Empirical interpolation: thin plate splines

D G Rossiter

Cornell University

November 20, 2016

Outline

1. Spatial prediction

2. 1D Splines
   - 1D natural splines
   - 1D smoothing splines

3. 2D: Thin-plate splines

4. Comparing kriging and spline interpolation

5. References

Spatial prediction

Aim: Prediction of (unknown) values at **unsampled** points. . .
. . . based on (known) values at **sampled** points

- **interpolation**: inside the convex hull of observations
- **extrapolation**: outside

Geostatistical vs. empirical prediction

geostatistical: predict based on a geostatistical **model** with fitted **parameters**

- e.g., parameters of a variogram
- e.g., trend surface (polynomial of coördinates)

empirical: predict by an **empirical** adjustment to known points

- e.g., Thiessen polygons (a.k.a. Voronoi tessellation, nearest neighbour)
- e.g., Triangulated irregular network (TIN)
- here: **Thin-plate splines**

- Similar to **feature-space** model-based statistical vs. empirical prediction
  - e.g., linear regression model vs. random forest

Assumptions of geostatistical methods

Ordinary Kriging (OK) the observations are the result of a locally spatially-correlated **second-order stationary random process**

- variogram model

Trend surface (TS) the observations are the result of a regional process

- polynomial regression on the coördinates

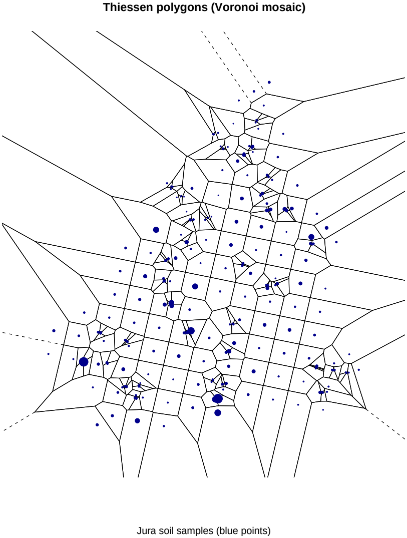
Universal Kriging (UK) some of the variation from regional processes, some residual local variation explained by a locally spatially-correlated second-order stationary random process

- residual (from trend surface) variogram model

Empirical methods of spatial prediction

- No model, just adjustment to observations
- Different methods have different adjustments
  - **Theissen polygons**: predict with value of nearest observation
  - **TIN**: compute position on triangular facet, predict from three corner observations on the (sloping) triangular facet
  - **Splines**: fit a "smooth" surface to observations, predict on surface

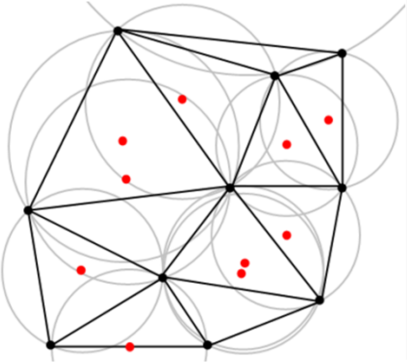
Empirical: Thiessen polygons

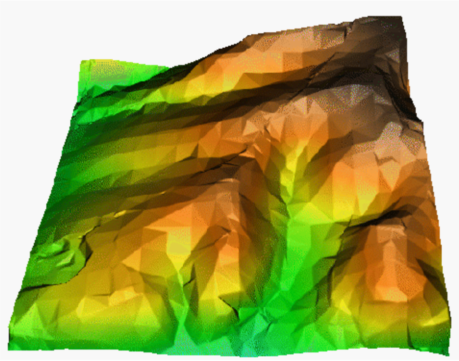


Predict at all locations within in each polygon with value of centroid point

Sharp boundaries between nearest-neighbour polygons

Empirical: TIN





Control points

Interpolate on each triangular facet

Source: https://docs.qgis.org/2.2/en/docs/gentle_gis_introduction/spatial_analysis_interpolation.htm

Splines

Splines: **piecewise** "**smooth**" functions to **approximate** a set of **control points**

1D fit a "smooth" curve to a set of points along a **transect**

2D fit a "smooth" surface to a set of points in **2D space**

In both cases:

- points have known **coördinate** (1D) or **coördinates** (2D)
- points have known **attribute value** which is to be interpolated

We begin with 1D splines to illustrate the smoothing procedure

1D Splines

A type of **basis expansion**:

- a **piecewise polynomial** function
  - each piece is defined only over some range
- pieces are joined at **knots**
- they have a defined degree of **continuity** between the pieces
- most common: 4th order: **continuous 1st and 2nd derivatives**
  - values, slopes and curvatures match at the knots
- These look smooth to the human eye and also correspond to an intuitive concept of smoothness.

Piecewise cubic polynomial

One for each piece of the range, joined at knots.
Example: with one knot at c:

$$y_i = \begin{cases} \beta_{01} + \beta_{11} x_i + \beta_{21} x_i^2 + \beta_{31} x_i^3 + \beta_4 h(x, \xi) + \varepsilon_i & : \mathrm{if} x_i < c; \\ \beta_{02} + \beta_{12} x_i + \beta_{22} x_i^2 + \beta_{32} x_i^3 + \beta_4 h(x, \xi) + \varepsilon_i & : \mathrm{if} x_i \geq c. \end{cases} \tag{1}$$

The β must be chosen so that the values, 1st, and 2nd derivatives are equal at the knot. This is ensured with the $\beta_4 h(x, \xi)$ term, as is explained below.

Spline basis representation

A cubic spline with K knots; this can be fit with least squares.
For each known observation (x_i, y_i):

$$y_i = \beta_0 + \beta_1 b_1(x_i) + \beta_2 b_2(x_i) + \cdots \beta_{K+3} b_{K+3}(x_i) + \varepsilon_i \qquad (2)$$

The basis uses one *truncated power basis* per knot; this is the h function of Equation (1):

$$h(x, \xi) = (x - \xi)^3_+ = \begin{cases} (x - \xi)^3 & \text{if} x > \xi \\ 0 & \text{otherwise} \end{cases} \qquad (3)$$

Smoothing vs. non-smoothing splines

- A **non-smoothing** spline goes through each knot, i.e., the values at each knot are considered exact
  - Knots can be at observation points or set by the analyst
  - It is usual to set up knots at equal intervals over the range of the predictor
  - The resulting curve is still smooth but the knots are respected
- A **smoothing** spline can deviate from the knots to make a smoother curve, based on a **roughness penalty**

1D natural splines

- Fit by solution of a linear equation expressing the constraints
- Includes constraint that function is linear outside the range of observations
- Spline **basis**: $n \times k$ weights matrix (k knots at each of n observation)
- Basis values are the coefficients of a cubic polynomial
- Basis values $\neq 0$ only for piecewise functions which affect that station
- y vector is values of dependent variable at the knots
- X design matrix are the values of the basis function evaluated for the independent variable at the knots

Solution by ordinary least squares (orthogonal projection):

$$y = XB + \varepsilon \longrightarrow \hat{B} = (X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}y \tag{4}$$

Computing a 1D natural spline

To compute a spline, the analyst must specify:

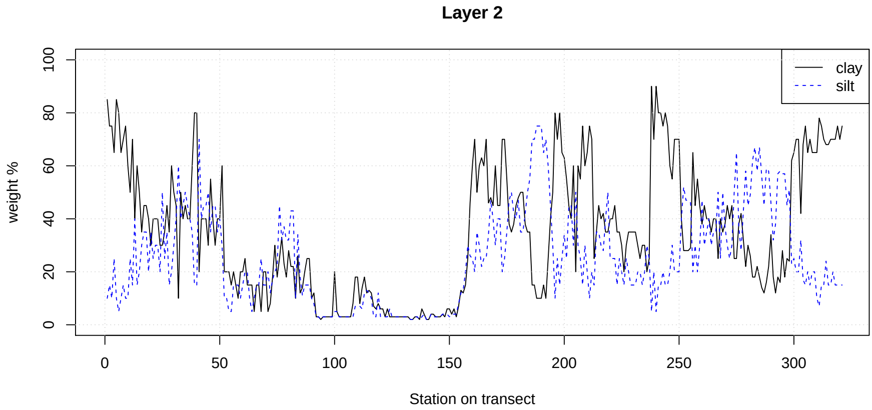
1. the **order** of the spline; most commonly cubic splines (4th order);
2. the **number** of knots;
3. the **placement** of knots in the range of the variable.

The knots can be at the **known data points**

More common: **equally-spaced knots** through the range of the sequence, make a linear interpolation from adjacent known points, and fit the spline through the knots.

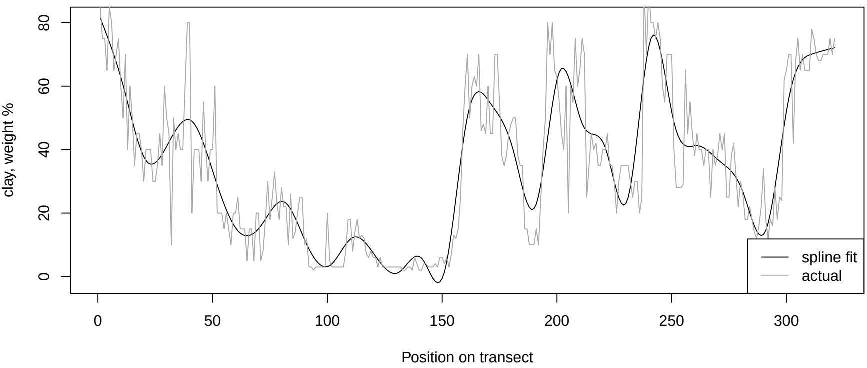
Degree of smoothness determined by the number of knots

Example: particle-size classes along a transect



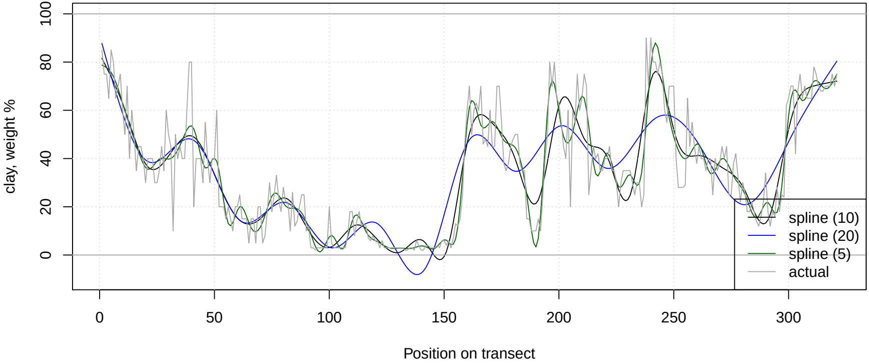
source: Webster, R.., & Cuanalo, H.E. de la C. 1975. *Soil transect correlograms of North Oxfordshire and their interpolation*. **Journal of Soil Science** 26, 176–194.

Natural spline fit to clay concentration – 1



Knots every 10 stations (320 stations total, so 32 knots)

Natural spline fit to clay concentration – 2



Different numbers of knots: every 5, 10, 20 stations

More knots → closer to data values, less smoothing, closer to extremes

1D smoothing splines

Optimal solution to this minimization problem:

$$\min_f \left\{ \sum_{i=1}^{N} \{y_i - f(x_i)\}^2 + \lambda J[f] \right\} \tag{5}$$

- f is the smoothing spline function (e.g., cubic piecewise polynomial); so $\sum_{i=1}^{N} \{y_i - f(x_i)\}^2$ is the **loss function**; we want to minimize this.
- J is the **penalty term**; λ is a **tuning parameter** which controls how important it is:
  - $\lambda = 0$: no roughness penalty, data will be fit exactly
  - $\lambda \to \omega$: the solution is an OLS linear fit to the data

So smoothness is determined by the **penalty**, not by number of knots; **all data values are used**

1D penalty function

$$J[f] = \int_{\mathbb{R}} \left\{ f''(t) \right\}^2 dt \tag{6}$$

This is the integral of the squared second derivative over the interval

So, so the more curvature, the higher the penalty.

(In practice the integral is discretized)

Determining the roughness penalty

- can specify directly, but more common to . . .
- . . . determine by **generalized cross-validation**
    - Wahba, G. 1990. Spline models for observational data. Society for Industrial and Applied Mathematics, Philadelphia.
    - Remove one or more points, re-fit without them, compute prediction error, summarize (see next slide)
- i.e., try different penalties and pick the one which **minimizes the cross-validation error**
    - penalty too *small*: will more closely fit remaining data but will likely be far from the removed cross-validation point
    - penalty too *large*: will be far from all points

Cross-validation

- A general term for:
  1. **removing** one or more observations;
  2. **fitting** a function or applying a model with the remaining observations;
  3. **predicting** with this function or model using the values of the predictor variables at the removed observations;
  4. computing the **prediction error** $(z - \hat{z})$ at the removed points
  5. **summarizing** these errors for all splits of the observation set.
- a reasonable estimate of the expected error if model is applied to another dataset
- K-fold cross-validation: split the dataset into K parts
- Leave-one-out cross-validation (LOOCV): $K = n$
- **Generalized** cross-validation: a quick method for linear fits with squared-error loss as the optimum (as here)

Smoothing vs. natural splines

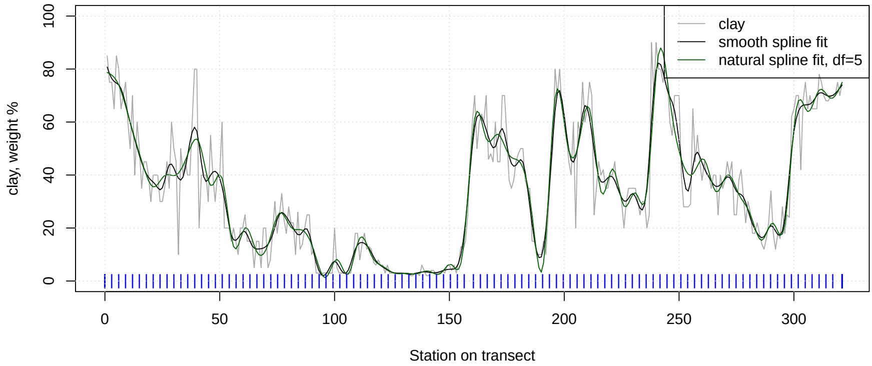
- The function that minimizes Equation (5) is a natural cubic spline with knots at all the data points
- However, it is a “shrunken” version of the natural cubic spline computed directly
- The tuning parameter λ controls the amount of shrinkage

Computing a smoothing spline

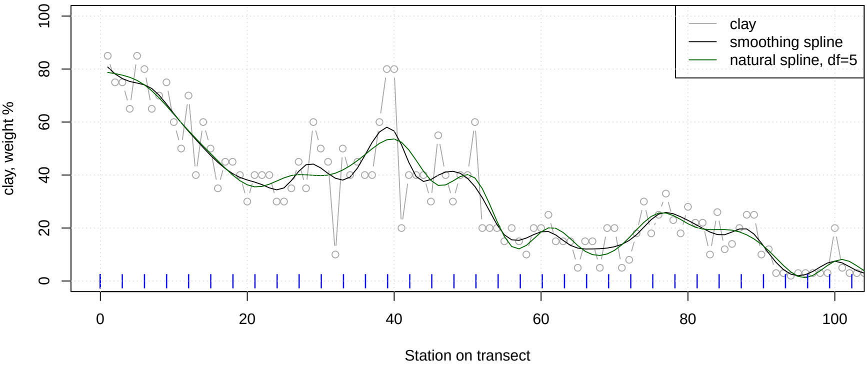
```
> # smooth.spline is in default 'stats' package
> (spl.fit <- smooth.spline(ds$clay2))

Smoothing Parameter  spar= 0.2544574
                     lambda= 1.204002e-07 (12 iterations)
Equivalent Degrees of Freedom (Df): 73.46845
Penalized Criterion: 18226.75
GCV: 95.48895
```

Df equivalent to a non-smooth fit with $320/73.5 = 4.35$ stations between knots (c.f. natural spline with knots every 5 stations)

Smoothing spline for clay concentration



Smoothing spline for clay concentration: detail

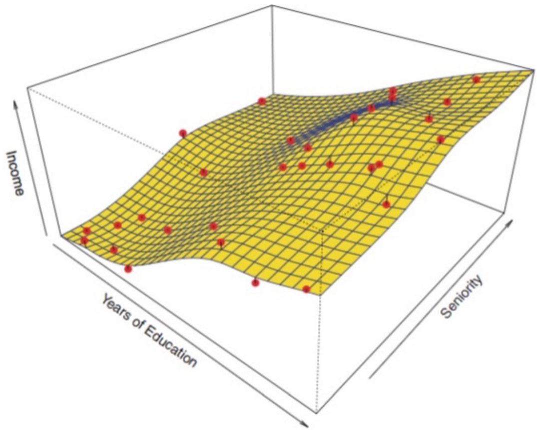


2D: Thin-plate splines (TPS)

We now consider splines in 2D (i.e., over a **surface**).

- The analogy is with a thin (so, flexible) plate that is warped to (more or less) fit the observations.
- This can range from very **“rigid”** . . .
  - i.e., a single surface: the least-squares plane of a first-order trend surface
- . . . to very **“flexible”**
  - i.e., perfectly fitting every observation.
- We want something in between: local *noise* should be somehow locally removed while local *structure* should be preserved.
- Also called **“minimum curvature”** method

Example thin-plate spline fit



Source: Fig. 2.5 in James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). **An introduction to statistical learning: with applications in R**. New York: Springer.

Fitting a TPS

- Fit is based on the k data points with known coördinates and attribute values.
- These can be described by $2(k+3)$ parameters
  - six overall **affine transformation** parameters to centre the function in 2D
  - $2 \cdot k$ **links to the control points**.
- Aim: **minimize the residual sum of squares** (RSS) of the fitted function, subject to a **constraint** that the function be "smooth"
- Constraint is expressed by a **roughness penalty** which balances the fit to the observations with smoothness.

MInimization problem

2D equivalent of 1D Equation (5):

$$\min_f \sum_{i=1}^{N} \{y_i - f(\mathbf{x_i})\}^2 + \lambda J[f] \tag{7}$$

where J is the penalty function and λ controls how important it is.

- $\lambda = 0$: no roughness penalty, data will be fit exactly
- $\lambda \to \omega$ the solution approximates the least-squares plane, i.e., the trend surface averaged over all the points.

2D roughness penalty

2D equivalent of 1D Equation (6):

$$J[f] = \int_{\mathbb{R}} \int_{\mathbb{R}} \left[\left(\frac{\partial^2 f(\mathbf{x})}{\partial x_1^2} \right)^2 + 2 \left(\frac{\partial^2 f(\mathbf{x})}{\partial x_1 \partial x_2} \right)^2 + \left(\frac{\partial^2 f(\mathbf{x})}{\partial x_2^2} \right)^2 \right] \mathrm{d}x_1 \mathrm{d}x_2 \tag{8}$$

where (x_1, x_2) are the two coördinates of the vector $\mathbf{x}$.

- Double integral is **total curvature**
- In practice, this is **discretized**
- Penalty determined by **cross-validation**

2D TPS solution

$$f(\mathbf{x}) = \beta_0 + \beta^T \mathbf{x} + \sum_{j=1}^{N} \alpha_j h_j(\mathbf{x}) \qquad (9)$$

- $\mathbf{x}$ is the 2D position vector
- the β account for the overall trend
- the α are the coefficients of the warping, one for of the N control points

The value $f(\mathbf{x})$ at each prediction point $\mathbf{x}$ is a result of the overall affine transformation and the sum of N basis functions evaluated at the point.

Radial basis functions

The set of functions $h_j(\mathbf{x})$ is the **basis kernel**, also called a **radial basis function** (RBF), for thin-plate splines:

$$h_j(\mathbf{x}) = \|\mathbf{x} - \mathbf{x}_j\|^2 \log \|\mathbf{x} - \mathbf{x}_j\| \tag{10}$$

- The norm distance $r = \|\mathbf{x} - \mathbf{x}_j\|$ is also called the **radius** of the basis function.
- The norm is usually the Euclidean (straight-line) distance.
- Note that the value decreases as the square of the distance from a control point.
- The proper radius is evaluated by cross-validation, based on the roughness penalty.

Computing a TPS

```
> # example dataset
> library(gstat)
> data(jura)
> class(jura.pred) # a dataframe, coordinates are fields
> # from the National Center for Atmospheric Research
> library(fields)
> # Tps expects coordinates as a matrix
> jura.pred$coords <- matrix(c(jura.pred$Xloc,
                               jura.pred$Yloc),
                               byrow=F, ncol=2)
> # Tps is a special case of the fields 'Krig' function
> surf.1 <-Tps(jura.pred$coords, jura.pred$Co)
```

Computed TPS – 1/3

```
Number of unique points:                259
Number of parameters in the null space 3
Parameters for fixed spatial drift     3
Effective degrees of freedom:          156.1
Residual degrees of freedom:           102.9
MLE sigma                              0.9095
GCV sigma                              1.095
Smoothing parameter lambda             1.21e-05
```

- Smoothing parameter λ
- fit has reduced the degrees of freedom
- MLE is the standard error of the internal goodness-of-fit
- GCV is the standard error of the cross-validation

Computed TPS – 2/3

```
Residual Summary:
      min      1st Q    median      3rd Q        max
-3.151000 -0.266200 -0.006246  0.201500  2.901000
```

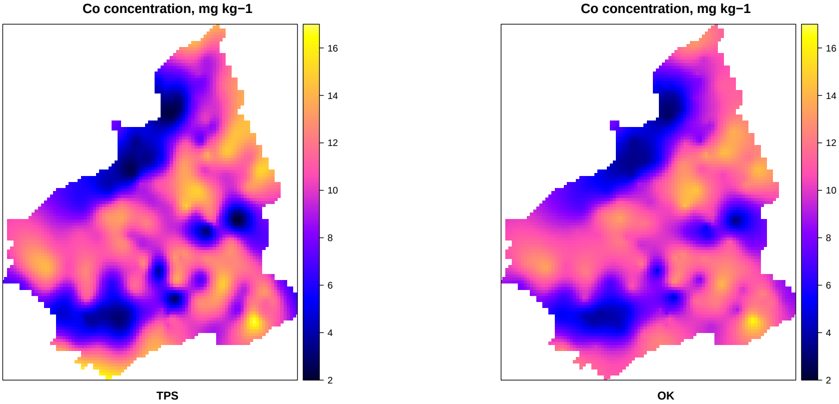
- Residuals from TPS fit

Computed TPS – 3/3

```
Covariance Model: Rad.cov
DETAILS ON SMOOTHING PARAMETER:
 Method used:   GCV    Cost:  1
   lambda        trA        GCV    GCV.one GCV.model       shat
1.210e-05 1.561e+02 3.016e+00 3.016e+00        NA 1.095e+00
```

- **Covariance model** is the spatial structure
- Here we chose the radial basis function
- The Generalized Cross Validation (GCV) at this λ is the **external** lack of fit (i.e., estimated prediction error).

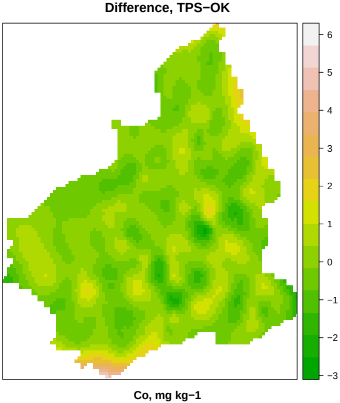
Kriging vs. splines, Jura cobalt



Thin-plate splines

OK

Difference TPS – OK, Jura cobalt



OK adjusted more locally (smaller “hot spots”)

TPS predicted more extreme high and low

This depends on the **fitted variogram model**, especially the range of spatial dependence

Also depends on the **roughness penalty** of TPS determined by cross-validation

Relation between spline and kriging interpolation

"... **kriging** is performed in **two steps**,

(1) a **structural analysis**, which fits a covariance and a degree of trend to the variable under study, then
(2) the **interpolation** itself, which uses the results of the structural analysis.

With **spline** interpolation, **no preliminary structural analysis** is performed . . . this should result in a loss of accuracy of splines compared to kriging."

– O. Dubrule. *Comparing splines and kriging*. **Computers & Geosciences**, 10(2-3):328–338, 1984

Advantages of TPS

1. No need to select a variogram model
   - but can select a different form of the radial basis function
2. No need to parameterize the variogram model
3. Automatic selection of optimum smoothness for minimum curvature
4. No need to decide on order of trend surface
   - But there is still an implicit assumption of **2nd order stationarity**, because a single smoothness parameter over the whole map
5. Can be used with small datasets where it is impossible to reliably fit a variogram model

Disadvantages of TPS

1. No model of spatial structure (whether a trend surface or local autocorrelation) to **interpret** in terms of **processes**
2. No internal model of prediction error (e.g., kriging prediction variance)

Choosing between approaches

"Kriging, by **minimizing the estimation variance**, is designed to provide estimates which are as close as possible to the actual values. . .
. . . Spline interpolation, by **minimizing the total curvature**, is designed to provide maps which have nice cosmetic properties."

– O. Dubrule. Reply: comparing splines and kriging. Computers & Geosciences, 12(5):729–730, 1986. doi: 10.1016/0098-3004(86) 90051-8

"If the map is going to be used for **future calculations**, one needs accuracy; kriging is good in this situation.

If one wants to quickly obtain a clear map showing the **main features of the variable**, splines are a good tool."

References: theory

- Hastie T, Tibshirani R, Friedman JH (2009) The elements of statistical learning data mining, inference, and prediction, 2nd ed. Springer, New York
- Wood, S. N. (2003). Thin plate regression splines. Journal of the Royal Statistical Society Series B-Statistical Methodology, 65, 95–114. http://doi.org/10.1111/1467-9868.00374
- Mitasova H, Mitas L (1993) Interpolation by regularized spline with tension: I. Theory and implementation. Mathematical Geology 25:641–655. doi: 10.1007/BF00893171
- James G, Witten D, Hastie T, Tibshirani R (2013) An introduction to statistical learning: with applications in R. Springer, New York

References: applications

- Hofierka, J., Mitasova, H., Parajka, J., & Mitas, L. (2002). Multivariate interpolation of precipitation using regularized spline with tension. Transactions in GIS, 6(2), 135–150. http://doi.org/10.1111/1467-9671.00101
- Hutchinson MF (1995) Interpolating mean rainfall using thin plate smoothing splines. International Journal of Geographical Information Science 9:385 – 403. x1
- Mitasova H, Hofierka J (1993) Interpolation by regularized spline with tension: II. Application to terrain modeling and surface geometry analysis. Mathematical Geology 25:657–669. doi: 10.1007/BF00893172