Spatial Data Analysis

Week 4: Geostatistics II

Meredith Franklin

Department of Statistical Sciences and School of the Environment

September 26th, 2025 and October 3, 2025

Geostatistical Data II

- Kriging (Simple, Ordinary, Universal)
- Spatial regression with covariance functions
- Inverse Distance Weighting
- Spline smoothing

Kriging

Typically, the final stage of a classical geostatistical analysis is to predict values of Z(s) at unobserved locations in $D \in \mathbb{R}^2$. Recall that unobserved locations could be specific x,y locations, or all locations (i.e. an infinitely small grid) in the domain. The classical spatial prediction method used for this purpose is called Kriging. Kriging:

- is a technique for spatial prediction.
- uses the estimated spatial process model to predict values at unobserved locations.
- is based on the fitted covariance function and the spatial regression model $Z(s) = \mu + \epsilon(s)$ where $\mu = X\beta$ can be $X\boldsymbol{\beta}$ and $\epsilon(s) \sim N(0, \Sigma(\theta))$

Objective of kriging: To estimate the value of $Z(s)$ at one or more unobserved locations in our region D based on our observed samples $z(s_1), z(s_2), ..., z(s_n)$

Kriging Recipe

- The basic kriging recipe:
  1. Choose a parametric model for the semivariance (covariance) function.
  2. stimate semivariogram (or covariance function) parameters (e.g., by WLS, ML, or REML).
  3. Gather a set of unobserved locations (points or grid).
  4. Make predictions and uncertainty estimates given the parameter estimates.
- The kriging predictions are weighted averages of the observations. The covariance/semivariogram governs the weights.
- The issue is how heavily to weight the observations based on distance from the location.

Kriging Theory: Best linear unbiased predictor (BLUP)

For an unobserved spatial location, s_0 we want to estimate $\hat{Z}(s_0)$. Kriging gives us the **BLUP** at any new location s_0. It is BLUP because:

- the prediction variance $E[\hat{Z}(s_0) - Z(s_0)]^2$ is minimized ("best").
- it is a linear prediction based on a weighted average of the observations ("linear"). The weighted average is $\hat{Z}(s_0) = \sum_{i=1}^{N} \omega_i Z(s_i)$
- the expected value of the prediction $E[\hat{Z}(s_0)]$ is equal to the expected value at s_0 ("unbiased"). The unbiased condition $E[\hat{Z}(s_0)] = E[Z(s_0)]$ implies that $\sum_i \omega_i = 1$. The condition $\sum_i \omega_i = 1$ ensures unbiasedness when the mean is constant, and also ensures non-negative and finite variance.

Kriging Theory

- Goal is to minimize squared error $E[(\hat{Z}(s_0) - Z(s_0))^2]$ subject to the unbiasedness constraint $\sum_{i=1}^{n} \omega_i = 1$. This ensures that the weighted sum of observations equals the true value on average.
- The problem boils down to finding the set of coefficients $\omega_1, ..., \omega_n$ that serve as the weights of our observations.
- The Lagrange multiplier method is used for finding the minimum subject to this constraint.
- This calculation assumes we know the moments, i.e. the mean $\mu(s)$ and the semivariance/covariance (or we have estimated it) $\gamma(h)$ or $C(h)$.

Kriging Theory

Mathematically,

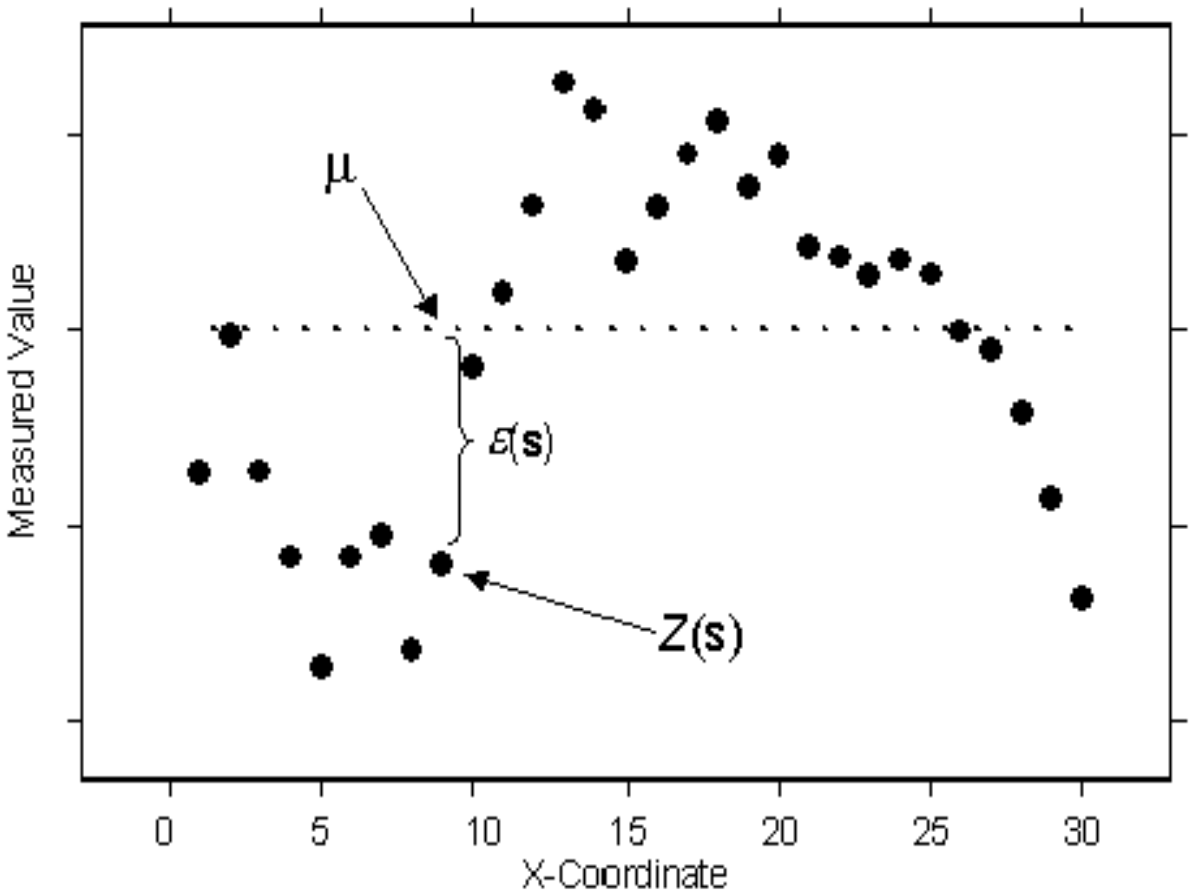
- $\hat{Z}(s_0) = \sum_{i=1}^{N} \omega_i Z(s_i)$
- $E[\hat{Z}(s_0)] = \mu = E[Z(s_0)]$
- with constraint $\sum_{i=1}^{N} \omega_i = 1$
- Minimize $E[(\hat{Z}(s_0) - Z(s_0))^2]$
- See derivations in handout on Quercus.

Kriging Types

There are different kriging types for different assumptions and analytical goals. The types are:

- Simple Kriging: assumes a known global mean $\mu = c$, which is often unrealistic so it is not commonly used. Furthermore, for the unbiasedness constraint to be applicable in kriging equations, we must estimate the expected value.
- Ordinary Kriging: assumes a constant unknown mean (mean needs to be estimated) $Z(s) = \mu + \epsilon(s)$.
- Universal Kriging: assumes a trend in x and y, and may include other spatially varying covariates $Z(s) = \mu(s) + \epsilon(s)$ where $\mu(s) = \sum_{k=1}^{p} \beta_k x_k(s)$.

Ordinary Kriging



Ordinary Kriging Equations

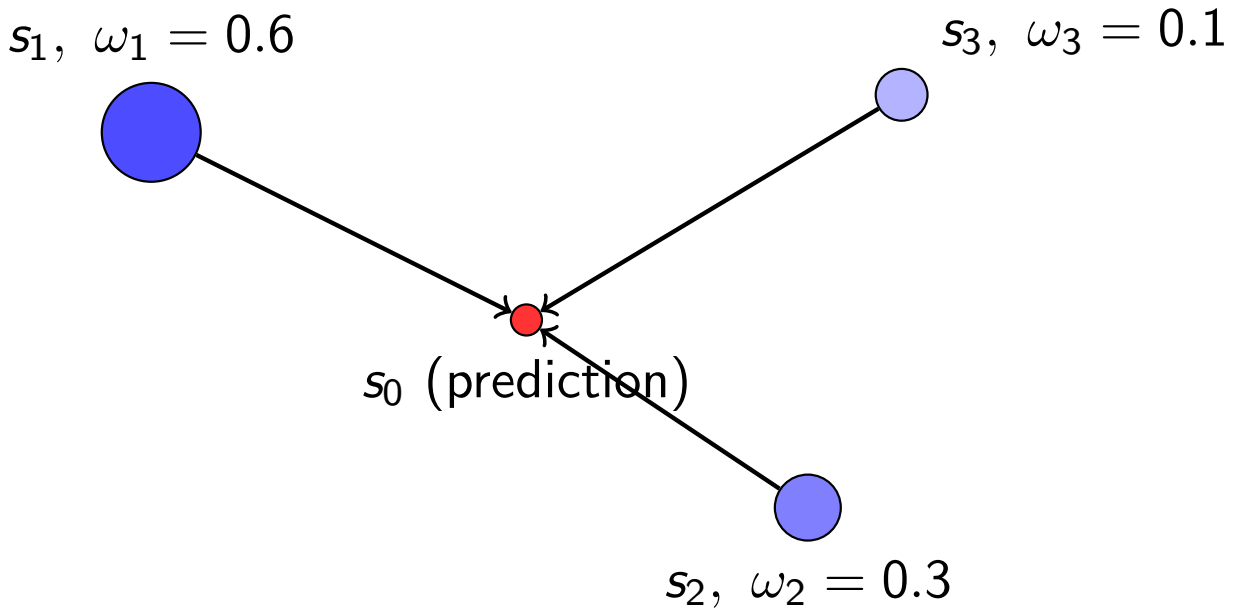
From the Lagrange multiplier approach, the kriging equations are $\sum_j \omega_j C_{ij} + \lambda = C_{i0}$ where C_{i0} and C_{ij} are evaluated based on our data and chosen covariance function.

$$\begin{pmatrix} \omega_1 \\ \omega_2 \\ \vdots \\ \omega_N \\ \lambda \end{pmatrix} = \begin{bmatrix} C_{11} & C_{12} & \dots & C_{1N} & 1 \\ C_{21} & C_{22} & \dots & C_{2N} & 1 \\ \vdots & \vdots & \dots & \vdots & \vdots \\ C_{N1} & C_{N2} & \dots & C_{NN} & 1 \\ 1 & 1 & \dots & 1 & 0 \end{bmatrix}^{-1} \times \begin{pmatrix} C_{10} \\ C_{20} \\ \vdots \\ C_{N0} \\ 1 \end{pmatrix}$$

This is the kriging system of equations of N+1 equations with N+1 unknowns; λ is the Lagrange multiplier enforcing unbiasedness. Solving for the weights, $\boldsymbol{\omega} = \mathbf{C}^{-1}\mathbf{D}$.

Note that $\mathbf{C}$ only needs to be estimated once, but $\mathbf{D}$ is found for every prediction location s_0. $\mathbf{C}$ is the covariance matrix among observed locations, $\mathbf{D}$ is the vector of covariances between s_0 and the observed locations

Kriging Weights Schematic



Weights decrease with distance:
$\omega_1 > \omega_2 > \omega_3, \quad \sum_i \omega_i = 1$

Nearby observations receive larger weights in kriging; distant ones contribute less.

Ordinary Kriging Variance

The variance of the prediction $\hat{Z}(s_0)$ is important as a measure of uncertainty in our predictions. It is obtained by the MSE:

$$MSE = \sum_{i=1}^{N}\sum_{j=1}^{N}\omega_i\omega_j Cov[Z(s_i), Z(s_j)] + Var[Z(s_0)]$$
$$-2Cov[\sum_{i=1}^{N}\omega_i Z(s_i), Z(s_0)]$$

which results in

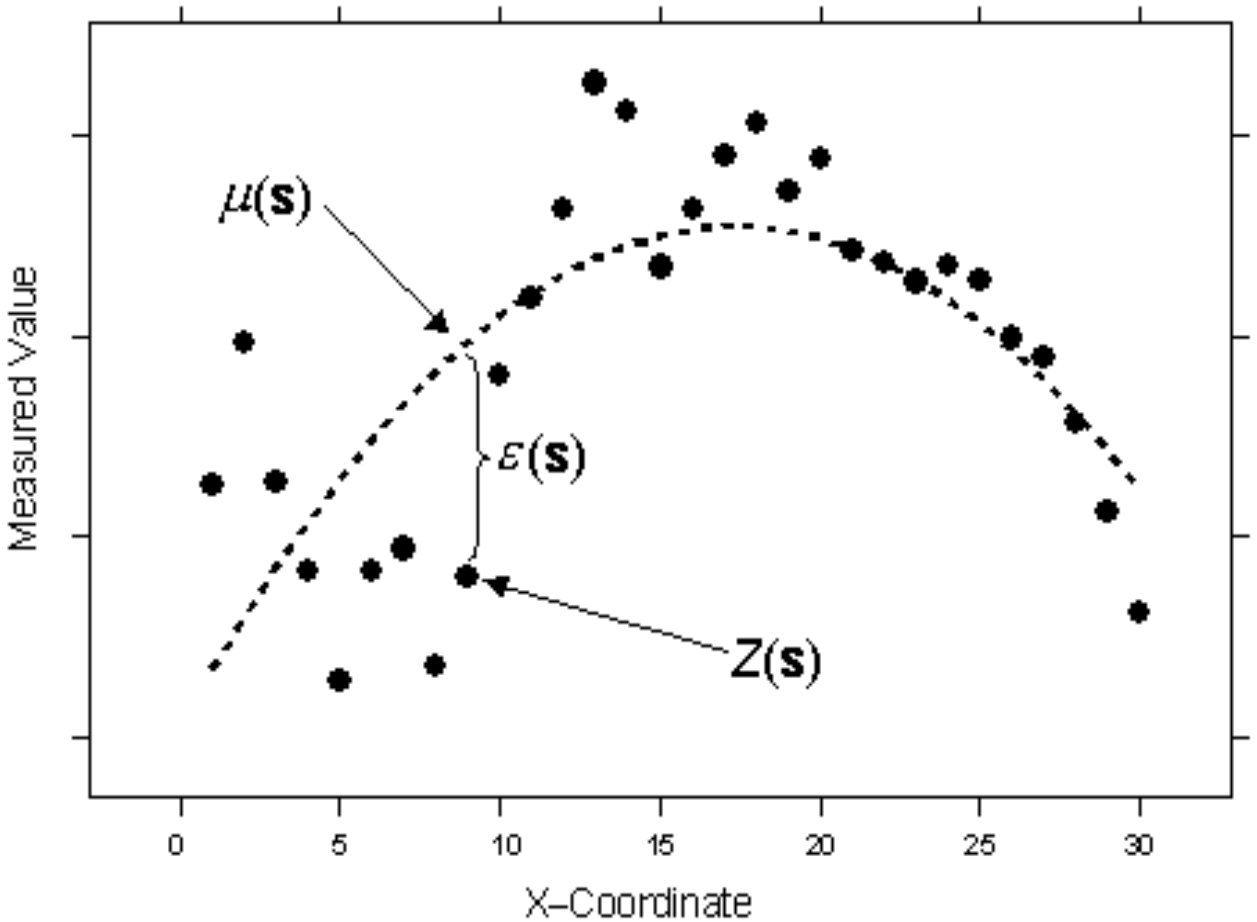
$$\sigma^2_{OK} = \sigma^2 - \sum_{i=1}^{N}\omega_i, Cov[Z(s_i), Z(s_0)] - \lambda$$

and in matrix form is

$$\sigma^2_{OK} = \sigma^2 - \boldsymbol{w}'\boldsymbol{D} + \lambda$$

w is the vector of weights and **D** is the covariance vector.

Universal Kriging



Universal Kriging

In universal kriging we don't assume that μ is estimated as a constant, but rather we assume a trend in x and y, and we may possibly include other spatially varying covariates. The general equation is $Z(s) = \mu(s) + \epsilon(s)$ where $\mu(s) = \sum_{k=1}^{p} \beta_k x_k(s_i)$.

- In addition to estimating the parameters of the spatially varying covariates, we have spatial correlation in the data that is not explained by the covariates.
- $Z(s) \sim \text{MVN}(X(s)\beta, \Sigma)$
- The term $X(s)\beta$ is referred to as the mean structure (often describing large scale variation or trend) and is distinguished from Σ which is the residual small-scale variation.
- The residual process adjusts the model for any residual spatial variation after accounting for the covariates.
- Universal kriging reduces to ordinary kriging when X contains only an intercept.
- Universal kriging is sometimes called spatial regression.

Universal Kriging

With $Z(s) \sim \text{MVN}(X\beta, \Sigma)$, under intrinsic stationarity Σ is our spatial covariance function $C(h)$ for spatial lags $h = \|s_i - s_j\|$, and $X\beta$ represents the $k = 1, \ldots, p$ covariates.

- In principle, spatially varying covariates can account for all large-scale variation in the response. However, in practice, residual spatial covariance usually remains and must be modeled via Σ.
- **Spatial confounding:** If covariates vary spatially at the same scale as the residual process, estimates of β can be biased and variance inflated.

Review of Regression (OLS)

- Recall the OLS estimator: $\hat{\beta}_{OLS} = (X^\top X)^{-1} X^\top Y$
- Here $Y(s_i)$ denotes regression responses, while $Z(s_i)$ denotes the spatial process in kriging. Both represent the process of interest.
- Variance estimate:

$$\hat{\sigma}^2_{OLS} = \frac{(Y - X\hat{\beta}_{OLS})^\top (Y - X\hat{\beta}_{OLS})}{n - p}$$

- Under correct model specification, $\hat{\beta}_{OLS}$ is unbiased and confidence intervals are valid.

Review of OLS Assumptions

- Variance of parameter estimates:

$$Var(\hat{\beta}_{OLS}) = \hat{\sigma}^2_{OLS}(X^\top X)^{-1}$$

- Assumes error variance-covariance matrix $\Sigma = \sigma^2 I$
- Assumes no correlation between errors.
- If residual spatial correlation exists, these assumptions are violated. $\hat{\beta}_{OLS}$ remains unbiased, but standard errors and inference are incorrect.

Solution 1: Generalized Least Squares

- With spatial covariance: $Cov(Y(s_i), Y(s_j)) \neq 0$
- Variance-covariance matrix: $\Sigma = Var(\epsilon) = \sigma^2 V$
- Generalized Least Squares (GLS) estimator:

$$\hat{\beta}_{GLS} = (X^\top \Sigma^{-1} X)^{-1} X^\top \Sigma^{-1} Y$$

- Variance estimate:

$$\hat{\sigma}^2_{GLS} = \frac{(Y - X\hat{\beta}_{GLS})^\top \Sigma^{-1} (Y - X\hat{\beta}_{GLS})}{n - p}$$

Generalized Least Squares

- Variance of parameter estimates:

$$Var(\hat{\beta}_{GLS}) = (X^\top \Sigma^{-1} X)^{-1}$$

- If OLS is used when spatial correlation is present, $\hat{\beta}_{OLS}$ is still unbiased but variance estimates are wrong.
- GLS is the **minimum variance linear unbiased estimator** (MVLUE), and thus more efficient than OLS.

Generalized Least Squares: Limitations

- We want to use GLS but the problem is we don't know Σ
- As before, we need to parameterize our spatial covariance, but we don't know the parameters!
- We need to estimate βs and Σ

There are two methods for approaching this problem:

1. Iteratively Re-weighted Generalized least squares
2. Maximum Likelihood (or Restricted Maximum Likelihood)

Solution 2a: Iteratively Re-weighted GLS

Steps:

1. Estimate $\hat{\beta}$ using OLS.
2. Compute residuals $r = Y - X\hat{\beta}$.
3. Fit covariance parameters $\theta = (\tau^2, \sigma^2, \phi)$ to $r \;\Rightarrow\; \Sigma(\hat{\theta})$.
4. Update $\hat{\beta}$ using GLS with $\Sigma(\hat{\theta})$.
5. Iterate steps 2–4 until convergence (e.g., $< 10^{-5}$).

Solution 2b: Maximum Likelihood

- Model: $Y \sim \text{MVN}(X\beta, \Sigma(\theta))$
- Likelihood:

$$L(\beta, \theta; Y) = (2\pi)^{-n/2}|\Sigma(\theta)|^{-1/2} \exp\left\{-\tfrac{1}{2}(Y - X\beta)^\top \Sigma(\theta)^{-1}(Y - X\beta)\right\}$$

- Log-likelihood:

$$\log L(\beta, \theta; Y) = -\tfrac{n}{2}\log(2\pi) - \tfrac{1}{2}\log|\Sigma(\theta)| - \tfrac{1}{2}(Y - X\beta)^\top \Sigma(\theta)^{-1}(Y - X\beta)$$

- $\Sigma(\theta)$ has parameters $\theta = (\tau^2, \sigma^2, \phi)$ (nugget, partial sill, range).

Maximum Likelihood (Profile)

- For fixed θ_0, the MLE of β is

$$\hat{\beta} = (X^\top \Sigma^{-1}(\theta_0) X)^{-1} X^\top \Sigma^{-1}(\theta_0) Y$$

- Plugging this in, the **profile log-likelihood** depends only on θ:

$$\log L(\theta; Y) = -\tfrac{1}{2} \log |\Sigma(\theta)| - \tfrac{1}{2} Y^\top P(\theta) Y$$

with

$$P(\theta) = \Sigma^{-1}(\theta) - \Sigma^{-1}(\theta) X (X^\top \Sigma^{-1}(\theta) X)^{-1} X^\top \Sigma^{-1}(\theta).$$

- Optimized numerically via Newton–Raphson or Fisher scoring.

$$\theta^{(k+1)} = \theta^{(k)} + (B^{(k)})^{-1} \nabla^{(k)}$$

where $B^{(k)}$ = information matrix (2nd derivatives), $\nabla^{(k)}$ = gradient (1st derivatives).

Progression: OLS $\rightarrow$ GLS $\rightarrow$ ML

OLS
Assumes $\Sigma = \sigma^2 I$ (independent errors).
Simple but ignores correlation.

GLS
Assumes known $\Sigma = \sigma^2 V$.
Accounts for correlation if Σ given.

ML / REML
Estimates β and covariance parameters θ.
Most general, but computationally heavier.

Hypothesis Testing and Model Comparison

- Likelihood ratio test (LRT): compare **full** vs. **reduced** model.

$$-2 \log \Lambda \sim \chi^2_q \quad (q = \text{df difference})$$

- For non-nested models: use information criteria

$$\text{AIC} = -2 \log L(\hat{\beta}, \hat{\theta}) + 2(p + m)$$

$$\text{BIC} = -2 \log L(\hat{\beta}, \hat{\theta}) + (p + m) \log n$$

- Smaller AIC or BIC $\Rightarrow$ better model.

Application: OK and UK

- Two equivalent approaches:
  1. **Detrend data**, fit residuals with Ordinary Kriging (OK).
  2. **Universal Kriging (UK)** directly.
- Detrending: estimate $\hat{\beta}_{GLS}$, then predict

$$\hat{Z}(s_0) = x(s_0)^\top \hat{\beta}_{GLS} + \hat{\epsilon}_{OK}(s_0).$$

- Predictions are the same if the covariance model is identical.
- BUT: prediction variances differ.
  - OK with detrended data ignores uncertainty in $\hat{\beta}_{GLS}$.
  - UK accounts for it.

Detrending + OK vs Universal Kriging

Detrend + OK

- Estimate $\hat{\beta}_{GLS}$ (trend).
- Krige residuals $\hat{\epsilon}(s)$ with OK.
- Prediction:

$$\hat{Z}(s_0) = x(s_0)^\top \hat{\beta}_{GLS} + \hat{\epsilon}_{OK}(s_0)$$

- Variance **ignores** $\hat{\beta}_{GLS}$ uncertainty.

Universal Kriging (UK)

- Estimate β and $\Sigma(\theta)$ jointly.
- Prediction:

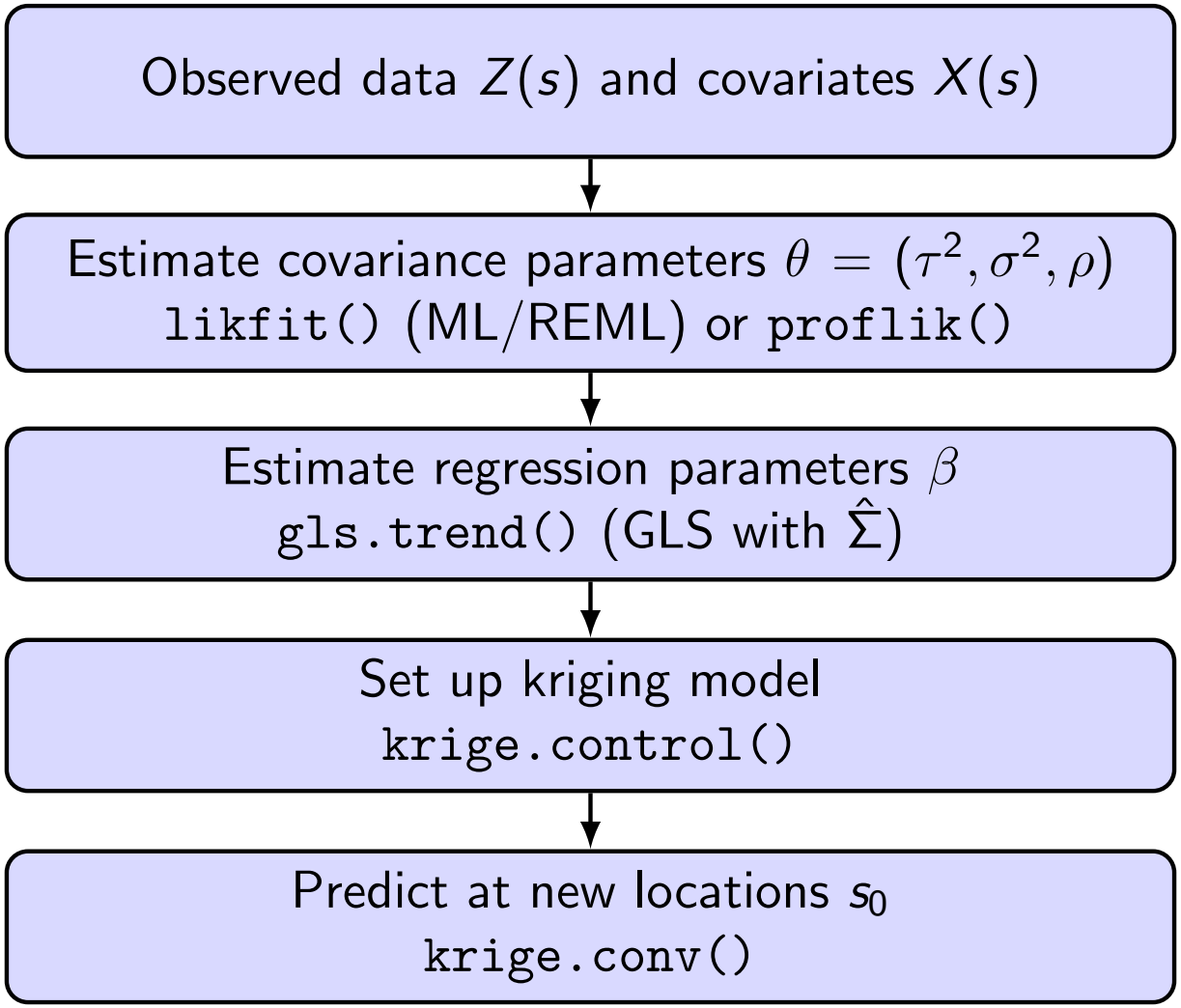
$$\hat{Z}(s_0) = x(s_0)^\top \hat{\beta} + \hat{\epsilon}_{UK}(s_0)$$

- Variance **includes** $\hat{\beta}$ uncertainty.

GLS and ML in geoR

- **Covariance parameter estimation**
  - `proflik()`: profile likelihood estimates of $\theta = (\tau^2, \sigma^2, \rho)$.
  - `likfit()`: ML or REML estimation of covariance parameters.
- **Regression parameter estimation**
  - OLS or GLS estimates of β can be obtained depending on whether Σ is taken as I or estimated from θ.
  - `gls.trend()` implements GLS estimation of the trend given covariance parameters.
- **Kriging prediction**
  - `krige.control()`: specify trend/covariates, covariance model, and pass fitted $\hat{\theta}$.
  - `krige.conv()`: compute predictions and variances at unobserved locations s_0.

Workflow in geoR: GLS / ML and Kriging



geoR workflow: Data $\rightarrow$ Covariance estimation $\rightarrow$ Regression (GLS/ML) $\rightarrow$ Kriging setup $\rightarrow$ Predictions.

Other Types of Kriging

- Primary kriging approaches:
  - Simple kriging (known mean)
  - Ordinary kriging (constant unknown mean)
  - Universal kriging (trend or covariates in mean)
- These can be viewed as point interpolation, producing a prediction surface (grid/raster).
- Other extensions exist for special goals:
  - **Cokriging**: multivariate kriging, exploits correlation between multiple spatial processes.
  - **Indicator kriging**: probability mapping of exceedances of a threshold (binary data).

Cokriging

- Cokriging is the **multivariate version** of kriging.
- Uses two or more spatial variables measured at (possibly different) locations.
- Goal: improve prediction of a variable that is sparsely observed by borrowing strength from a correlated variable that is more densely observed.
- Example: predict heavy metal concentrations using both sparse soil samples and abundant remote sensing data.
- Exploits **cross-covariance functions** to reduce prediction variance.

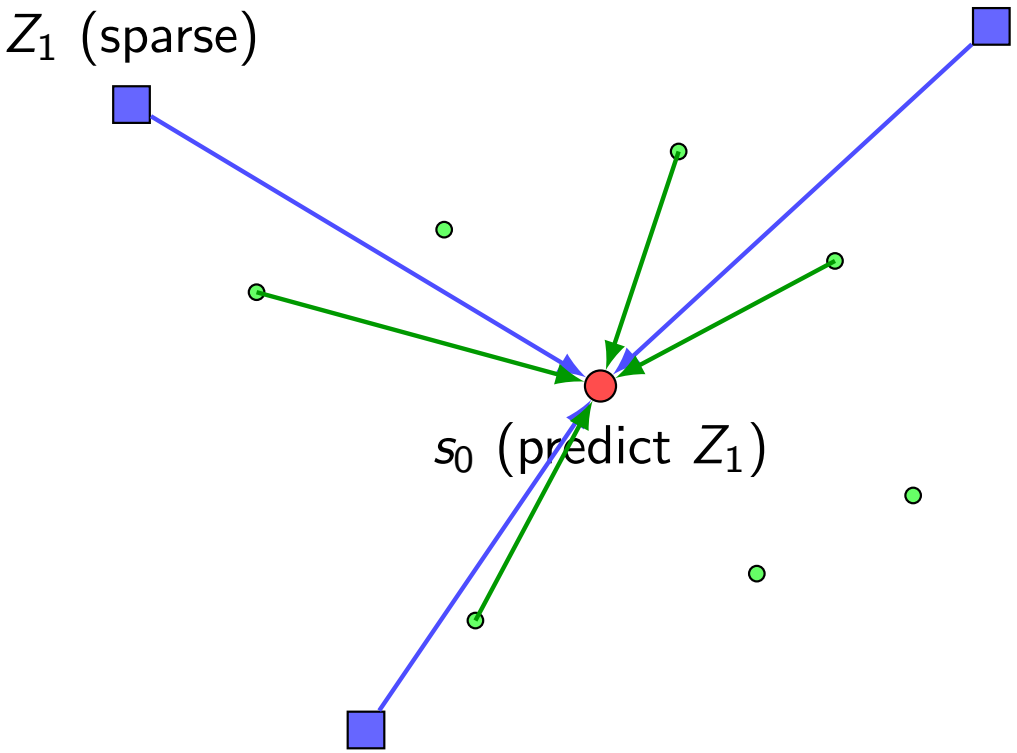
Cokriging & Cross-Variograms

- **Cokriging**: multivariate kriging that uses two (or more) correlated spatial variables to improve prediction of one of them.
- Requires modeling both **auto-variograms** and **cross-variograms**:

$$\gamma_{11}(h) = \tfrac{1}{2} Var\big(Z_1(s+h) - Z_1(s)\big) \qquad \text{(auto for } Z_1\text{)}$$
$$\gamma_{22}(h) = \tfrac{1}{2} Var\big(Z_2(s+h) - Z_2(s)\big) \qquad \text{(auto for } Z_2\text{)}$$
$$\gamma_{12}(h) = \tfrac{1}{2} E\big[(Z_1(s+h) - Z_1(s))(Z_2(s+h) - Z_2(s))\big] \qquad \text{(cross)}$$

- Valid models require that the covariance/variogram matrix formed from $\gamma_{11}(h), \gamma_{22}(h), \gamma_{12}(h)$ is **positive semidefinite** for all h.
- Prediction at s_0 combines observed values of both variables, with weights determined by these auto- and cross-variograms.
- Implemented in R via `gstat` (not `geoR`); see Bivand, Pebesma, & Gómez-Rubio (Applied Spatial Data Analysis with R, Ch. 8).

Cokriging



Z_1 (squares) and Z_2 (circles) are measured at different locations,
but both are georeferenced to the same spatial domain.
Cokriging uses auto-variograms $\gamma_{11}(h)$, $\gamma_{22}(h)$ and cross-variogram $\gamma_{12}(h)$
to borrow strength from Z_2 and improve prediction of Z_1 at s_0.

Indicator Kriging

- Used when interest is in the **probability** of exceeding a threshold, not the expected value.
- Transform original data $Z(s)$ into indicators:

$$I(Z(s) > z_0) = \begin{cases} 1 & \text{if } Z(s) > z_0 \\ 0 & \text{otherwise} \end{cases}$$

- Kriging these indicators yields

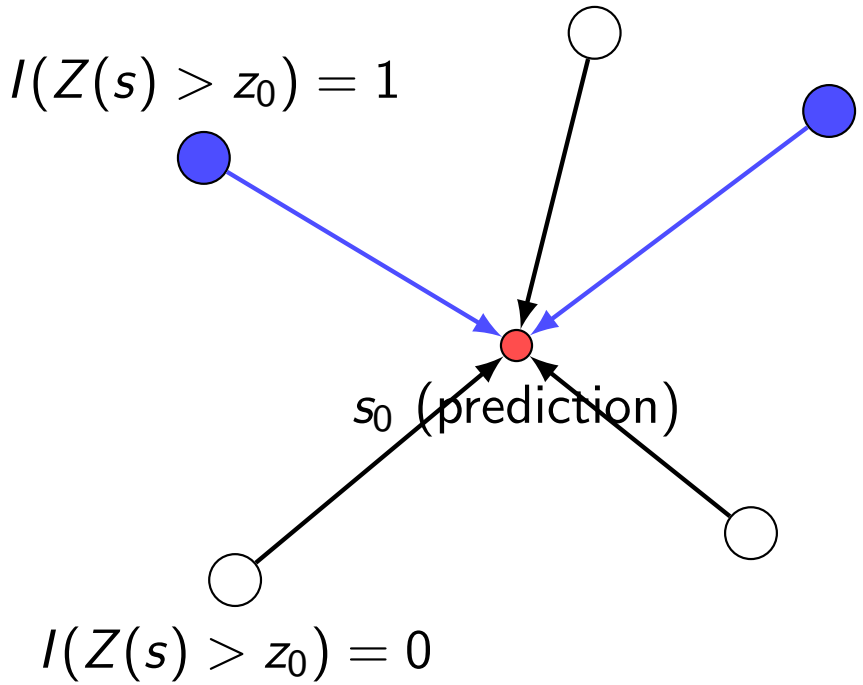
$$\hat{I}(s_0) = \sum_{i=1}^{n} \lambda_i I(Z(s_i) > z_0)$$

- $\hat{I}(s_0)$ is an estimate of $P(Z(s_0) > z_0 \mid \text{data})$.

Indicator Kriging: Interpretation

- Produces a **probability map** rather than a mean prediction surface.
- Useful for:
  - Mapping risk of contamination above regulatory thresholds.
  - Probability of soil nutrient deficiency or disease exceedance.
- Advantage: makes no assumption about Gaussianity of the underlying variable.
- Limitation: predictions may be less smooth and harder to interpret as continuous fields.

Indicator Kriging



Continuous data $Z(s)$ transformed to indicators $I(Z(s) > z_0) \in \{0, 1\}$.
Kriging these indicators gives $\hat{I}(s_0) \approx P(Z(s_0) > z_0 \mid \text{data})$.
Result: a spatial probability map of exceedance.

Interpolation & Smoothing

Other types of spatial prediction (non-statistical)

- **Inverse Distance Weighting (IDW)** Deterministic interpolation: weighted average of nearby data.
- Kernel smoothing Local weighted averaging with a kernel function.
- Loess smoothing Local polynomial fitting.
- **Spline smoothing** Smooth surface that minimizes curvature.

Inverse Distance Weighting (IDW)

- A simple deterministic interpolation method.
- Prediction at s_0 is a weighted average of observations:

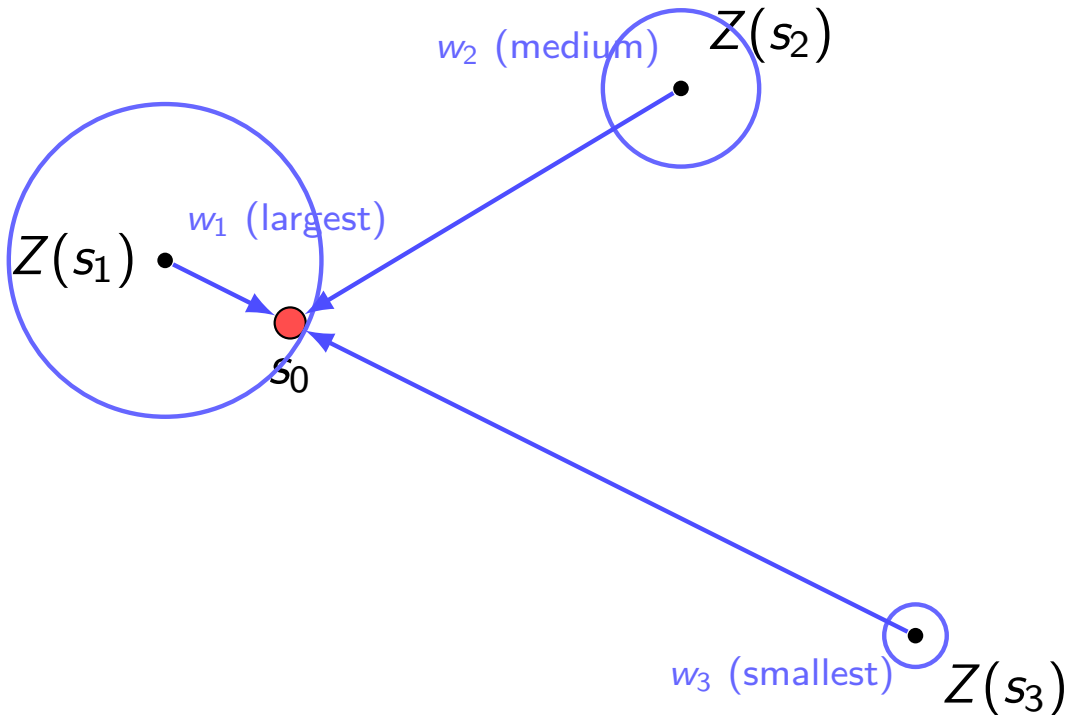
$$\hat{Z}(s_0) = \frac{\sum_i Z(s_i)\, d(s_i, s_0)^{-p}}{\sum_i d(s_i, s_0)^{-p}}$$

- Weights decrease with distance: closer points matter more.
- The parameter p controls smoothness:
  - Small p: distant points still influence $\hat{Z}(s_0) \Rightarrow$ smoother surface.
  - Large p: prediction dominated by nearest neighbors $\Rightarrow$ surface shows local variability.

IDW: Practical Considerations

- **Power parameter** p:
  - $p = 1$: smoother interpolation, distant points still contribute.
  - $p = 3$: moderate localization.
  - $p = 12$: very localized, predictions close to nearest observation.
- **Neighborhood definition**:
  - Can restrict to k nearest neighbors or points within a radius.
  - Prevents distant points from contributing unrealistically.
- **Limitations**:
  - Sensitive to clustering of observations.
  - Does not account for spatial trends or anisotropy.
  - No formal uncertainty quantification.

IDW

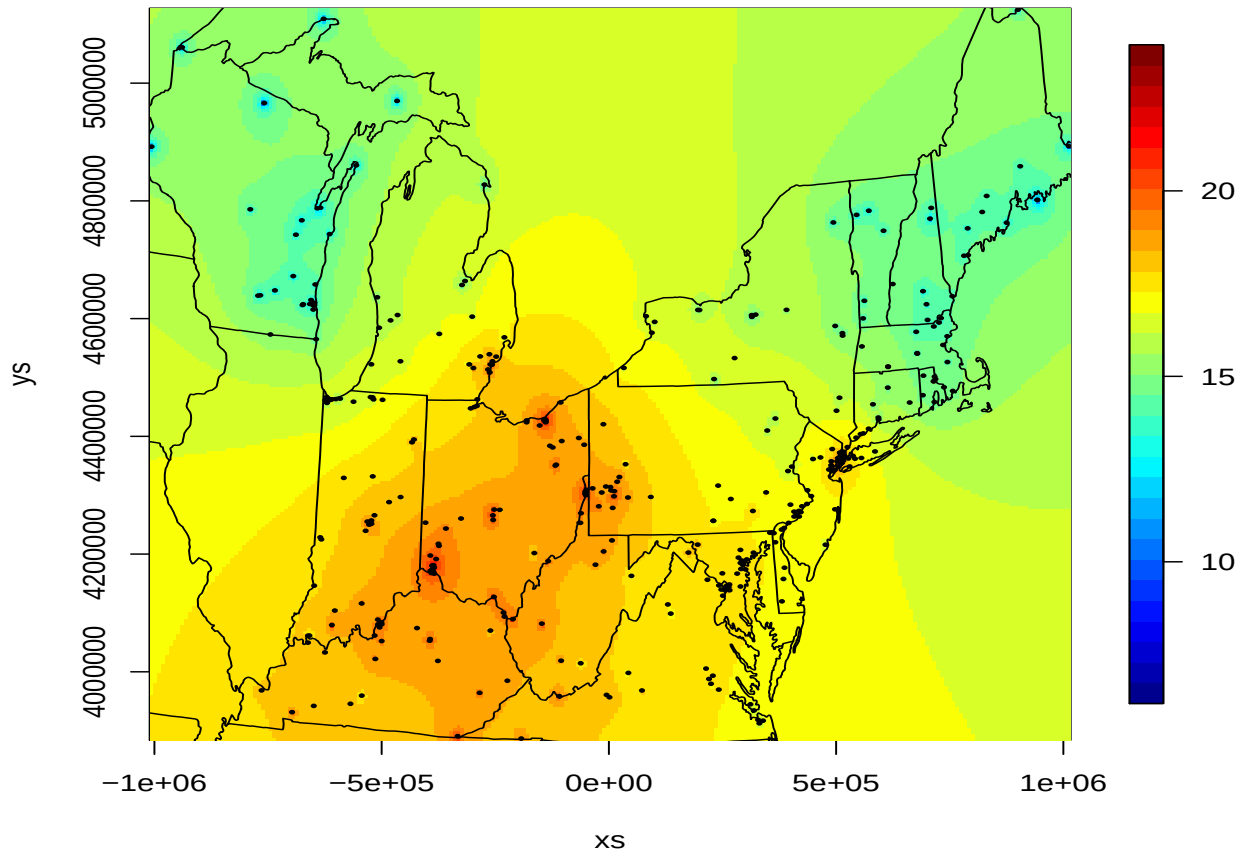


Weights: $w_i = d(s_i, s_0)^{-p} \Big/ \sum_j d(s_j, s_0)^{-p}$

Closer points $\Rightarrow$ much larger weights.

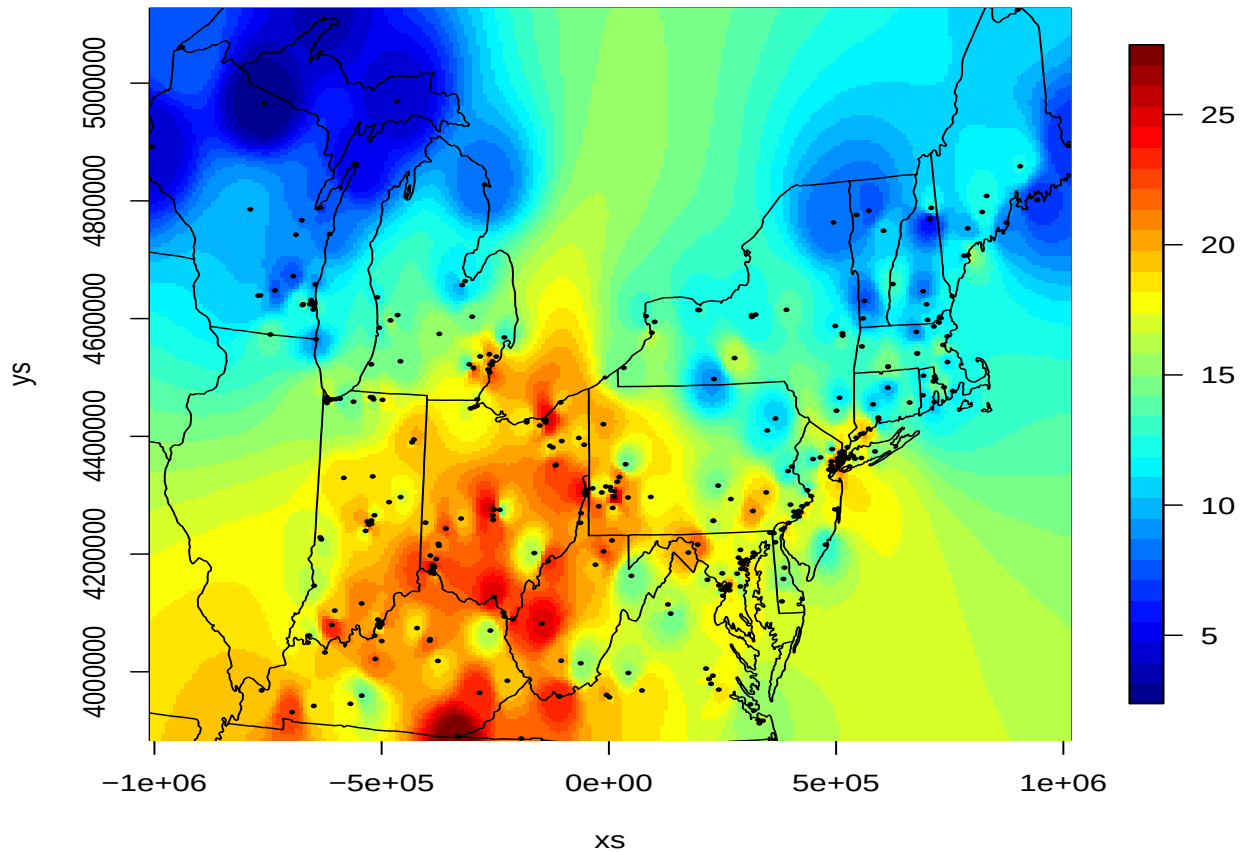
Here: $w_1 > w_2 > w_3$.

IDW Example: $p = 1$



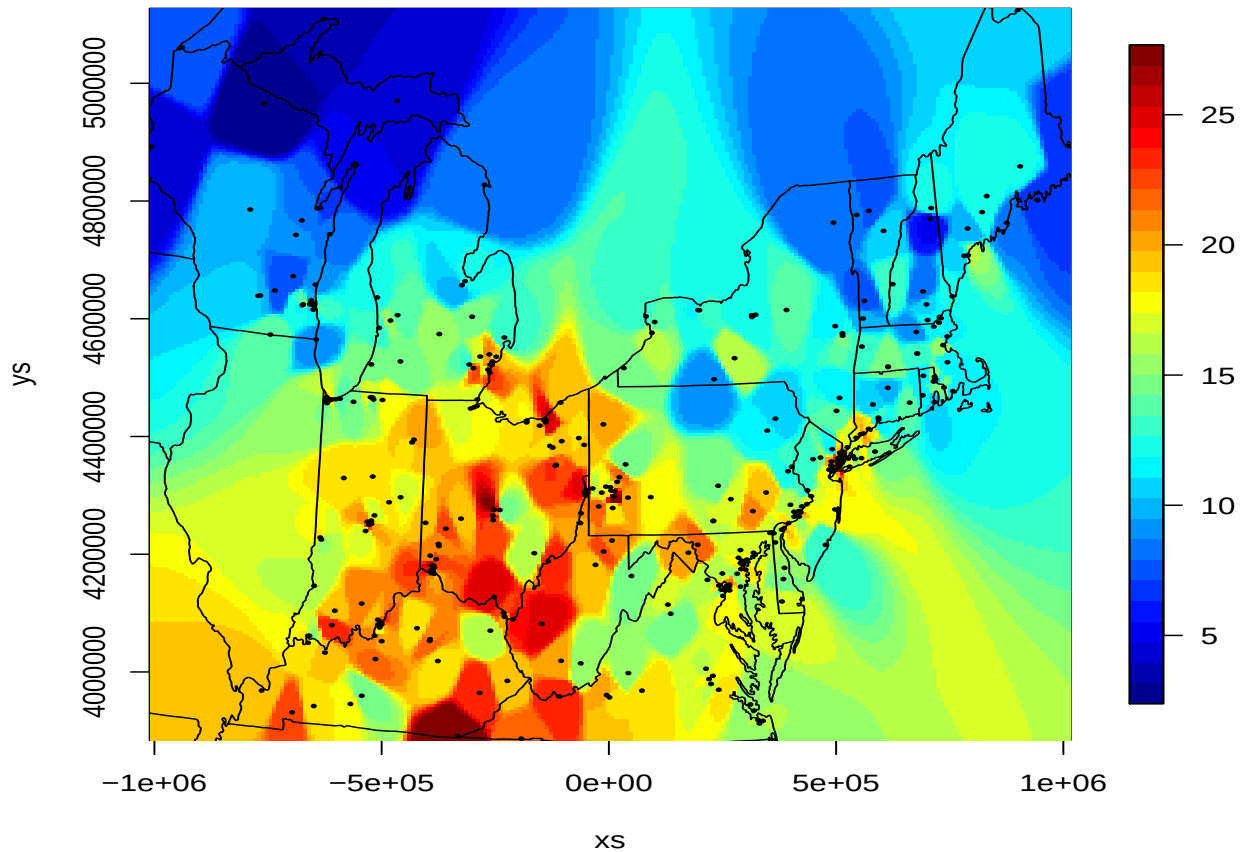
Smooth surface: distant points still have influence.

IDW Example: $p = 3$



Intermediate case: balance of smoothness and locality.

IDW Example: $p = 12$



Highly localized: prediction dominated by nearest neighbor(s).

IDW vs. Kriging

- IDW:
  - Deterministic: no model of spatial correlation.
  - Simple and fast to compute.
  - No measure of uncertainty.
- Kriging:
  - Statistical: weights derived from estimated variogram/covariance.
  - Provides prediction *and* prediction variance.
  - More flexible, but requires model assumptions and fitting.
- Use IDW when simplicity or speed is the goal, kriging when statistical inference is needed.

Why Splines?

- Kriging: models spatial correlation via the covariance function.
- Splines: model spatial structure in the **mean function** instead.
- General setup:

$$Z(s) = f(s) + \epsilon, \quad \epsilon \sim N(0, \sigma^2 I)$$

- Goal: estimate a smooth function $f(s)$ that captures spatial trend.
- Benefits:
  - Flexible, nonparametric modeling of spatial trend.
  - Still yields prediction errors (statistical smoothing).
  - Naturally extended to generalized additive models (GAMs).

Basis Functions: Intuition

- A **basis** is a set of simple functions $\{b_j(x)\}$ that can be combined (with coefficients β_j) to approximate more complex $f(x)$:

$$f(x) = \sum_{j=1}^{k} \beta_j b_j(x).$$

- Example: polynomials $b_1(x) = 1,\ b_2(x) = x,\ b_3(x) = x^2, \ldots$
- Analogy: like combining Lego blocks to build different shapes — basis functions are the blocks, coefficients β_j are how much of each block we use.
- The regression coefficients β_j control the contribution of each basis function.

Basis Functions: Intuition

- Represent a complicated function $f(x)$ as a linear combination of simpler **basis functions**.

$$f(x) = \sum_{j=1}^{k} \beta_j b_j(x)$$

- Example: polynomial basis in 1-D

$$y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \beta_3 x_i^3 + \beta_4 x_i^4 + \epsilon_i$$

- Basis functions:

$$b_1(x) = 1,\ b_2(x) = x,\ b_3(x) = x^2,\ b_4(x) = x^3,\ b_5(x) = x^4$$

Basis Functions: Matrix Form

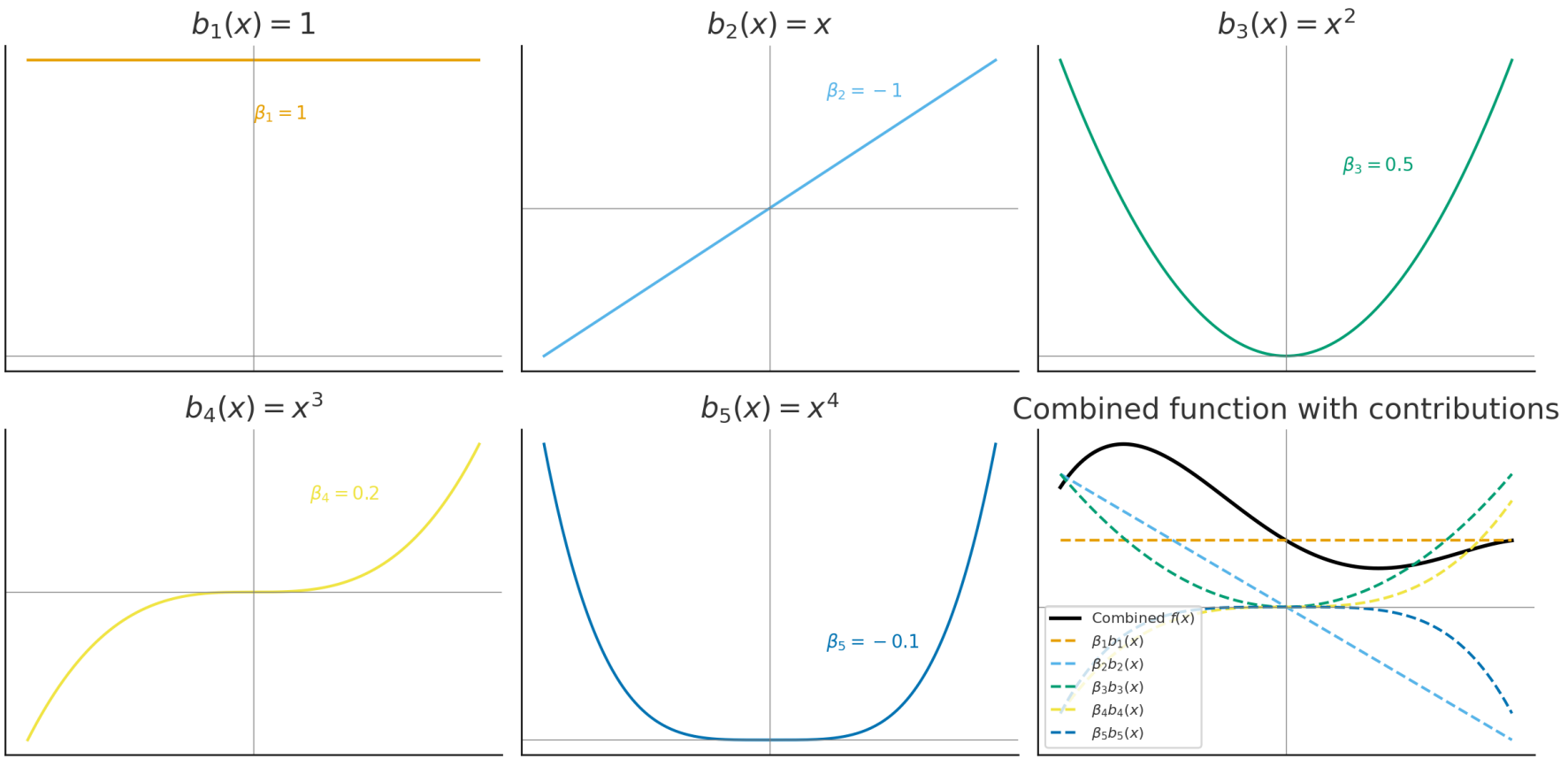
- Collect basis functions into a **basis matrix B**.

$$\mathbf{f} = \mathbf{B}\boldsymbol{\beta}$$

- Example (polynomial basis):

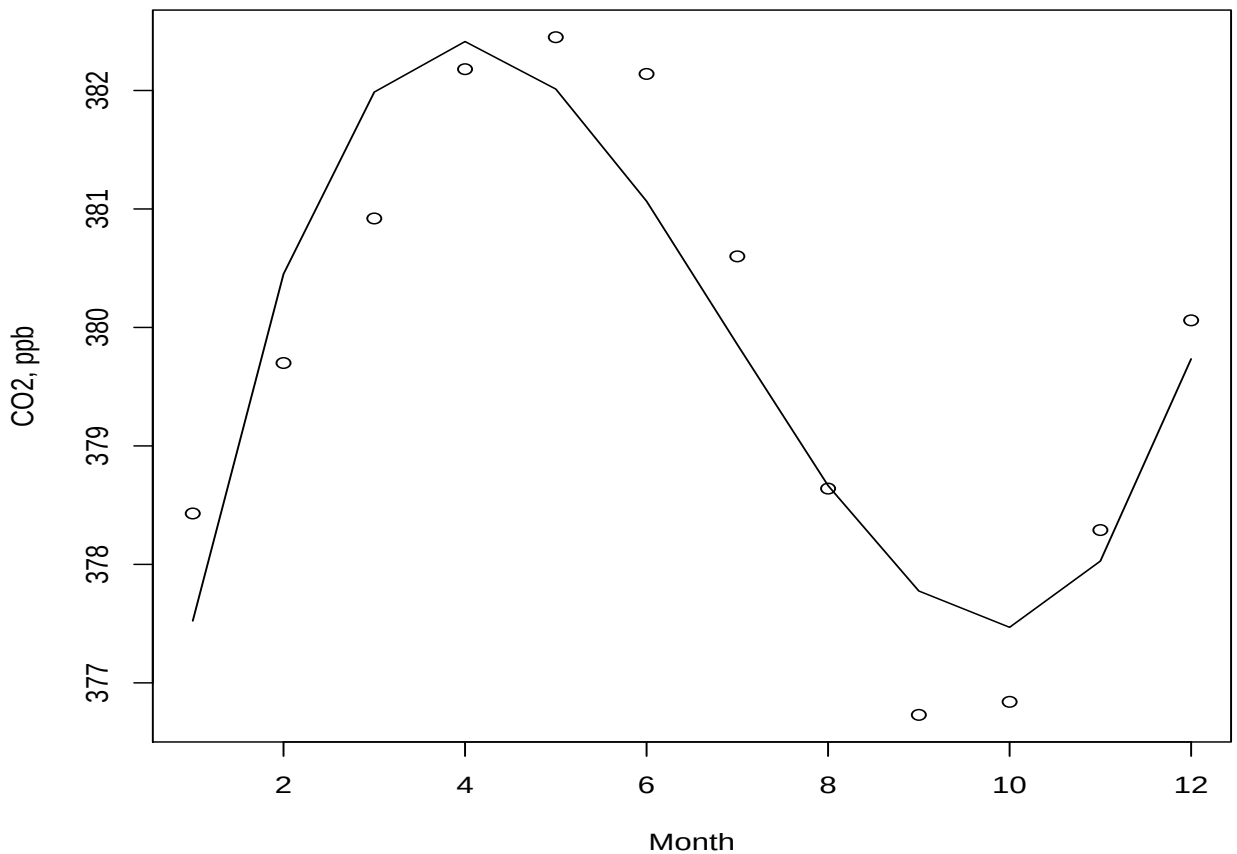
$$\begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_n) \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_1^2 & x_1^3 & x_1^4 \\ 1 & x_2 & x_2^2 & x_2^3 & x_2^4 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & x_n & x_n^2 & x_n^2 & x_n^4 \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \beta_3 \\ \beta_4 \end{bmatrix}$$

Polynomial Basis



Polynomial Basis

The basis functions are each multiplied by β_j and then summed to give the final curve $f(x)$. In the previous slide, this is shown in the bottom left figure. Below, we show this concept in terms of an example of CO_2 concentrations over a year (monthly data).



Polynomial Basis Recap

- A function $f(x)$ can be written as a weighted sum of polynomial basis functions:

$$f(x) = \sum_{j=0}^{p} \beta_j x^j$$

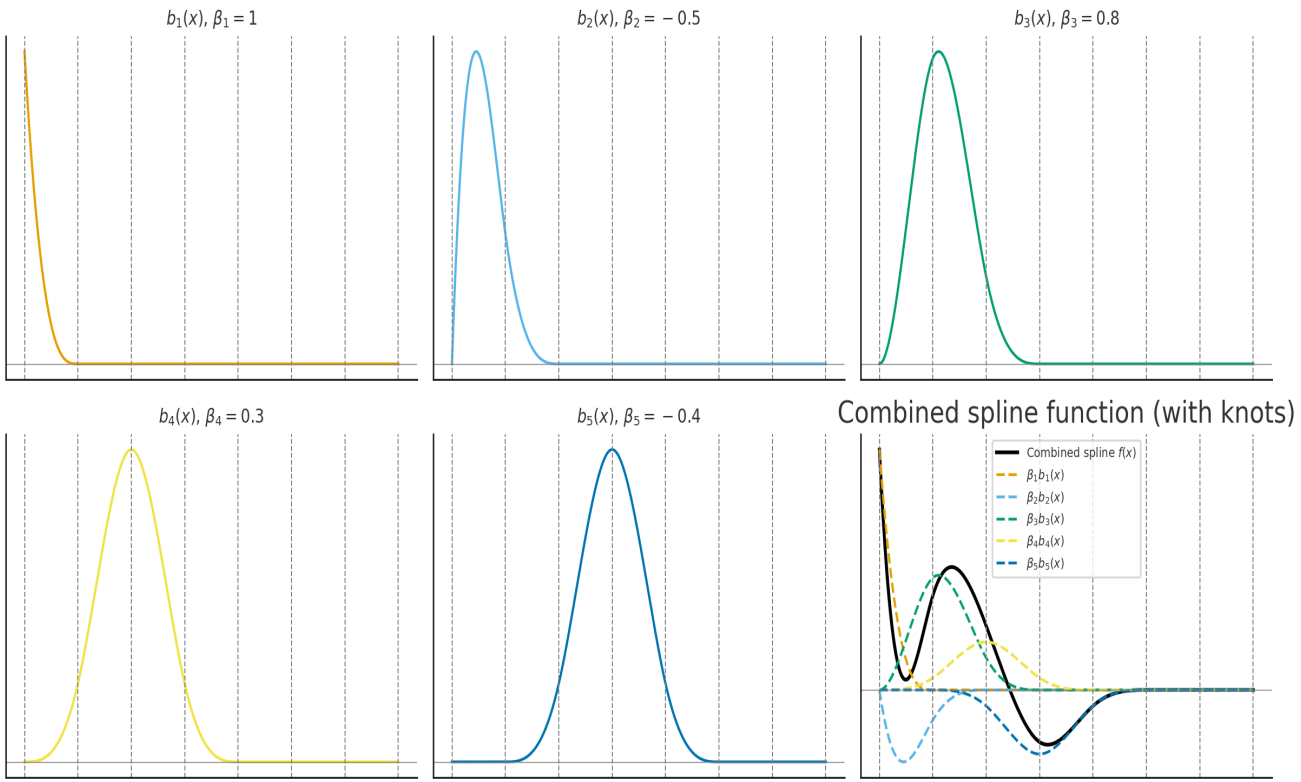
- The coefficients β_j control the shape of the curve.
- Polynomials provide flexibility, but they can behave poorly:
  - High-order polynomials become unstable ("wiggly").
  - Poor fit near boundaries (Runge's phenomenon).
- This motivates using **splines**: piecewise polynomials joined smoothly at **knots**.

Splines: Concepts & Types

- A **spline** is a function made up of **piecewise polynomials**.
  - Defined between **knots** (points where polynomial pieces join).
  - Joined smoothly (continuous first and second derivatives).
- Advantages over high-order polynomials:
  - Flexible but stable (avoids oscillations at boundaries).
  - **Local control**: changing one knot only affects the nearby region.
- Types of splines:
  - **Cubic splines**: piecewise cubic polynomials joined smoothly.
  - **Natural splines**: cubic splines with linear constraints at the boundaries (reduces edge wiggles).
  - **B-splines**: basis representation of splines using local polynomial pieces.
  - **Smoothing splines**: knots at every observation, with a penalty λ to control wiggliness.
- General form:

$$f(x) = \sum_{j=1}^{K} \beta_j b_j(x), \quad b_j(x) \text{ are spline basis functions.}$$

Cubic Splines Example



Choosing the number and placement of knots controls smoothness:

- Few knots $\Rightarrow$ smoother curve.
- Many knots $\Rightarrow$ wiggly curve.

Smoothing Splines (Concept)

- Goal: Fit a smooth curve $f(t)$ through noisy data (t_i, y_i).
- We want to balance:
  - **Goodness of fit**: match the data closely.
  - **Smoothness**: avoid overly wiggly functions.
- Define an objective function:

$$\text{RSS}(f) + \lambda J(f)$$

where

$$\text{RSS}(f) = \sum_i (y_i - f(t_i))^2, \quad J(f) = \int (f''(t))^2 \, dt$$

- $\lambda \geq 0$ is the **smoothness parameter**.
  - Small λ: fit the data very closely (wiggly curve).
  - Large λ: force f to be smoother (nearly linear).

Smoothing Splines (Optimization)

- We minimize the penalized residual sum of squares:

$$\min_f \left\{ \sum_i (y_i - f(t_i))^2 + \lambda \int (f''(t))^2 \, dt \right\}$$

- This is a trade-off between:
  - $\sum_i (y_i - f(t_i))^2$ = data fidelity
  - $\int (f''(t))^2 dt$ = roughness penalty
- Solution: a **natural cubic spline** with knots at each unique t_i.
- The "best" curve is obtained by penalizing the wiggliness of the spline.

Smoothing Splines (Basis Representation)

- Represent $f(t)$ using spline basis functions:

$$f(t) = \sum_{j=1}^{K} \beta_j b_j(t)$$

- Let B be the basis matrix, S the penalty matrix:

$$B_{ij} = b_j(t_i),$$
$$S_{jk} = \int b_j''(t) b_k''(t)\, dt$$

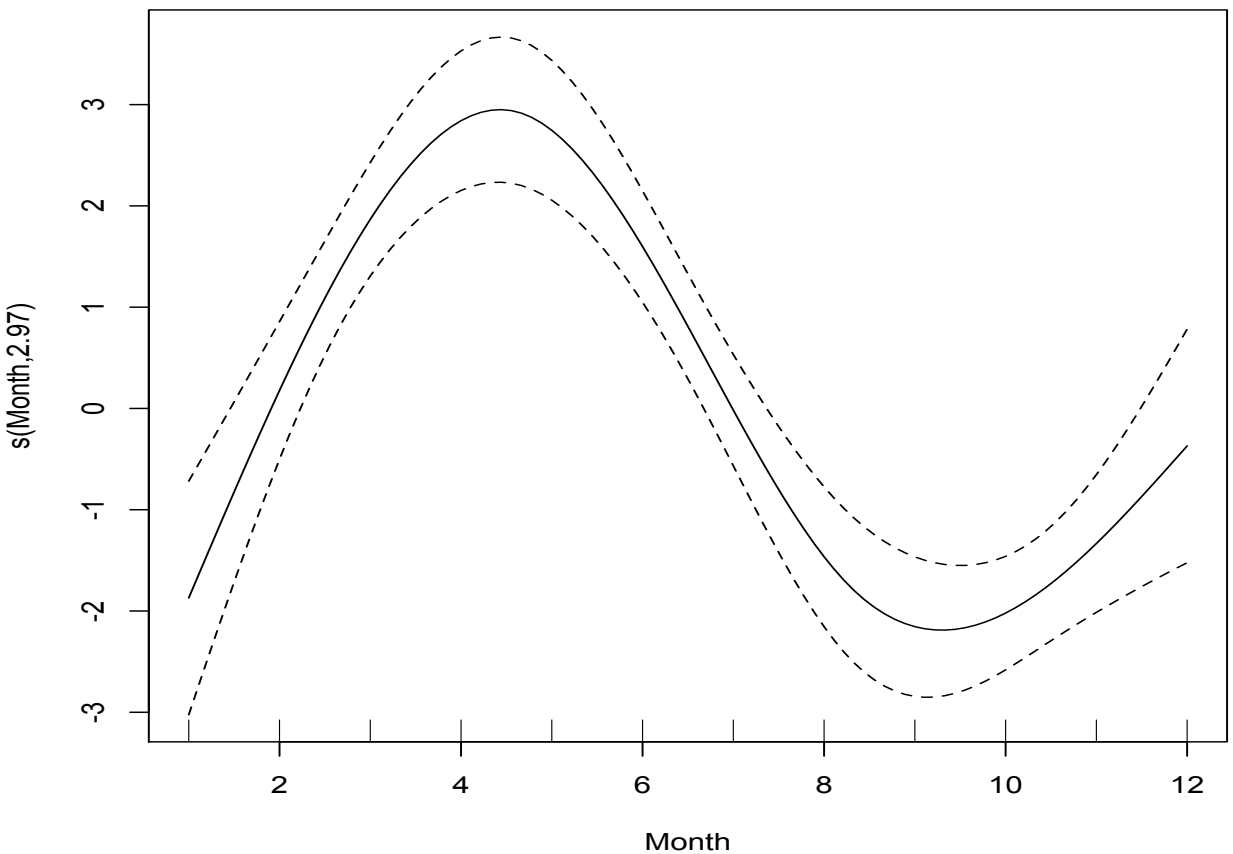
- The optimization problem becomes:

$$\min_{\beta} (y - B\beta)^\top (y - B\beta) + \lambda \beta^\top S \beta$$

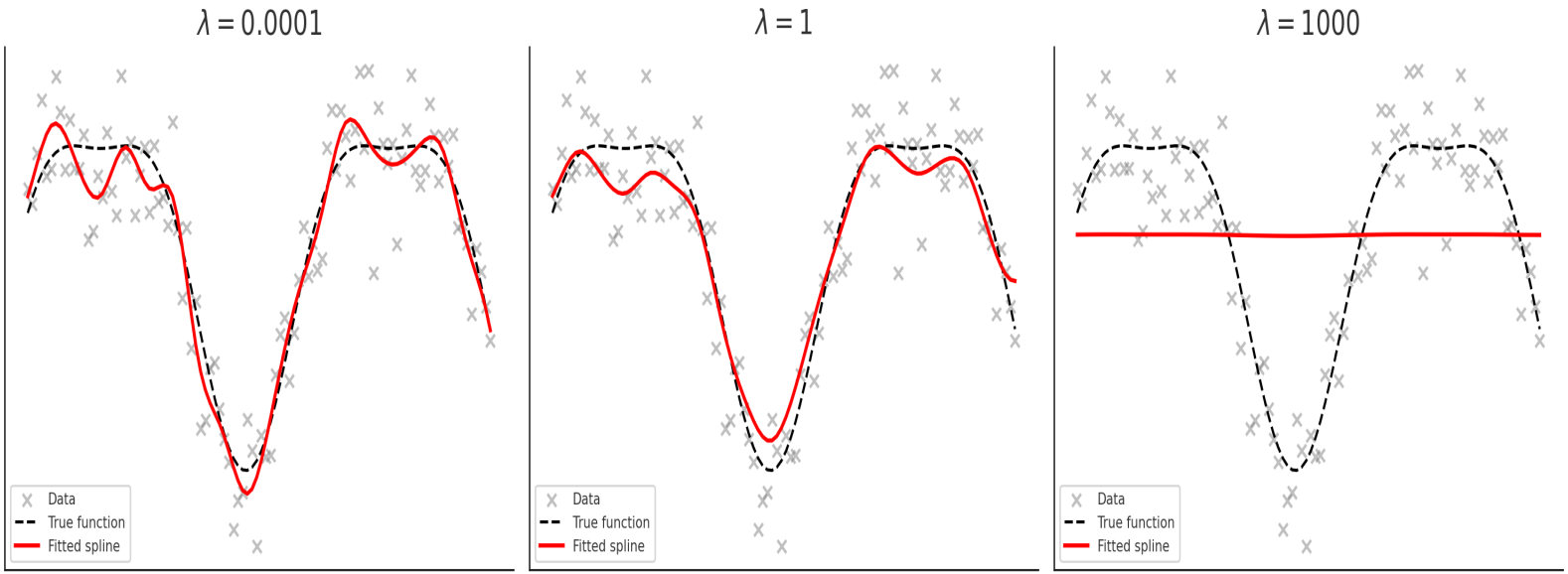
- Solution:

$$\hat{\beta} = (B^\top B + \lambda S)^{-1} B^\top y$$

Splines



Splines

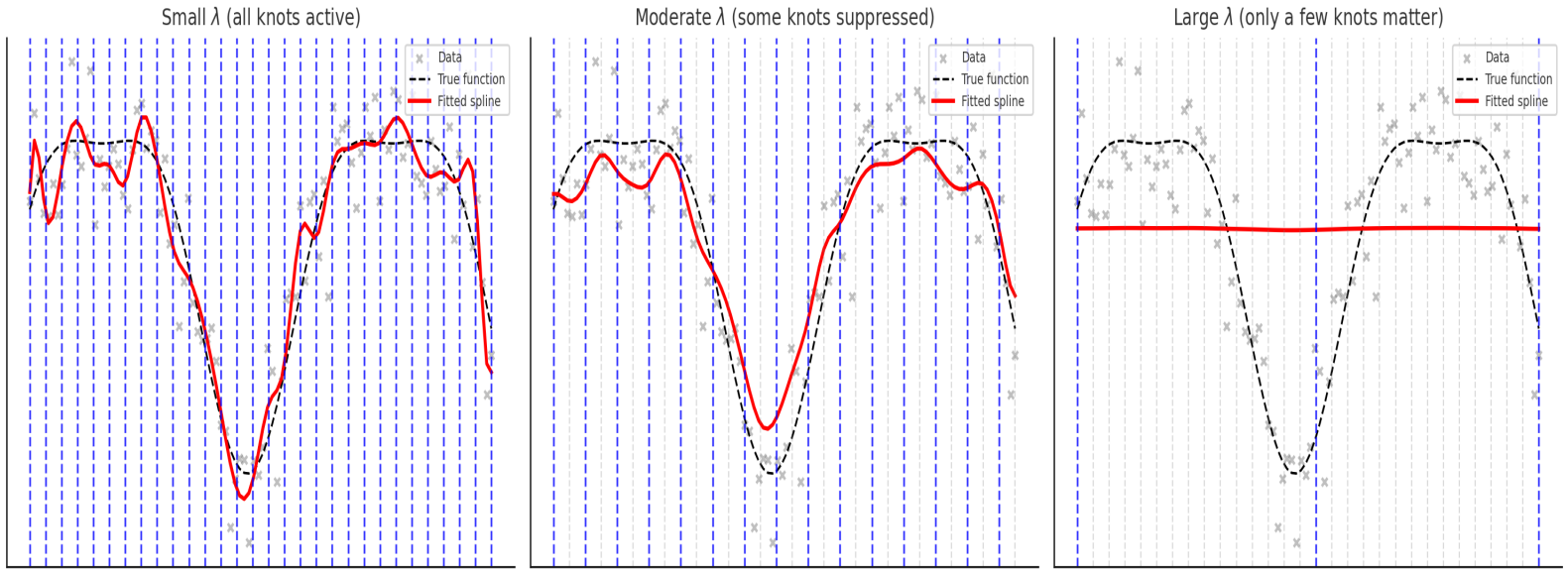


- Left (λ very small $=$ 1e-4): spline is very wiggly, almost interpolates the noise.
- Middle ($\lambda = 1$): good balance between smoothness and fit.
- Right (λ very large $=$ 1000): curve is almost straight (linear trend).

Knots vs. Penalty Parameter λ

- In smoothing splines:
  - A knot is placed at each observation t_i.
  - This gives the spline the potential to fit the data exactly.
- The penalty parameter λ controls how much we use that flexibility:
  - $\lambda \to 0$: little penalty $\Rightarrow$ spline interpolates the data (very wiggly).
  - $\lambda \to \infty$: heavy penalty $\Rightarrow$ spline approaches a straight line.
- Interpretation:
  - **Knots**: define the "vocabulary" of possible wiggles.
  - λ: decides how much of that vocabulary is actually used.

Knots vs. Penalty Parameter λ



- Small λ: all knots are active (blue), spline wiggles to fit data closely.
- Moderate λ: only some knots are active (blue), others are suppressed (grey), leading to a smoother curve.
- Large λ: only a few key knots remain active, spline is very smooth (almost linear).

Splines

Types of 1-D splines include:

- cubic splines (basically piecewise cubic polynomials)
- cyclic splines (cubic splines with connected ends)
- basis splines (B-spline) with other polynomial orders
- cardinal splines (where knot placement is always a certain distance)
- wavelets (often cardinal wavelet splines)

From 1-D to 2-D Splines

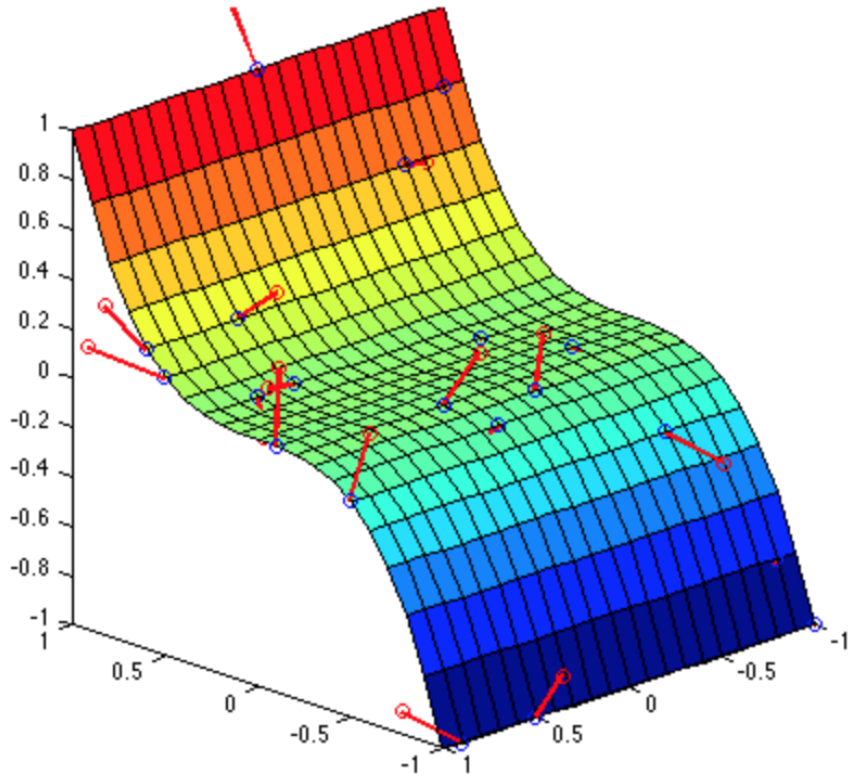
- Recall in 1-D smoothing splines we minimize:

$$\sum_i (y_i - f(x_i))^2 + \lambda \int (f''(x))^2 dx$$

- First term = **fit to data**, second = **roughness penalty**.
- In 2-D, we want to model a smooth surface $g(s_1, s_2)$ over space.
- Idea: extend the penalty to include curvature in both directions.

Thin Plate Splines (TPS)

- A thin plate spline is the natural 2-D generalization of a smoothing spline.
- Think of placing a bendable plate over the data points $Z(s)$.
- Points exert an upward/downward pull, like weights on the plate.
- The spline solution balances fidelity to data vs. bending energy.



Thin Plate Spline Penalty

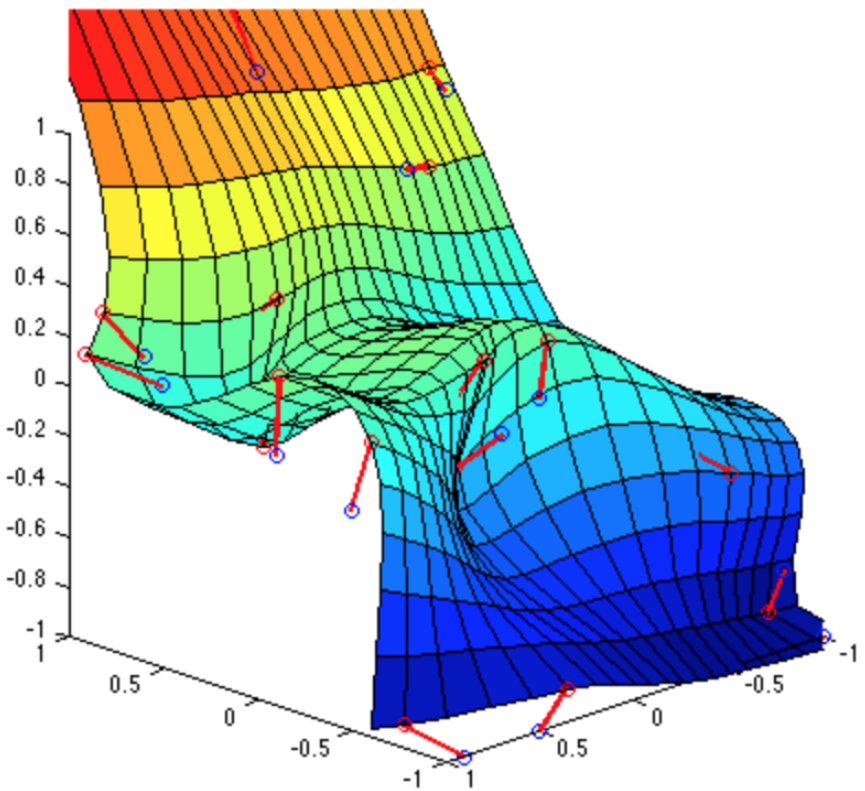
We minimize:

$$\sum_i \left(z_i - g(s_{i1}, s_{i2})\right)^2 + \lambda \iint \left[\left(\frac{\partial^2 g}{\partial s_1^2}\right)^2 + 2\left(\frac{\partial^2 g}{\partial s_1 \partial s_2}\right)^2 + \left(\frac{\partial^2 g}{\partial s_2^2}\right)^2 \right] ds_1\, ds_2$$

- The integral is the **bending energy** of the surface.
- λ controls trade-off: large λ = smoother, flatter surface.
- Without penalty ($\lambda = 0$), the surface interpolates exactly.

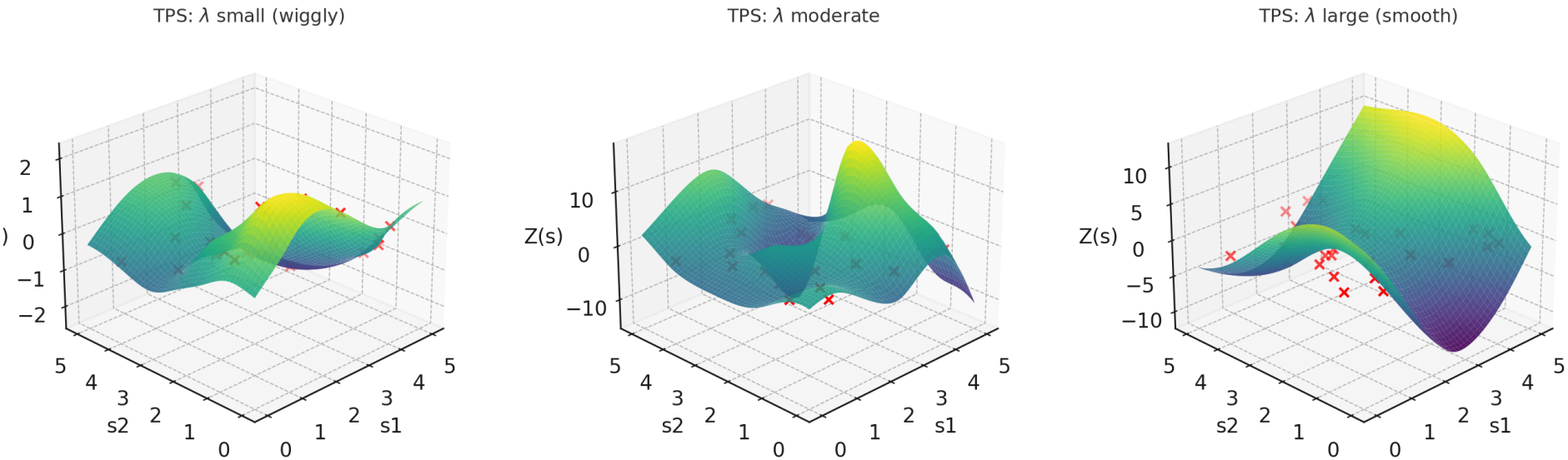
Thin Plate Splines: Interpretation

- Surface is pulled most where data are dense.
- Many knots $\Rightarrow$ wiggly surface, few knots $\Rightarrow$ smoother.
- Predictions extend smoothly to unsampled areas.
- Provides both estimates and standard errors (unlike IDW).



Thin Plate Splines and the Penalty Parameter λ

- In 2-D, λ controls the **smoothness** of the surface:
  - Small λ: very flexible, surface follows the data closely.
  - Moderate λ: balances fit and smoothness.
  - Large λ: surface is very smooth, close to a plane.



Effect of λ on thin plate spline smoothing. Left: small λ (wiggly). Middle: moderate λ. Right: large λ (smooth).

Thin Plate Splines in Practice

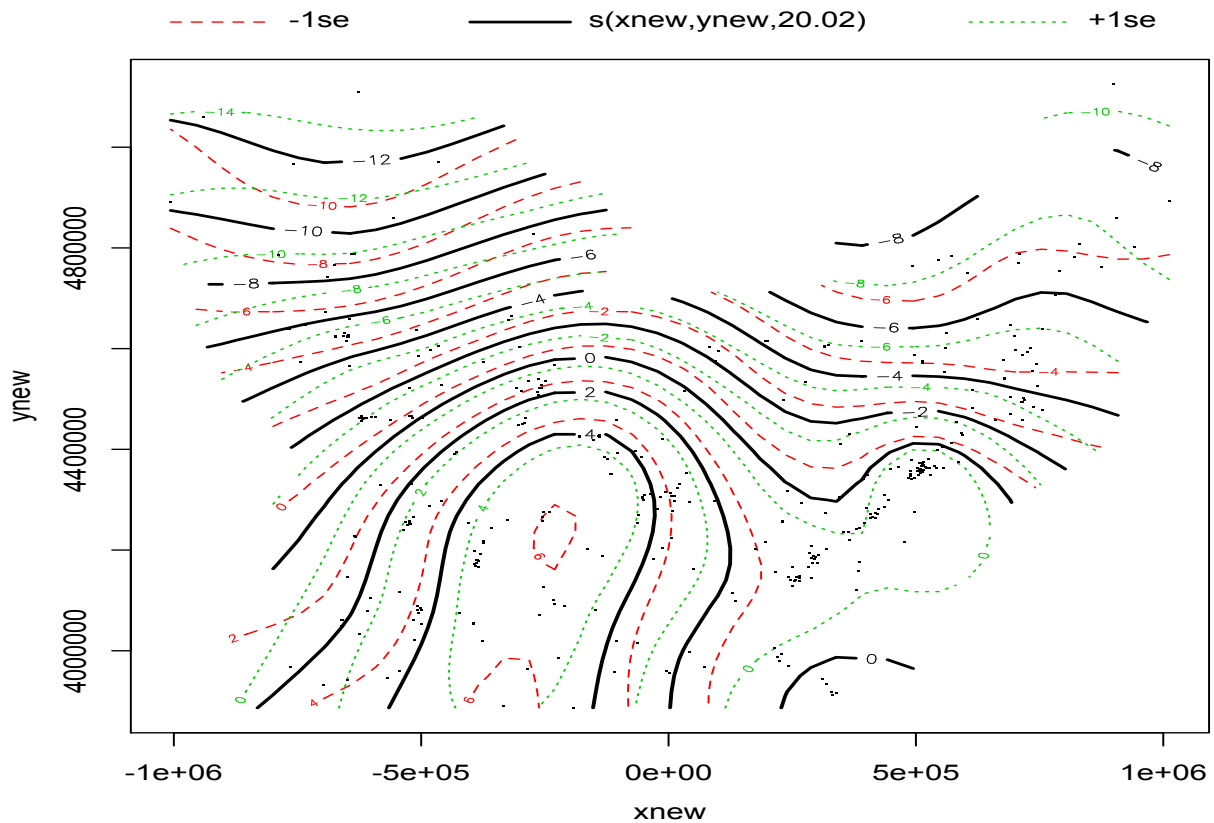
- Implemented in R using `mgcv::gam()` with `s(x,y,bs="tp")`.
- Other 2-D spline families:
  - Tensor-product splines.
  - Radial basis splines.
- Widely used for spatial smoothing, e.g. $PM_{2.5}$ exposure maps.

Thin Plate Splines in Practice

- So far: we introduced splines as smoothers using basis functions and penalties.
- Now: extend this to spatial data by fitting a smooth surface $g(s_1, s_2)$ to locations (s_1, s_2).
- Example: $PM_{2.5}$ measurements across a region.
- Goal: interpolate a smooth surface of concentrations and quantify uncertainty.

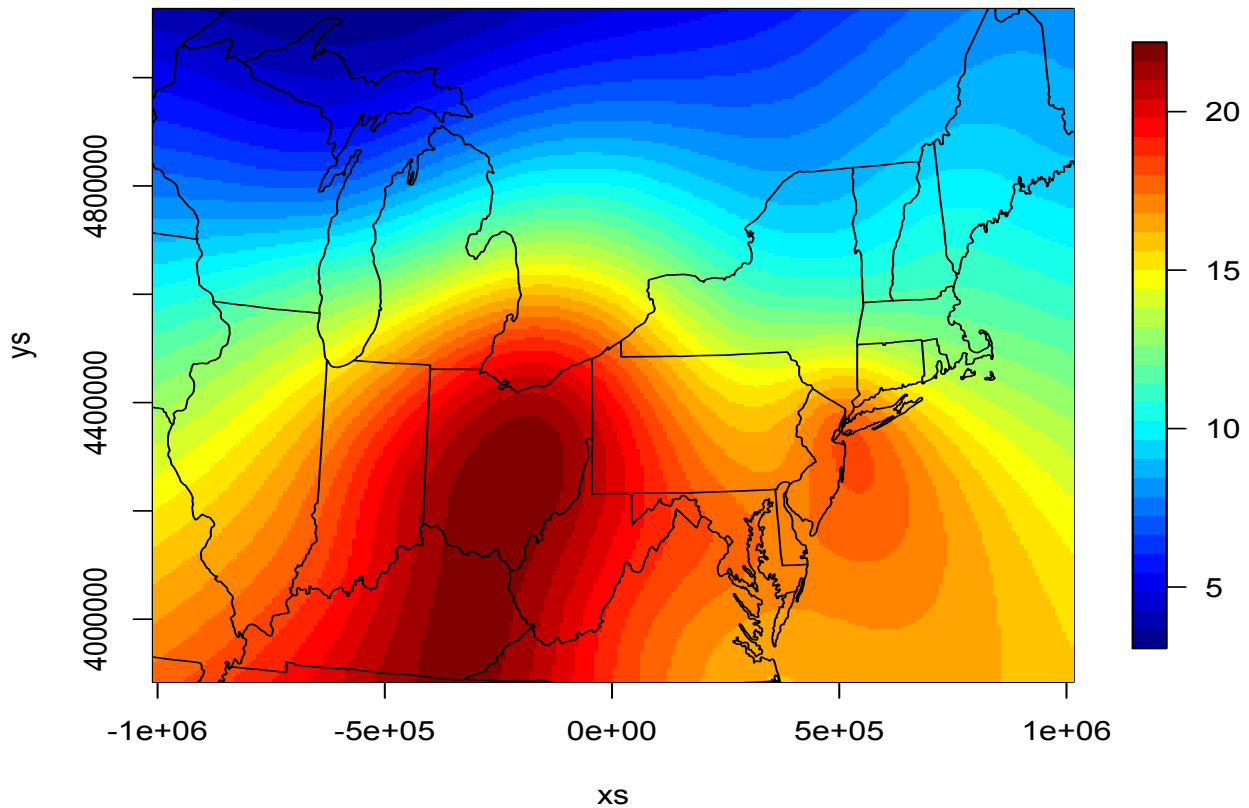
Thin Plate Spline Regression (Application to $PM_{2.5}$)

- Apply TPS to $PM_{2.5}$ monitoring data.
- Fitted spline surface captures both large-scale spatial trends and local variation.
- Contour map illustrates higher concentrations where monitors are denser or values are elevated.



Thin Plate Spline Regression Predictions

- Like kriging, TPS produces predictions $\hat{Z}(s_0)$ at any location.
- Predictions reflect smooth surfaces estimated from basis functions + penalty.
- Can be visualized as a continuous raster surface.



Thin Plate Spline Prediction Uncertainty

- Recall TPS/GAM model:

$$Y = B\beta + \epsilon, \quad \epsilon \sim N(0, \sigma^2 I),$$

where B is the spline basis matrix, β are the basis coefficients.

- Estimation is penalized:

$$\hat{\beta} = \arg\min_{\beta} \left\{ (Y - B\beta)^\top (Y - B\beta) + \lambda \beta^\top S \beta \right\},$$

where S is the penalty matrix (roughness penalty).

- The covariance of $\hat{\beta}$ is

$$Var(\hat{\beta}) = \sigma^2 (B^\top B + \lambda S)^{-1} B^\top B (B^\top B + \lambda S)^{-1}.$$

- Prediction at a new location s_0:

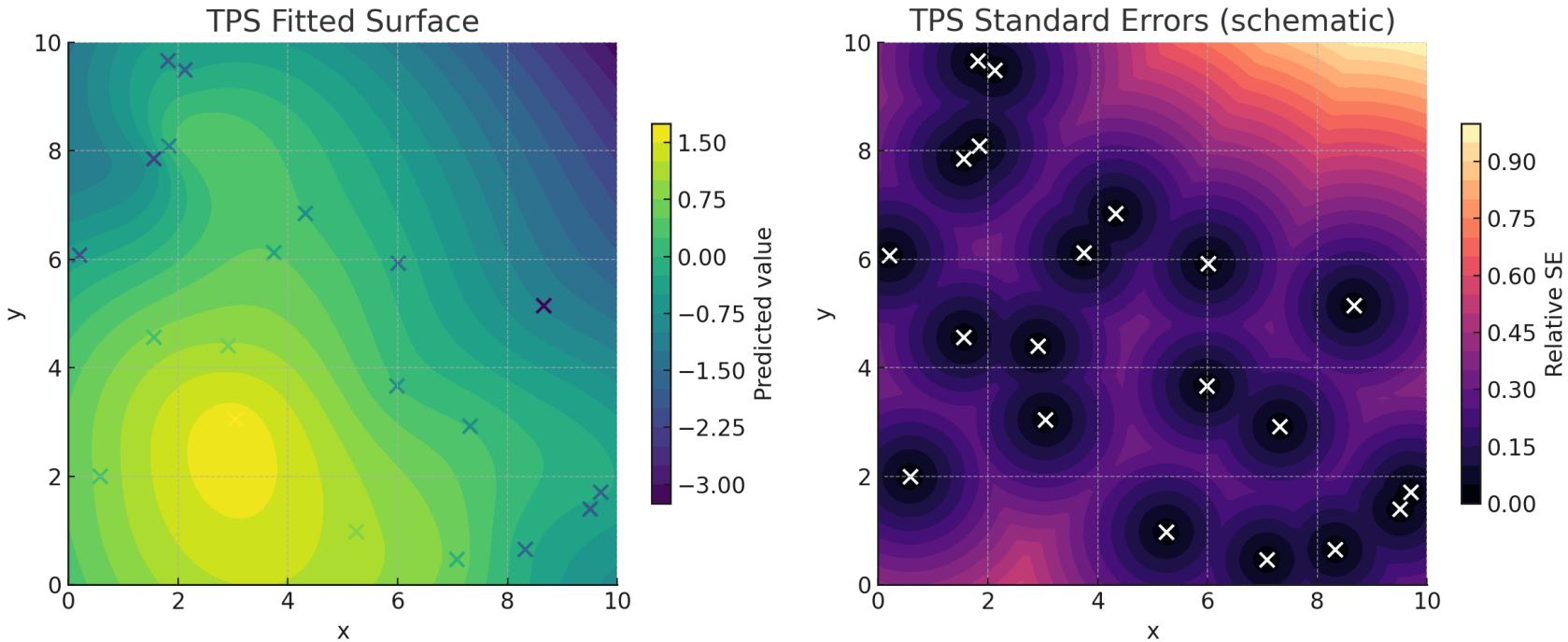
$$\hat{f}(s_0) = b(s_0)^\top \hat{\beta},$$

with variance

$$Var(\hat{f}(s_0)) = b(s_0)^\top \, Var(\hat{\beta}) \, b(s_0).$$

Uncertainty in Thin Plate Splines

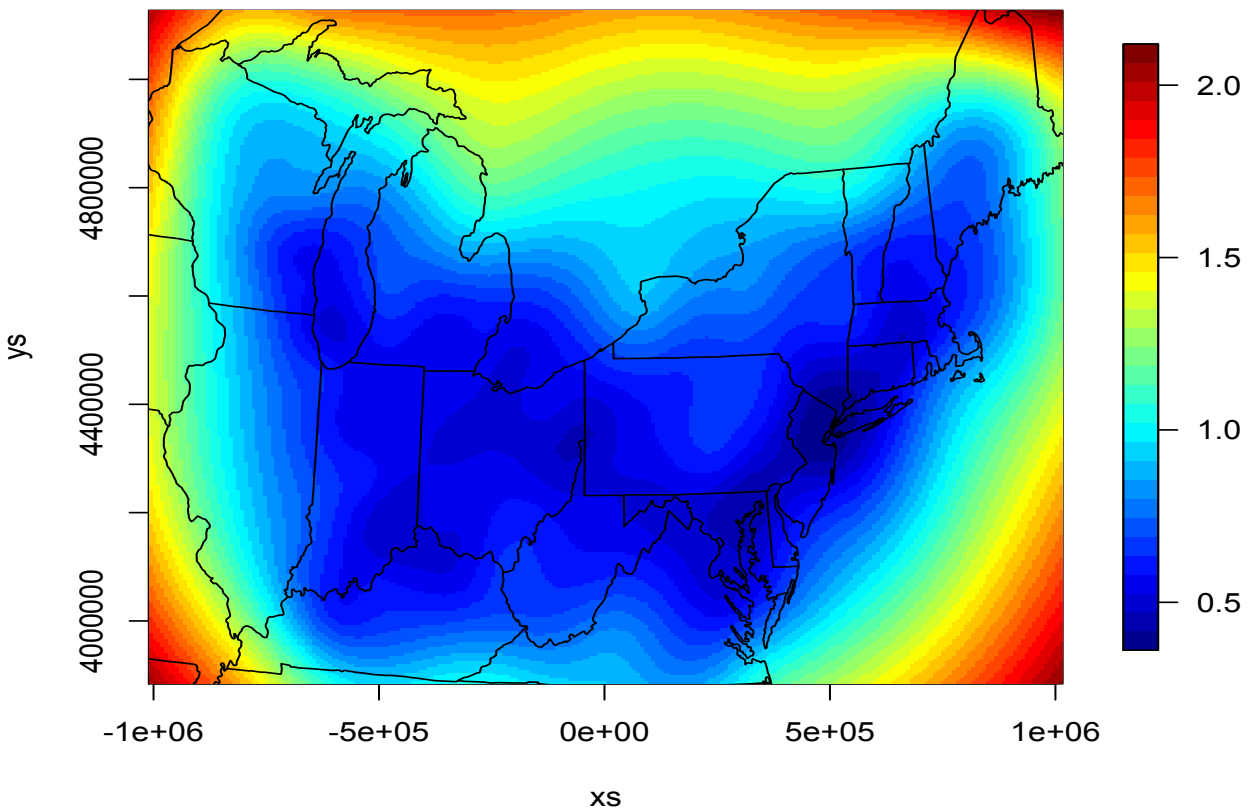
- TPS predictions $\hat{Z}(s_0)$ come with **standard errors**, derived from the covariance of spline coefficients.
- Interpretation:
  - Small SE near data points (surface well-constrained).
  - Larger SE where data are sparse (extrapolation).
  - SEs reflect estimation uncertainty, not stochastic process variance (contrast with kriging).



Left: TPS fitted surface. Right: schematic standard errors (low near data, high away).

Thin Plate Spline Regression Prediction Uncertainty

- Unlike IDW or deterministic splines, TPS provides **standard errors**.
- Uncertainty is typically lower near monitoring sites and higher where data are sparse.
- This feature makes TPS closer in spirit to kriging than simple interpolation.



Thin Plate Spline vs Kriging: Key Differences

- **Kriging**: models spatial dependence via covariance function $C(h)$.
- **TPS**: models smooth mean function $g(s)$ with a penalty on curvature.
- Both yield predictions + uncertainty, but come from different assumptions.
- In practice: TPS is often easier to implement in regression frameworks (e.g., GAM).