Inverse modeling of Moving Average Kernels for 3D Gaussian Simulation

Oscar Peredo[1] & Julián M. Ortiz[1,2] & Oy Leuangthong[3]

[1]Advanced Lab for Geostatistical Supercomputing, Advanced Mining Technology Center, University of Chile
[2]Department of Mining Engineering, University of Chile
[3]SRK Consulting (Canada) Inc.

10th Conference on Geostatistics for Environmental Applications, July 9-11, 2014 - Paris, France

Outline

Outline

Motivation

- Many methods exist for generating geostatistical simulations once a structural model (a variogram) has been determined.

Motivation

- Many methods exist for generating geostatistical simulations once a structural model (a variogram) has been determined.

- A constant challenge is to simplify this process and minimize the time required for generating the realizations.

Motivation

- Many methods exist for generating geostatistical simulations once a structural model (a variogram) has been determined.

- A constant challenge is to simplify this process and minimize the time required for generating the realizations.

- Given an input dataset,

  is there a non-parametric and automatic way to generate simulations without having to fit a model?

Outline

Introduction

- We propose an inverse approach to Gaussian simulation, based in the ***Moving Averages*** (MA) method as decribed by Journel and Huijbregts (1978) and Oliver (1995).

Introduction

- We propose an inverse approach to Gaussian simulation, based in the ***Moving Averages*** (MA) method as decribed by Journel and Huijbregts (1978) and Oliver (1995).

- A non-linear optimization problem is assembled and solved to find the best kernel that generates simulations, whose variogram matches experimental values.

Introduction

- We propose an inverse approach to Gaussian simulation, based in the ***Moving Averages*** (MA) method as decribed by Journel and Huijbregts (1978) and Oliver (1995).

- A non-linear optimization problem is assembled and solved to find the best kernel that generates simulations, whose variogram matches experimental values.

- With the optimal kernel at hand, several simulations can be generated delivering similar variograms, without having to fit a model to the experimental variogram.

Outline

Forward MA: theory

If we assume the covariance can be written as:

$$C(r) = \{f * f\}(r) \tag{1}$$

Forward MA: theory

If we assume the covariance can be written as:

$$C(r) = \{f * f\}(r) \tag{1}$$

we can recover the kernel function f applying:

$$f(r) = \mathcal{F}^{-1}\left(\sqrt{\mathcal{F}(C(r))}\right) \tag{2}$$

Forward MA: theory

If we assume the covariance can be written as:

$$C(r) = \{f * f\}(r) \tag{1}$$

we can recover the kernel function f applying:

$$f(r) = \mathcal{F}^{-1}\left(\sqrt{\mathcal{F}(C(r))}\right) \tag{2}$$

and using it to obtain correlated RFs $\mathbf{Y}$ with covariance $C(r)$:

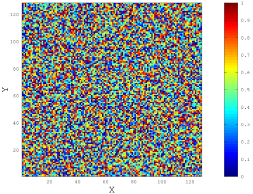
$$\mathbf{Y}(\mathbf{x}) = \int_U f(\mathbf{x} - \mathbf{x}')\mathbf{Z}(\mathbf{x}')d\mathbf{x}' = \{f * \mathbf{Z}\}(\mathbf{x}) \tag{3}$$

with $\mathbf{Z}$ a stationary RF with a Dirac covariance measure.

Forward MA: examples

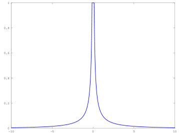
$\mathbf{Z}$	$\rightarrow$	f	$\rightarrow$	$\mathbf{Y} = f * \mathbf{Z}$

Pure nugget

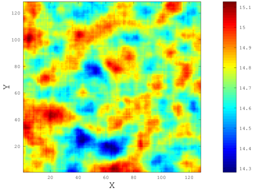


Exponential

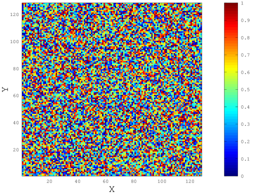
$$f(r) = \frac{1}{r\sqrt{2\pi a}} e^{\frac{-r}{a}}$$



$$C(r) = e^{\frac{-r}{a}}$$

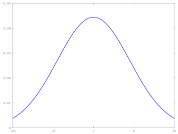


Pure nugget

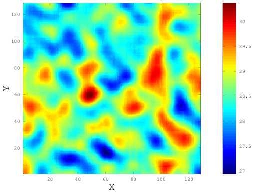


Gaussian

$$f(r) = \left(\frac{4}{a^2\pi}\right)^{\frac{3}{4}} e^{\frac{-2r^2}{a^2}}$$



$$C(r) = e^{\frac{-r^2}{a^2}}$$



Outline

Outline

- Motivation
- Introduction
- Forward Moving Averages
- Inverse Moving Averages
  - Theory
  - Implementation
  - Examples
- Conclusions / Future work

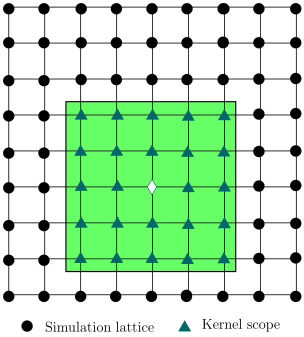
Inverse MA: theory

Relation between $\gamma(h)$ and $f(r)$ when $\mathbf{Y} = f * \mathbf{Z}$:

$$\begin{aligned} \gamma(h) &= \frac{1}{2N_h} \sum_{\{(p,q):x_q=x_p+h\}} [y(x_p) - y(x_q)]^2 \\ \gamma(h) &\approx \mathbf{w}^T \mathbf{M}(z,h) \mathbf{w} \end{aligned} \tag{4}$$

with:

- N_h: # pairs in the set $\{(p,q) : x_q = x_p + h\}$ from the simulation lattice.
- N_w: # points lying into kernel scope.
- $\mathbf{M}(z,h) \in \mathbb{R}^{N_w \times N_w}$: dense semidefinite matrix depending on z and h.
- $\mathbf{w} \in \mathbb{R}^{N_w}$: kernel values at discrete points, $w_i \in \{f(r) \in \mathbb{R} : r \geq 0\}, i = 1, \ldots, N_w$.



Inverse MA: theory

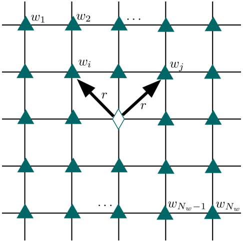
Forward MA	$\mathbf{w} := f(r) \quad \rightarrow \quad \gamma$
Inverse MA	$\gamma^{target} \quad \rightarrow \quad \mathbf{w}^{opt}$

Inverse MA: theory

Forward MA	$\mathbf{w} := f(r) \quad \rightarrow \quad \gamma$
Inverse MA	$\gamma^{target} \quad \rightarrow \quad \mathbf{w}^{opt}$

$$\underbrace{\begin{array}{ll} \underset{\mathbf{w}\in\mathbb{R}^{N_w}}{\text{minimize}} & \displaystyle\sum_{i=1}^{nlags} \left(\underbrace{\mathbf{w}^T \mathbf{M}(z,h_i)\mathbf{w}}_{\gamma^{\mathbf{Y}}(h_i)} - \gamma^{target}(h_i) \right)^2 \\ \text{subject to} & \begin{array}{llll} w_i & \leq & B & , i = 1,\ldots,N_w \\ w_i & \geq & 0 & , i = 1,\ldots,N_w \\ w_i & = & w_j & , i,j \text{ are equidistant to the origin} \end{array} \end{array}}_{\text{(OPT)}}$$

i,j are equidistant to the origin



Inverse MA: theory

Practitioner's work-flow:

$\gamma^{target} \leftarrow$ Obtain experimental variogram values
$\mathbf{w}^0 \leftarrow$ Set initial kernel weights
$\Omega \leftarrow$ Set a simulation lattice
$z \leftarrow$ Obtain realization of $\mathbf{Z}$ (pure nugget) in Ω
$(y, \mathbf{w}^{opt}) \leftarrow$ Solve (OPT) with inputs γ^{target}, $\mathbf{w}^0$, Ω and z
while More realizations of $\mathbf{Y}$ in Ω are needed **do**
    $z \leftarrow$ Obtain realization of $\mathbf{Z}$ (pure nugget) in Ω
    $y \leftarrow$ Convolute z with $\mathbf{w}^{opt}$ in Ω (Forward MA)
end while

Inverse MA: theory

Practitioner's work-flow:

$\gamma^{target} \leftarrow$ Obtain experimental variogram values
$\mathbf{w}^0 \leftarrow$ Set initial kernel weights
$\Omega \leftarrow$ Set a simulation lattice
$z \leftarrow$ Obtain realization of $\mathbf{Z}$ (pure nugget) in Ω
$(y, \mathbf{w}^{opt}) \leftarrow$ Solve (OPT) with inputs γ^{target}, $\mathbf{w}^0$, Ω and z
while More realizations of $\mathbf{Y}$ in Ω are needed **do**
    $z \leftarrow$ Obtain realization of $\mathbf{Z}$ (pure nugget) in Ω
    $y \leftarrow$ Convolute z with $\mathbf{w}^{opt}$ in Ω (Forward MA)
end while

Inverse MA: theory

Practitioner's work-flow:

1: $\gamma^{target} \leftarrow$ Obtain experimental variogram values
2: $\mathbf{w}^0 \leftarrow$ Set initial kernel weights
3: $\Omega \leftarrow$ Set a simulation lattice
4: $z \leftarrow$ Obtain realization of $\mathbf{Z}$ (pure nugget) in Ω
5: $(y, \mathbf{w}^{opt}) \leftarrow$ Solve (OPT) with inputs γ^{target}, $\mathbf{w}^0$, Ω and z
6: **while** More realizations of $\mathbf{Y}$ in Ω are needed **do**
7: $\quad z \leftarrow$ Obtain realization of $\mathbf{Z}$ (pure nugget) in Ω
8: $\quad y \leftarrow$ Convolute z with $\mathbf{w}^{opt}$ in Ω (Forward MA)
9: **end while**

Inverse MA: theory

Practitioner's work-flow:

$\gamma^{target} \leftarrow$ Obtain experimental variogram values
$\mathbf{w}^0 \leftarrow$ Set initial kernel weights
$\Omega \leftarrow$ Set a simulation lattice
$z \leftarrow$ Obtain realization of $\mathbf{Z}$ (pure nugget) in Ω
$(y, \mathbf{w}^{opt}) \leftarrow$ Solve (OPT) with inputs γ^{target}, $\mathbf{w}^0$, Ω and z
while More realizations of $\mathbf{Y}$ in Ω are needed **do**
    $z \leftarrow$ Obtain realization of $\mathbf{Z}$ (pure nugget) in Ω
    $y \leftarrow$ Convolute z with $\mathbf{w}^{opt}$ in Ω (Forward MA)
end while

Inverse MA: theory

Practitioner's work-flow:

$\gamma^{target} \leftarrow$ Obtain experimental variogram values
$\mathbf{w}^0 \leftarrow$ Set initial kernel weights
$\Omega \leftarrow$ Set a simulation lattice
$z \leftarrow$ Obtain realization of $\mathbf{Z}$ (pure nugget) in Ω
$(y, \mathbf{w}^{opt}) \leftarrow$ Solve (OPT) with inputs γ^{target}, $\mathbf{w}^0$, Ω and z
while More realizations of $\mathbf{Y}$ in Ω are needed **do**
    $z \leftarrow$ Obtain realization of $\mathbf{Z}$ (pure nugget) in Ω
    $y \leftarrow$ Convolute z with $\mathbf{w}^{opt}$ in Ω (Forward MA)
end while

Inverse MA: theory

Practitioner's work-flow:

$\gamma^{target} \leftarrow$ Obtain experimental variogram values
$\mathbf{w}^0 \leftarrow$ Set initial kernel weights
$\Omega \leftarrow$ Set a simulation lattice
$z \leftarrow$ Obtain realization of $\mathbf{Z}$ (pure nugget) in Ω
$(y, \mathbf{w}^{opt}) \leftarrow$ Solve (OPT) with inputs γ^{target}, $\mathbf{w}^0$, Ω and z
while More realizations of $\mathbf{Y}$ in Ω are needed **do**
    $z \leftarrow$ Obtain realization of $\mathbf{Z}$ (pure nugget) in Ω
    $y \leftarrow$ Convolute z with $\mathbf{w}^{opt}$ in Ω (Forward MA)
end while

Inverse MA: theory

Practitioner's work-flow:

$\gamma^{target} \leftarrow$ Obtain experimental variogram values
$\mathbf{w}^0 \leftarrow$ Set initial kernel weights
$\Omega \leftarrow$ Set a simulation lattice
$z \leftarrow$ Obtain realization of $\mathbf{Z}$ (pure nugget) in Ω
$(y, \mathbf{w}^{opt}) \leftarrow$ Solve (OPT) with inputs γ^{target}, $\mathbf{w}^0$, Ω and z
while More realizations of $\mathbf{Y}$ in Ω are needed **do**
    $z \leftarrow$ Obtain realization of $\mathbf{Z}$ (pure nugget) in Ω
    $y \leftarrow$ Convolute z with $\mathbf{w}^{opt}$ in Ω (Forward MA)
end while

Inverse MA: theory

Practitioner's work-flow:

1: $\gamma^{target} \leftarrow$ Obtain experimental variogram values
2: $\mathbf{w}^0 \leftarrow$ Set initial kernel weights
3: $\Omega \leftarrow$ Set a simulation lattice
4: $z \leftarrow$ Obtain realization of $\mathbf{Z}$ (pure nugget) in Ω
5: $(y, \mathbf{w}^{opt}) \leftarrow$ Solve (OPT) with inputs γ^{target}, $\mathbf{w}^0$, Ω and z
6: **while** More realizations of $\mathbf{Y}$ in Ω are needed **do**
7: $\quad z \leftarrow$ Obtain realization of $\mathbf{Z}$ (pure nugget) in Ω
8: $\quad y \leftarrow$ Convolute z with $\mathbf{w}^{opt}$ in Ω (Forward MA)
9: **end while**

Inverse MA: theory

Practitioner's work-flow:

$\gamma^{target} \leftarrow$ Obtain experimental variogram values
$\mathbf{w}^0 \leftarrow$ Set initial kernel weights
$\Omega \leftarrow$ Set a simulation lattice
$z \leftarrow$ Obtain realization of $\mathbf{Z}$ (pure nugget) in Ω
$(y, \mathbf{w}^{opt}) \leftarrow$ Solve (OPT) with inputs $\gamma^{target}, \mathbf{w}^0, \Omega$ and z
while More realizations of $\mathbf{Y}$ in Ω are needed **do**
    $z \leftarrow$ Obtain realization of $\mathbf{Z}$ (pure nugget) in Ω
    $y \leftarrow$ Convolute z with $\mathbf{w}^{opt}$ in Ω (Forward MA)
end while

We have obtained 1 or more realizations of $\mathbf{Y}$ with variograms similar to γ^{target} in an **automatic and non-parametric way.**

Outline

Inverse MA: implementation

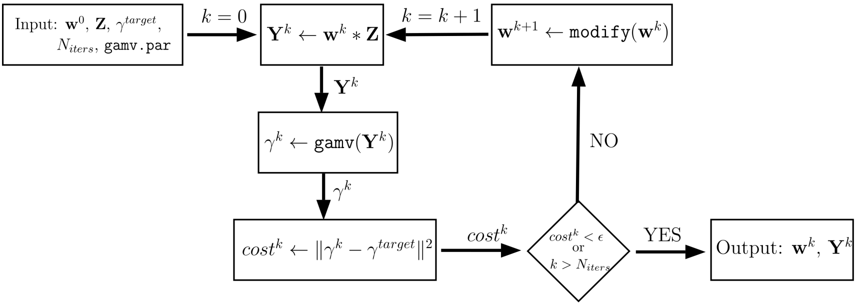
Resolution of (OPT):

Inverse MA: implementation

Resolution of (OPT):
Simulated annealing scheme, implemented in ANSI C + Fortran calls to GSLIB routines `gam/gamv`.

Inverse MA: implementation

Resolution of (OPT):
Simulated annealing scheme, implemented in ANSI C + Fortran calls to GSLIB routines `gam/gamv`.

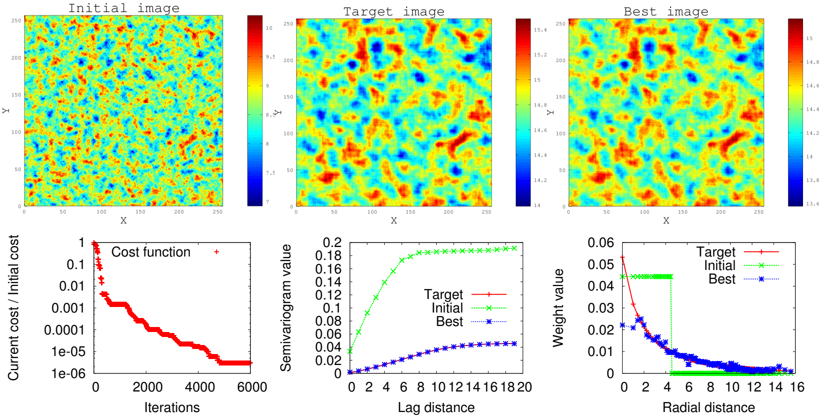


Outline

Inverse MA: example 1

Synthetic scenario:

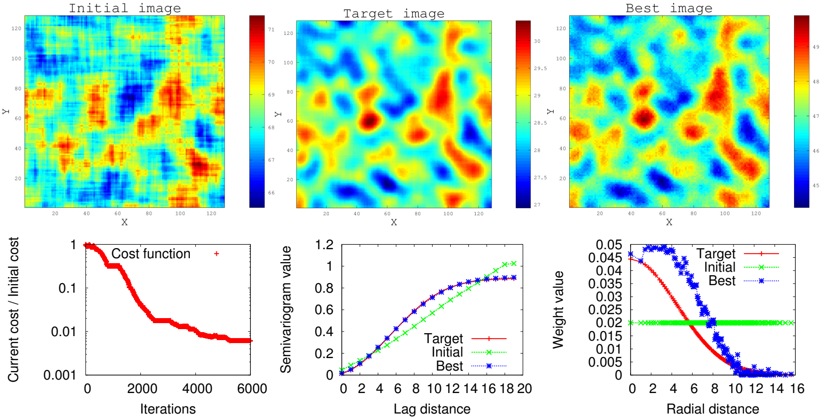
$\mathbf{w}^0$ spherical (initial image) , γ^{target} exponential (target image) $\rightarrow$ $\mathbf{w}^{opt}$ exponential (best image)



Inverse MA: example 2

Synthetic scenario (using `nscore` before applying `gam/gamv`):

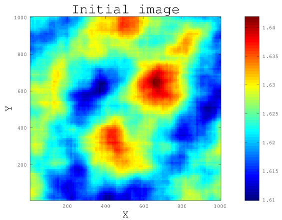
$\mathbf{w}^0$ constant	,	γ^{target} gaussian	$\rightarrow$	$\mathbf{w}^{opt}$ gaussian
(initial image)		(target image)		(best image)

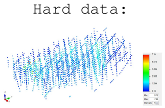


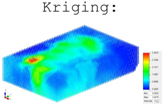
Inverse MA: example 3

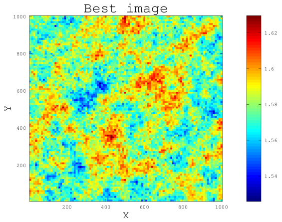
Real data scenario (using `nscore` before applying `gam/gamv`):

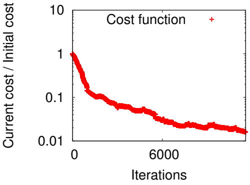
$\mathbf{w}^0$ exponential (initial image) , γ^{target} experimental (real data) $\rightarrow$ $\mathbf{w}^{opt}$ (best image)

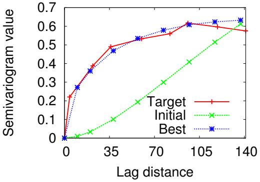


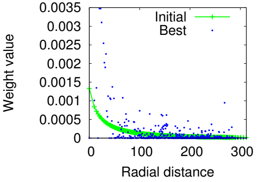








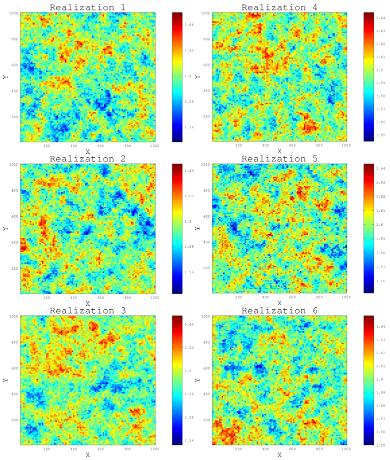


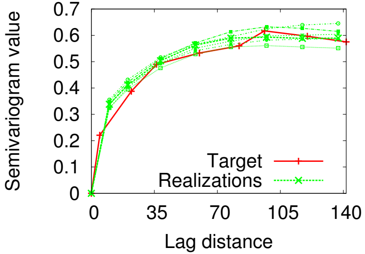


Inverse MA: example 3 (cont'd)

Real data scenario (using `nscore` before applying `gam/gamv`):

$$\mathbf{w}^{opt} \quad \rightarrow \quad \mathbf{Y}^1, \ldots, \mathbf{Y}^n \text{ with } \gamma^1 \approx \gamma^{target}, \ldots, \gamma^n \approx \gamma^{target}$$





Outline

Conclusions

- New approach to Gaussian simulation that **does not require explicit variogram modelling by the practitioner.**
- Early application of this methodology to synthetic and real data examples show reasonable convergence and kernel weight values.
- Simulated annealing is used to solve the optimization problem, but in principle any method for non-linear optimization can be used.

Future work

- Allow anisotropy in kernel weights (relaxing equidistance constraint).
- Add hard data conditioning to the simulated images.
- Accelerate the current implementation of convolution and variogram calculation.
  - Convolution: straight-forward parallelization, multi-core and GPU acceleration.
  - Variogram: slightly more difficult, but recent acceleration advances are promising[1,2].

[1] Peredo, Ortiz: *Resurrecting GSLIB by code optimization and multi-core programming*, to appear in IAMG2014. Obtained up to 12x with 1×10^6 data points.

[2] Baeza, Peredo, Ortiz: *A fast implementation of gamv using GPUs*, in preparation. Obtained up to 49x with 1×10^6 data points (single precision).

Thanks for your attention!

Contact: `operedo [at] alges.cl`

`http://www.alges.cl`

`http://www.amtc.cl`

References (1)

A. G. Journel and C. J. Huijbregts, *Mining Geostatistics*, Academic Press, London, 1978.

Dean S. Oliver, *Moving averages for gaussian simulation in two and three dimensions.*, Mathematical Geology **27** (1995), no. 8, 939–960.