Spatial Analysis and Modeling

Modeling Spatial Dependence

Corso di formazione su ML e DL
Fondazione LINKS
25/09/2025

ISI Foundation

Learning Objectives

- Understand **spatial dependence** theory and its implications
- Master **measurement** techniques for different data types (areal vs point)
- Apply three major strategies for **handling spatial dependence**
- Implement solutions using Python spatial analysis ecosystem

Module Structure

1. **Foundations** — Theory, types, and consequences
2. **Measurement Methods** — Areal data indices and point data variograms
3. **Spatial Strategies** — Feature engineering, model structure, regularization

Part 1: Foundations

Spatial Dependence

Definition:
Spatial dependence (or **spatial autocorrelation**) exists when the value of a variable at one location is correlated with values at nearby locations, violating the independence assumption of classical statistics.

$$\mathrm{Cov}(Y(s_i), Y(s_j)) \neq 0 \text{ for } s_i \neq s_j$$

Intuitive Interpretation:

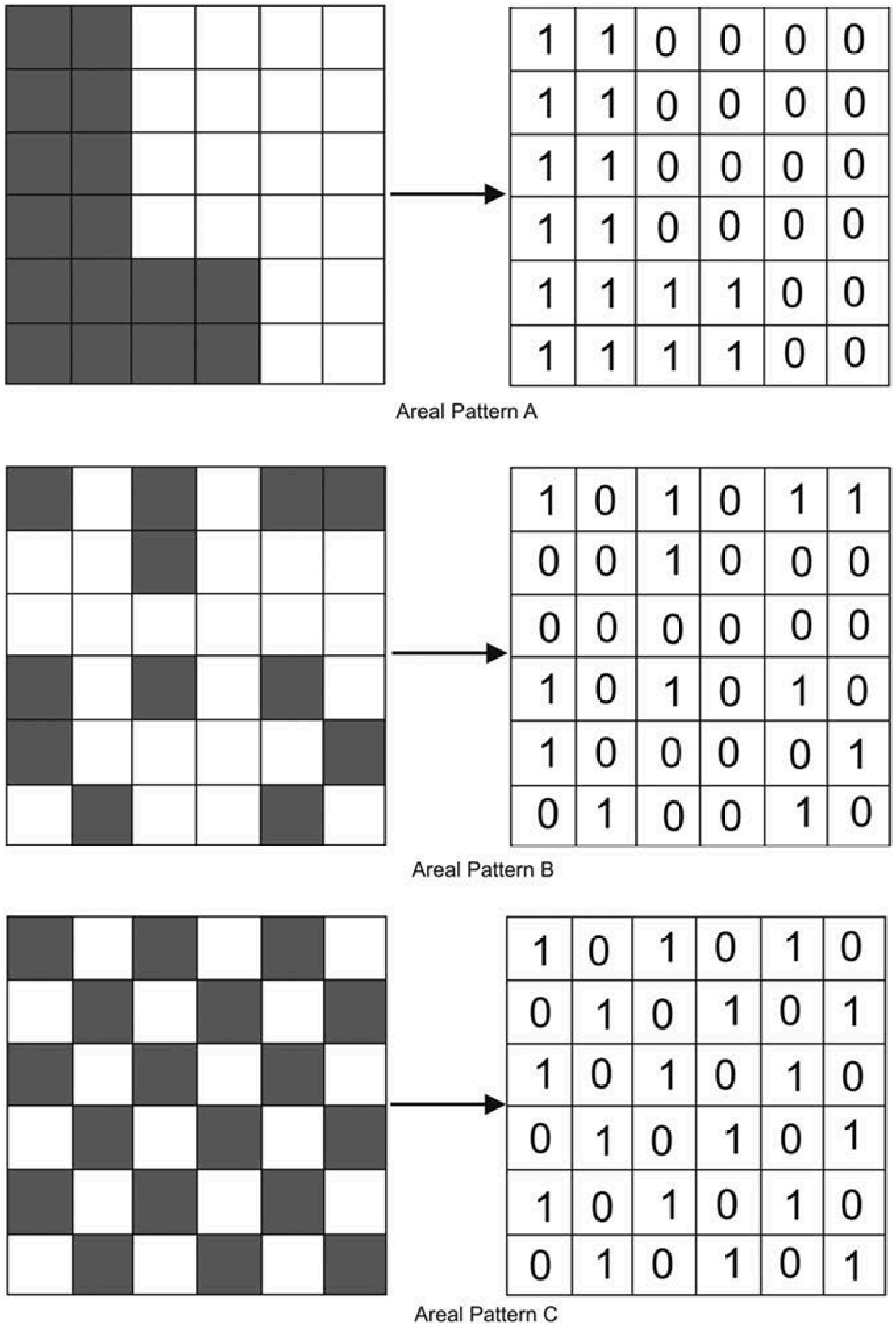
> *Everything is related to everything else, but near things are more related than distant things* — **Tobler's First Law of Geography**

Spatial dependence reveals **mechanisms** and **process dynamics** that operate through space:

- **Diffusion** / **contagion**: outcomes spread via contact or temporal propagation (e.g., disease, crime contagion).
- **Spillovers** (contextual effects): neighbours' attributes influence local outcomes (WX), without direct outcome feedback.
- **Environmental** / **abiotic gradients**: smooth physical covariates (climate, elevation, soils) create broad spatial trends.
- **Transport** / **flow** / **dispersion**: movement along networks or media produces directional or non-Euclidean dependence (pollution plumes, river flows).
- **Social** / **behavioral networks**: influence travels along social ties or institutions (peer effects, norms) rather than pure geography.
- **Shared-context confounding**: a latent spatial factor affects multiple variables, producing apparent spatial autocorrelation.

Types of Spatial Dependence

- **Positive:** Similar values cluster together
  - **Examples**: Housing prices, disease rates, pollution levels
- **Negative:** Dissimilar values are neighbors (rare)
  - **Examples:** Competitive retail locations, territorial animals
- **No Dependence:** Random spatial distribution
  - **Null hypothesis** in most spatial analyses



Consequences of Ignoring Spatial Dependence (1)

Statistical Issues

- **Standard errors** underestimated $\longrightarrow$ inflated significance
  - When nearby observations are correlated, the sample contains less independent information; i.i.d. SE then understate uncertainty and p-values are too optimistic.
- **Residuals** spatially correlated $\longrightarrow$ model misspecification
  - Spatial structure in residuals signals omitted spatial processes (missing covariates, spillovers, wrong trend); re-specify the model or model the residual covariance.
- **Predictions** suboptimal $\longrightarrow$ missed spatial patterns
  - Ignoring spatial covariance wastes information that kriging/Gaussian-process style models can use — expect poorer accuracy and miscalibrated prediction intervals.

Consequences of Ignoring Spatial Dependence (2)

Scientific Issues

- **Spurious relationships** from shared spatial factors
  - X and Y appear correlated because both vary with an unobserved spatial factor (confounder) rather than X causing Y.
- **Ecological fallacy** from inappropriate aggregation
  - relationships observed at an aggregated spatial unit do not necessarily hold at the individual level; aggregation can create or hide associations.
  - compare aggregate vs micro-level analyses when possible; use multilevel/hierarchical models that separate within- and between-unit effects; avoid making individual-level claims from areal analyses
- **Missing mechanisms** that operate through space

> Spatial dependence is a **challenge** (methodological) and an **opportunity** (insights)

Part 2: Measurements (Areal)

Spatial Autocorrelation

Definition: Correlation between values at different areal units, weighted by spatial proximity
Key Requirements:

- Spatial weights matrix W or Graph defining **neighborhood relationships**
- Choice of W affects results $\rightarrow$ need for a sensitivity analysis

Two Main Approaches:

1. **Global measures**: Single statistic summarizing dependence across entire study area
2. **Local measures**: Identify specific locations contributing to global patterns

Spatial connectivity: Weight Matrix W vs Graph

Two equivalent perspectives for encoding spatial relationships:

- **Matrix view** (classic): Spatial weights matrix $W \in \mathbb{R}^{n \times n}$ with entries w_{ij} quantifying the strength of connection from j to i.
- **Graph view** (modern): A sparse network of nodes (observations) and edges (relationships) with edge weights. In libpysal: **libpysal.graph.Graph**.

Why prefer Graph in practice?

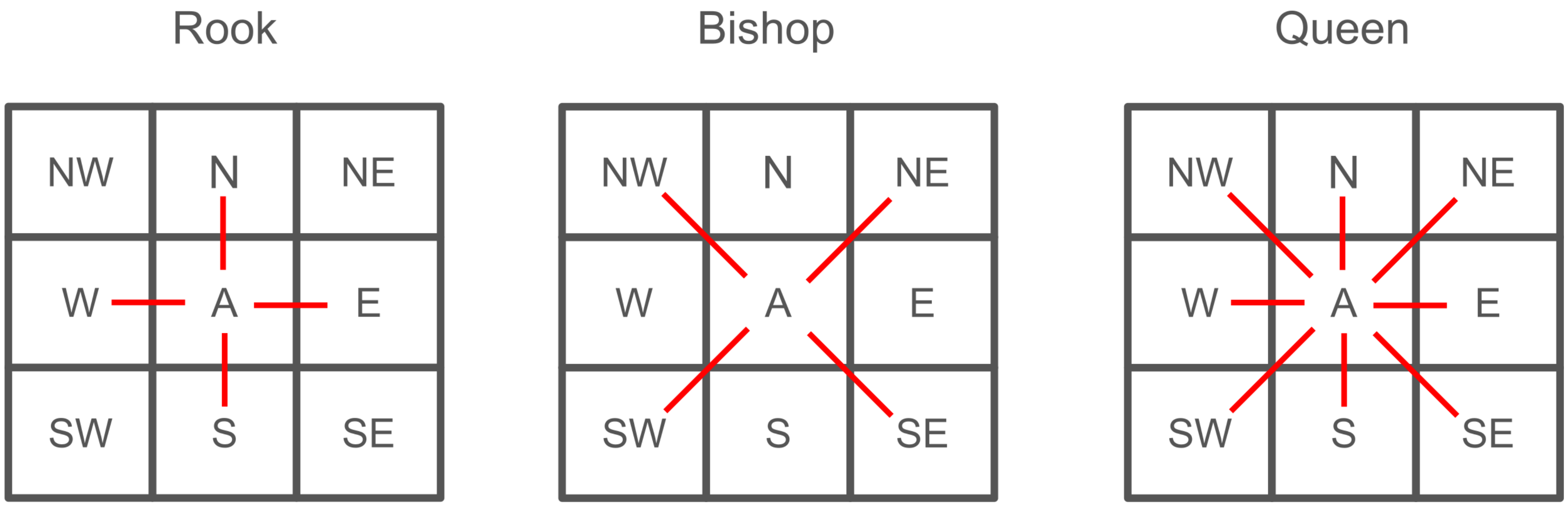
- **Natural API** for construction, inspection, plotting, and set operations
- **Efficient sparse representation**; easy access to neighbors, degrees, components, asymmetry
- **Simple normalization** via **transform** and seamless plotting, NetworkX export
- **Backward compatible**: can produce a legacy W object for packages that require it

> Key equivalence: the matrix W is just the adjacency matrix of the spatial graph, possibly after a transformation (e.g., row-standardization).

Building spatial graphs (libpysal >= 4.9)

Common constructions (all produce undirected graphs unless noted):

- **Contiguity** (areal polygons)
  - **Rook** : neighbors share a border
  - **Queen** : neighbors share a border or a vertex

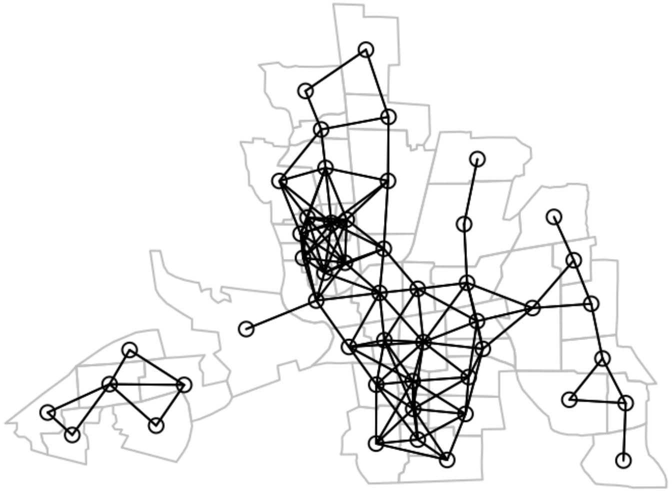


- **K-nearest neighbors** (KNN; points/centroids)
  - Each node connects to its k closest neighbors (can be asymmetric)

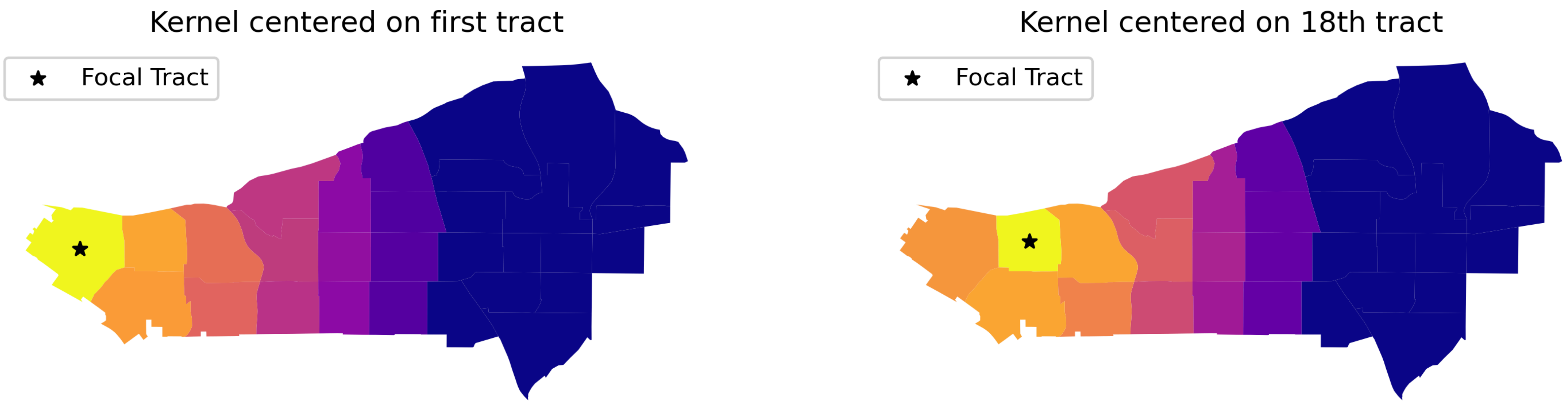
K nearest neighbours, k = 4

- **Distance band** (points or polygon centroids)
  - Edge exists if distance $\leq$ threshold (CRS units)

Distance based neighbours 0-0.6188642



- **Kernels** (distance-decay)
    - Continuous edge weights decreasing with distance (e.g., triangular, gaussian)



```python
from libpysal.graph import Graph

# gdf: GeoDataFrame of polygons; use gdf.centroid for point-based rules

g_rook   = Graph.build_contiguity(gdf)               # rook by default
g_queen  = Graph.build_contiguity(gdf, rook=False)   # queen
g_band   = Graph.build_distance_band(gdf.centroid, threshold=1000)   # meters if projected CRS
g_knn10  = Graph.build_knn(gdf.centroid, k=10)
g_kernel = Graph.build_kernel(gdf.centroid, kernel='triangular', bandwidth=1000)

# Row-standardize
g_rs = g_rook.transform('r')  # sum of outgoing weights for each node equals 1
```

Notes:

- Use a `projected CRS` for distance-based graphs (meters/feet). For polygons, graphs use centroids for distance rules.

Inspecting and validating graphs

- **Neighbors** and **weights**
  - **g[unit_id]** or **g.adjacency.loc[unit_id]** $\longrightarrow$ neighbor weights (pandas Series)
  - **g.neighbors[unit_id]**, **g.weights[unit_id]** $\longrightarrow$ tuple views
- **Degrees** and **sparsity**
  - **g.cardinalities** $\longrightarrow$ degree per node; **g.pct_nonzero** $\longrightarrow$ sparsity of W
- **Connectivity** and **isolates**
  - **g.n_components**, **g.isolates**, **g.component_labels**
- **Asymmetry** (KNN)
  - **g_knn10.asymmetry()** lists edges lacking reciprocals
- Matrix forms and **export**
  - **g.sparse** $\longrightarrow$ SciPy CSR; **g.to_networkx()** $\longrightarrow$ NetworkX graph

Transformations (aka standardizations)

In spatial econometrics it’s common to transform weights:

- `Row-standardization` ("r"): $w_{ij}^* = w_{ij} / \sum_j w_{ij}$
- `Double-standardization` ("d"): weights across all pairs sum to 1
- `Binary` vs `continuous`: contiguity is binary; kernels or border-length allow continuous weights

```
g_r = g.transform('r')       # row-standardized
g_d = g.transform('d')       # doubly-standardized
```

Remember: `Graph` is immutable; reassign to keep transformed versions.

Advanced graph concepts (brief)

- Set operations (**composability**)
  - E.g., "Bishop" = Queen minus Rook: **bishop = g_queen.difference(g_rook)**
- **Network-based distances** (walk/drive times)
  - Build graphs from travel cost matrices (e.g., from OSM networks)
- **Flow-based graphs**
  - Build from observed flows (trade, commuting) as weighted, possibly directed edges
- **Coincident nodes** (points at exact same location)
  - Strategies: **jitter** (random displacement) or **clique** (connect co-located nodes)

These provide realistic alternatives to Euclidean proximity when infrastructure or movement data drive interactions.

Choosing and justifying a spatial graph

- **Match mechanism**:
  - **edge-sharing (contiguity)**: use rook/queen when transmission happens across shared borders (policy spillovers, adjacency diffusion).
  - **distance / transport**: use distance bands, KNN, or travel-time kernels when influence decays with separation or travel cost (accessibility effects).
  - **flows**: use observed flow matrices (directed/weighted) when measured exchanges (commutes, trade, migration) drive interactions.
- **Sensitivity analysis**: test conclusions across multiple reasonable graphs
- **Standardize consistently**: row-standardize when interpreting SAR/SDM impacts; document your choice
- **Check connectivity/islands**: isolates may require KNN or higher thresholds

Reference and examples: **knaaptime** "The Spatial Graph" tutorial

Moran's I — Global Spatial Autocorrelation

$$I = \frac{n \sum_i \sum_j w_{ij}(y_i - \bar{y})(y_j - \bar{y})}{(\sum_i \sum_j w_{ij}) \sum_i (y_i - \bar{y})^2}$$

Intuitive Interpretation:

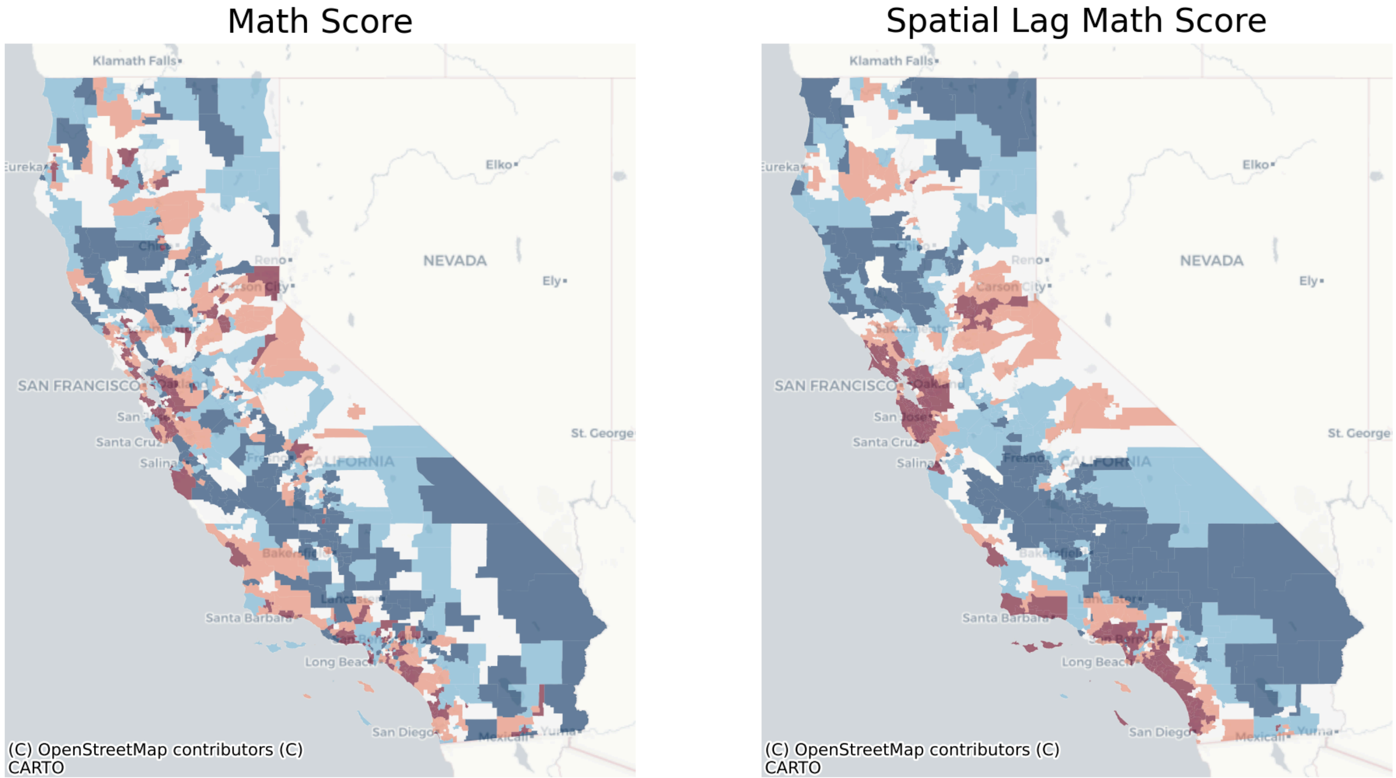
- Measures tendency for similar values to cluster spatially
- Compares observed covariation with the expected under spatial randomness

Range and Interpretation:

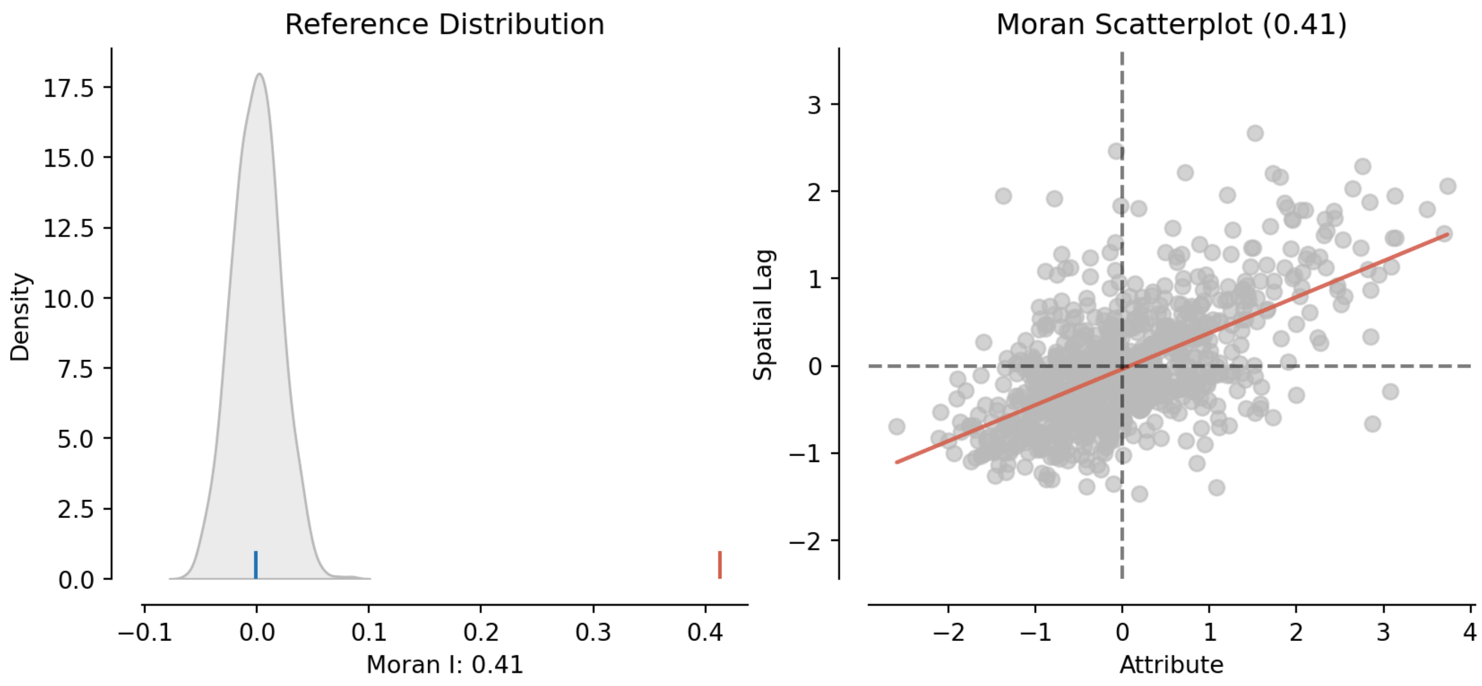
- $I \in [-1, 1]$ approximately (exact bounds depend on W)
- $I > 0$: **Positive autocorrelation** (similar values cluster)
- $I < 0$: **Negative autocorrelation** (dissimilar values cluster)
- $I \approx 0$: **No spatial autocorrelation** (random pattern)

Global Moran's I — Visual diagnostics

- **Spatial lag map**: the neighbors' average (only when W is row-standardized); otherwise Wy is a weighted sum.
  - Visual cue for clustering vs randomness (**more spiky behavior**)



- **Reference distribution** : In the significance testing, the observed I is compared to a reference distribution generated by randomly permuting the data across locations (**Complete Spatial Randomness, CSR**).
  - It shows what values of I are expected under no spatial autocorrelation. A **pseudo p-value** quantifies how unlikely the observed I is under the null hypothesis.
- **Moran scatterplot** : standardized y vs standardized Wy; slope $\approx I$



Moran's I — Variations and Extensions

Local Moran's I (LISA):

$$I_i = \frac{(y_i - \bar{y})}{\sigma^2} \sum_j w_{ij}(y_j - \bar{y})$$

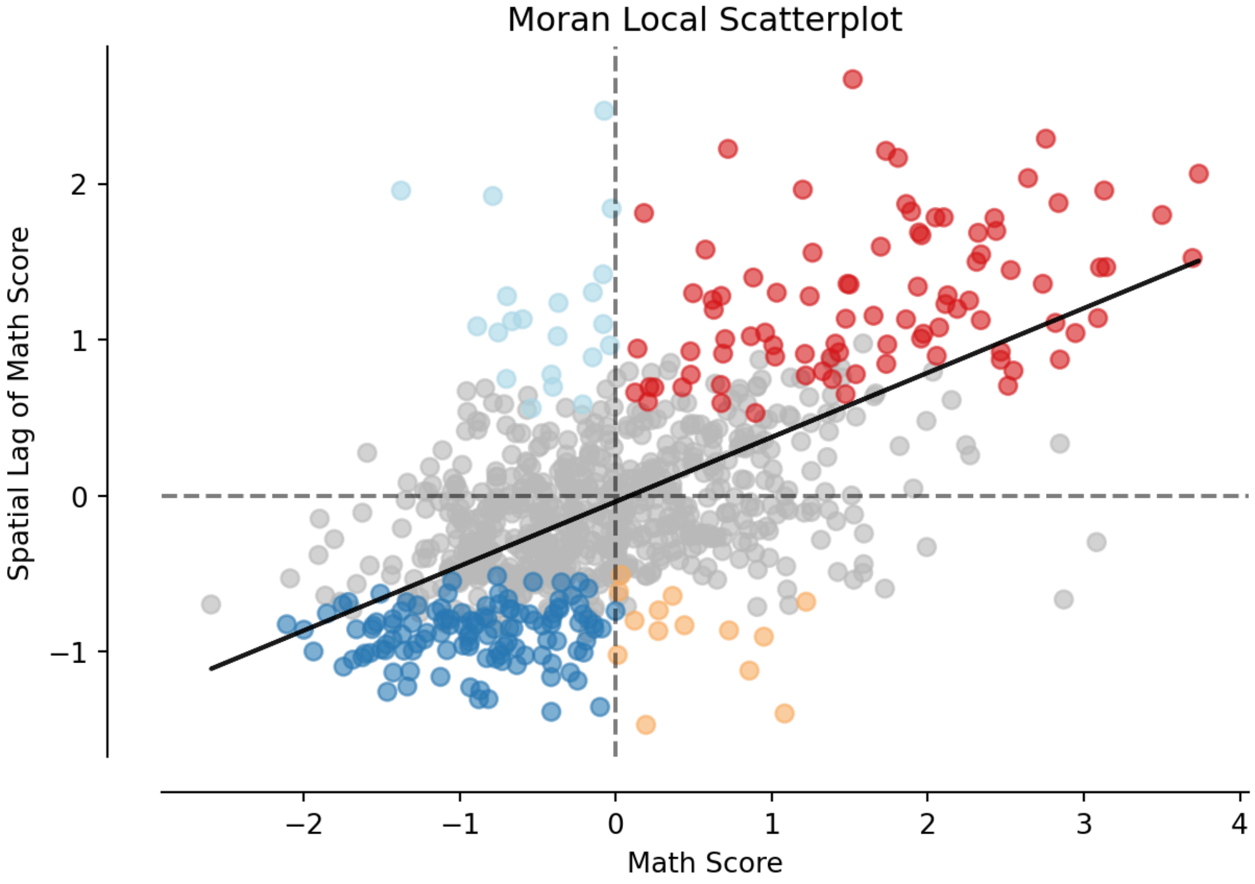
- Identifies **local clusters** and **spatial outliers**
- Four categories: **HH** (high-high), **LL** (low-low), **HL** (high-low), **LH** (low-high)

Interpretation

- HH/LL: local clusters (**hotspots**/**coldspots**)
- HL/LH: spatial outliers (**diamonds in the rough** or **holes in a donut**)

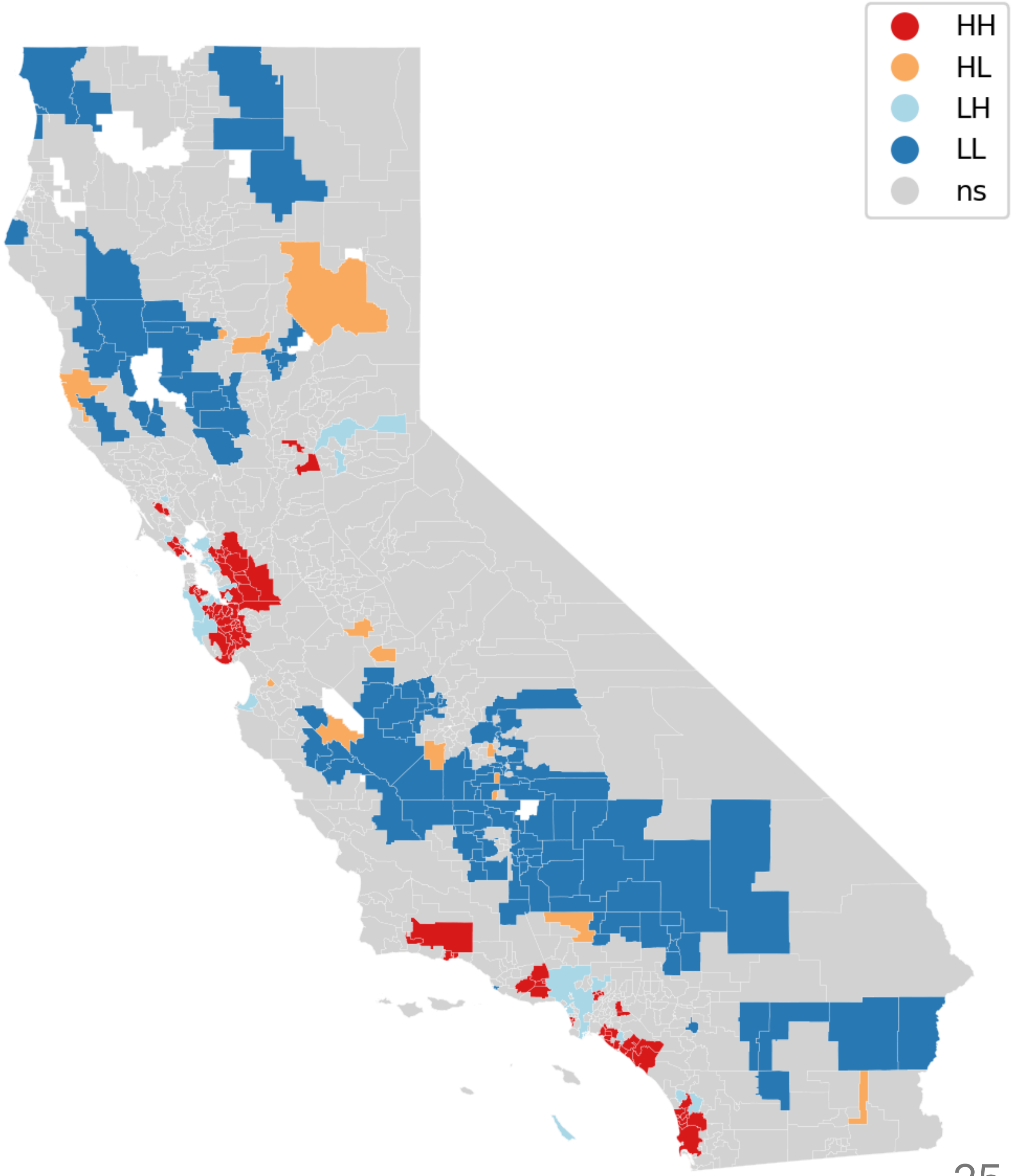
LISA — Visual diagnostics

- **Local Moran scatterplot**: highlights significant observations by quadrant
  - x axis: standardized value $z_i = (y_i - \bar{y})/s$;
  - y axis: spatial lag $\sum_j w_{ij} z_j$.
  - Points are **colored by quadrant** (HH/LL/HL/LH) and typically **masked** when not significant (permutation p_{sim}).



LISA — Visual diagnostics (2)

- **LISA cluster map**: shows HH, LL, HL, LH clusters; gray = not significant



Mapping LISA — a practical workflow

1. Compute **local p-values** by conditional permutations ($R \geq 999$)
2. Apply **multiplicity correction** (Holm for FWER or BH for FDR)
3. **Classify HH/LL/HL/LH** using signs, then **mask by adjusted significance**
4. **Report** R, sidedness (encodes your alternative hypothesis, i.e., positive autocorrelation, negative autocorrelation, or any deviation), and correction; add a legend describing classes

> Reference and examples: knaaptime "The Spatial Graph" tutorial

Part 2: Measurements (Point Data)

Introduction

Context: Continuous spatial fields observed at **point locations** (no built-in neighborhood).
Goal: Summarize how **similarity decays with distance** so we can **predict** and **plan sampling**.
Use-cases

- Environmental monitoring (temperature, rainfall, air quality)
- Natural resources (ore grades, soil properties)
- Ecology (biomass, chlorophyll, moisture)

Key idea: Use **distance-based** measures of spatial continuity rather than adjacency lists.

Assumptions

Stationarity (local enough)

- Mean and correlation structure don't change dramatically across the study area.
- If strong trends exist $\rightarrow$ **detrend** first (e.g., regress on elevation, lat/lon smooths) and variogram the **residuals**.

Isotropy vs. Anisotropy

- **Isotropy:** dependence depends on **distance only**.
- **Anisotropy:** dependence also depends on **direction** (wind, valleys, rivers).
  $\rightarrow$ Use **directional** semivariograms to detect/model it.

Practical note: The semivariogram is robust to mild mean drift (it uses **differences**), but strong trends should be handled before modeling.

(Semi)Variogram — Motivation

Question: Up to what distance do nearby points still share useful information?

Semivariogram = distance $\rightarrow$ average dissimilarity

- Near 0: low (or a **nugget jump** if micro-noise exists)
- Increases as similarity decays
- Levels off at the **sill** (beyond this, points are effectively independent)

For a distance class (lag) **h**,

$$\gamma(h) = \frac{1}{2|N(h)|} \sum_{(i,j)\in N(h)} \left(z(s_i) - z(s_j)\right)^2$$

In simple words: Half the **average squared difference** of all pairs whose separation falls in that distance bin.

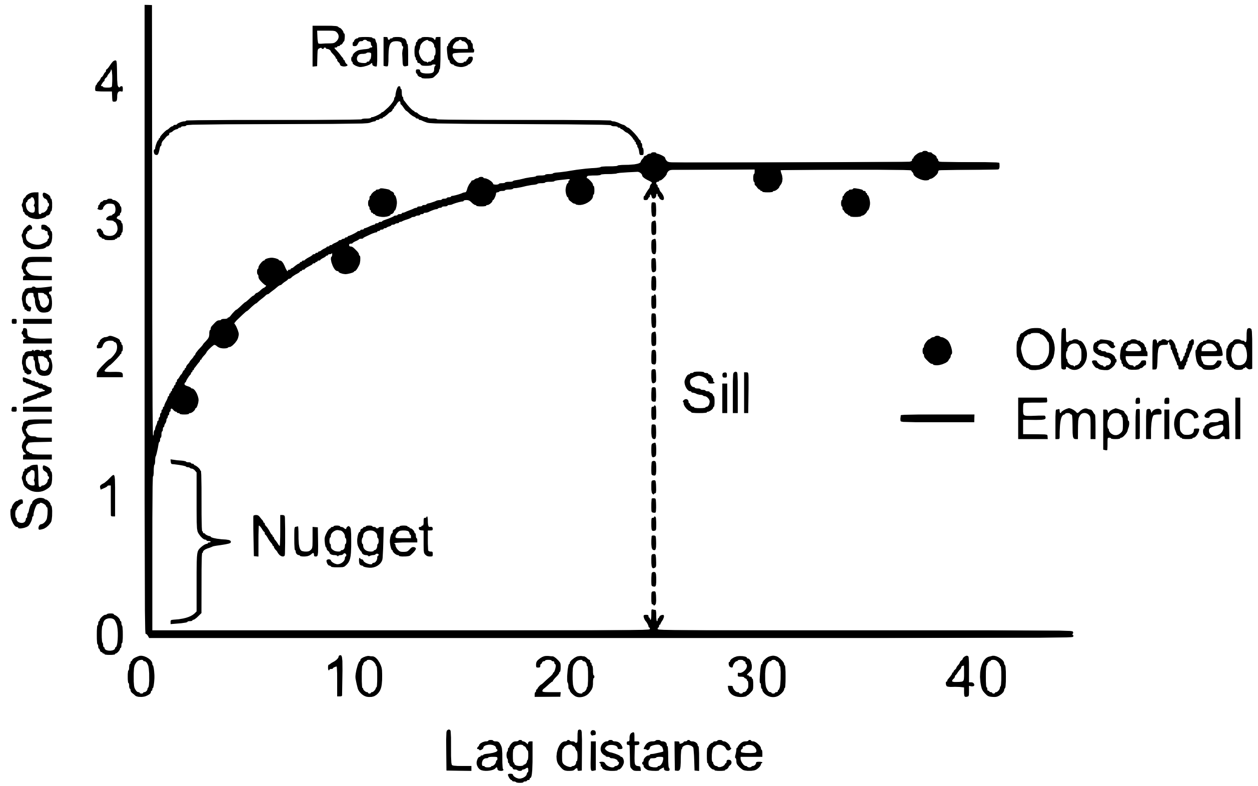
1. We create an **empirical curve** selecting 10-20 distance bins
2. We **fit a smooth model** (spherical/exponential/Gaussian/Matérn) to this empirical curve.

Parameters

- **Nugget** : value at (almost) zero distance $\rightarrow$ **micro-scale variability + measurement noise**.
- **Partial sill** : height of **spatial signal** above the nugget.
- **Sill (total)** : nugget + partial sill ≈ **total variance** of the process.
- **Range** : distance where $\gamma(h) \approx 95\%$ of sill $\rightarrow$ beyond **correlation fades**.

Rules of thumb

- **Nugget/Sill** < 0.25 $\rightarrow$ strong spatial dependence; > 0.75 $\rightarrow$ weak.
- Sensor spacing ≈ **~$\frac{1}{2}$ range** is often efficient.



Best Practices & Takeaways

Best practices

- Metric CRS; sensible max lag; enough pairs per bin
- Check trend & anisotropy; if present, **detrend** or use **universal/regression kriging**
- Prefer **simple, plausible** models (spherical/exponential/gaussian) before Matérn
- Report **nugget, partial sill, sill, range**, and **nugget/sill ratio**
- Validate with **spatial CV**; report accuracy **and** interval coverage

Takeaways

- Semivariogram = **distance vs. average dissimilarity**.
- **Nugget** separates **noise/micro-scale** from structure.
- **Range** = practical **spatial horizon**; guides sampling.
- Keep models simple; focus on **interpretability + validation**.

Point patterns (discrete events)

- Study of the **spatial distribution of events** (trees, crimes, disease cases, earthquakes).
- A **point pattern** = set of event locations, possibly with attributes (*marked point patterns*).
- Key questions:
  - Are points **clustered**, **regular**, or **random**?
  - At what **scale** do patterns appear?
  - Are clusters associated with proximity to sources or risk factors?

Motivations & Applications

- **Detect clustering/repulsion** beyond random placement.
- **Understand processes** generating patterns (ecological, social, physical).
- Applications:
  - Disease mapping
  - Crime analysis
  - Seismic activity
  - Plant ecology

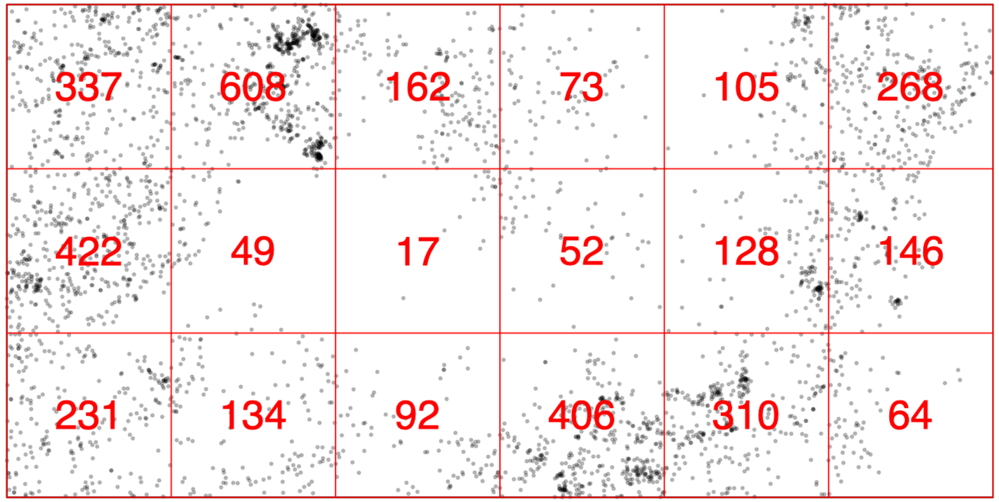
> **Goal:** Move from descriptive visualization → quantitative models.

First-Order Methods (Intensity)

- Focus on **absolute location of events** (variation in density).
- Reveal **large-scale variation**, not interactions.
- **Quadrat methods**: partition area → count points → estimate intensity.
- **Kernel Density Estimation (KDE):** smooth density surface.

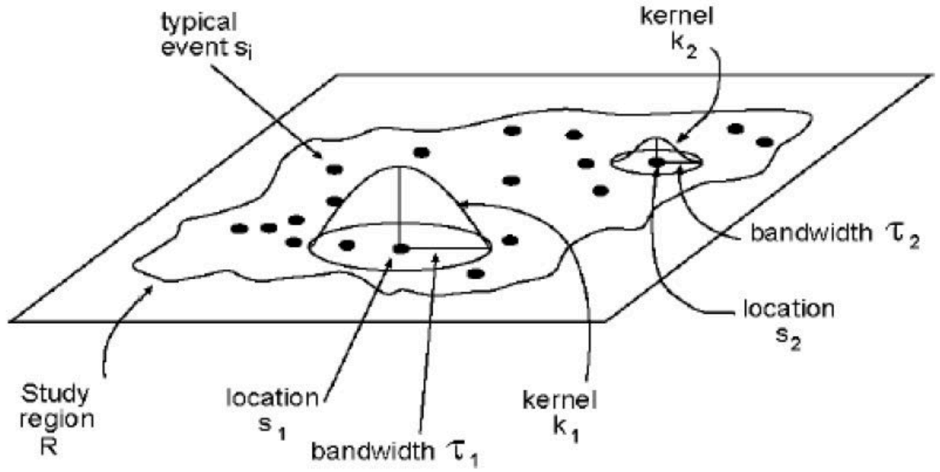
Quadrat Method

- **Procedure**:
  i. Divide study region into equal-area quadrats.
  ii. Count number of points per quadrat.
  iii. Compute local intensity $\hat{\lambda}(A_l) = \dfrac{n(A_l)}{|A_l|}$.
- **Insights**:
  - Larger quadrats → smoother maps.
  - Smaller quadrats → “spiky” results.
- **Limitations**:
  - Sensitive to size/shape (MAUP).
  - Captures **only first-order effects**.



Kernel Density Estimation

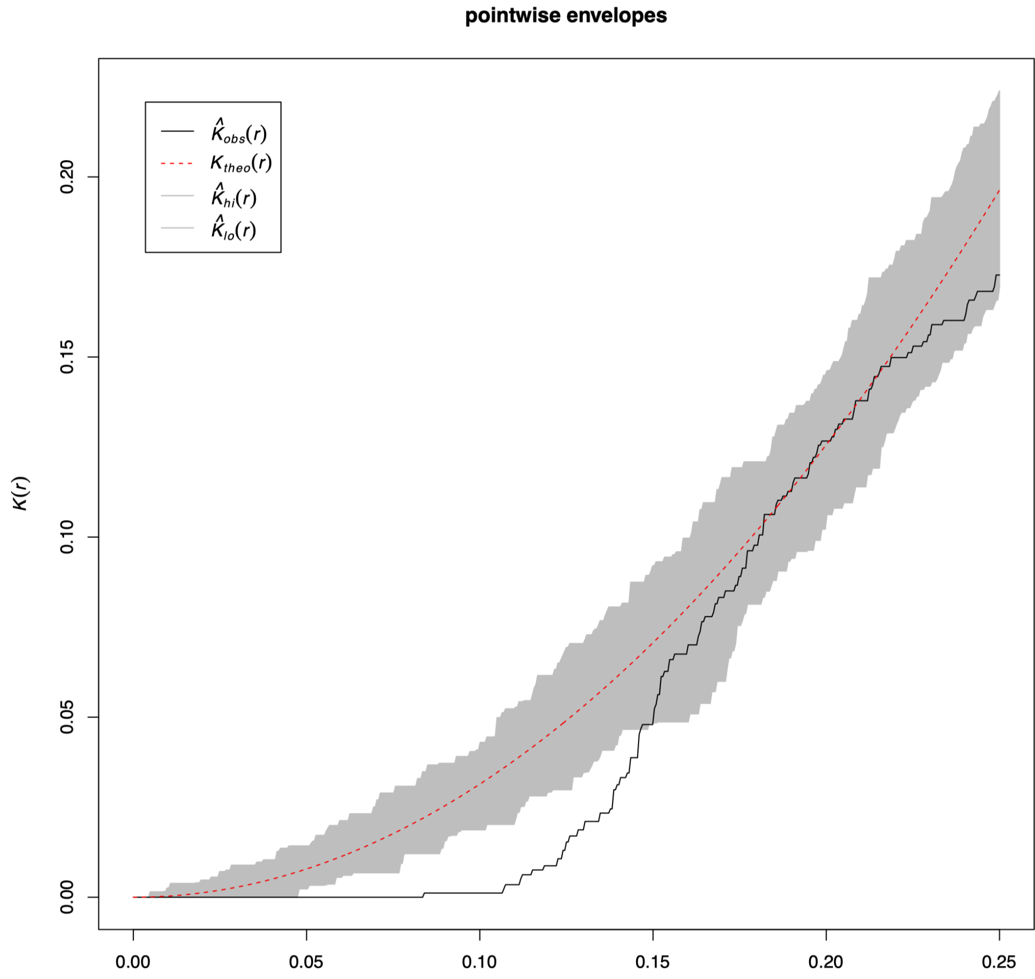
- **Idea**: Smooth point contributions with a kernel function.
- **Bandwidth choice critical**:
  - Small → noisy, spiky.
  - Large → oversmoothing.
- **Adaptive kernels** adjust to local density.
- **Applications**: crime hotspots, disease risk surfaces.:contentReference[oaicite:8]{index=8}



Second-Order Methods (Interactions)

- Examine **event–event relationships** (clustering vs repulsion).
- Tools:
  - **NNI**: observed vs expected mean nearest neighbor distance.
  - **G function**: CDF of nearest-neighbor distances.
  - **F function**: empty space distribution, i.e., CDF of the distance from an arbitrary location to the nearest event
  - **K function**: cumulative neighbors within radius *d*.

> **R/spatstat**: canonical, full-featured.
> **Python/pointpats**



Part 3: Spatial Dependence Strategies

Three Main Approaches:

1. **Feature Engineering**: Add spatial context to input variables
2. **Model Structure**: Embed dependence in model equations/assumptions
3. **Regularization**: Add spatial smoothness penalties to objective function

Choice Criteria:

- **Data type**
- **Computational resources**
- **Interpretability needs**
- **Uncertainty quantification**

Approach 1: Spatial Feature Engineering

Definition: Transform spatial relationships into input **features** for standard ML/statistical models
Core Idea: Make spatial dependence **explicit** in the feature space

Common Approaches:

- **Spatial lags**: Include neighbors' values as features
- **Kernel features**: Gaussian/exponential weighted neighborhoods
- **Spatial aggregations**: Local averages, densities, gradients
- **Distance features**: Distances to landmarks, boundaries, centroids
- **Graph-derived embeddings**: learn compact node representations (via node2vec/DeepWalk or GNN encoders) that capture multiscale connectivity and local structure
- **Texture features**: Local variance, local Moran's I, edge density

Moran Spatial Eigenvector Filtering (MSEF)

- **Definition:** Uses **eigenvectors** derived from a (centered) spatial weights matrix to represent principal spatial patterns (from broad to fine scales). These eigenvectors capture orthogonal spatial structures and can be included as additional regressors to model spatial variation or remove residual spatial autocorrelation.

- **How it works (brief):**
  i. Build a spatial weights matrix W encoding the neighborhood (contiguity, KNN, kernel, or flow-based).
  ii. Center W using the mean-centering matrix $C = I - \frac{1}{n}\mathbf{1}\mathbf{1}^\top$, then compute $B = CWC$.
  iii. Compute eigenpairs of B: $Bv_k = \lambda_k v_k$. Eigenvectors v_k with large positive eigenvalues λ_k reflect positive spatial autocorrelation at different scales.
  iv. Select a **subset of eigenvectors** (see selection criteria) and include them as additional regressors or features in your model.

When to use:

- As a flexible, model-agnostic way to add spatial features to ML pipelines.
- To orthogonalize spatial patterns so that standard regression/ML methods can handle spatial structure without changing their core estimators.

Selection tips:

- **Rank eigenvectors by their associated Moran's I** (compute Moran for each v_k) and pick those with strong positive autocorrelation.
- **Use forward selection** (AIC/BIC/CV) to avoid **overfitting** and **multicollinearity**.
- **Limit the number of eigenvectors** (e.g., tens, not hundreds) based on sample size and parsimony.

Caveats:

- Results depend on the choice of W; always run **sensitivity checks** across plausible W.
- Selected eigenvectors can be correlated with covariates — check multicollinearity and interpret coefficients cautiously.
- Eigenvectors capture spatial structure but do not identify causal mechanisms.

References: Griffith (2003) on ESF; Dray et al. (2006) for eigenfunction methods.

Feature Engineering — Advantages and Limitations

Pros:

- **Model agnostic**: Works with any ML algorithm
- **Computational efficiency**: Standard optimization, no special software
- **Easy interpretation**: Features have clear spatial meaning
- **Flexible**: Can combine multiple spatial representations

When to Use:

- Rapid **prototyping** and benchmarking
- Large-scale prediction systems
- When spatial structure is well-understood
- **Integration** with existing ML pipelines

Cons:

- **Manual engineering**: Requires domain knowledge and experimentation
- **Feature proliferation**: May create high-dimensional, correlated features
- **Static neighborhoods**: Fixed spatial relationships, not adaptive
- **No uncertainty quantification**: Point estimates only

Strategy 2: Spatial Model Structure

Definition: Incorporate spatial dependence directly into model equations/assumptions
Core Idea: **Model the process generating spatial dependence** rather than just adding spatial features

Examples (there are SO many alternatives!):

1. **Spatial econometrics models**: **SLX, SAR, SEM, SDM, SDEM, SAC**
2. **Kriging**: Gaussian Process-based optimal prediction for point-reference data
3. **Hierarchical Models**: Multi-level spatial frameworks combining structured and unstructured effects

Spatial Econometric Models

Concepts first: what is “spatial” in these models?

- Two mechanisms
  - **Spillover in outcomes** (endogenous): neighboring outcomes interact $\longrightarrow$ global feedback possible
  - **Spillover in covariates** (exogenous): neighboring X influence local y
- Two spatial scales
  - **Local**: effects fall only on immediate neighbors
  - **Global**: effects propagate to neighbors-of-neighbors (**feedback loops**)

Model mapping (quick mental model)

- **SLX**: local exogenous spillovers
- **SAR**: global endogenous spillovers
- **SEM**: global error diffusion (correlated residuals)
- **SDM**: SAR + SLX (global endogenous + local exogenous)
- **SDEM**: SEM + SLX (global diffusion + local exogenous)
- **SAC/SARAR**: SAR + SEM (both endogenous and error diffusion)

— choose based on theory and diagnostics, not only fit.

SLX — Spatial Lag of X (local exogenous spillovers)

Equation:

$$Y = X\beta + W X\,\theta + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2 I)$$

Interpretation:

- Coefficients in θ measure how neighbors' characteristics affect Y_i (local spillovers)
- Linear model $\to$ coefficients have usual interpretation (no feedback multiplier)

When to use:

- **Expect neighborhood attributes to matter, but not feedback** in Y
- Clean, interpretable spillovers needed; **baseline** for design-based work

Pros/Cons:

- Pros: OLS-estimable, transparent interpretation; great starting point
- Cons: No endogenous feedback; ignores global propagation

SAR — Spatial Autoregressive (Lag) Model

Equation:

$$Y = \rho WY + X\beta + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2 I)$$

Interpretation:

- Endogenous interaction/feedback among outcomes
- **Spillover** effects via $(I - \rho W)^{-1}$ (**spatial multiplier**)
- **Direct** and **indirect** impact decomposition

When to Use:

- Peer effects, diffusion processes
- Network spillovers in outcome variable
- Policy impact analysis with spatial feedback

SEM — Spatial Error Model

Equations:

$$Y = X\beta + u, \quad u = \lambda W u + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2 I)$$

Interpretation:

- Captures **omitted spatially structured shocks**
- Corrects inference when residuals cluster spatially
- Focus on consistent β estimation

When to Use:

- **Spatial correlation due to unobserved variables**
- Model **misspecification** creates spatial residual patterns
- Primary interest in covariate effects, not spatial spillovers

SDM — Spatial Durbin Model

Equation:

$$Y = \rho WY + X\beta + WX\theta + \varepsilon$$

Interpretation:

- Allows spillovers through both outcomes and covariates
- WX captures exogenous neighbor effects
- Nests SAR/SEM under parameter restrictions

When to Use:

- Suspect both endogenous and exogenous spillovers
- Policy diffusion analysis
- Robust to some W misspecification

Advantage: Detailed direct/indirect/total impact decomposition

SDEM — Spatial Durbin Error

Equations:

$$Y = X\beta + WX\,\theta + u, \quad u = \lambda W u + \varepsilon, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2 I)$$

Interpretation:

- Local exogenous spillovers (via WX) + global error diffusion (via λ)
- Still linear in $Y \rightarrow$ coefficients in β, θ interpretable as usual

When to use:

- No theoretical need for endogenous feedback in Y, but residual diffusion likely
- Prefer linear interpretation of coefficients without impact multipliers

Pros/Cons:

- Pros: Interpretable parameters; efficient vs SLX when errors are spatial
- Cons: Misspecified if true process has endogenous feedback (favor SDM then)

SAC / SARAR — Spatial combo (lag + error)

Equations:

$$Y = \rho WY + X\beta + u, \quad u = \lambda W u + \varepsilon$$

Interpretation:

- Endogenous outcome feedback and spatially correlated errors simultaneously

Notes:

- Can be useful when residual autocorrelation remains after SAR/SDM
- Identification and interpretation can be delicate; avoid fully unrestricted “Manski” models

Explanation and Interpretation of Parameters ρ, θ, and ε

ρ **(rho):**

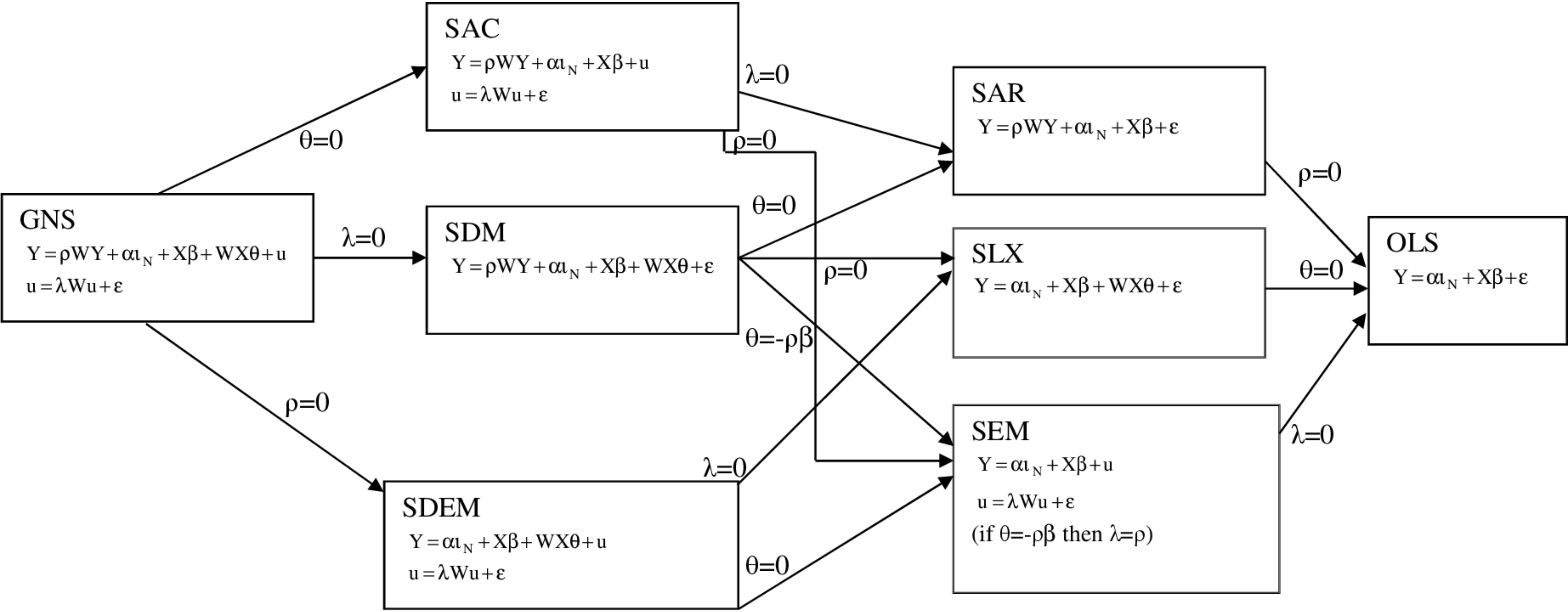
- Represents the strength of endogenous spatial dependence (feedback) in outcome variable Y.
- In SAR/SDM/SAC models, ρ quantifies how much neighboring outcomes (WY) influence each location.
- **Interpretation:**
  - $\rho = 0$: No spatial feedback; outcomes are independent.
  - $0 < \rho < 1$: Positive spatial spillovers; high values in neighbors increase local Y.
  - $\rho < 0$: Negative spatial spillovers (rare); high values in neighbors decrease local Y.
- **Magnitude:** Larger $|\rho|$ means stronger spatial autocorrelation and more global propagation of effects.

θ (theta):

- Measures exogenous spatial spillovers from neighbors' covariates (WX).
- In SLX/SDM/SDEM models, θ captures how neighbors' characteristics affect local outcomes.
- **Interpretation:**
  - $\theta_j > 0$: Higher values of covariate X_j in neighbors increase local Y.
  - $\theta_j < 0$: Higher values of X_j in neighbors decrease local Y.
- **Magnitude:** Indicates the average effect of neighboring covariates, controlling for local X.

ε **(epsilon):**

- Represents random error or noise, assumed to be normally distributed and independent across locations.
- In SEM/SDEM/SAC models, ε is the innovation term after accounting for spatial structure.
- **Interpretation:**
  - Captures unexplained variation not modeled by X, WX, or spatial feedback.
  - If residuals show spatial autocorrelation, model may need SEM/SAC structure.



SAC
$Y = \rho WY + \alpha \iota_N + X\beta + u$
$u = \lambda Wu + \varepsilon$
GNS
$Y = \rho WY + \alpha \iota_N + X\beta + WX\theta + u$
$u = \lambda Wu + \varepsilon$
SDM
$Y = \rho WY + \alpha \iota_N + X\beta + WX\theta + \varepsilon$
SDEM
$Y = \alpha \iota_N + X\beta + WX\theta + u$
$u = \lambda Wu + \varepsilon$
SAR
$Y = \rho WY + \alpha \iota_N + X\beta + \varepsilon$
SLX
$Y = \alpha \iota_N + X\beta + WX\theta + \varepsilon$
SEM
$Y = \alpha \iota_N + X\beta + u$
$u = \lambda Wu + \varepsilon$
(if $\theta=-\rho\beta$ then $\lambda=\rho$)
OLS
$Y = \alpha \iota_N + X\beta + \varepsilon$
$\theta=0$
$\lambda=0$
$\rho=0$
$\lambda=0$
$\rho=0$
$\theta=0$
$\rho=0$
$\theta=-\rho\beta$
$\lambda=0$
$\theta=0$
$\rho=0$
$\theta=0$
$\lambda=0$

Picking a model: diagnostics and strategy

- **Start from OLS** $\longrightarrow$ **run spatial diagnostics on residuals**
  - Moran's I on residuals; LM lag and LM error; robust LM versions; SARMA joint test
- **Specific-to-general search (often recommended)**
  - If LM-error wins $\longrightarrow$ try SEM/SDEM
  - If LM-lag wins $\longrightarrow$ try SAR/SDM
  - If both significant $\longrightarrow$ prefer Durbin specs (SDM/SDEM) and test down
- **General-to-specific complement**
  - Start from SDM or SAC; test common-factor restrictions to reduce to SAR/SLX
- **Always re-check residual spatial autocorrelation after fitting**

Interpreting coefficients and impacts

- Models with ρWY (SAR, SDM, SAC) are **non-linear** in Y
  - Coefficients β are not marginal effects
  - Report **direct**, **indirect**, and **total impacts** from $(I - \rho W)^{-1}$
- SLX/SDEM
  - θ on WX are interpretable as average neighbor effects (no multiplier)

> Practice: Always report impacts (and their uncertainty) when $\rho \neq 0$

Quick decision cheat sheet (advanced)

- Start: OLS + diagnostics (Moran's I, LM/Robust LM, Durbin tests)
- Robust LM-Error only $\longrightarrow$ SEM (or SDEM if SLX plausible)
- Robust LM-Lag only $\longrightarrow$ SAR (or SDM if SLX plausible)
- Both robust tests significant $\longrightarrow$ SDM; test down via common-factor restrictions
- AK significant in 2SLS $\longrightarrow$ spatial structure remains; consider SLX/SDM/SDEM with valid instruments

References: spreg "Specification tests" notebook; Koley & Bera (2024); Anselin & Rey (2014).

> **Practical Example**
> More worked examples (hedonics)

Best practices for applied work (checklist)

- **Theory first**: decide whether endogenous feedback is plausible
- Weights W: **try multiple reasonable graphs** (contiguity, KNN, distance) and **row-standardize**
- **Diagnostics**: LM tests, robust variants, and residual Moran's I
- **Prefer Durbin models** (SDM/SDEM) **when in doubt about exogenous spillovers**
- **Impacts**: compute and report direct/indirect/total effects for lag models
- **Sensitivity**: test conclusions across alternative W and sample definitions
- **Validation**: use spatial cross-validation; check for remaining spatial structure in residuals
    - After fitting: residual Moran's I should not be significant; otherwise revise W/specification

Conditional Autoregressive Model (CAR)

- Designed for **areal** (polygon) data.
- Typical workflow: fit a regression with predictors X; if residuals remain spatially autocorrelated (Moran's I significant), include a **spatial random effect** ϕ to model that leftover structure.
- CAR supplies a **prior** for ϕ that borrows strength from neighboring areas, effectively **modeling the spatial pattern left unexplained** by X and improving inference and prediction.

How CAR differs from spatial-econometric models (1)

- **Modeling target**
  - **CAR:** models a spatially structured latent effect ϕ (residual/process) in a hierarchical model — it explains leftover spatial structure after X.
  - **Spatial-econometric models (SAR/SEM/SDM):** directly specify spatial dependence in the regression (e.g., WY, Wu, WX) and focus on spillovers among observed outcomes or covariates.

- **Mechanism vs residual structure**
  - **CAR:** treats spatial dependence as a prior for latent spatial variation (smoothing/borrowing strength).
  - **SAR/SDM:** model endogenous feedback or explicit neighbor effects in the outcome equation (causal spillover interpretation requires identification).

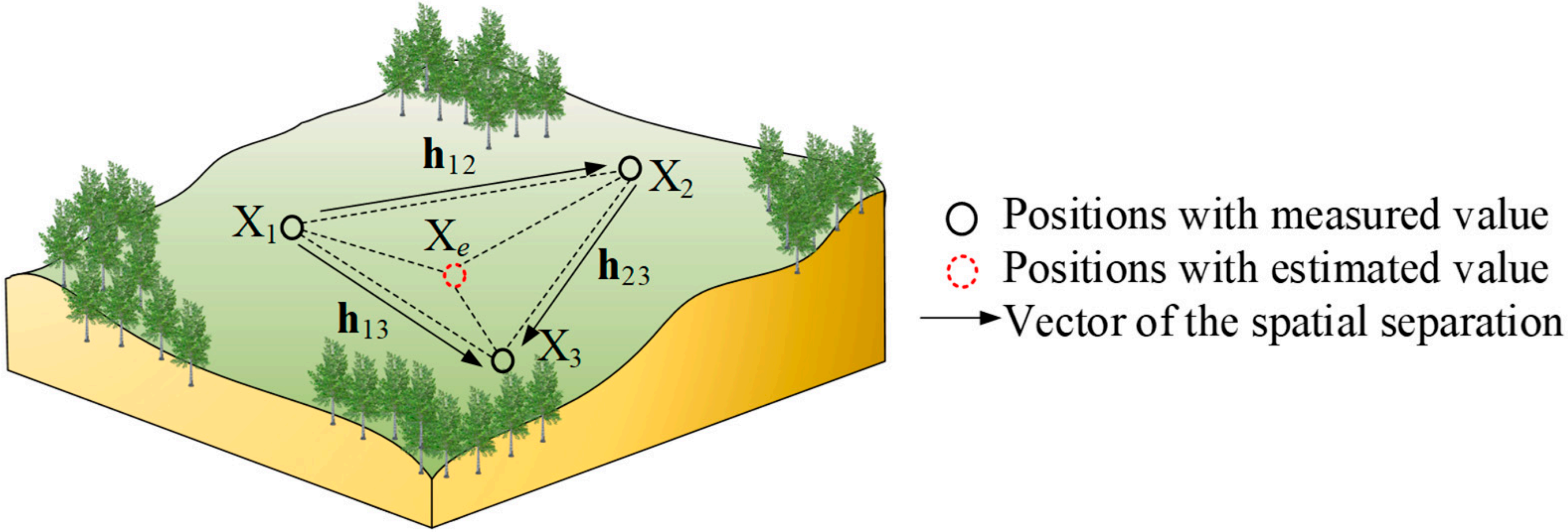
How CAR differs from spatial-econometric models (2)

- **Inference & usage**
  - CAR is commonly used in Bayesian hierarchical frameworks (non-Gaussian likelihoods, small-area estimation, disease mapping) — emphasis on uncertainty of latent field.
  - SAR/SEM/SDM are common in spatial econometrics with frequentist and Bayesian estimators focused on estimating spillover parameters and impact decomposition.

- **Practical implication for applied work**
  - If residual autocorrelation remains after X, prefer CAR (or CAR + fixed effects) to model latent spatial structure and improve inference.
  - If theory suggests outcome feedback or neighbor-to-neighbor causal spillovers, use SAR/SDM and report direct/indirect impacts (and handle identification/instruments as needed).

Kriging for Point-Reference Data

Concept: **Interpolate a spatial process at unsampled locations** using a covariance (or semivariogram) model.



Kriging — Quick Overview

- Prediction at an unsampled location as a **weighted average** of nearby observations.
- The weights are chosen to be **optimal** so the predictor is **unbiased** and has **minimum variance** given the assumed covariance/variogram model.
- Outputs: **a point prediction** (the kriged value) and a **per-location uncertainty measure** (the kriging variance).

Ordinary Kriging — prediction system

Objective: predict Z at location s0 as a linear unbiased estimator with minimum variance.

Prediction (linear estimator)

$$\hat{Z}(s_0) \;=\; \sum_{i=1}^{n} \lambda_i \, Z(s_i)$$

Unbiasedness constraint

$$\sum_{i=1}^{n} \lambda_i \;=\; 1$$

- How to interpret the kriging weights and prediction
  - **Weights reflect the spatial configuration and the fitted variogram/covariance**: closer and more informative neighbors get larger weights; distant or noisy neighbors get low weights.
  - **The prediction is not a simple smoothing average**: it accounts for spatial correlation structure (range, sill, nugget) so it adapts to the effective spatial horizon.

- How to interpret the kriging variance
    - **The kriging variance is the model-based estimate of uncertainty for the predicted value at that location** (smaller where observations are dense and correlations are strong; larger in sparse areas or beyond the range).
    - It is a function of the variogram/covariance and sampling geometry only — **it does not depend on the observed values** themselves (only on their **locations** and the fitted model).
    - Use it to build approximate prediction intervals (under normality or large-sample approximations) and to prioritize further sampling (**high variance → useful new sample**).

Practical Workflow

1. **Compute empirical semivariogram** (choose distance bins).
2. **Fit** theoretical model (spherical, exponential, Gaussian, Matérn).
3. Build covariance/variogram matrix and **solve** kriging system $\rightarrow$ **weights** $\rightarrow$ **predictions**.
4. Optionally **detrend** or include trend terms (universal kriging).
5. **Validate** with spatial CV (blocked CV) or leave-one-out.

Outputs & Validation

- Outputs: predicted value and kriging variance at each target location.
- Validation: spatial CV, LOO errors, and check residual variogram (should be near pure nugget).

Strengths

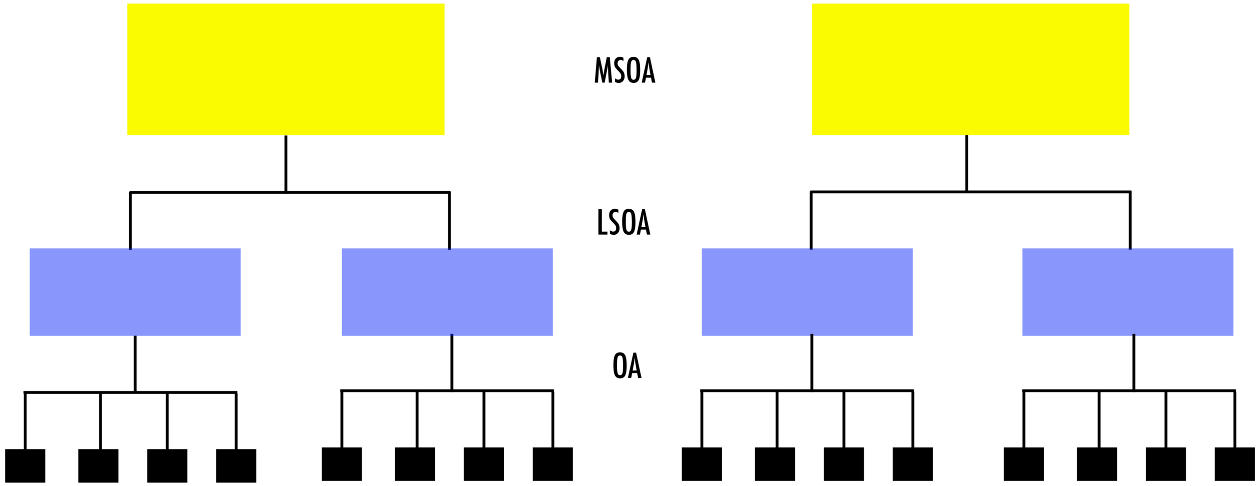
- Optimal linear interpolation under model assumptions.
- Explicit **uncertainty quantification** (kriging variance).
- **Flexible**: supports trends and auxiliary variables (cokriging / regression kriging).

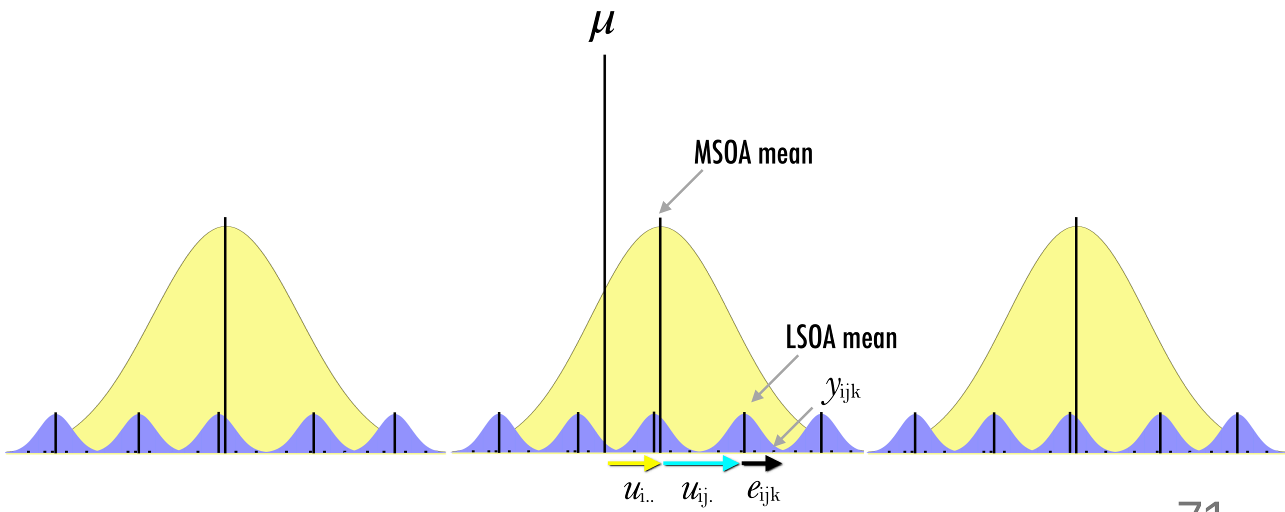
Limitations & Scalability

- **Sensitive** to variogram choice and fitting.
- Assumes **stationarity** (or **local stationarity**); detrend if needed.
- **Computationally heavy** for large n.

Note: Hierarchical Spatial Models (optional)

- **Multi-level** / **mixed-effects** frameworks to model grouped data with spatial dependence (e.g., CAR/ICAR for areal data; GP/SPDE for point-referenced).
- Out of scope for today; recommended readings: Multilevel Modelling - Part 1 Multilevel Modelling - Part 1





Model Structure: Recap

Advantages

- **Theoretical foundation**: Grounded in spatial process theory
- **Interpretable parameters**: Range, spillover intensity, etc.
- **Uncertainty quantification**: Full probabilistic framework
- **Scientific insight**: Direct vs indirect effects, spatial mechanisms

Limitations

- **Strong assumptions** (stationarity, parametric dependence forms)
- **Computational complexity** (dense matrix ops, factorization)
- **Model selection challenge** (many competing variants)
- **Scalability concerns**

When to Prioritize Model Structure

- Need scientific mechanism insight (spillovers, diffusion)
- Formal inference / hypothesis testing required
- Uncertainty quantification critical
- Spatial structure strong & theory-supported
- Dataset size manageable or sparse methods available

Strategy 3: Spatial Regularization

Definition: **Add spatial smoothness penalties to learning objectives**
Core Idea: Encourage **similar predictions** for **nearby locations**
Mathematical Framework:

$$\min_{\theta} \underbrace{\sum_{i} \ell(y_i, f_\theta(x_i))}_{\text{Data fit}} + \lambda \underbrace{\sum_{(i,j)\in E} w_{ij} \|\theta_i - \theta_j\|^2}_{\text{Spatial penalty}}$$

Regularization Types:

- **Parameter smoothing:** Local model parameters vary smoothly
- **Prediction smoothing:** Similar predictions for neighbors
- **Graph total variation:** Piecewise-constant spatial structure
- **Embedding regularization:** Smooth latent representations

Flexibility: Can be added to most objective functions

Regularization — Advantages and Limitations

Pros:

- **Model flexibility**: Compatible with any differentiable objective
- **Computational efficiency**: Standard optimization with additional term
- **Tunable smoothness**: Control via regularization parameter λ
- **Scales well**: Linear complexity in graph edges

Cons:

- **Hyperparameter tuning**: Choosing λ and graph structure
- **Over-smoothing risk**: May blur important spatial discontinuities
- **No probabilistic interpretation**: Point estimates, no uncertainty
- **Graph dependence**: Results sensitive to spatial graph construction

When to Use:

- Complex, non-parametric models (neural networks, trees)
- Large datasets where full spatial models infeasible
- When local smoothness assumption reasonable

Strategy Comparison and Selection Guide

Aspect	Feature Engineering	Model Structure	Regularization
Ease of Implementation	High	Medium	Medium
Computational Cost	Low	High	Medium
Theoretical Foundation	Low	High	Medium
Uncertainty Quantification	None	Full	Limited
Scalability	Excellent	Poor	Good
Interpretability	High	High	Medium

Decision Framework:

1. **Start with feature engineering** for baseline and rapid iteration
2. **Use model structure** when interpretation and uncertainty quantification are critical
3. **Apply regularization** for complex models and large datasets
4. **Combine approaches** for best performance (hybrid methods)

Implementation Decision Framework

Step 1: Data Diagnosis

- Visualize spatial patterns and covariate relationships
- Test for spatial dependence (Moran's I, variograms) and check stationarity
- Note mechanisms to guide model choice (spillover, diffusion, shared context)

Step 2: Method Selection

- Feature engineering — fast baselines and easy integration with ML
- Model structure — SAR/SDM/SEM/CAR when theory or inference requires it
- Regularization / hierarchical models — for scale, complexity, or multilevel processes

Step 3: Implementation Strategy

- Try multiple plausible Ws for sensitivity
- Recompute graph-derived features (e.g., WX) inside CV folds when needed
- Use spatial cross-validation for performance and hyperparameter tuning
- Balance interpretability vs compute cost; document W, transforms, and seeds

Step 4: Model Validation

- Residual checks: Moran’s I and variogram of residuals (should be near zero)
- Compare approaches on spatial CV and uncertainty calibration (pred intervals)
- Sensitivity checks for W, model form, and key parameters

Python Software Ecosystem

Core Spatial Analysis Libraries

Spatial Weight Matrices & Econometrics:

- `libpysal`: Spatial weights, autocorrelation measures
- `spreg`: SAR, SEM, SDM estimation
- `pysal`: Comprehensive spatial analysis suite

Geostatistics & Kriging:

- `scikit-gstat`: Variogram modeling and kriging
- `pykrige`: Ordinary, universal, regression kriging
- `gstools`: Advanced geostatistical modeling
- `geostatspy`: Educational geostatistics workflows

Hierarchical & Bayesian Models:

- `pymc`: Bayesian spatial models with MCMC
- `stan`: Advanced hierarchical spatial modeling
- `inla`: Fast approximate Bayesian inference (R interface)