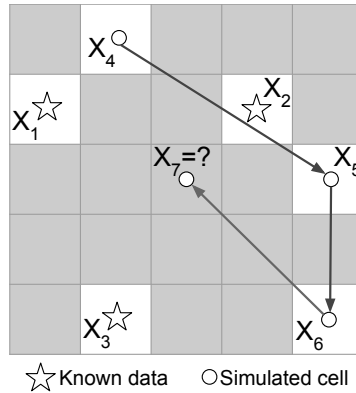


Figure 6: Illustration of SGS Operation. Adapted from (Doyen, P., 2007).

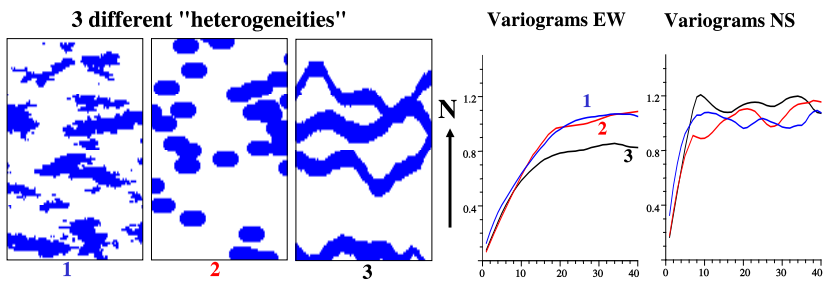


Figure 7: The variogram as a poor descriptor of geological heterogeneity. Three different geological heterogeneities result in three similar variograms. Adapted from: (CAERS; ZHANG, 2004).

direction between two samples. However, shapes of geological bodies, such as channels, and curvilinear structures cannot be correctly captured by the variogram model (JOURNAL, 1992; STREBELLE, 2002). Figure 7 shows three images with very different spatial correlation patterns but with similar variograms.

Therefore, Multiple-Point Statistics (MPS) was created to address this issue. MPS offers a new set of tools that can model complex and heterogeneous geological environments through the use of Training Images (TI). A TI may be a photograph, a draw, or anything that generally describes the geometrical spatial characteristics of a region. For example, the images shown in Figure 7 could be used as input for a MPS simulation method.

Simulation methods based on MPS allow capturing geological ele-

ments like channels, reefs, bars, dikes or different oriented facies, while honouring data information. Two popular MPS simulations methods are the SNESIM (STREBELLE, 2002) and, more recently, the Direct Sampling (DS) (MARIETHOZ; RENARD; STRAUBHAAR, 2010). We describe the DS method with more details in the next chapter.

2.2 ARTIFICIAL NEURAL NETWORKS

An Artificial Neural Network (ANN) can be defined as a computing system composed by many highly interconnected simple processing units (called neurons), which process information by their dynamic state response to external inputs (CAUDILL, 1989). There are many kinds of ANN models and they can be classified as supervised or unsupervised learning approaches. Two examples of supervised learning models are the Multi-Layer Perceptron (MLP) (RUMENLHART; HINTON; WILLIAMS, 1986), which is by far the most popular model, and the Radial Basis Function (RBF), which is commonly used for non-linear regression tasks. On the unsupervised learning side, one example is the Self-Organizing Maps (SOM) (KOHONEN; SCHROEDER; HUANG, 2001), which can detect patterns from the input data and generate a topological map. Another interesting model is the ART, which is an ANN model based on the adaptive resonance theory (CARPENTER; GROSSBERG, 1987).

One interesting feature present in ANNs is their ability to learn from data without requiring too much knowledge about the dataset or the ANN model. This allows us to use them as black boxes by inputting some training data and tweaking a few parameters. Also, the MLP model for example, is an universal approximator and very powerful for either function approximation or classification tasks (HORNIK; STINCHCOMBE; WHITE, 1989).

2.2.1 Multi-Layer Perceptron

The Multi-Layer Perceptron is one of the most popular ANN models. Its architecture is organized in several layers of neurons which are fully connected with the subsequent layer. It has a minimum of 3 layers, the input layer, one or more hidden layers and the output layer. Each neuron return as output the result of an activation function computed over a weighted combination of its inputs as shown in equation 2.13

$$y_j(n) = \varphi \left(\sum_{i=0}^D \omega_{ji}(n) y_i(n) \right) \quad (2.13)$$

where $\varphi(x)$ is the activation function, D is the of inputs applied to neuron j , ω_{ji} is the synaptic weight that connects the neuron j from the previous layer to the neuron i in the current layer and $y_i(n)$ is the input signal at neuron j (or the output of neuron i from previous layer). The logistic function (equation 2.14) is commonly used as activation function.

$$\varphi(n) = \frac{1}{1 + \exp \left(- \sum_{i=0}^D \omega_{ji}(n) y_i(n) \right)} \quad (2.14)$$

The training process in MLP is usually done using the Backpropagation algorithm (RUMENLHART; HINTON; WILLIAMS, 1986), which is a supervised learning method that performs a gradient descent towards a global minimum along the steepest vector of the error surface. In other words, the training process can be summarized as (SATHYANARAYANA, 2014):

1. The Neural Network weights must be adjusted such that the error calculated on a known training set is minimized.
2. The strategy to minimize the error is: iteratively change the weights by a small amount proportional to the partial derivative of the error with respect to that weight.
3. Keep refining the weights until the error is low enough or the error does not fall anymore.

2.3 GAUSSIAN MIXTURE MODELS

We can define a Gaussian Mixture Model (GMM) as a parametric probability density function that assumes all the data points are generated from a mixture of finite number of Gaussian distributions. This mixture is represented as weighted sum of Gaussian component densities as given by equation 2.15 (REYNOLDS, 2015).

$$p(x|\lambda) = \sum_{i=1}^M \omega_i g(x|\mu_i, \Sigma_i) \quad (2.15)$$

$$g(x|\mu_i, \Sigma_i) = \frac{1}{(2\pi)^{D/2} |\Sigma_i|^{1/2}} \exp \left\{ -\frac{1}{2} (x - \mu_i)' \Sigma_i^{-1} (x - \mu_i) \right\} \quad (2.16)$$

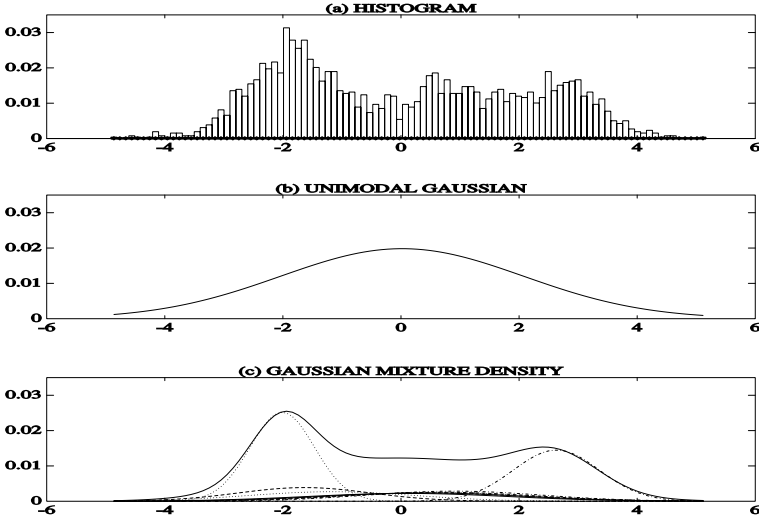


Figure 8: Comparison of distribution modeling. a) histogram of an audio data; b) maximum likelihood uni-modal gaussian; c) GMM and its underlying 10 components. Adapted from (REYNOLDS, 2015).

where x is a D-dimensional data vector, $\omega_i, i = 1, \dots, M$, are the mixture weights, $g(x|\mu_i, \Sigma_i), i = 1, \dots, M$ are the component Gaussian densities as defined in equation 2.16 with mean μ_i and covariance matrix Σ_i . The weights ω_i are non-negative and sum to 1 (convex combination). As Reynolds (2015) states, the complete GMM is parameterized by its mean vectors, covariance matrices and mixture weights:

$$\lambda = \{\omega_i, \mu_i, \Sigma_i\} \quad i = 1, \dots, M. \quad (2.17)$$

There are many configurations of GMMs, they may have full rank covariance matrices or be constrained to diagonal covariance matrices. Also, their covariance matrices may be unique for each component or a shared one for all components. Figure 8 compares the adjust of an unimodal gaussian and a GMM over some audio data. We can observe that the GMM is able to clearly model the multimodal nature of the data besides smoothly fitting the distribution.

2.3.0.1 Maximum Likelihood Parameter Estimation

The most popular technique to estimate the parameters of a GMM is the Maximum Likelihood (ML) estimation. It aims at maximizing the likelihood of the GMM given the training data. The GMM likelihood can be calculated as:

$$p(X|\lambda) = \prod_{t=1}^T p(x_t|\lambda) \quad (2.18)$$

where T is a sequence of training vectors $X = \{x_1, \dots, x_T\}$ and λ are the GMM parameters. Because equation 2.18 is non-linear of the parameters λ , we cannot maximize it directly. An alternative solution is to use the Expectation-Maximization (EM) iterative algorithm. The elementary idea of EM is to start with a model λ and, at each iteration, obtain a new model $\tilde{\lambda}$ such that $p(X|\tilde{\lambda}) \geq p(X|\lambda)$ until some convergence criteria is met. EM re-calculates the weights, means and covariances using (REYNOLDS, 2015):

Mixture Weights

$$\bar{\omega} = \frac{1}{T} \sum_{t=1}^T Pr(i|x_t, \lambda) \quad (2.19)$$

Means

$$\bar{\mu}_i = \frac{\sum_{t=1}^T Pr(i|x_t, \lambda)x_t}{Pr(i|x_t, \lambda)} \quad (2.20)$$

Variances (diagonal covariance)

$$\bar{\sigma}_i^2 = \frac{\sum_{t=1}^T Pr(i|x_t, \lambda)x_t^2}{Pr(i|x_t, \lambda)} - \bar{\mu}^2 \quad (2.21)$$

$Pr(i, x_t, \lambda)$ is the *a posteriori* probability for the component i

$$Pr(i, x_t, \lambda) = \frac{\omega_i g(x_t|\mu_i, \Sigma_i)}{\sum_{k=1}^M \omega_k g(x_t|\mu_k, \Sigma_k)} \quad (2.22)$$

3 RELATED WORK

3.1 SPATIAL INTERPOLATION

Recently, our research group started investigating the possibility of replacing geostatistical methods with Neural Networks for spatial interpolation tasks. Fazio and Roisenberg (FAZIO; ROISENBERG, 2013; FAZIO, 2013) analytically proved that RBF networks and Kriging are equivalent when using the variogram model as activation function of the RBF network. Also, they showed that training an RBF network is faster than solving the Kriging system of equations.

Other studies evaluated the performance of neural networks and other methods in spatial interpolation problems. Gumus, K.; Sen, A. (2013) compared MLP networks, Ordinary Kriging, IDW (Inverse Distance Weighting) and MRBF (Multiquadratic Radial Basis Function) in the problem of interpolating a digital elevation model. Regarding RMSE (Root Mean Square Error), MLP obtained the worst results. Similarly, Nevtipilova, V., et al (2014) evaluated MLP against Kriging and IDW in the approximation of three surface functions and MLP's performance was even worse than IDW while Kriging obtained the best performance in both studies.

On the other hand, Deligiorgi and Philippopoulos (Deligiorgi, D.; Philippopoulos, K., 2011) achieved better performance using MLP, rather than Kriging, in the spatial interpolation of air pollution in Athens, Greece. However, the authors ignored potential anisotropy in the variogram model, choosing an isotropic model, which may be responsible for the poor performance of Kriging. This result endorses one of the motivations for this work: develop a method that provides good performance without the need of expert knowledge in geostatistics.

3.2 GEOSTATISTICAL SIMULATION AND NEURAL NETWORKS

Some studies relate Gaussian Mixture Models and geostatistical simulation. Gilardi, N. ; Bengio, S. ; Kanevski, M. (2002) compared Sequential Gaussian Simulation (SGS) and Gaussian Mixture Models (GMMs) for environmental risk mapping applications. Their results show that the simulation grid generated from the GMM suffers from a high degree of smoothness when compared to SGS. We attribute this high level of smoothness to the fact that their method seems to adjust only a single global the GMM and do not se-

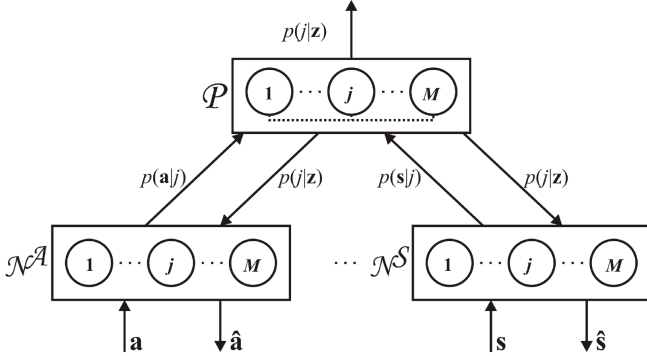


Figure 9: General architecture of IGMN. Source: (HEINEN; ENGEL; PINTO, 2012).

quentially update the model to incorporate every new simulated point.

Grana and Mukerji (GRANA; MUKERJI et al., 2012) proposed a sequential simulation approach that uses Gaussian Mixture Models in the estimation of the prior distribution for seismic inversion applications. Their approach uses the Expectation-Maximization (EM)(DEMPSTER; LAIRD; RUBIN, 1977) algorithm to adjust the GMM and a variogram model to control the spatial correlation. Although they have obtained interesting results regarding the correctness of the simulation, their approach still requires variogram modeling and uses the non-incremental EM algorithm to adjust the GMM, which may lead to a high computational cost.

3.3 THE INCREMENTAL GAUSSIAN MIXTURE NETWORK

This work builds upon IGMN; a neural network model proposed in (HEINEN, 2011; HEINEN; ENGEL, 2011) that features particularly interesting characteristics which may be explored and improved to solve geostatistical problems. IGMN aims at approximating the results of the iterative EM algorithm (DEMPSTER; LAIRD; RUBIN, 1977) for adjusting Gaussian Mixture Models in an incremental way, using only a single scan through the data. In this section, we briefly present the most relevant aspects of IGMN.

3.3.1 Learning Process

Figure 9 shows the general architecture of IGMN. It consists of cortical regions, $\mathcal{N}^A, \mathcal{N}^S$, for each variable and an association region \mathcal{P} . Initially, the Gaussian Mixture Model inside IGMN has no Gaussian components. New components are incrementally added to the model when necessary, according to a novelty criteria. When a new training tuple x is given to IGMN, it calculates the squared Mahalanobis distance $d_M^2(x, j)$ to every existing Gaussian components j of its mixture model (PINTO; ENGEL, 2015). If this tuple is not close enough to any existing component, according to a chi-square test, with significance level being the novelty criteria controlled by the user, a new Gaussian component is created, and all parameters of the model are updated. Some of the parameters are: the means, covariance matrices, age and the prior probability of each Gaussian component. Update equations for all parameters are derived from the Robbins-Monro stochastic approximation (ROBBINS, 1951) and the derivations are available in (ENGEL; HEINEN, 2010; ENGEL, 2009).

3.3.2 Adding New Gaussian Components to the Model

When a new training data x does not meet the novelty criteria, i.e., it is far from all gaussian components in the mixture model, IGMN creates a new gaussian component with the following initial values: mean $\mu_j = x$, covariance matrix $C_j = \sigma_{ini}^2 I$, age $sp_j = 1$, $v_j = 1$ and prior probability $p(j) = \frac{1}{\sum_{i=1}^K sp_i}$. Where K is the number of components, including the newly created (PINTO; ENGEL, 2015) one. The parameter σ_{ini} is provided by the user and it controls the initial values of the covariance matrices. An important observation regarding the creation of new components is that IGMN assigns a diagonal covariance matrix. Therefore, initially, the covariance matrix of a new Gaussian component is aligned with the X, Y axis. As more training data is provided to IGMN, the better it will identify and update each component's covariance matrices to reflect potential anisotropy in other directions.

3.3.3 Inference on IGMN

During training, IGMN does not distinguish between inputs and targets, they are presented together, as an input tuple. For making an inference, IGMN interpolates the center (means) of the target variable conditioned to

the posterior probabilities of the given variables. For example, if we have a tridimensional dataset, where each single data is $x = (a, b, c)$. If we want to estimate c , given a, b , first we split this tuple in two parts: $x_t = c$ (target) and $x_i = a, b$ (input). Then, the posterior probabilities are calculated as in equation 3.1. The target x_t can be estimated using the conditional mean equation presented in equation 3.2.

$$p(j|x_i) = \frac{p(x_i|j)p(j)}{\sum_{q=1}^K p(x_i|q)p(q)} \forall j \quad (3.1)$$

$$\hat{x}_t = \sum_{j=1}^K p(j|x_i)(\mu_{j,t} + C_{j,ti}C_{j,i}^{-1}(x_i - \mu_{j,i})) \quad (3.2)$$

$C_{j,ti}$ represents the sub-matrix of the j th gaussian component covariance matrix relating the input and targets. $C_{j,i}$ represents the known part only and $\mu_{j,t}$ is the mean of the j th component of the target variable and $\mu_{j,i}$ is the mean of the j th component of the input data (PINTO; ENGEL, 2015). The covariance matrix with the mentioned sub-matrices is show below:

$$C_j = \left(\begin{array}{c|c} C_{j,i} & C_{j,it} \\ \hline C_{j,ti} & C_{j,t} \end{array} \right) \quad (3.3)$$

IGMN is also able to provide the (co)variance of its estimates (equation 3.4). This will be particularly important for our sequential gaussian simulation implementation, described in section III.

$$\hat{C}_t = \sum_{j=1}^K \left\{ C_{j,t} - C_{j,ti}C_{j,i}^{-1}C_{j,ti}^T + \|\bar{\mu}_{j,t} - \hat{x}_t\| \right\} \quad (3.4)$$

3.4 MULTIPLE-POINT SIMULATION

A few years ago, Mariethoz, Renard e Straubhaar (2010) proposed a powerful method called the Direct Sampling (DS). It is a Multiple-Point Simulation algorithm that can deal either with categorical or continuous variables and can also perform co-simulation. This method has been attracting a lot of attention, mainly because it is straightforward to implement, easy to parallelize and to use. One problem, though, is that DS generates noise in its realizations. Meerschman et al. (2013) suggests post-processing each realization to remove the noise. The post-processing step consists in re-simulating every point with an informed neighborhood. While solving the problem of the noise, post-processing increases computational cost. We describe the DS

algorithm in the next section. In Section 4.3 we modify this algorithm and mix it with IGMN to reduce noise and speed it up.

3.4.1 The Direct Sampling Multiple-Point Simulation Algorithm

The DS algorithm aims at simulating a random function $Z(x)$. As input, it receives a simulation grid (SG), whose nodes are denoted x , a training image (TI), whose nodes are denoted y and, optionally, a set of N conditioning points $z(x_i), i \in [1, \dots, N]$. The steps of the algorithm are:

1. If there are conditioning points, assign them to the closest simulation grid cells.
2. Define a path through the remaining unsimulated points in the SG. The path specifies an order to visit each location. It may be Random (STREBELLE, 2002), unilateral or any other kind of path.
3. For each location x in the path:
 - (a) Find the neighbors of x (n closest grid nodes $\{x_1, x_2, \dots, x_n\}$ already assigned in SG). If no neighbor exists, randomly take a node y from TI and assign its value $Z(y)$ to $Z(x)$ in the SG. Then proceed to the next unsimulated node.
 - (b) If it found neighbors, compute the lag vector $L = \{h_1, \dots, h_n\} = \{x_1 - x, \dots, x_n - x\}$ which define the neighborhood of x , $N(x, L) = \{x + h_1, \dots, x + h_n\}$. Figure 10a shows an example where x has three neighbors and the lag vectors are: $L = \{(1, 2), (2, 1), (-1, 1)\}$. It represents the relative locations of the already simulated points.
 - (c) Compute the data event $d_n(x, L) = \{Z(x + h_1), \dots, Z(x + h_n)\}$, which is a vector with the values of the variable of interest at the neighborhood. In the example of Figure 10a, $d_n(x, L) = \{0, 0, 1\}$.
 - (d) Calculate the dimension of the search window in the TI. It is defined by the maximum and minimum values at each dimension of the lag vectors (Figure 10b).
 - (e) Randomly choose a location y in the search window and from this location, start scanning the whole window. For each location y :
 - i. Calculate the data event $d_n(y, L)$ in the TI. In Figure 10c, a random grid node was selected and its data event is $d_n(y, L) = \{1, 0, 1\}$.

- ii. Compute the distance $d\{d_n(x, L), d_n(y, L)\}$ between data events found in the SG and the TI. The distance may be computed in many different ways. This is one of the most important steps of DS, because this calculation is what allows it to simulate either continuous or discrete variables and also perform cossimulation by joining different variables in the same calculation. In our example, we perform a simple mean of the categorical values as show in Equation 4.5.

$$d\{d_n(x), d_n(y)\} = \frac{1}{n} \sum_{i=1}^n a_i \in [0, 1], \quad (3.5)$$

$$\text{where } a_i = \begin{cases} 0 & \text{if } Z(x_i) = Z(y_i) \\ 1 & \text{if } Z(x_i) \neq Z(y_i) \end{cases}$$

- iii. Store $y, Z(y)$ and $d\{d_n(x, L), d_n(y, L)\}$ if it is the lowest distance calculated so far for the current point being simulated.
- iv. If $d\{d_n(x, L), d_n(y, L)\}$ is smaller than the acceptance threshold t , the value $Z(y)$ is assigned to $Z(x)$. Figure 10d illustrates this step. In that case, the data event in the TI matches exactly the one in the SG, so the distance is zero and the value $Z(y) = 1$ is copied to the SG.
- v. Else, if the number of iterations exceeds a certain fraction of the TI, the lowest distance of node y obtained so far is accepted and the value $Z(y)$ is copied to $Z(x)$.

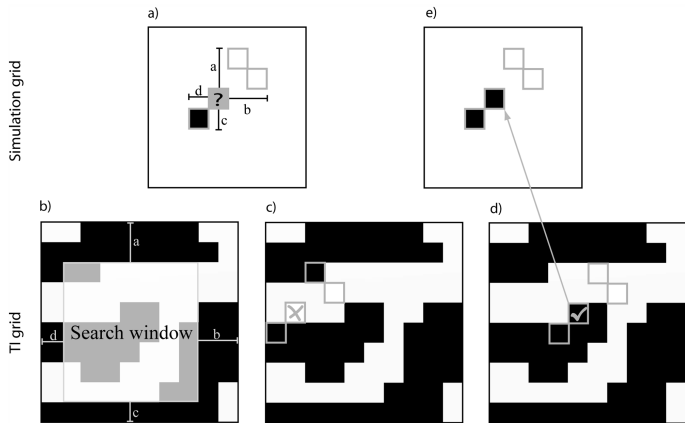


Figure 10: Illustration of the DS method. (a) Define the data event in the simulation grid. The question mark is the node to be simulated. The white and black pixels represent nodes that have been previously simulated. (b) Define a search window in the TI grid by using the dimensions a , b , c , d of the data event. (c) scan the search window starting from a random location until (d) the simulation data event is satisfactorily matched. (e) Assign the value of the central node of the first matching data event to the simulated node. Source: (MARIETHOZ; RENARD; STRAUBHAAR, 2010).

4 PROBLEM FORMULATION AND PROPOSED SOLUTION

In this Chapter, we will present specific problems and proposed solutions to apply IGMN to spatial interpolation and geostatistical simulation applications. First, in Section 4.1, we describe an improvement to the IGMN estimation process, inspired by how Kriging deals with trend and anisotropy in its Variogram model. Then, in Section 4.2, provided with this improved variant of IGMN, we describe our modified version of the SGS algorithm that uses IGMN instead of Kriging for the interpolation and variance calculation step. This version of SGS, using IGMN is unable to generate exact conditioned simulations because IGMN does not store the original data values. So, it is better suited to unconditioned simulations. Finally, in Section 4.3.2.1, we describe our Multiple-Point Simulation algorithm, which can generate conditioned simulations. This algorithm is a modified version of the Direct Sampling (MARIETHOZ; RENARD; STRAUBHAAR, 2010) and uses IGMN to reduce noise and speed up the simulation.

4.1 SPATIAL INTERPOLATION WITH IGMN

4.1.1 Problem Description: Lack of Trend Component in the IGMN

Figure 11 shows a simple example comparing the interpolation of IGMN and Ordinary Kriging methods over a 1-dimensional small data set. As can be seen, Ordinary Kriging estimations tend to a constant mean value when the point being estimated distances from known points. In this example, when the location being estimated is out of the range of influence (determined by the variogram) of all known points, the interpolated value becomes equal to the arithmetic mean of the data. On the other hand, IGMN estimations tend to the closest known value, no matter how far it is.

This behavior is due to the normalization term of IGMN's inference equation (eq. 3.2), which uses Bayes' rule. These posterior probabilities represent the statistical distance between each component j of the Gaussian mixture and the point to be estimated. When the point is far from all Gaussian components of the mixture, all the components will have posterior probability very close to zero. But, the closest Gaussian, will still have a probability orders of magnitude greater than the other components, even if they are only slightly further. Thus, after normalizing the posterior probabilities, the closest Gaussian will have a probability close to 1, which generates the saturated

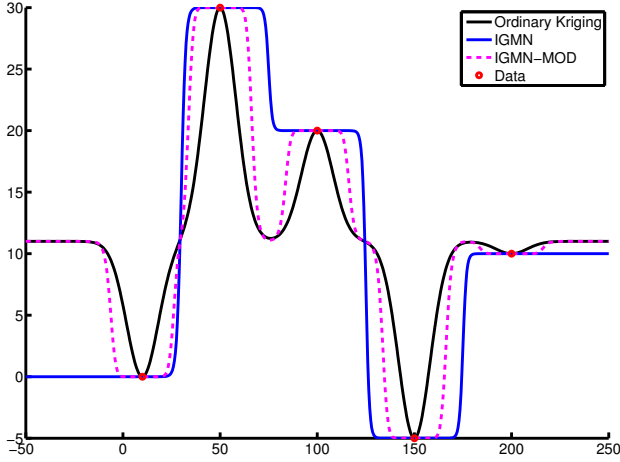


Figure 11: Ordinary Kriging, IGMN and Modified-IGMN interpolating a simple 1-dimensional data set.

appearance observed in figure 11.

4.1.2 Proposed Solution

Inspired by how Kriging handles its trend, we created the concept of *moving Gaussian trend component*. This works as follows: firstly, a new Gaussian component with mean $\mu_{trend} = [\mu_{trend,i}, \mu_{trend,t}]$ is created. The user provides a constant value to the $\mu_{trend,t}$ part of the mean (when not given by the user, we set to the arithmetic mean of existing Gaussian components. Similar to Ordinary Kriging). The other part of the mean, $\mu_{trend,i}$, represents the location X, Y of this Gaussian trend component. Then, we insert this newly created Gaussian component in a distance inversely proportional to the Mahalanobis distance from the closest Gaussian to the point being estimated (Equation 4.1). So, if the closest Gaussian is too far, the Gaussian trend will be placed near the point being estimated and, consequently, the trend will have a higher influence in the estimate. Similarly, if the closest Gaussian is near the location being estimated, the Gaussian trend will have a small impact.

$$\mu_{trend,i} = x_i + (s_i - x_i) * \frac{R}{M_d(s_i, x_i)} \quad (4.1)$$

In equation 4.1, x_i is the point being estimated and s_i is the mean of the

closest Gaussian component. $M_d(s_i, x_i)$ is the Mahalobis distance between the location being estimated and the closest Gaussian. R is a constant that defines the minimum distance where the trend will start to have a higher influence in the estimate. In this experiment, we set $R = 6$, which means that whenever the closest Gaussian is distant from a Mahalanobis distance greater or equal to 6 from the point being estimated, the trend Gaussian will be placed closer and will have a higher influence on the estimation result.

Figure 12 illustrates how the concept of the *moving Gaussian trend component*. The yellow X , represents the location being estimated. The red circle represents the closest Gaussian component. The green circles represent three potential locations to place the center (X, Y) of the moving Gaussian trend. In the first example, μ_1 is the location X, Y where to place the moving Gaussian trend component. As we can see, it will be placed very close to X_i because the closest Gaussian component S_i is far away, which means $R < M_d(s_i, x_i)$, in this example $R/M_d(s_i, x_i) = 0.5$. On the other hand, when $R > M_d(s_i, x_i)$, like the example of μ_3 , where $R/M_d(s_i, x_i) = 2$, the trend component will be placed twice as far as the closest component s_i . In this case, the influence of the moving Gaussian component will be much smaller than when placed in the μ_1 position. The covariance matrix of the trend Gaussian component is the same of a newly created component.

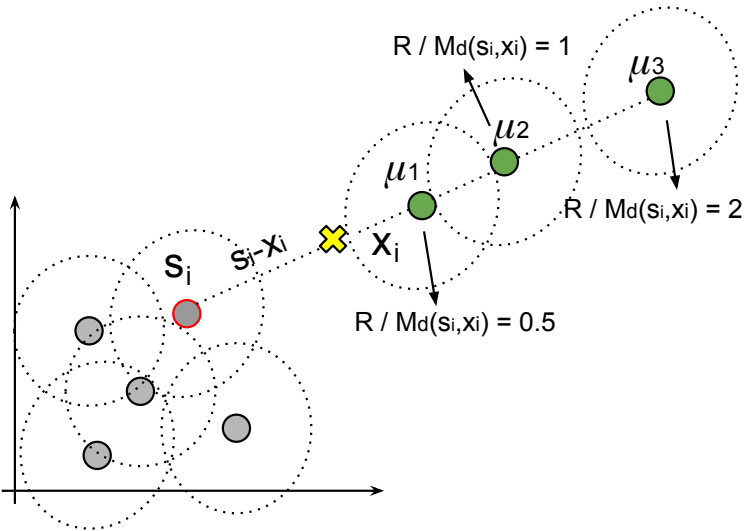


Figure 12: Example of placing the moving Gaussian trend component in three different locations

4.1.3 Problem Description: Diagonal Covariance Matrix for New Gaussian Components

As noted in Chapter III, IGMN initializes new Gaussian components with a diagonal covariance matrix. Which means that every new component will have a covariance matrix aligned to the axis of reference as shown in figure 13a. Potential anisotropies in other directions are learned incrementally, as soon as more data is provided for training. For example, if the dataset has a high continuity in the direction $\theta = 45^\circ$ from the north and a lower continuity in the perpendicular direction, IGMN will incrementally update the covariance matrix to make it look more like figure 13b during training.

The problem of relying solely on IGMN to learn anisotropies is that if the data set available is too sparse, IGMN will not have many opportunities to adjust the covariance matrix of each component incrementally. Consequently, it may not have the chance to learn it correctly. Data sets with clear anisotropies are not rare in environmental science studies. Figure 16 shows an example of a data set with direction of maximum correlation around $\theta = 45^\circ$ from the north.

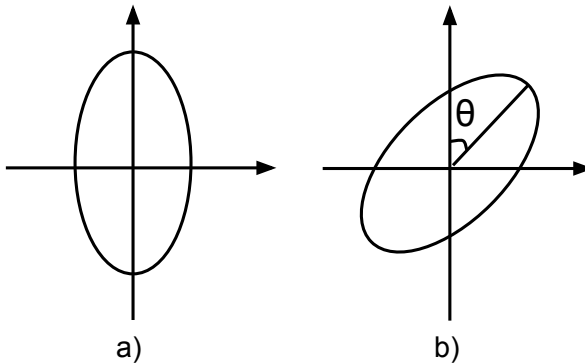


Figure 13: Top view illustration of a 2-d gaussian function with: a) covariance matrix aligned to the reference axis; b) covariance matrix rotated by θ .

4.1.4 Proposed Solution

To overcome this issue, we modified IGMN's training procedure to perform an initial rotation in the covariance matrix of new Gaussian components. The rotation is performed as shown in equation 4.2 before this new

Gaussian component is inserted into the mixture model. The angle of initial rotation is pre-defined by the user.

Therefore, when solving a problem with a clear direction of anisotropy, as in figure 16, the user can provide the initial angle of rotation which will be applied to the covariance matrix. This shortcuts IGMN's job of identifying the anisotropies and gives the opportunity for the user to insert expert knowledge about the problem.

$$C' = RCR^T \quad (4.2)$$

$$R = \begin{bmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{bmatrix} \quad (4.3)$$

4.2 SEQUENTIAL GAUSSIAN MIXTURE SIMULATION WITH IGMN

In Section 4.1 we proposed modifications to IGMN to improve its spatial interpolation performance. The proposed improvements allow the user to insert expert knowledge by configuring the angle of anisotropy and the trend value. Also, the proposed solutions make the estimation result of IGMN a little more similar to the estimation result of Kriging, as will be shown in Chapter 5. Therefore, one natural next step is to use IGMN in the place of Kriging for performing geostatistical simulation. In this section, we show how we can unite the Sequential Gaussian Simulation (SGS) method and IGMN to perform unconditional geostatistical simulation. Using IGMN instead of Kriging in SGS means we do not need to model a variogram.

IGMN is not able to directly perform a geostatistical simulation. But, it has three characteristics that make it particularly attractive for this task: 1. provides variance of its estimates; 2. on-line continuous learning and 3. learns and incorporates spatial correlations automatically, through the Gaussian Mixture Model. Based on that, we adapted the SGS algorithm by replacing the estimation through Kriging for the estimation through IGMN and implemented it inside IGMN. Also, we used the incremental learning characteristic of IGMN to incorporate recent simulated values into the model. So, instead of having to repeat the whole training procedure, as it would be necessary for a non-incremental model, such as MLP, we only add a new training tuple to the model. The detailed procedure is presented in algorithm 1. In this example, the simulated cell is taken from a normal distribution with mean and variance obtained from IGMN estimation. An alternative way is to take the simulated cell from the mixture distribution identified by IGMN, but we chose to maintain the simulation procedure as close as possible to the SGS.

Algorithm 1 Incremental Gaussian Mixture Sequential Simulation

```

function IGMN_SIMULATION
  for all realization  $r$  do
    {Initializes the grid to be simulated}
    build( $grid$ )
    while  $grid$  not empty do
       $cell$  = random cell from  $grid$ 
      remove  $cell$  from  $grid$ 
      {Calculates mean and variance (eq. 3.2 and 3.4)}
       $\hat{x}_t = \sum_{j=1}^K p(j|x_t)(\mu_{j,t} + C_{j,t}C_{j,i}^{-1}(x_t - \mu_{j,i}))$ 
       $\hat{C}_t = \sum_{j=1}^K \left\{ C_{j,t} - C_{j,t}C_{j,i}^{-1}C_{j,i}^T + \|\bar{\mu}_{j,t} - \hat{x}_t\| \right\}$ 
      {Takes value from a normal dist.}
       $simulated[cell, r] = gaussian(\hat{x}_t, \hat{C}_t)$ 
      {Incrementally incorporates new value}
       $IGMN\_train([cell, simulated[cell, r]])$ 
    end while
  return  $simulated$ 
end for
end function

```

One significant limitation of this algorithm is that it is not capable of performing exact conditional simulations. The reason is the same for spatial interpolation: IGMN does not store the training points and may use one single Gaussian component to represent multiple points. So, this method is better applied to perform simulations that do not require exact reproduction of conditioning points.

4.3 MULTIPLE-POINT SIMULATION WITH IGMN AND DS

4.3.1 Problem: Noise and Speed of the Direct Sampling Method

As presented in Chapter III, the Direct Sampling (DS) Multiple-Point Simulation method has been attracting a lot of attention in the last few years. Especially, because it can handle discrete or continuous variables, perform co-simulation, and it is relatively easy to parameterize. However, one common problem with DS is that its realizations end up having noise and outliers. Figure 14 shows an example. The right side of Figure 14 is a realization of DS, which was generated having as training image Figure 14a. As can be

seen, the realization contains a lot of noise and outliers. To solve this problem, Meerschman et al. (2013) suggests smoothing the result after each realization by performing a post-processing step. This step consists in re-simulating the whole grid with a known neighborhood.

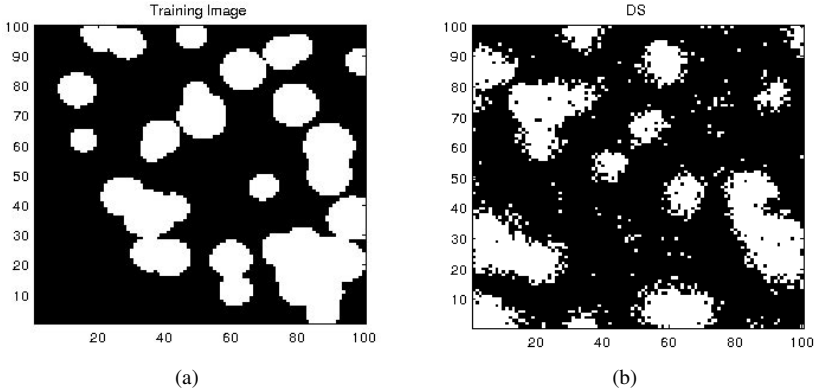


Figure 14: a) Training Image (TI); b) A realization of Direct Sampling based on the provided TI.

Although post-processing each realization might help to remove the noise, it slows down, even more, the simulation. In chapter III, we saw that, for each simulated point (pixel), DS scans the training image looking for a pattern match. The speed of this scanning process depends on two parameters: the acceptance threshold and the maximum fraction of the TI that DS is allowed to scan. So, setting the threshold too low and the fraction to scan to 1, would increase the chance that DS scan the whole image for every simulated point, which is very slow. On the other hand, setting the threshold too high and the fraction of TI to a number close to zero would make DS runs faster but produce poor simulations.

One of the main issues in the DS method is that it copies only a single point from the training image to the simulation grid at each step. This issue influences both speed and quality of the realizations. Instead, if DS copied multiple points, the simulation could be much faster. From this insight, we decided to use IGMN to help DS decide when and how many points to copy at once.

4.3.2 Proposed Solution

First, we pre-process the training image with IGMN to identify clusters. These clusters are represented by multivariate Gaussians on IGMN which are discovered incrementally during the IGMN training process. There is no need to determine the number of clusters beforehand. As a result, we obtain clusters with different sizes and we can find out the cluster of each TI point. We can also determine the spatial dimensions of each cluster by looking at the covariance matrices of the Gaussian components adjusted by IGMN. Figure 15b shows the IGMN Gaussian components after the training process with a TI (Figure 15a) as input. Note that IGMN has Gaussians with different sizes, depending on the spatial continuity of the region.

Second, we run a modified version of DS that, instead of copying only a single point at each step, copies multiple points at once. The criterion to copy multiple points is that they must belong to the same cluster and be located inside a radius R . R is proportional to the cluster spatial dimensions. Then, when DS finds a match, if the current simulated point is inside a big cluster, more of its neighbors will be copied together. On the other hand, if that point matched is inside a small cluster, just a few (or none) of its neighbors will be copied together. The concept behind this idea is that clusters represent continuity regions. So, the simulation can copy and paste more points in a region with high continuity than in one with low continuity.

The gain of speed comes with the possibility of copying multiple points and avoiding re-scanning the TI many times. Also, the noise present in DS is significantly reduced because almost no point is copied alone. So, it is less likely to have single outliers being copied over continuous regions.

Our solution inserts an overhead of the pre-processing step of clustering the TI using IGMN. But, this cost is accounted only once because it is not executed for every realization. Instead, it is run only once for each TI. So, in a typical simulation study, in which multiple realizations are required, this becomes a constant cost and tends to become less important as the number of realizations grows. In the following subsection, we describe the steps to perform this simulation.

4.3.2.1 The Multiple-Point Simulation Algorithm with IGMN and DS

Similarly to the DS algorithm presented in Chapter III, this algorithm aims at simulating a random function $Z(x)$. Some of the steps are similar to what was presented in chapter 3. As input, it requires a simulation grid (SG), whose nodes are denoted x , a training image (TI), whose nodes are denoted

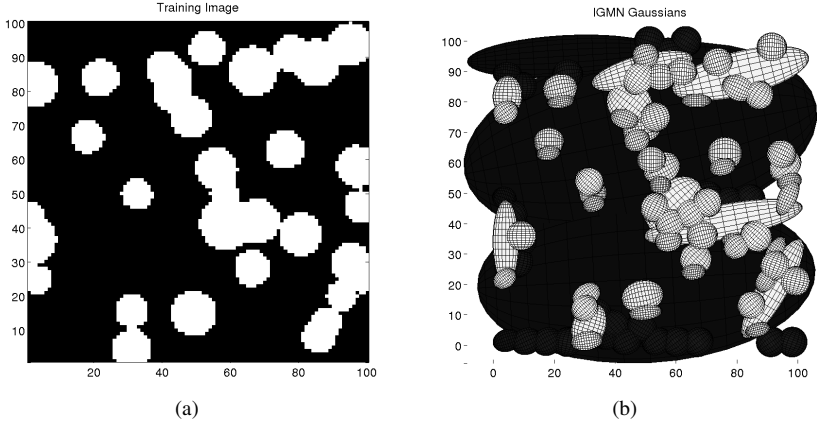


Figure 15: a) Training Image (TI); b) IGMN multi-variate gaussian components after training

y , a radius range R_{min}, R_{max} , and, optionally, a set of N conditioning points $z(x_i), i \in [1, \dots, N]$. The radius range defines a circular region around a node where neighbors can be copied together. Then, the algorithm goes through the following steps:

1. Train an IGMN model with the points of the Training Image as input. Each training point consists of its 2d location y and its value $Z(y)$.
2. Use the trained IGMN model to classify every point into clusters and build a cluster grid $C(y)$. This cluster grid is an image with the same dimensions of the TI. But, instead of having $Z(y)$ at every location, it has $C(y)$, which is a number that identifies whose cluster each location y belongs to.
3. From the IGMN model, calculate the spatial dimension $dim(c)$ of the cluster c . In this work, we used $dim(c)$ as the maximum value of variance between $var(x, x)$ or $var(y, y)$. These values are directly found in the main diagonal of the covariance matrix of the corresponding Gaussian component that represents cluster c on IGMN.
4. If there are conditioning points, assign them to the closest simulation grid cells.
5. Define a path through the remaining unsimulated points in the SG. The path specifies an order to visit each location. It may be Random (STREBELLE, 2002), unilateral or any other kind of path.

6. For each location x in the path:

- (a) Find the neighbors of x (n closest grid nodes $\{x_1, x_2, \dots, x_n\}$ already assigned in SG).
- (b) If no neighbor exists:
 - i. Randomly take a node y from TI, identify its cluster $C(y)$ and the spatial dimension of the cluster $dim(C(y))$.
 - ii. Calculate the radius R proportionally to the cluster $C(y)$ dimension:

$$R = R_{min} + (R_{max} - R_{min}) * \frac{dim(C(y))}{max(dim)} \quad (4.4)$$

- iii. Get all neighbors $N(y, R, C(y))$ that are in a circle of radius R centered in y and in the same cluster $C(y)$
 - iv. Calculate their relative location with respect to y (the center of the circle) and assign their $Z(y)$ values to the SG in the respective $Z(x)$ locations, relative to x .
- (c) Else, if neighbors were found, compute the lag vector $L = \{h_1, \dots, h_n\} = \{x_1 - x, \dots, x_n - x\}$ which defines the neighborhood of x , $N(x, L) = \{x + h_1, \dots, x + h_n\}$. Figure 10a shows an example where x has three neighbors and the lag vectors are: $L = \{(1, 2), (2, 1), (-1, 1)\}$. It represents the relative locations of the already simulated points.
 - (d) Compute the data event $d_n(x, L) = \{Z(x + h_1), \dots, Z(x + h_n)\}$, which is a vector with the values of the variable of interest at the neighborhood. In the example of Figure 10a, $d_n(x, L) = \{0, 0, 1\}$.
 - (e) Calculate the dimension of the search window in the TI. It is defined by the maximum and minimum values at each dimension of the lag vectors.
 - (f) Randomly choose a location y in the search window and from this location, start scanning the whole window. For each location y :
 - i. Calculate the data event $d_n(y, L)$ in the TI. In Figure 10c, a random grid node was selected and its data event is $d_n(y, L) = \{1, 0, 1\}$.
 - ii. Compute the distance $d\{d_n(x, L), d_n(y, L)\}$ between data events found in the SG and the TI. In our example, we perform a simple mean of the categorical values as show in Equation

4.5.

$$d\{d_n(x), d_n(y)\} = \frac{1}{n} \sum_{i=1}^n a_i \in [0, 1], \quad (4.5)$$

$$\text{where } a_i = \begin{cases} 0 & \text{if } Z(x_i) = Z(y_i) \\ 1 & \text{if } Z(x_i) \neq Z(y_i) \end{cases}$$

- iii. Store $y, Z(y)$ and $d\{d_n(x, L), d_n(y, L)\}$ if it is the lowest distance calculated so far for the current point being simulated.
- iv. If $d\{d_n(x, L), d_n(y, L)\}$ is smaller than the acceptance threshold t , then, repeats steps 5(b)i-iv to assign the accepted value and neighbors to the SG.
- v. Else, if the number of iterations exceeds a certain fraction of the TI, the lowest distance of node y obtained so far is accepted and the algorithm repeats steps 5(b)i-iv to assign the accepted value and neighbors to the SG.

5 EXPERIMENTS AND RESULTS

To evaluate the proposals presented in Chapter 4 we performed three experiments. The first one aims at evaluating how the solutions proposed in Section 4.1 can improve the performance of IGMN in spatial interpolation tasks. The second experiment aims at evaluating if the algorithm proposed in Section 4.2 is, visually, able to generate different equally probable realizations (scenarios) that preserve important characteristics of spatial correlation and anisotropy present in the dataset. Also, this second experiment aims at evaluating if the multiple realizations, when averaged, honor (look like) the mean values obtained through Ordinary Kriging and IGMN. The third experiment intent to evaluate our MPS simulation algorithm that mixes IGMN and DS proposed in Section 4.3.2.1. The first two experiments use the Meuse River data set, which comprises 155 samples of four heavy metals measured in the top soil of a flood plain along the river Meuse (Pebesma, Edzer, 2015). The experiments described in this section use the variable Zinc, along with its coordinates X and Y . Figure 16 shows a plot of the Zinc values.

5.1 SPATIAL INTERPOLATION COMPARISON EXPERIMENT

5.1.1 Methodology

In this experiment, we compared our modified version of IGMN (IGMN-MOD) with Ordinary Kriging (Krig), IDW, MLP networks and the original IGMN in a spatial interpolation problem. For doing so, we randomly split the Meuse River dataset into two parts: test set and training set. For the test set, we selected 55 samples, which we did not use for training. The remaining 100 samples, were arranged into four different training set sizes: 25, 50, 75 and 100 samples. The goal of using different training set sizes is to evaluate the robustness of the spatial interpolation methods when the data set becomes gradually more sparse.

We trained all methods with each of these training set sizes. We then made each method estimate the Zinc values in the 55 locations (X, Y) of the test set. As a metric of comparison, we used two criteria suggested in (ISA-AKS, E. H.; SRIVASTAVA, R. M., 1989), the RMSE (root-mean-square error, equation 5.1) and CC (correlation coefficient, equation 5.2). Where σ_o and σ_e are the standard deviation of observed and estimated data. We calculated these statistics between the 55 real values of the test set and the estimated

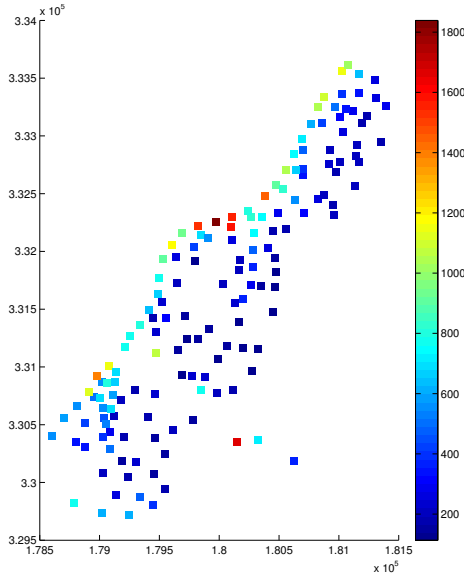


Figure 16: Meuse River Data Set - Concentration of Zinc (ppm).

values from each interpolation method.

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^N [\hat{z}(x_i) - z(x_i)]^2} \quad (5.1)$$

$$CC = \frac{\sum_{i=1}^N [z(x_i) - z_m][\hat{z}(x_i) - \hat{z}_m]}{\sigma_e \sigma_o N} \quad (5.2)$$

The variogram model used to perform ordinary kriging is presented in equation 5.3. *Nug* and *Sph* represent the Nugget effect and the Spherical variogram using the popular GSTAT notation (PEBESMA; WESSELING, 1998; PEBESMA, 2004). The variogram is anisotropic and has its direction of maximum continuity at 45° from the north.

$$Variogram = 0.05Nug(0) + 0.59Sph(1200, 45, .4) \quad (5.3)$$

The variogram modeling process for this dataset is available in (Pe- besma, Edzer, 2015). In Figure 17 we show the plot of the directional variograms in four different directions: 0° , 45° , 90° , and 135° (0° is North and 90° is East). The directional tolerance is $\pm 22.5^\circ$, so the variogram in the 0 direction, for example, includes points from -22.5° to $+22.5^\circ$. The nugget, sill

and model type (spherical) were derived from an omnidirectional variogram plot. As we can see in Figure 17, the 45° direction has the largest range, thus, largest continuity. The anisotropy ratio (0.4) was estimated by dividing the range of the 135° direction by the range of the 45° one.

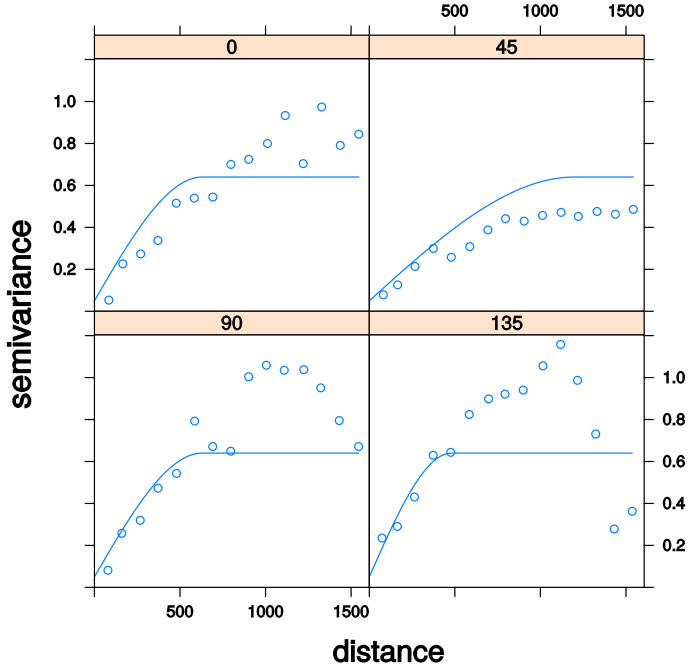


Figure 17: Directional variograms (0° , 45° , 90° and 135°) of the Meuse River dataset. Source: (Pebesma, Edzer, 2015).

The architecture of the MLP network was 2-H-1, with the locations X, Y as inputs and the Zinc values as output. For each training set size (25, 50, 75 and 100), we trained the network 80 times, where the number of neurons H in the hidden layer varied from 2 to 10 and the training was repeated ten times for each architecture, to avoid potential problems with random weight initialization. Only the best result for each training set was selected for comparison.

The IGMN parameters were set to $\delta = 0.06$, $\tau = 0.05$, $range = \max([x, y, zinc]) - \min([x, y, zinc])$. The initial covariance matrix for each new Gaussian component is initially defined as $diag(\delta * range)$ and is later rotated according to θ . The IGMN modified parameters were set to $\theta = 45^\circ$ (direction of anisotropy) and $\mu_{trend} = [\bar{x}, \bar{y}, \bar{zinc}]$ (arithmetic mean of the da-

taset). τ is the predefined level of significance used by IGMN in a χ^2 test to decide when to create a new gaussian component. More details about other parameters of IGMN can be found in (HEINEN, 2011). Finally, for the IDW method we set the power parameter to 2.

5.1.2 Results

Figure 18 shows RMSE and CC mean results of the experiments for 30 runs of each of the four training set sizes (25, 50, 75 and 100). We rounded the values of RMSE and truncated the values of CC to 2 decimal places. We calculate RMSE and CC between the 55 real samples of in the test set and the estimated values calculated from each of the four methods. As we can see, the modified IGMN outperformed original IGMN, MLP, and IDW, in both RMSE and CC. It does not outperform Kriging with but follows it closely.

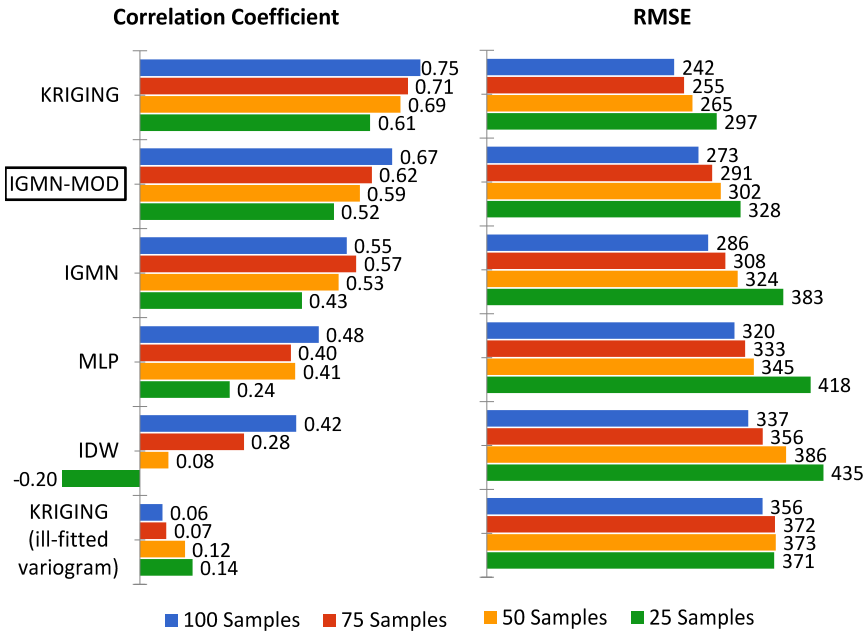


Figure 18: Correlation Coefficient (Left) and RMSE (Right) between 55 real samples and interpolated values by Kriging, IDW, MLP, IGMN and IGMN-MOD. The known samples were divided into training set sizes of 25, 50, 75 and 100 samples.

The RMSE results pointed out another important aspect present in Kriging and modified IGMN: their robustness to sparse datasets. To further support this evidence, we can look again at the values of RMSE in figure 18. The best result of Kriging, with 100 samples training set, is 242. The worst, with only 25 samples in the training set, is 297, which gives a difference of 55 between the best and worst case. Similarly, modified IGMN has its best case RMSE of 273 and the worst one equals to 328, also 55 of difference between best and worst cases. On the other hand, original IGMN has the best RMSE of 286 and worst of 383, a difference of almost 100, where most of the worsening in performance happens with the training set sparse, with only 25 samples. A similar poor performance with the 25-sample training set is observed in MLP and IDW as well. We credit the improvement in robustness with sparse data sets found in our method mainly to the inclusion of the moving Gaussian trend component, proposed in Section 4.1.

Moreover, it is important to note that IGMN has only a few configuration parameters that do not require any expertise in geostatistics. On the contrary, the performance of Kriging directly depends on the quality of the modeled variogram. To support this observation we modified the range of the variogram presented in equation 5.3 from 1200 to 120 and repeated the experiment for Kriging. The RMSE and CC results are shown in the lower part of figure 18. As we can see, the performance of Kriging became the worst when using a poorly modeled variogram.

5.2 SEQUENTIAL GAUSSIAN MIXTURE SIMULATION EXPERIMENT

5.2.1 Methodology

The goal of this experiment was to observe two aspects, regarding the sequential simulation algorithm proposed in Section 4.2: 1. If IGMN simulation was capable of generating equally probable different scenarios (realizations) that maintain the main characteristics of spatial correlation found in the dataset; 2. If the mean of multiple realizations of IGMN sequential simulation tends to approximate the interpolation using Kriging and the modified IGMN as presented in figure 19.

For this experiment, we chose the same dataset of the previous experiment, the Meuse River data set (figure 16). We trained IGMN using all 155 samples available. The simulation grid size was $[77 \times 55] = 4235$ samples, with a cell size of $[50 \times 50]$, starting at $X_0 = 178605, Y_0 = 329714$. The grid covers the whole rectangular area that englobes the Meuse River data set presented in figure 16.

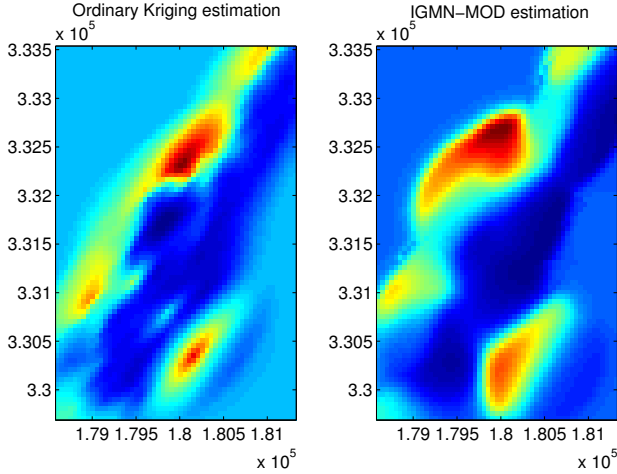


Figure 19: Left: Ordinary Kriging interpolation over the simulation grid. Right: IGMN interpolation over the simulation grid.

Figure 20 shows nine realizations performed with IGMN sequential simulation algorithm. In these realizations, we can observe that the characteristic of high spatial correlation in the direction of approximately 45° from the north is maintained. Other important result presented in figure 20 is that the realizations generated are significantly different from each other. For example, the realization shown in the first column of the second row is very different from the one beside it.

In addition, figure 21 presents the mean values of 30 realizations in each point of the grid. As shown, this image looks like the estimation of Kriging and IGMN presented in figure 19, which means that the simulation process honors the mean values. These results suggest that even though IGMN was able to generate very heterogeneous realizations, it still honors the mean values and spatial correlation present in the original dataset.

5.3 MULTIPLE-POINT SIMULATION EXPERIMENT

5.3.1 Methodology

This experiment aims at comparing the performance of our proposal (DS+IGMN) and the Direct Sampling (DS) method, regarding speed and qua-

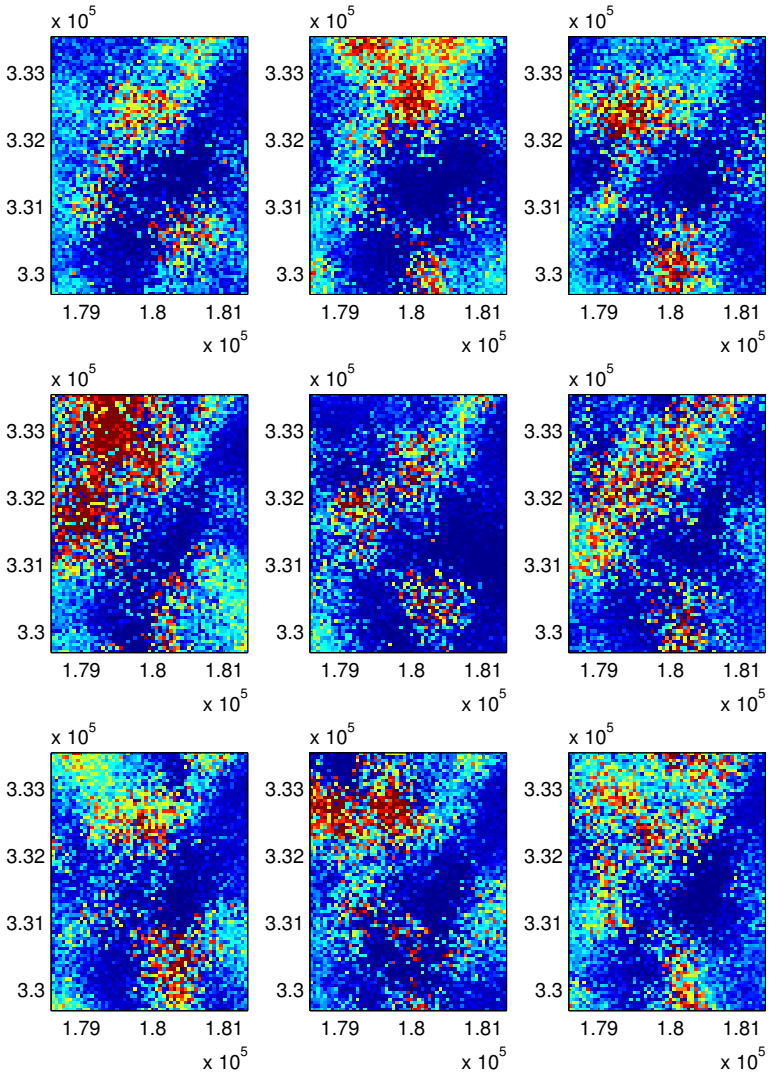


Figure 20: Nine realizations of the IGMN sequential simulation algorithm.

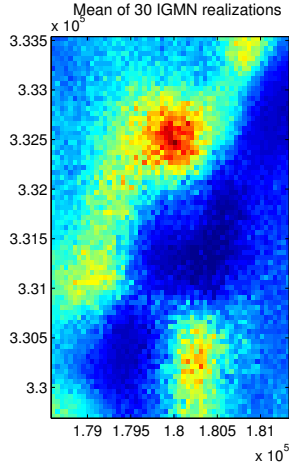


Figure 21: Mean of 30 realizations performed with IGMN over the simulation grid.

lity of realizations. As described in Chapter 4, we proposed to use the modified IGMN together with DS to copy multiple points from the training image to the simulation grid at each step. The two principal goals of our method are to speed up the simulation and to reduce the noise present in DS. To use as Training Image (TI) in this experiment, we generate a black square image composed of randomly placed white circles with a radius between 4% and 8% of the side of the image. We chose this procedure because this is the classical example that is distributed with the DS Matlab implementation.

For comparing the noise produced by the two methods, we used the TI shown in Figure 22a, with size 100×100 pixels. We performed 12 realizations, six with DS and six with DS+IGMN. Then, we visually compare the amount of noise of them. The size of the simulated grid is also 100×100 and the configured parameters for both methods were: *acceptance_threshold* = 0.1 and 0, *fract_of_ti_to_scan* = 0.5, *fixed_template_window* = [20,20]. The specific parameters configured for IGMN were: *radius_range*=[2,5], *range*=*ti_size*/4, *delta*=0.1, *tau*=0.1, *vmin*=0, *spmin*=0, *covtype*=full.

To evaluate speed, we ran an experiment with different simulation grids, ranging from 50×50 to 150×150 . We recorded the total time and the mean time of 5 realizations for each grid size. The TI generated for this experiment was of size 150×150 and it is shown in Figure 22b. We ran all the experiments on top of a Matlab environment, in an Intel Core i7-4500U machine with 8gb of RAM running Ubuntu 16.04 Linux. The authors of DS (MARIETHOZ; RENARD; STRAUBHAAR, 2010) provided the source

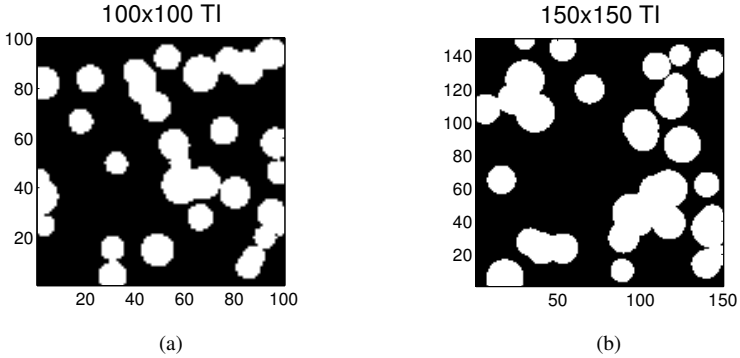


Figure 22: Training images used to evaluate DS+IGMN: a) 100x100 TI used to evaluate noise; b) 150x150 TI used to evaluate speed

code for running the simulation and generating TIs at their website ¹. Also, IGMN authors provided us their source code upon our request.

5.3.2 Results

Figure 23 shows 12 realizations generated with DS and DS+IGMN. In Figure 23a, we can see three realizations produced by DS+IGMN with parameter *acceptance_threshold*= 0.1. When compared to the three realizations of DS Figure 23b, DS+IGMN generate realizations that reproduces the main characteristics of the training image (Figure 22a) as well as DS does. On the other hand, DS+IGMN realizations have less noise than DS ones. We credit this noise reduction to our strategy of copying multiple points at once, which reduces the chance of leaving unsimulated isolated locations.

In Figures 23c and 23d we show the result of the same experiment but setting *acceptance_threshold*= 0. The goal was to observe the quality of the realizations in an extreme case of very low tolerance. Again, DS produce more noise than DS+IGMN.

In addition to the noise reduction, the mean time spent to generate each single realization presented in Figure 23 was of around 8 seconds for DS+IGMN with *acceptance_threshold*= 0.1 and 245s for DS with *acceptance_threshold*= 0.1. When setting *acceptance_threshold*= 0, DS+IGMN was able to generate each realization in 28s, on average, and DS took 728s.

¹www.minds.ch/gm/DS.htm

So, DS+IGMN with *acceptance_threshold*= 0 is still much faster than DS with *acceptance_threshold*= 0.1, which allows us to set parameters very tight and obtain better quality simulations. In this example, with the same set of parameters, DS+IGMN was more than 25 times faster than DS, besides generating higher quality realizations. It is important to notice that we are using the didactic implementation of DS and more optimized implementations may exist.

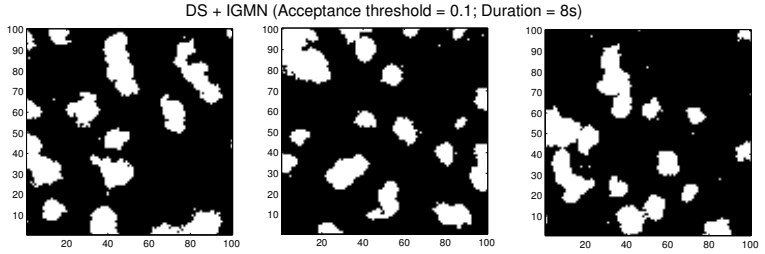
We also measured the duration to produce realizations with different grid sizes. Table 1 shows the mean time of 5 realizations for grid sizes varying from 50x50 to 150x150, using the 150x150 training image shown in Figure 22b. DS+IGMN is again faster than DS in all grid sizes. This improvement is possible because the most costly operation involved in DS is the search for a pattern match in the TI. So, simulating multiple points at once, as we propose, has the potential of speedup in a rate inversely proportional the number of points that are copied (simulated) together at each simulation step.

The time spent training and clustering the training images using IGMN was of 130s for the 100x100 TI and 292s for the 150x150 TI. This procedure is executed only once when the image is created. So it is a constant cost and gets less important as the number of realizations grows. For example, the total time to generate five realizations in the 150x150 grid was around 3 minutes with DS+IGMN and 64 minutes for DS. So, even adding the training time of IGMN to the DS+IGMN total time, does not make it even close to the 64 minutes DS took to run. This huge improvement in speed, together with the improvement in the simulation quality by reducing noise, gives a significant contribution to the state of art of Multiple-Point Simulation.

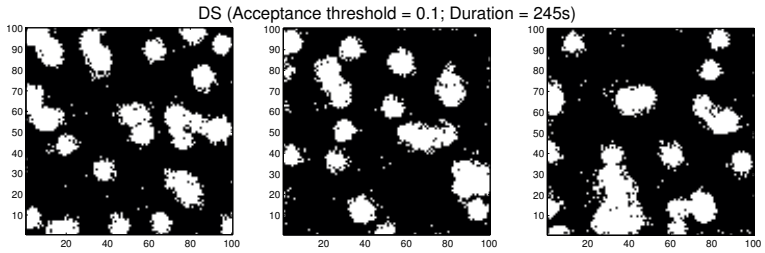
Table 1: Mean duration of 5 realizations in various grid sizes using DS and DS+IGMN with a 150x150 TI. Parameters: *acceptance_threshold*= 0.1 and *fract_of_ti_to_scan* = 0.5

Grid size	DS		DS+IGMN	
	Mean	σ	Mean	σ
50x50	53.2	24.5	3.1	2.1
75x75	118.7	75.6	7.8	4.9
100x100	378.6	176.7	20.1	11.8
125x125	450.2	225.9	30.6	10.7
150x150	770.7	134.8	44.1	15.8

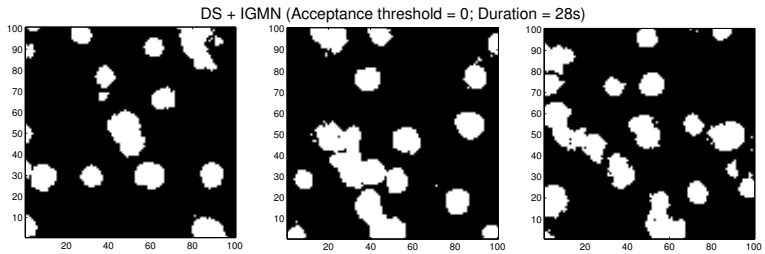
It is important to notice that setting the *radius_range* parameter too high, will make DS+IGMN copy more points at once and be faster. However, the algorithm may start to insert some undesirable artifacts because it will not be able to match the intersection between the big patches of points being



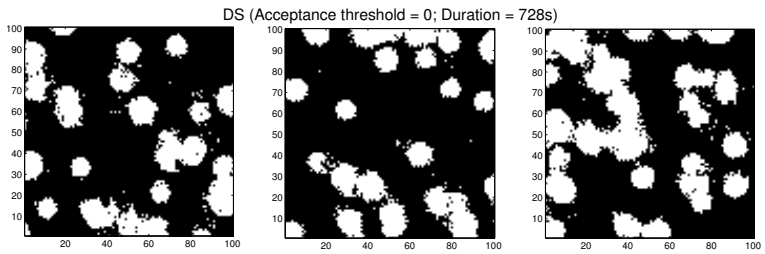
(a)



(b)



(c)



(d)

Figure 23: a) DS+IGMN Realizations with *acceptance_threshold*= 0.1; b) DS Realizations with *acceptance_threshold*= 0.1; c) DS+IGMN Realizations with *acceptance_threshold*= 0; d) DS Realizations with *acceptance_threshold*= 0.

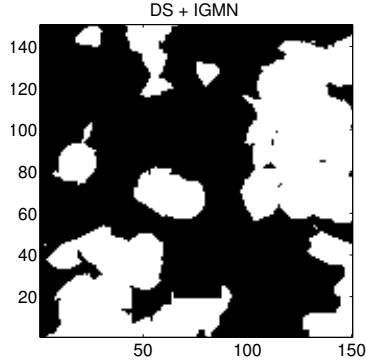


Figure 24: Example of a bad DS+IGMN realization with *radius_range* set too high [20,25]

copied. For example, the algorithm may copy a patch of points from the border of the TI to the middle of the simulation grid. Figure 24 shows an example of setting the *radius_range* parameter too high [20,25] in a 100x100 grid.

Currently, we still have not defined a general rule for setting the *radius_range* parameter because it depends on the continuity and size of the training image. This makes it harder to obtain a formula that generalizes. But, the basic idea is to set this parameter to the maximum possible value, as long as it does not affect the quality of the simulation. One way the user can successfully choose a good value for this parameter is by cropping a small portion of the TI, running simulations with different *radius_range* values and visually observing the quality of the produced realizations.

6 CONCLUSION AND FUTURE WORK

In this work, we studied how the Artificial Neural Network model IGMN, based on incremental Gaussian Mixture Models, could be adapted and improved to solve spatial interpolation, sequential simulation, and multiple-point simulation problems. We saw that geostatistical methods, such as Kriging and SGS, require modeling a Variogram, which is not a simple task for non-experts in Geostatistics. Therefore, one of the motivations of this study was to investigate the possibility of using the IGMN model to solve geostatistical problems, without the need to model a Variogram.

For spatial interpolation problems, we improved the performance of IGMN by adding the possibility to specify a trend component and the direction of maximum continuity. The concept of the moving Gaussian trend component makes the IGMN estimation result tend to a user specified value when the location being estimated is far from known points. Also, the user may set an angle of rotation to the initial covariance matrices, which help IGMN to identify the general anisotropy pattern of the training set.

As an alternative to the Sequential Gaussian Simulation (SGS) method, we proposed and implemented inside IGMN a modified version of SGS. Our proposed solution uses IGMN, instead of Kriging, in the interpolation and variance calculation step. Thus, it enables IGMN to perform an unconditional sequential simulation. This approach has the same limitation stated for spatial interpolation problems: it cannot reproduce exact conditioning points because IGMN may represent many training points with a single Gaussian component. Therefore, it is best applied to unconditional simulation.

Finally, we proposed a Multiple-Point Simulation algorithm that unites the IGMN model and the Direct Sampling (DS) method. Our algorithm has a pre-processing step that clusters the Training Image using IGMN. Then, it simulates multiple points at each simulation step by copying patches of similar points identified by IGMN. This algorithm maintains the same capabilities of DS, i.e., it can perform either conditional or unconditional simulation with continuous or discrete variables and can also perform cossimulation.

To validate the proposed spatial interpolation improvements on IGMN and our sequential simulation algorithm, we performed several experiments using the Meuse River dataset. The results obtained from these experiments demonstrated that: (i) Our modified version of IGMN has better performance (regarding RMSE and CC) than the original IGMN, MLP, and IDW, especially in sparse and anisotropic datasets; (ii) Although not beating the performance of Ordinary Kriging, our modifications in the IGMN estimation process make its performance much closer to Ordinary Kriging than before;

(iii) The proposed sequential simulation algorithm, which uses IGMN instead of Kriging, is capable of generating heterogeneous realizations that preserve important characteristics of the original data set, such as the direction of maximum continuity. Also, the mean values of various realizations honor the mean of the IGMN interpolation over the grid.

To evaluate our Multiple-Point Simulation algorithm that unites Direct Sampling (DS) and IGMN, we compared it with the DS method alone. Our results demonstrated that: (i) DS+IGMN produces better quality simulations with much less noise than the DS method; (ii) regarding speed, our approach is orders of magnitude faster than the DS method.

One limitation of IGMN for spatial interpolation and Geostatistical simulation tasks is that it does not store the training points and may use one single Gaussian component to represent multiple points. So, our sequential simulation approach is not able to perform simulations that require exact reproduction of conditioning points.

In future work, we will investigate strategies to automate the definition of some required parameters. For example, the *radius_range* parameter in the Multiple-Point Simulation algorithm. We think it is possible to achieve this by using two information: the size of the Training Image (TI) and the size of the clusters identified by IGMN, which indirectly describes the continuous regions contained in the TI.

We believe the contributions presented in this dissertation add value to existing works and represent an important step to make neural networks an effective and popular tool for solving geostatistics problems. Our Multiple-Point Simulation method may enable researchers to study more complex problems that were not viable before because of time complexity. We consider this one of the main contributions of this work. Also, our improvements in the spatial interpolation process of IGMN made its performance close to Kriging without the need of providing a variogram model. So, the modifications to IGMN proposed in this work might turn IGMN into a handy tool for non-experts in geostatistics.

Based on the contributions presented above, we believe that the main objective of this work was achieved, i.e., adapting the IGMN to solve spatial interpolation and geostatistical simulation problems. Besides studying how to apply IGMN to address these kinds of problems, we also had a chance to perform significant improvements in existing methods, such as the Direct Sampling, by using IGMN to speed it up and enhance the quality of its realizations.

REFERENCES

- AITOKHUEHI, I.; DURLOFSKY, L. J. Optimizing the performance of smart wells in complex reservoirs using continuously updated geological models. **Journal of Petroleum Science and Engineering**, v. 48, n. 3, p. 254 – 264, 2005. ISSN 0920-4105. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0920410505000963>>.
- BOHLING, G. Introduction to geostatistics and variogram analysis. **Kansas geological survey**, 20p, 2005.
- CAERS, J.; ZHANG, T. Multiple-point geostatistics: a quantitative vehicle for integrating geologic analogs into multiple reservoir models. AAPG Special Volumes, 2004.
- CARPENTER, G. A.; GROSSBERG, S. A massively parallel architecture for a self-organizing neural pattern recognition machine. **Computer vision, graphics, and image processing**, Elsevier, v. 37, n. 1, p. 54–115, 1987.
- CAUDILL, M. Neural nets primer, part vi. **AI Expert**, Miller Freeman, Inc., v. 4, n. 2, p. 61–67, 1989.
- CHILES, J.-P.; DELFINER, P. **Geostatistics: modeling spatial uncertainty**. [S.l.]: John Wiley & Sons, 2009.
- CRESSIE, N. Statistics for spatial data: Wiley series in probability and statistics. **Wiley-Interscience, New York**, v. 15, p. 105–209, 1993.
- Deligiorgi, D.; Philippopoulos, K. **Spatial interpolation methodologies in urban air pollution modeling: application for the greater area of metropolitan Athens, Greece**. [S.l.]: INTECH Open Access Publisher, 2011.
- DEMPSTER, A. P.; LAIRD, N. M.; RUBIN, D. B. Maximum likelihood from incomplete data via the em algorithm. **Journal of the royal statistical society. Series B (methodological)**, JSTOR, p. 1–38, 1977.
- Doyen, P. **Seismic reservoir characterization: An earth modelling perspective**. [S.l.]: EAGE, 2007.
- ENGEL, P. M. Inbc: An incremental algorithm for dataflow segmentation based on a probabilistic approach. **INBC: an incremental algorithm for dataflow segmentation based on a probabilistic approach**, Universidade Federal do Rio Grande do Sul Porto Alegre, 2009.

ENGEL, P. M.; HEINEN, M. R. Incremental learning of multivariate gaussian mixture models. In: **Advances in Artificial Intelligence–SBIA 2010**. [S.l.]: Springer, 2010. p. 82–91.

FAZIO, V. S.

Interpolação espacial: uma comparação analítica entre redes RBF e Krigagem — Universidade Federal de Santa Catarina, Florianópolis, Brazil, 2013. Disponível em:

<<https://repositorio.ufsc.br/handle/123456789/107339>>.

FAZIO, V. S.; ROISENBERG, M. Spatial interpolation: An analytical comparison between kriging and rbf networks. In: **Proceedings of the 28th Annual ACM Symposium on Applied Computing**. New York, NY, USA: ACM, 2013. (SAC '13), p. 2–7. ISBN 978-1-4503-1656-9. Disponível em: <<http://doi.acm.org/10.1145/2480362.2480364>>.

Gilardi, N. ; Bengio, S. ; Kanevski, M. Conditional gaussian mixture models for environmental risk mapping. p. 777–786, 2002.

GRANA, D.; MUKERJI, T. et al. Sequential bayesian gaussian mixture linear inversion of seismic data for elastic and reservoir properties estimation. In: SOCIETY OF EXPLORATION GEOPHYSICISTS. **2012 SEG Annual Meeting**. [S.l.], 2012.

GUARDIANO, F. B.; SRIVASTAVA, R. M. Multivariate geostatistics: Beyond bivariate moments. In: _____. **Geostatistics Tróia '92: Volume 1**. Dordrecht: Springer Netherlands, 1993. p. 133–144. ISBN 978-94-011-1739-5. Disponível em: <http://dx.doi.org/10.1007/978-94-011-1739-5_12>.

Gumus, K.; Sen, A. Comparison of spatial interpolation methods and multi-layer neural networks for different point distributions on a digital elevation model. **Geodetski vestnik**, v. 57, n. 3, p. 523–543, 2013.

HEINEN, M.; ENGEL, P.; PINTO, R. Using a gaussian mixture neural network for incremental learning and robotics. In: **Neural Networks (IJCNN), The 2012 International Joint Conference on**. [S.l.: s.n.], 2012. p. 1–8. ISSN 2161-4393.

HEINEN, M. R. **A connectionist approach for incremental function approximation and on-line tasks**. Tese (Doutorado) — Universidade Federal do Rio Grande do Sul, Porto Alegre, Brazil, march 2011. Disponível em: <<http://hdl.handle.net/10183/29015>>.

HEINEN, M. R.; ENGEL, P. M. An incremental connectionist approach for concept formation, reinforcement learning and robotics. **Journal of Applied Computing Research (JACR)**, v. 1, n. 1, p. 2–19, 2011.

HOFFMAN, B. T.; CAERS, J. History matching by jointly perturbing local facies proportions and their spatial distribution: Application to a north sea reservoir. **Journal of Petroleum Science and Engineering**, v. 57, n. 3–4, p. 257 – 272, 2007. ISSN 0920-4105. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0920410506002786>>.

HORNIK, K.; STINCHCOMBE, M.; WHITE, H. Multilayer feedforward networks are universal approximators. **Neural networks**, Elsevier, v. 2, n. 5, p. 359–366, 1989.

ISAAKS, E. H.; SRIVASTAVA, R. M. **Applied Geostatistics**. [S.l.], 1989.

JOURNEL, A. Geostatistics: roadblocks and challenges, Soares, A., geostatistics-troia. **Dordrecht: Kluwer Academic**, v. 1, p. 475–489, 1992.

KOHONEN, T.; SCHROEDER, M. R.; HUANG, T. S. (Ed.). **Self-Organizing Maps**. 3rd. ed. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2001. ISBN 3540679219.

KRIGE, D. G. A Statistical Approach to Some Basic Mine Valuation Problems on the Witwatersrand. **Journal of the Chemical, Metallurgical and Mining Society of South Africa**, Operational Research Society, v. 52, n. 6, p. 119–139, dez. 1951. Disponível em: <<http://dx.doi.org/10.2307/3006914>>.

LINCHTENSTERN, A. **Kriging methods in spatial statistics**. Tese (Doutorado) — Bachelor's Thesis, Technische Universität München, Department of Mathematics, Germany, 2013.

MARIETHOZ, G.; RENARD, P.; STRAUBHAAR, J. The direct sampling method to perform multiple-point geostatistical simulations. **Water Resources Research**, v. 46, n. 11, p. n/a–n/a, 2010. ISSN 1944-7973. W11536. Disponível em: <<http://dx.doi.org/10.1029/2008WR007621>>.

MATHERON, G. **Traité de géostatistique appliquée. 1 (1962)**. [S.l.]: Editions Technip, 1962.

MATHERON, G. variables régionalisées et leur estimation [les]. Masson et CIE, éditeurs, 1965.

MEERSCHMAN, E. et al. A practical guide to performing multiple-point statistical simulations with the direct sampling algorithm. **Computers and Geosciences**, v. 52, p. 307 – 324, 2013. ISSN 0098-3004. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0098300412003299>>.

Nevtipilova, V., et al. Testing artificial neural network (ann) for spatial interpolation. **J Geol Geosci**, v. 3, n. 145, p. 2, 2014.

OLIVER, M.; WEBSTER, R. A tutorial guide to geostatistics: Computing and modelling variograms and kriging. **{CATENA}**, v. 113, p. 56 – 69, 2014. ISSN 0341-8162. Disponível em: <<http://www.sciencedirect.com/science/article/pii/S0341816213002385>>.

PEBESMA, E. J. Multivariable geostatistics in s: the gstat package. **Computers & Geosciences**, Elsevier, v. 30, n. 7, p. 683–691, 2004.

PEBESMA, E. J.; WESSELING, C. G. Gstat: a program for geostatistical modelling, prediction and simulation. **Computers & Geosciences**, Elsevier, v. 24, n. 1, p. 17–31, 1998.

Pebesma, Edzer. The meuse data set: a brief tutorial for the gstat r package. 2015. Disponível em: <<https://cran.r-project.org/web/packages/gstat/vignettes/gstat.pdf>>. Acesso em: 25 out. 2015.

PetroWiki. **Geostatistical conditional simulation**. [S.l.], c2015. Acesso em: 20 out. 2015.

PINTO, R. C.; ENGEL, P. M. A fast incremental gaussian mixture model. **PLoS ONE**, Public Library of Science, v. 10, n. 10, p. e0139931, 10 2015. Disponível em: <<http://dx.doi.org/10.1371/journal.pone.0139931>>.

PNNL. Kriging variogram. 2016. Disponível em: <http://vsp.pnnl.gov/help/Vsample/Kriging_Variogram.htm>. Acesso em: 9 nov. 2016.

REYNOLDS, D. Gaussian mixture models. **Encyclopedia of biometrics**, Springer, p. 827–832, 2015.

ROBBINS, S. M. H. A stochastic approximation method. **The Annals of Mathematical Statistics**, Institute of Mathematical Statistics, v. 22, n. 3, p. 400–407, 1951. ISSN 00034851. Disponível em: <<http://www.jstor.org/stable/2236626>>.

RUMENLHART, D.; HINTON, G. E.; WILLIAMS, R. J. Learning internal representation by error propagation, parallel distributed processing. **Explor. Microstruct. Cognition**, v. 1, p. 318–362, 1986.

SATHYANARAYANA, S. A gentle introduction to backpropagation. 2014. Disponível em:

<http://numericinsight.com/uploads/A_Gentle_Introduction_to_Backpropagation.pdf>

Acesso em: 10 nov. 2016.

STREBELLE, S. Conditional simulation of complex geological structures using multiple-point statistics. **Mathematical Geology**, v. 34, n. 1, p. 1–21, 2002. Disponível em: <<http://dx.doi.org/10.1023/A:1014009426274>>.

WACKERNAGEL, H. **Multivariate Geostatistics, 387 pp.** [S.l.]: Springer, New York, 2003.

ZHANG, Y. Introduction to geostatistics course notes. **Dept. of Geology & Geophysics, University of Wyoming**, 2011. Acesso em: 4 nov. 2016.