

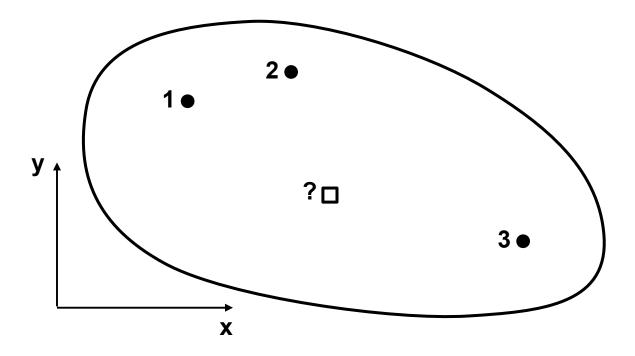
Lecture outline . . .

- Kriging
- Kriging Interactive Demo with GeostatsPy
- Kriging Workflow with GeostatsPy



We need to make predictions away from sampled locations.

• To determine where to sample next, find a resource, remediate the subsurface:







12b Geostatistics Course: Kriging

GeostatsGuy Lectures



12c Data Analytics: Kriging in R

GeostatsGuy Lectures



12d Python Data Analytics: Simple Kriging

GeostatsGuy Lectures

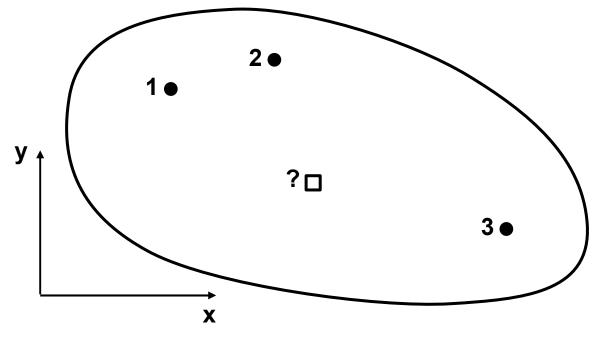


Lecture outline . . .

Kriging



Consider the case of estimating at an unsampled location:



 $z(u_{\alpha})$ is the data values

 $z^*(u_0)$ is an estimate

 λ_{α} is the data weights

 m_z is the global mean

How would you do this given data, $z(\mathbf{u}_1)$, $z(\mathbf{u}_2)$, and $z(\mathbf{u}_3)$?

$$z^*(\mathbf{u_0}) = \sum_{\alpha=1}^n \lambda_{\alpha} \mathbf{z}(\mathbf{u_{\alpha}}) + \left(1 - \sum_{\alpha=1}^n \lambda_{\alpha}\right) m_z$$
 Unbiasedne Constraint Weights su

Unbiasedness Weights sum to 1.0.



Simple Kriging System of Equations

We use kriging to calculate the best weights integrating:

- 1. spatial continuity: the variogram (expressed as covariance)
- 2. closeness: spatial correlation between samples and unknown location
- 3. redundancy: spatial correlation between samples and each other

The kriging system of equations to determine the three weights:

$$\lambda_1 \cdot C(\mathbf{u}_1, \mathbf{u}_1) + \lambda_2 \cdot C(\mathbf{u}_1, \mathbf{u}_2) + \lambda_3 \cdot C(\mathbf{u}_1, \mathbf{u}_3) = C(\mathbf{u}_0, \mathbf{u}_1)$$

$$\lambda_1 \cdot C(\mathbf{u}_2, \mathbf{u}_1) + \lambda_2 \cdot C(\mathbf{u}_2, \mathbf{u}_2) + \lambda_3 \cdot C(\mathbf{u}_2, \mathbf{u}_3) = C(\mathbf{u}_0, \mathbf{u}_2)$$

$$\lambda_1 \cdot C(\mathbf{u}_3, \mathbf{u}_1) + \lambda_2 \cdot C(\mathbf{u}_1, \mathbf{u}_2) + \lambda_3 \cdot C(\mathbf{u}_1, \mathbf{u}_3) = C(\mathbf{u}_0, \mathbf{u}_1)$$

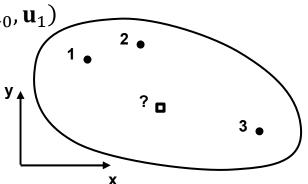
Note: $C(\mathbf{h}) = C(0) - \gamma(\mathbf{h})$

In matrix notation:

$$\begin{bmatrix} C(\mathbf{u}_1, \mathbf{u}_1) & C(\mathbf{u}_1, \mathbf{u}_2) & C(\mathbf{u}_1, \mathbf{u}_3) \\ C(\mathbf{u}_2, \mathbf{u}_1) & C(\mathbf{u}_2, \mathbf{u}_2) & C(\mathbf{u}_2, \mathbf{u}_3) \\ C(\mathbf{u}_3, \mathbf{u}_1) & C(\mathbf{u}_3, \mathbf{u}_2) & C(\mathbf{u}_3, \mathbf{u}_3) \end{bmatrix} \begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = \begin{bmatrix} C(\mathbf{u}_0, \mathbf{u}_1) \\ C(\mathbf{u}_0, \mathbf{u}_2) \\ C(\mathbf{u}_0, \mathbf{u}_3) \end{bmatrix}$$

redundancy

closeness





- Solution exists and is unique of matrix $\left[C(v_i, v_j)\right]$ is positive definite
- Kriging estimator is unbiased: $E\left\{\left[Z Z^*\right]\right\} = 0$
- Minimum error variance estimator (just try to pick weights, you won't bet it)
- Best Linear Unbiased Estimator
- Provides a measure of the estimation (or kriging) variance (uncertainty in the estimate):

$$\sigma_E^2(\mathbf{u}) = C(0) - \sum_{\alpha=1}^n \lambda_\alpha C(\mathbf{u} - \mathbf{u}_\alpha) \qquad \sigma_E^2 \to [\mathbf{0}, \sigma_x^2]$$



Lecture outline . . .

 Kriging Interactive Demo with GeostatsPy



Let's calculate spatial estimates with kriging:

- we can move the data and change the variogram model and observe:
 - estimate and uncertainty
 - data weights

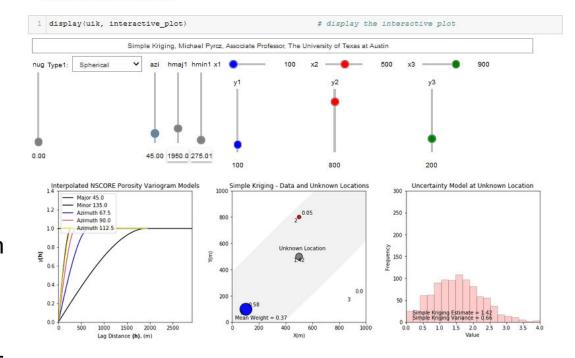
Observe the these for:

- 100% nugget effect
- isotropic with range of 9000m
- a data at the estimate location
- strong anisotropy
- two data very close to each other
- one data screened by another

Interactive Simple Kriging Demostration • select the variogram model and the data locations and observe the outputs from simple kriging Michael Pyrcz, Associate Professor, University of Texas at Austin Twitter | GitHub | Website | GoogleScholar | Book | YouTube | LinkedIn | GeostatsPy The Inputs Select the variogram model and the data locations:

• nua: nuaget effect

- c1 : contributions of the
- . hmaj1 / hmin1 : range in the major and minor direction
- . (x1, y1),...(x3,y3): spatial data locations



Interactive Python Jupyter kriging (Interactive Simple Kriging.ipynb).



Lecture outline . . .

Kriging Workflow with GeostatsPy



Let's walkthrough a more thorough a spatial estimation workflow:

facies by indicator kriging

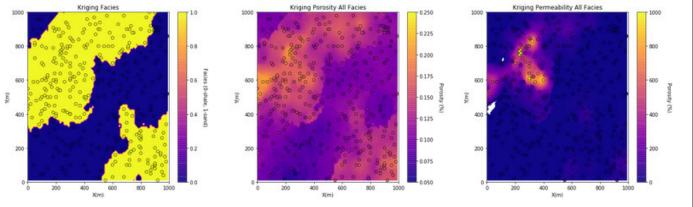
porosity by-facies with

simple kriging

ikmap = geostats.ik2d(df, 'X', 'Y', 'Facies', ivtype, 0, 2, thresh, gcdf, no_trend, tmin, tmax, nx, xmn, xsiz, ny, ymn, ysiz, ne GSLIB.locpix st(ikmap[:,:,0],xmin,xmax,ymin,ymax,xsiz,0.0,1.0,df,'X','Y','Facies','Probability Shale','X(m)',' GSLIB.locpix_st(ikmap[:,:,1],xmin,xmax,ymin,ymax,xsiz,0.0,1.0,df,'X','Y','Facies','Probability Sand','X(m)','Y plt.subplots adjust(left=0.0, bottom=0.0, right=2.0, top=2.5, wspace=0.2, hspace=0.2) Data for IK3D: Variable column Facies Number = 368 Setting up rotation matrices for variogram and search Working on the kriging currently on estimate 0 currently on estimate 1000 currently on estimate 2000 currently on estimate 3000 currently on estimate 4000 currently on estimate 5000 currently on estimate 6000 currently on estimate 7000 currently on estimate 8000 currently on estimate 9000

We are ready to run the indicator kriging with the 2 cateogries (sand and shale) and calculate the probability of sand and shale at all locations and plot the results.

null ndarray not of correct size so ik2d will not use a trend -



Python Jupyter kriging (GeostatsPy_kriging.ipynb).



Lecture outline . . .

- Kriging
- Kriging Interactive Demo with GeostatsPy
- Kriging Workflow with GeostatsPy