Improving Spatial Data Interoperability

A Framework for Geostatistical Support-To-Support Interpolation

Michael F. Goodchild, Phaedon C. Kyriakidis, Philipp Schneider, Matt Rice, Qingfeng Guan, Jordan Hastings
University of California, Santa Barbara

http://www.ncgia.ucsb.edu/projects/nga/

Overview

- Overall NGA project description
- Past work in support of interoperability
- Geostatistical support-to-support interpolation framework
  - Theoretical background
  - A simple application example
- Conclusions

Introduction

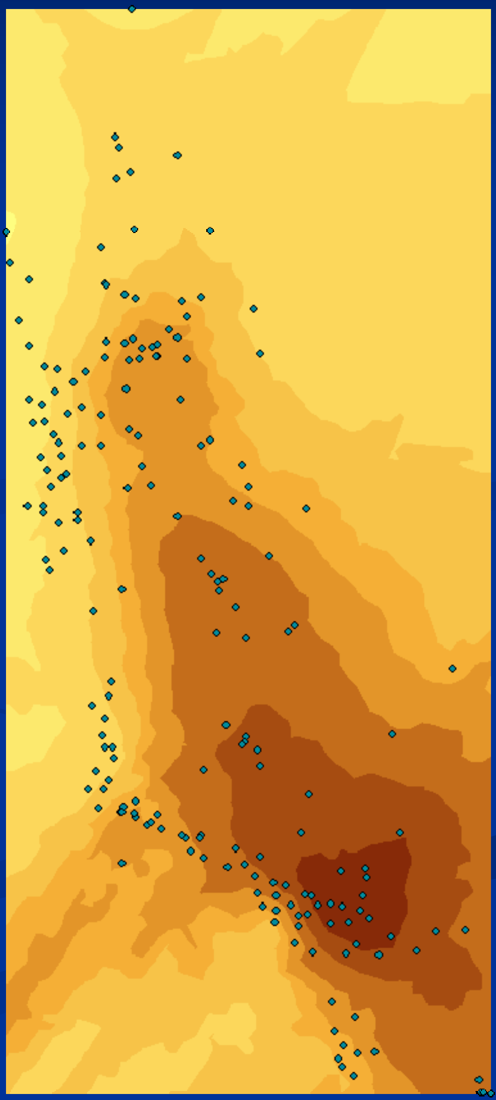
- GIS users are more and more sharing spatial data of different origins
- Geospatial data community is facing interoperability problems between data sets
- There are four main interoperability issues
  - Syntax
  - Semantics
  - Accuracy
  - Spatial Support
- Last aspect has not seen as much research as the first three

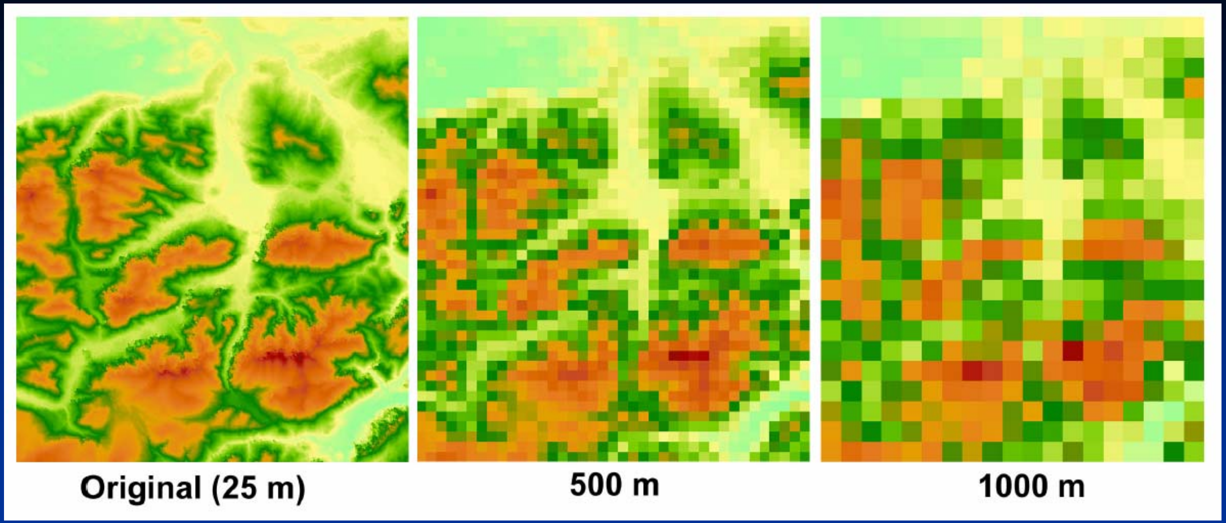
Overall Project Research Objectives

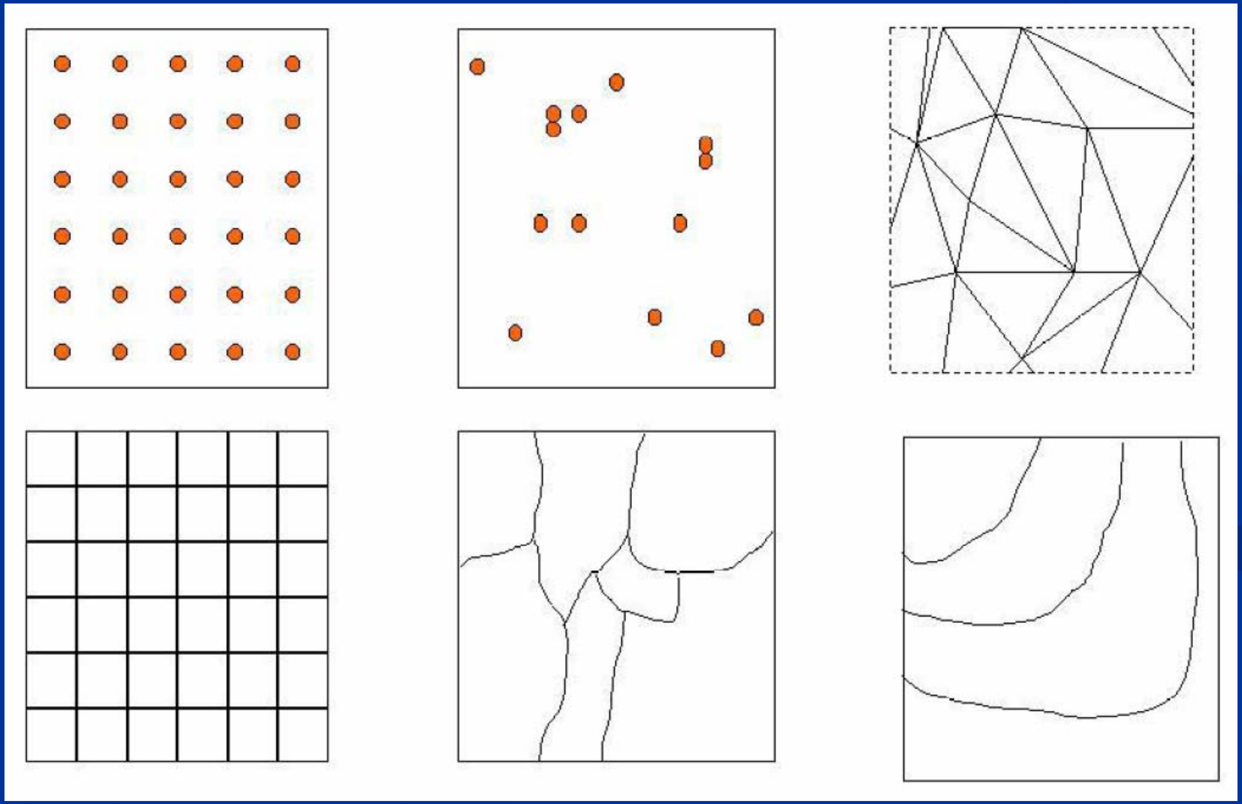
- Spatial Webs
  - Investigate problems occurring in Spatial Webs due to interoperability issues
  - Devise methods to overcome these problems
  - Test the methods on a prototype
- Data Integration
  - Investigate how datasets can be integrated into a single product or used to answer a single query (virtual product)
  - Devise methods of conflation for concatenation, averaging, and differencing
  - Test the developed methods on a prototype

Spatial support issues

- Discrete objects vs. continuous fields
- Six major discretization methods:
  - sampling at irregular points
  - sampling at point grid
  - averaging over grid of cells
  - digitized contours
  - TINs
  - irregular tesselations
- Each method creates its own spatial support (points, raster cells, polylines, polygons etc.)
- Interoperability issues when integrating fields with other fields or with discrete objects
  - Solution: Point or areal interpolation and resampling




Original (25 m)
500 m
1000 m

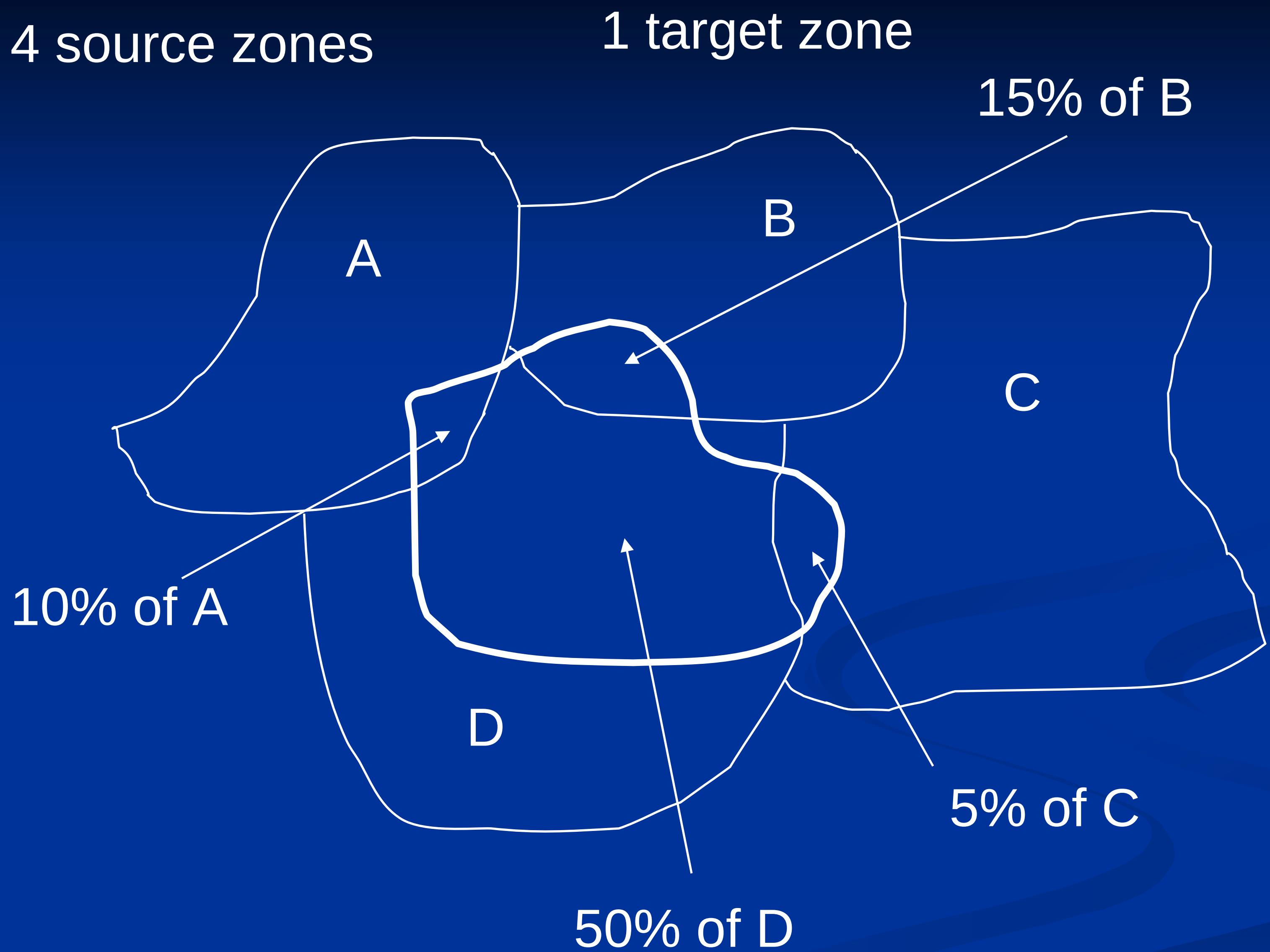


NGA Project Research agenda – Spatial support

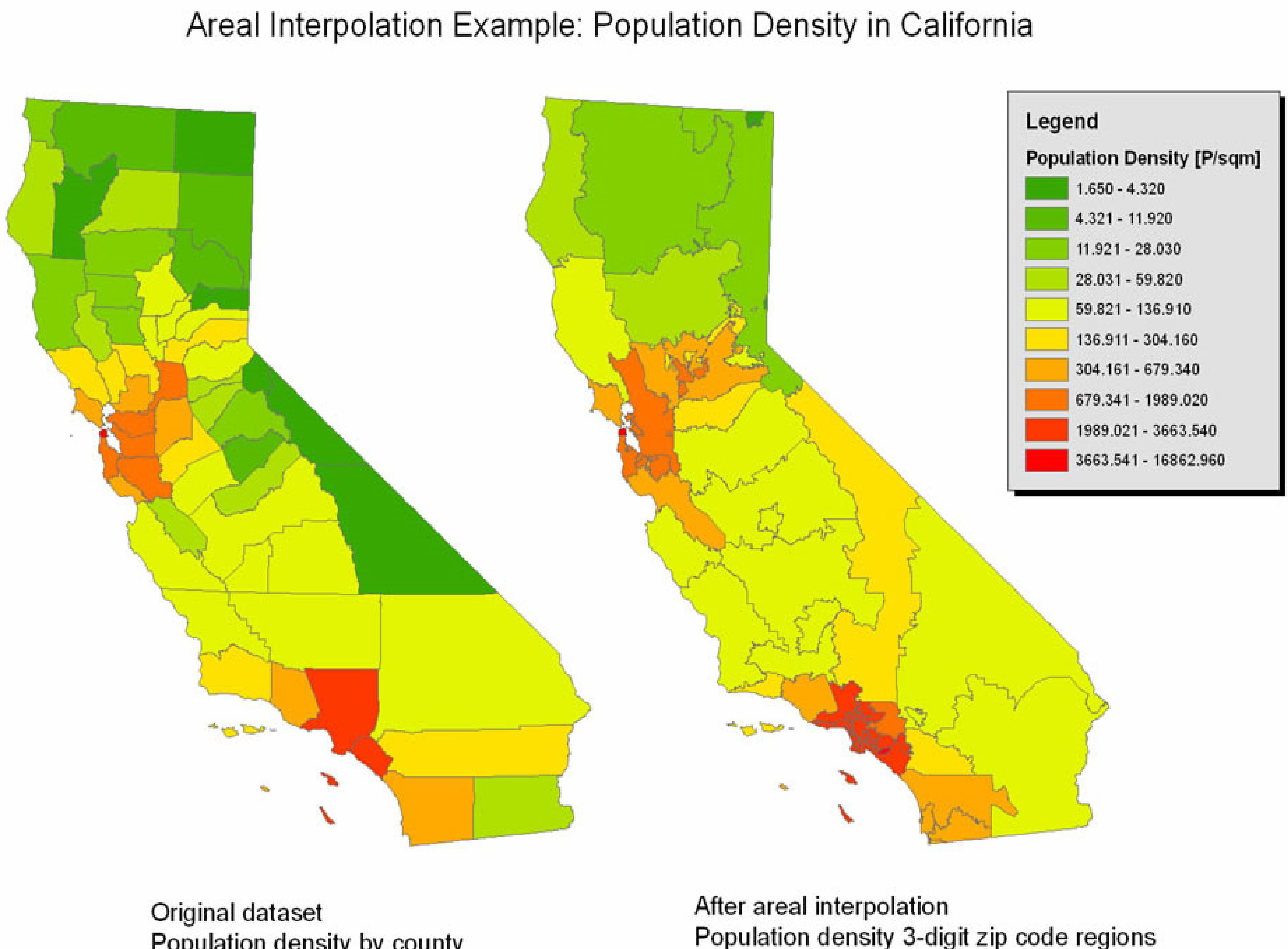
- Investigation of potential issues in connection with spatial support
  - Accounting for spatial correlation
  - Incorporation of known point measurements and boundary conditions
- Development of solutions for these issues
  - Enhanced areal interpolation
- Implementation of a prototype for transparent interoperability of datasets with different spatial support
- Demonstration of the use of the developed methods in case studies

Past research on support interoperability

- Areal interpolation has been around since 1980
  - Areal weighting (piecewise approximation)
  - Interpolation using control zones
- So far, no general and unifying framework has been developed
- Existing methods lack some desirable properties
  - Spatial correlation not accounted for
  - Internal consistency (mass preservation) not guaranteed
  - No uncertainty assessment



$$Pop_{TARGET} = 0.10\ Pop_A + 0.15\ Pop_B + 0.05\ Pop_C + 0.50\ Pop_D$$



Areal Interpolation Example: Population Density in California
Legend
Population Density [P/sqm]
1.650 - 4.320
4.321 - 11.920
11.921 - 28.030
28.031 - 59.820
59.821 - 136.910
136.911 - 304.160
304.161 - 679.340
679.341 - 1989.020
1989.021 - 3663.540
3663.541 - 16862.960
Original dataset
Population density by county
After areal interpolation
Population density 3-digit zip code regions

Spatial support – a geostatistical approach

- Definition
  - Support = Domain informed by each datum or unknown value
- Assumption of underlying point support field

$$\{z(\mathbf{x}), \mathbf{x} \in D\}$$

  - Actual value unknown
  - Viewed as realizations of stationary random field (RF) model
  - Parametrized by a mean and covariance function

$$E\{Z(\mathbf{x})\} = m_Z, \forall \mathbf{x} \qquad Cov\{Z(\mathbf{x}), Z(\mathbf{x}')\} = C_Z(\mathbf{x} - \mathbf{x}')$$

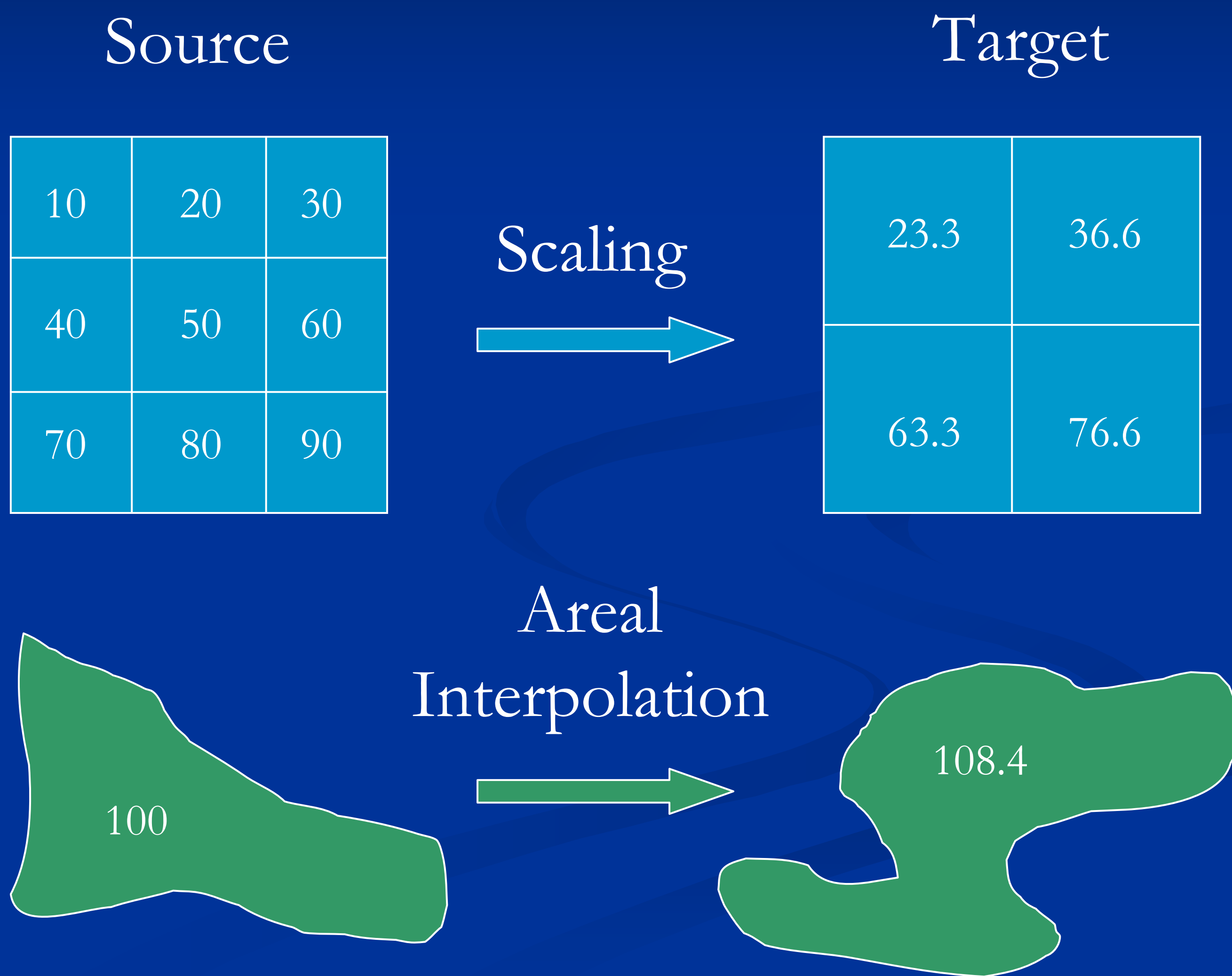
Geostatistical Support-To-Support Interpolation

- Framework for dealing with spatial support interoperability
- Distinguishes between two sets of supports
  - Source support with known attribute values
  - Target support with unknown attribute values
- Supports can be completely disjoint or partially overlapping
- Can have arbitrary shape, size, and orientation
- Single requirement: No two source supports may coincide

Support Examples

Raster data

Vector data (Polygons)



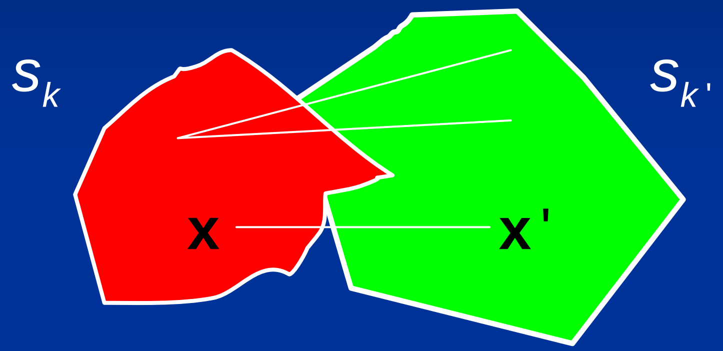
Procedure

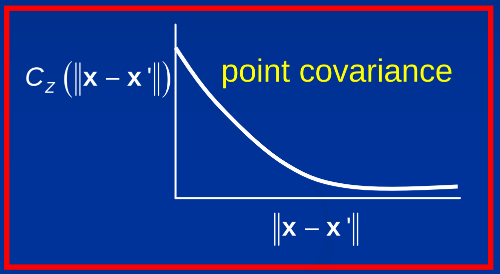
- Target predictions obtained by block-to-block Kriging using
  - Available source data
  - Auto-covariance between any two source supports
  - Cross-covariance between any two target and source supports
- Covariances computed as convolutions of the point support covariance with respective sampling kernels $g(x)$
- Sampling kernel $g(x)$ quantifies contribution of each point to the areal datum in whose support that point falls
- $g(x) = 1$ for intensive variables (e.g., population density)
  $g(x) = 1/A_s$ for extensive variables (A_s = area of support)

Auto-covariance between two source supports

$$C_Z\left(s_k, s_{k'}\right) = Cov\left\{Z\left(s_k\right), Z\left(s_{k'}\right)\right\} = \int_{\mathbf{x}\in s_k} g_k(\mathbf{x}) \int_{\mathbf{x}'\in s_{k'}} g_{k'}(\mathbf{x}')\, C_Z(\mathbf{x}-\mathbf{x}')\, d\mathbf{x}'\, d\mathbf{x}$$

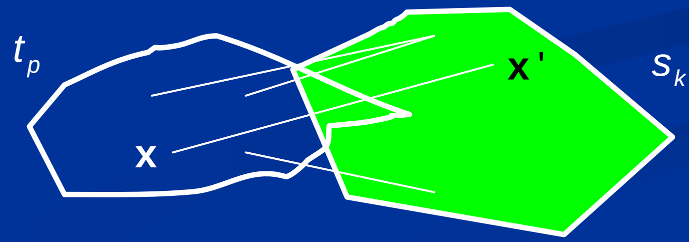
non-stationary





Cross-covariance between target and source supports

$$C_Z\left(t_p, s_k\right) = Cov\left\{Z\left(t_p\right), Z\left(s_k\right)\right\} = \int_{\mathbf{x}\in t_p} g_p(\mathbf{x}) \int_{\mathbf{x}'\in s_k} g_k(\mathbf{x}')\, C_Z(\mathbf{x}-\mathbf{x}')\, d\mathbf{x}'\, d\mathbf{x}$$



Predicting Target Values

Assuming a known point mean m_Z=0, for simplicity

$$\hat{z}\left(t_p\right) = \mathbf{w}_p^{'}\mathbf{z}_s = \left[w_p\left(s_1\right) \quad \cdots \quad w_p\left(s_K\right) \right] \begin{bmatrix} z\left(s_1\right) \\ \vdots \\ z\left(s_K\right) \end{bmatrix}$$

Recall that some source data might be of point support

<u>System of normal equations for finding weights:</u>

$$\begin{pmatrix} C_Z\left(s_1, s_1\right) & \cdots & C_Z\left(s_1, s_K\right) \\ \vdots & \ddots & \vdots \\ C_Z\left(s_K, s_1\right) & \cdots & C_Z\left(s_K, s_K\right) \end{pmatrix} \begin{pmatrix} w_p\left(s_1\right) \\ \vdots \\ w_p\left(s_K\right) \end{pmatrix} = \begin{pmatrix} C_Z\left(t_p, s_1\right) \\ \vdots \\ C_Z\left(t_p, s_K\right) \end{pmatrix}$$

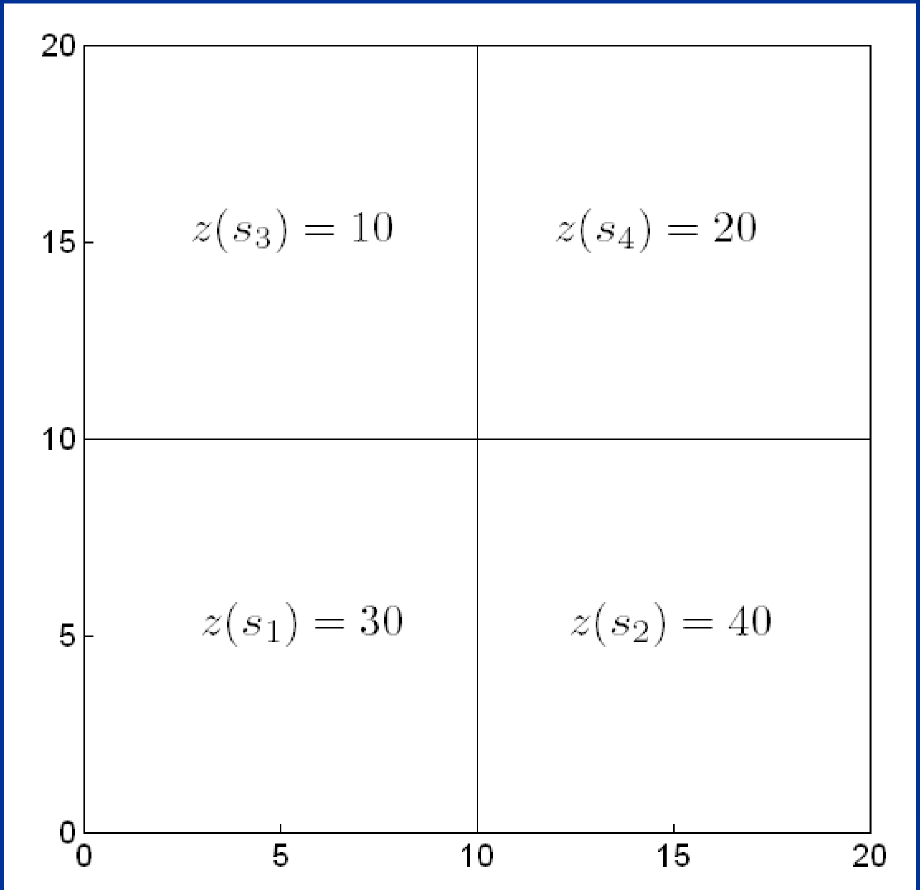
Unique solution exists, as long as point covariance model is positive definite

Framework properties

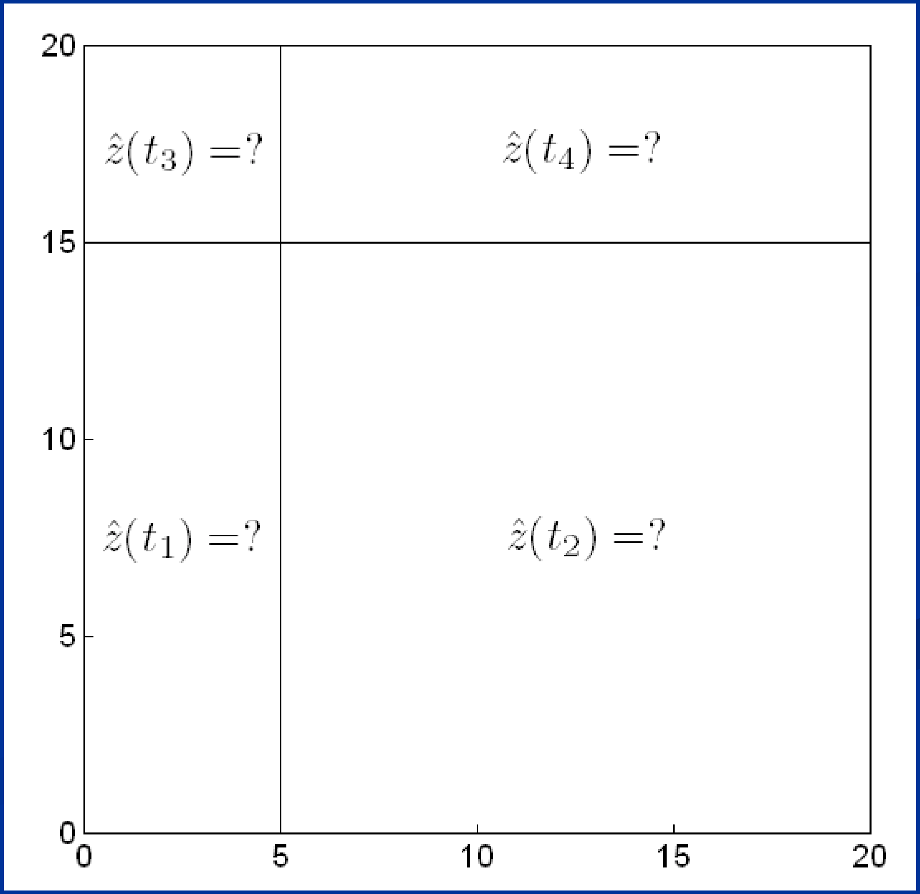
- General: Can handle integrated measurements over arbitrary domains
- Simple: Utilizes standard geostatistical theory with minor modifications
- Comprehensive: can handle alternative types of point covariance models
- Consistent: guarantees reproduction of data at larger scales (mass preserving)
- Providing uncertainty assessment: regarding target predictions

A simple example

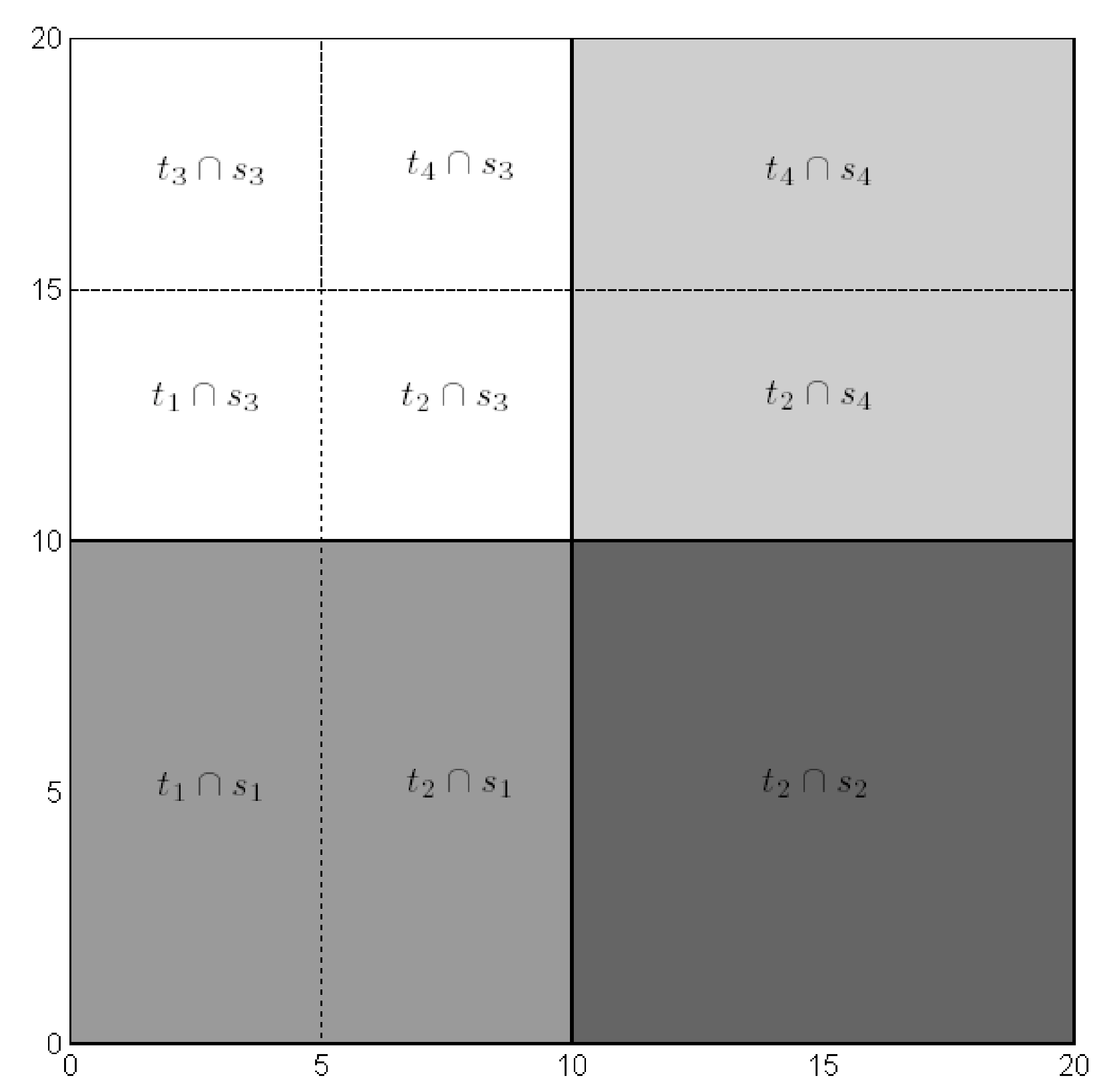
Source zone configuration and data values



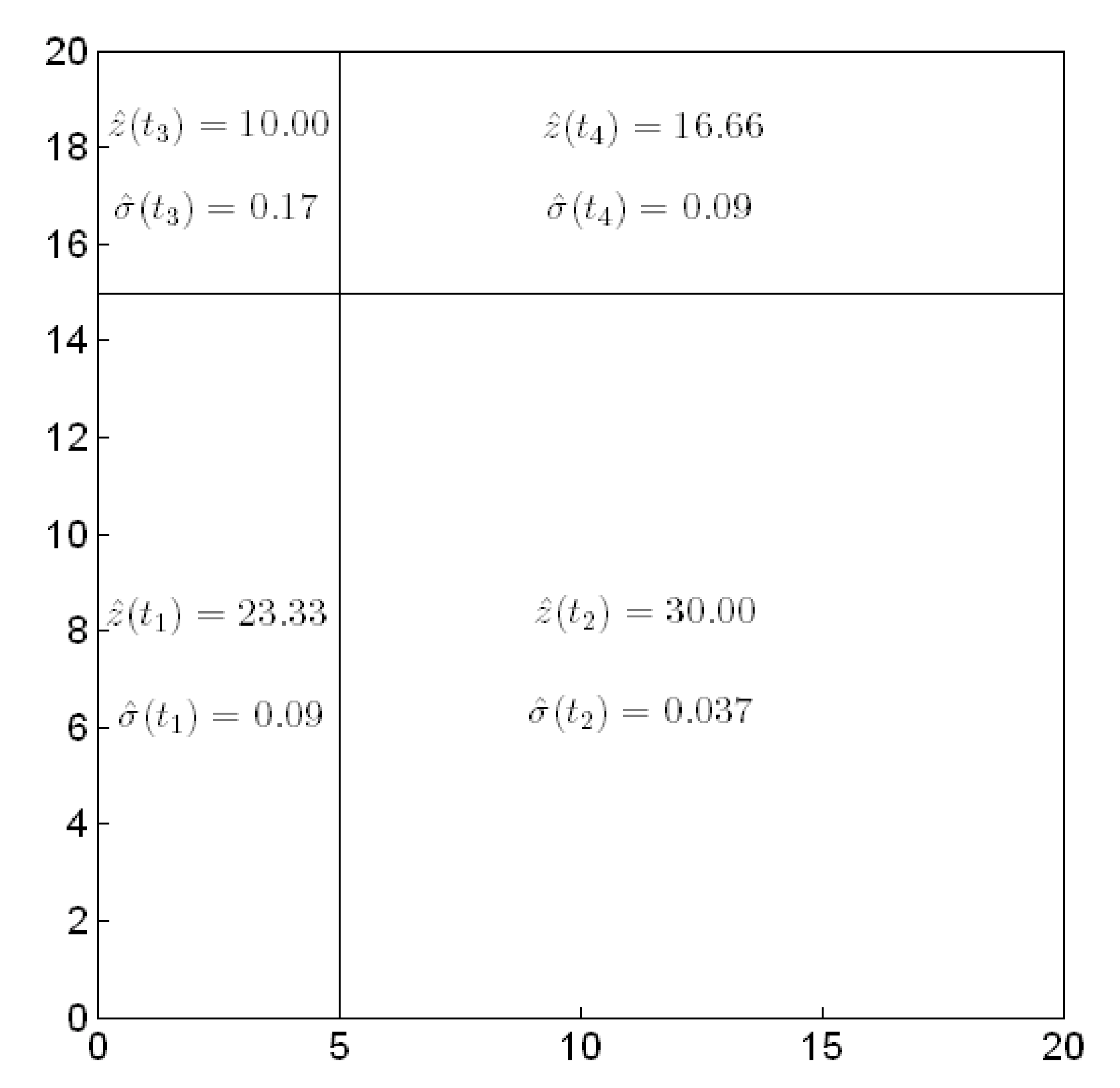
Target zone configuration



Source and target overlay



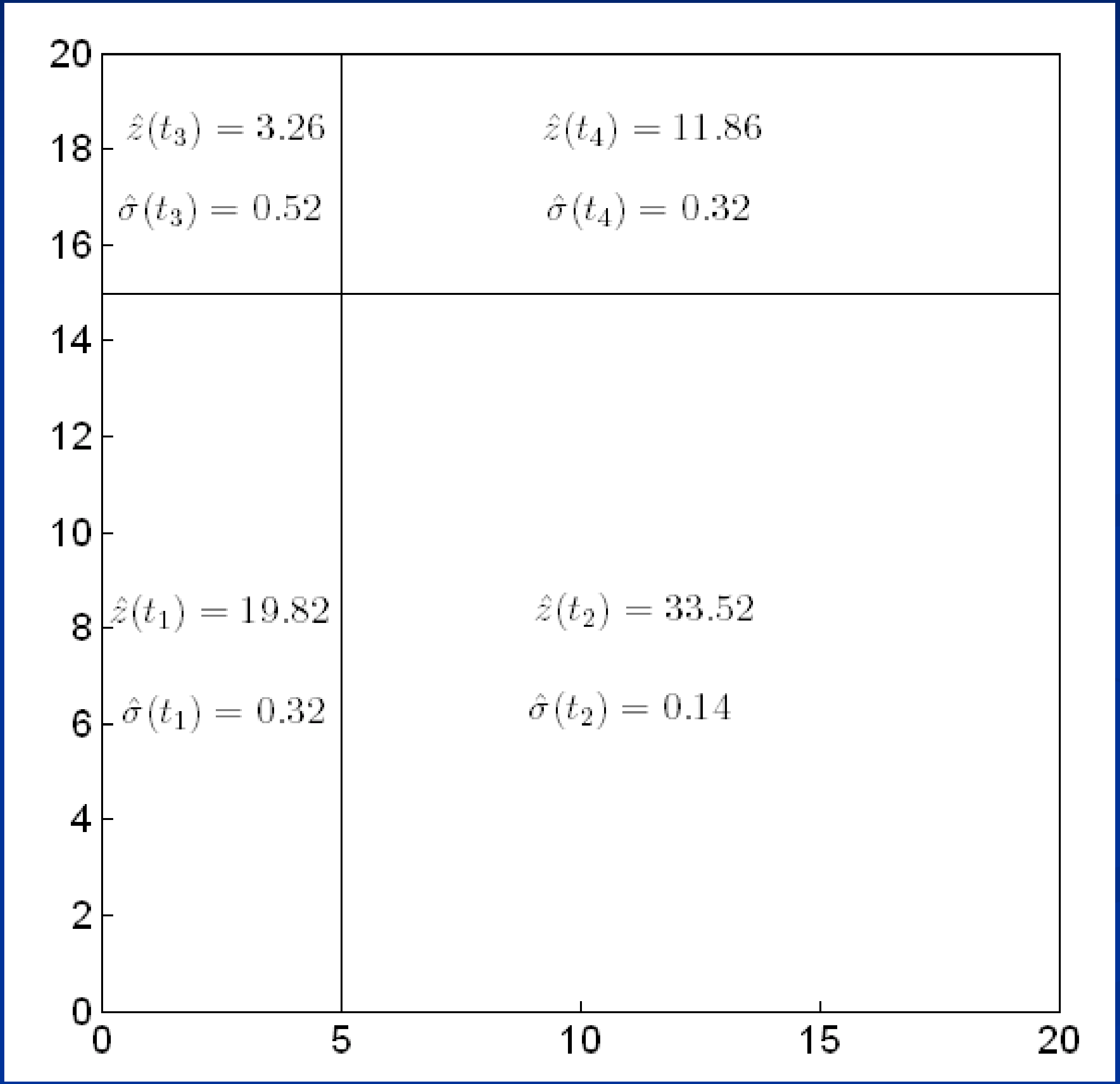
Results: No spatial correlation



Pure nugget effect

Unit sill

Spherical semivariogram model

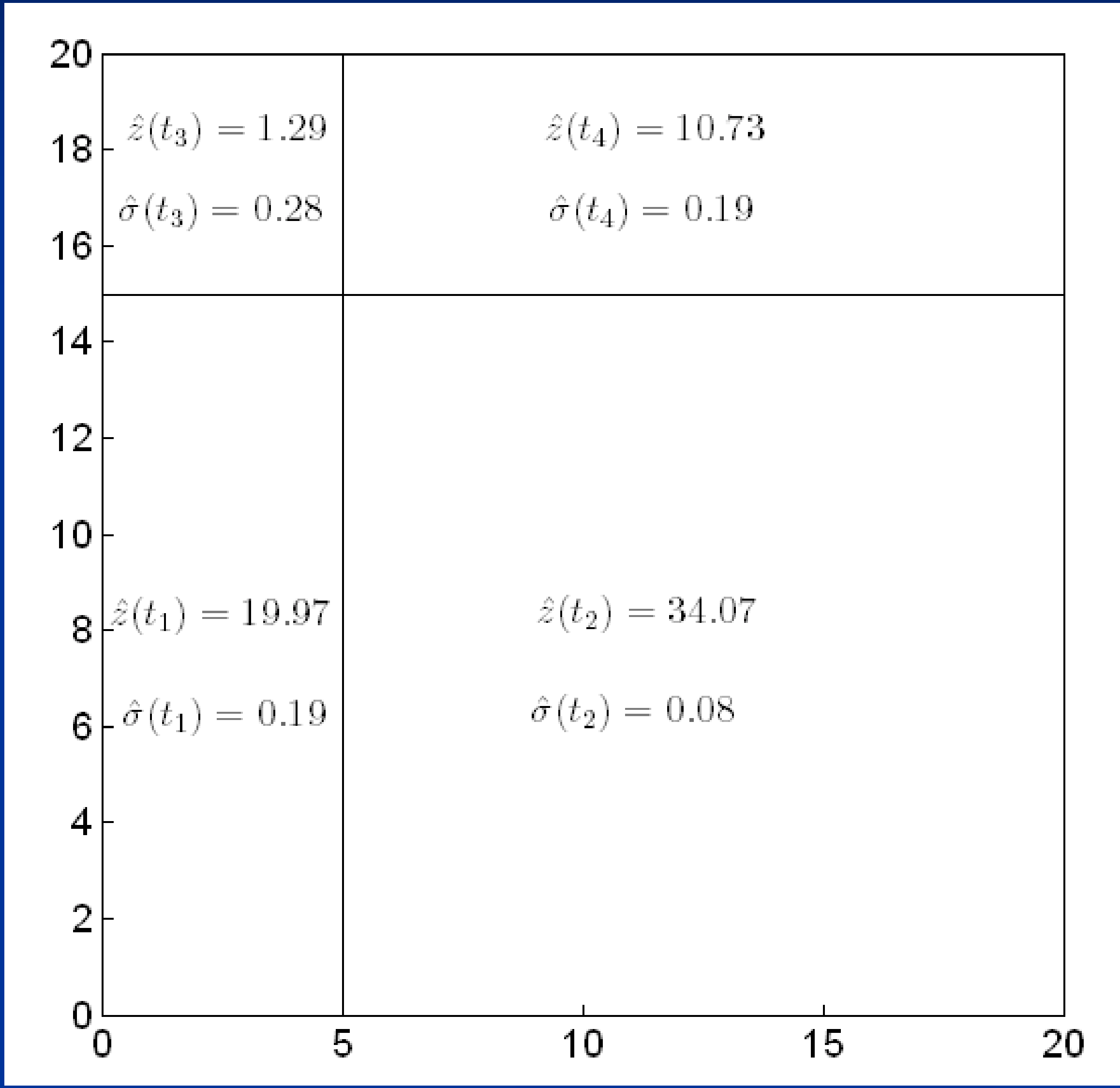


Spherical model

Unit sill

Range = 10

Gaussian semivariogram model



Gaussian model

Unit sill

Range = 20

Example results

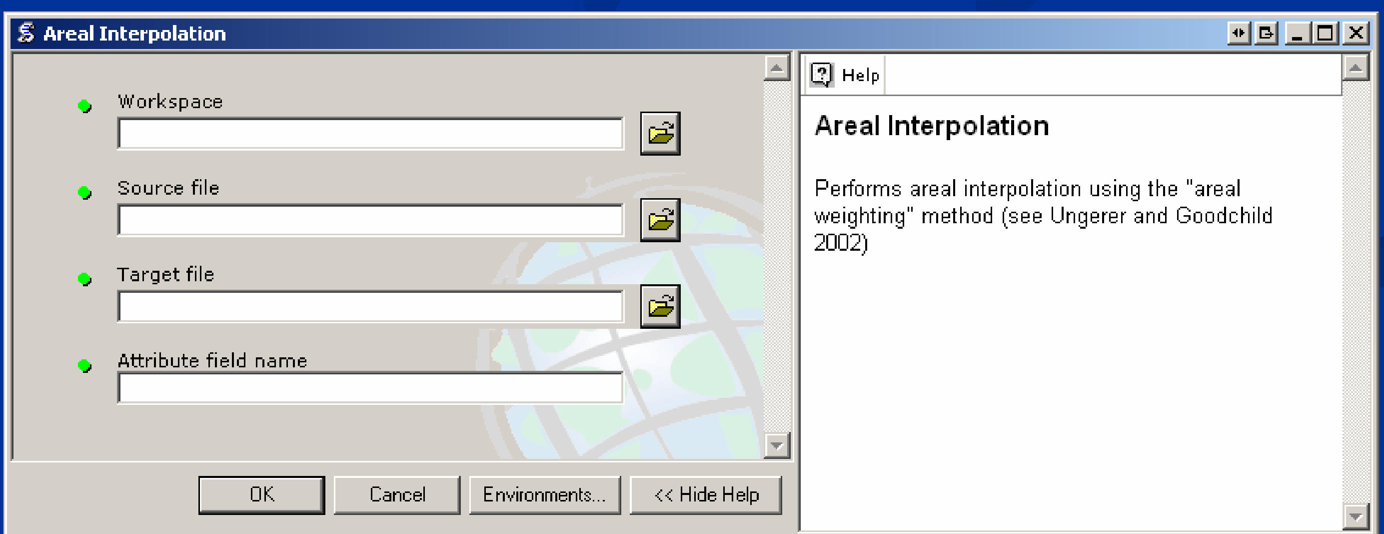
- "Geographic effect" more pronounced for smoother variogram models
- Prediction uncertainty is a function of target area
- Sum of product of source values and source areas equals sum of product of target predictions and target areas

$$\sum_{p=1}^{P} \hat{z}(t_p) \cdot |t_p| = \sum_{k=1}^{K} z(s_k) \cdot |s_k|$$

Tasks in progress and future work

- Possible extensions of the framework
  - Accounting for auxiliary variables (e.g. through Universal Kriging or CoKriging)
  - Accounting for non–gaussian data
  - Accounting for uncertainty in source data
- Implement the support-to-support interpolation in ArcGIS 9
  - Scripting language Python provides a suitable programming environment

```
#Import standard library modules
import win32com.client, sys, os
#Create the Geoprocessor object
gp = win32com.client.Dispatch("es

#Set the input workspace
gp.workspace = sys.argv[1]

#Set the clip feature class
clipFeatures = sys.argv[2]

#Set the output workspace
out_Workspace = sys.argv[3]

#Set the cluster tolerance
clusterTolerance = sys.argv[4]

try:
    #Get a list of the feature cl
    fcs = gp.ListFeatureClasses()
```



Conclusions

- A general framework for geostatistical support-to-support interpolation was presented
- The approach shows several desirable properties
- Existing methods for areal interpolation can be derived as particular cases of the proposed framework
  - Areal weighting
  - Dasymetric mapping
  - Tobler's pycnophylactic method
- Future work will focus on extensions of the framework and implementation