Principal curves: A versatile tool for data compression and beyond

Jochen Einbeck

Department of Mathematical Sciences, Durham University

jochen.einbeck@durham.ac.uk

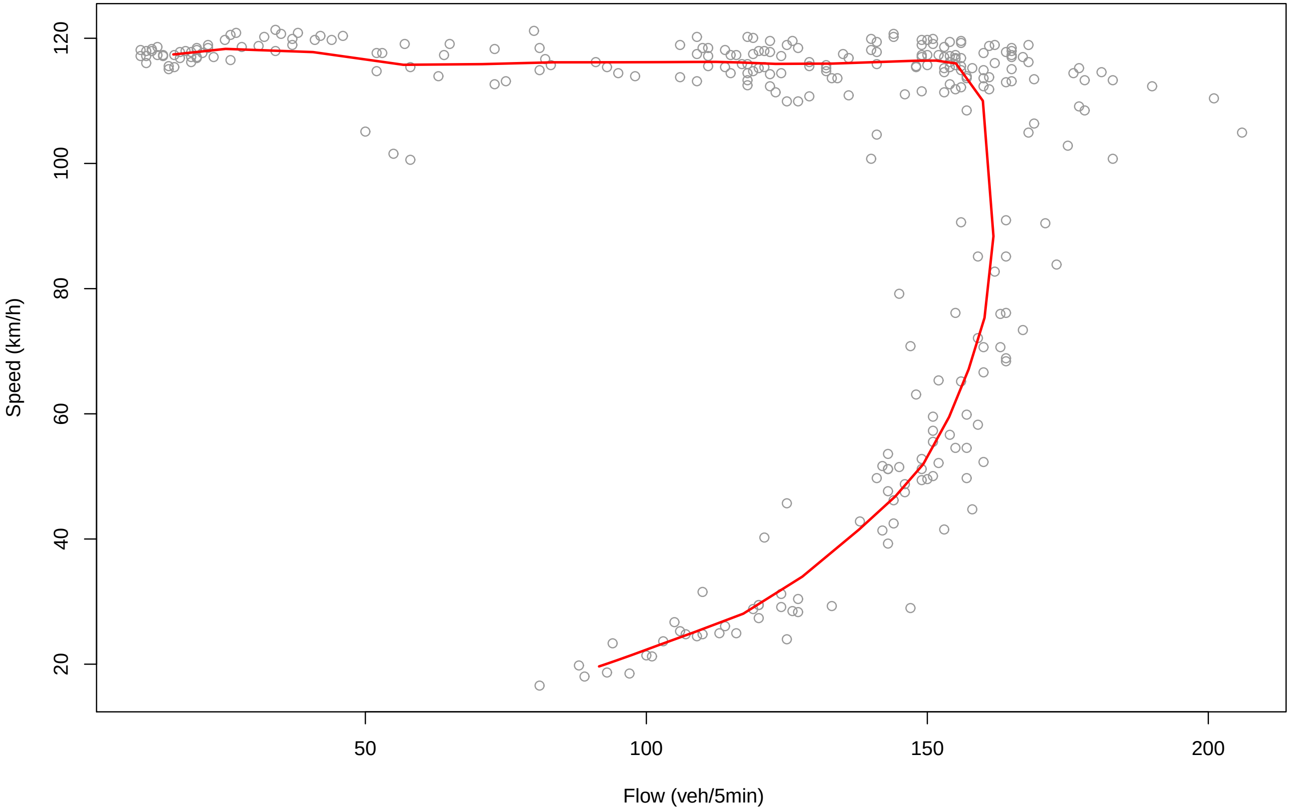
joint work with Ludger Evers (University of Glasgow),

Edinburgh, 22nd of October 2010

Principal curves

Principal Curves are smooth curves passing through the 'middle' of a multivariate data cloud $X = (X_1, \ldots, X_n)$, where $X_i \in \mathbb{R}^d$.

Example: Speed-Flow data from a Californian "freeway".



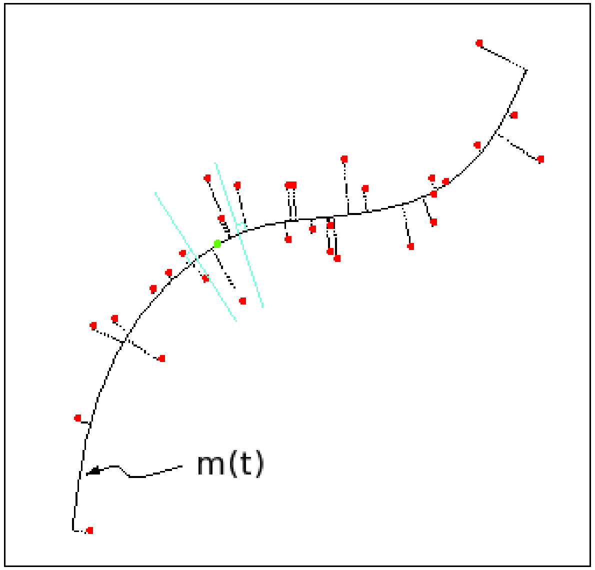
Principal curves: Original definition

Hastie & Stützle (HS, 1989) define each point on the principal curve m as the average of all points which project there ('self-consistency'), i.e.

$$m(t) = E(X|t_m(X) = t)$$

where $t_m(X)$ is the projection index of X onto the curve m.

- If the principal curve is linear, then it is a principal component.
- If a curve $m(t)$ is self-consistent, it is a critical point of the distance function
$$\triangle(m) = E\left(\inf_t ||X - m(t)||^2\right).$$
- However, it was later shown that the critical point is actually just a saddle point of $\triangle(m)$.
- If $X = g(T) + \epsilon$ with T uniform and $\epsilon \sim N(0, \sigma^2 I)$, then generally $m \neq g$!



(from: Hastie et al, 2001))

Types of principal curves

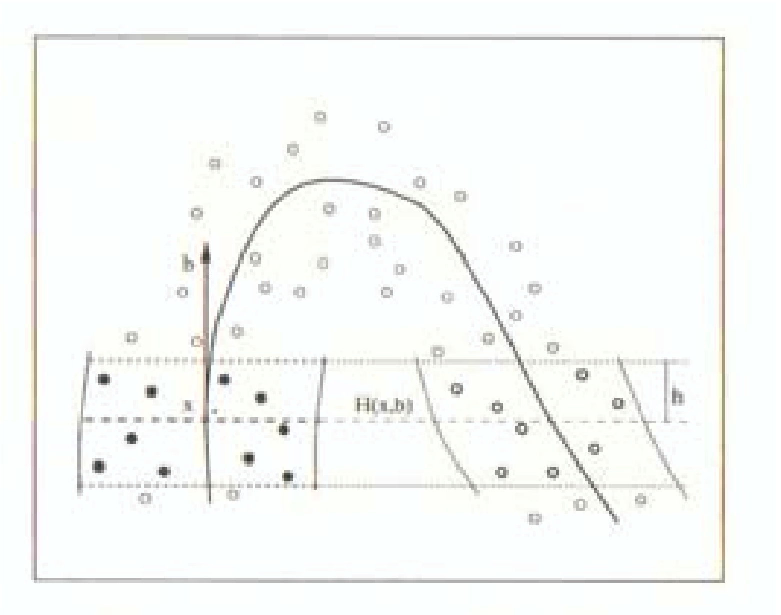
Today exist a variety of different notions of principal curves, which vary essentially in how the “middle” of the data cloud is defined/found:

- Global (‘top-down’) algorithms start with an initial line (usually the 1st PC line) and bend this line or concatenate other lines to it until some convergence criterion is met (HS, Tibshirani 1992, Kégl et al. 2002).
  - Allows theoretical analysis.
  - Goes wrong if initial order of projection indices is not right.
  - Extension to branched or disconnected data clouds difficult.
- Local (‘bottom-up’) algorithms estimate the principal curve locally moving step by step through the data cloud (Delicado 2001, Einbeck et al. 2005).
  - More flexible, but far more variable.
  - Extend straightforwardly to branched and disconnected data.
  - Theoretical investigations rather difficult.

Delicado's PCOPs

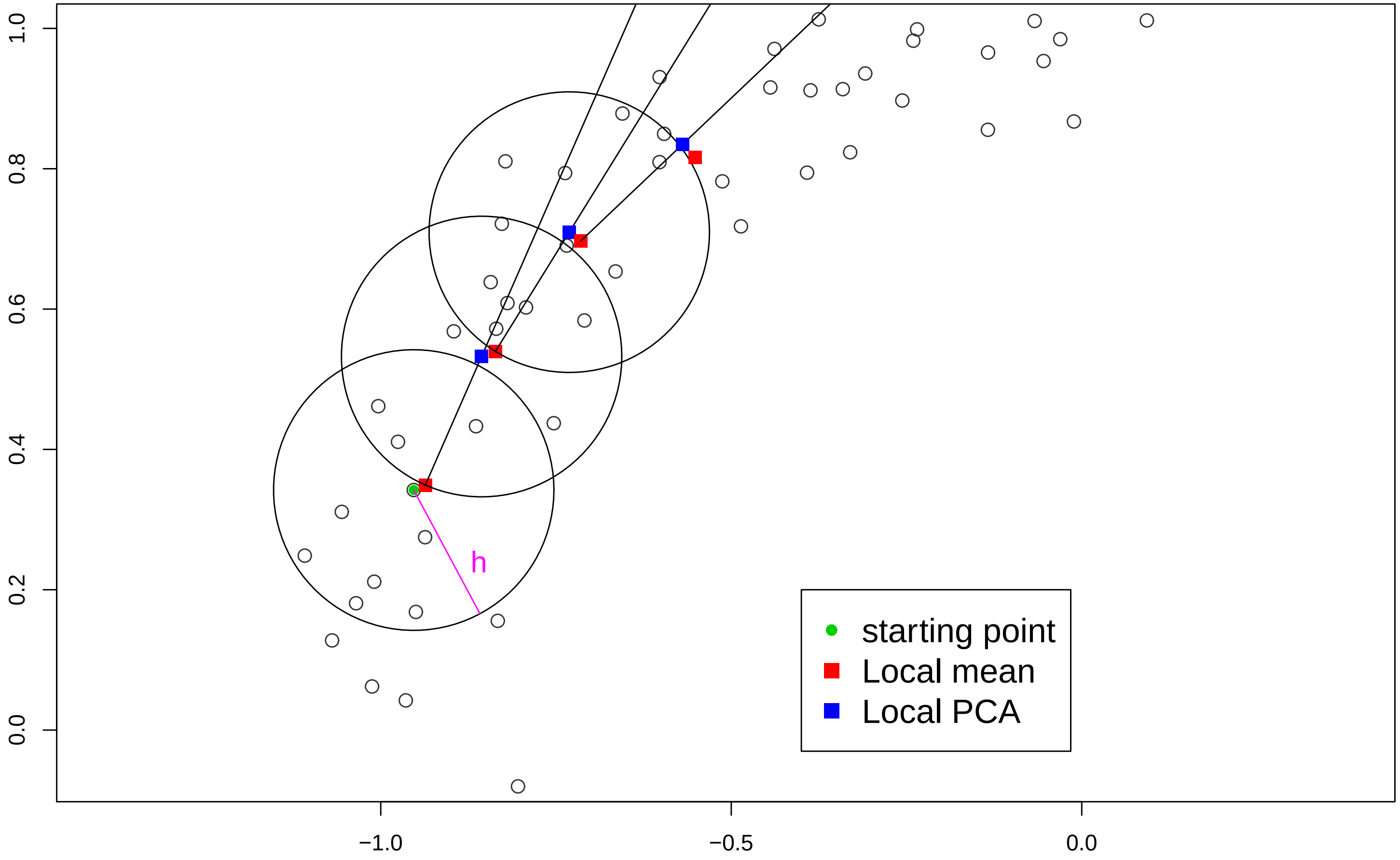
Delicado (2001) defines *principal curves of oriented points* (PCOPs) as a sequence of fixed points of the function $\mu^*(x) = E(X|X \in H)$, where H is the hyperplane through x minimizing locally the variance of the data points projected on it.

- Works fine for most (not too complex) data sets.
- Mathematically elegant
- However, quite complicated and computationally demanding.
- Requires a cluster analysis at every point of the principal curve.



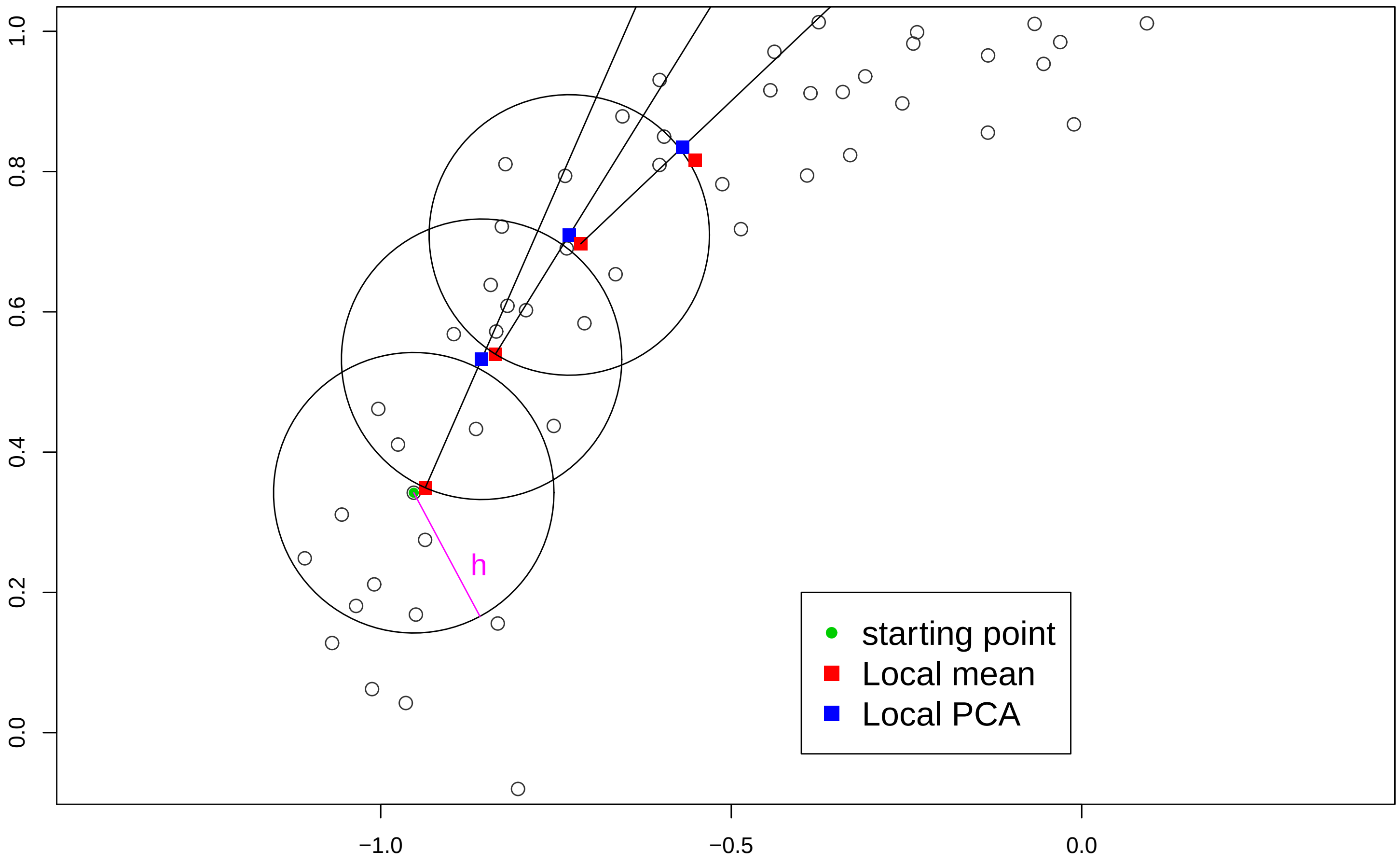
Local principal curves (LPC)

Einbeck, Tutz & Evers (2005): Calculate alternately a local mean and a first local principal component, each within a certain bandwidth h.



Local principal curves (LPC)

Einbeck, Tutz & Evers (2005): Calculate alternately a local mean and a first local principal component, each within a certain bandwidth h.



- The LPC is the series of local means.

Algorithm for LPCs

- Given: A data cloud $X = (X_1, \ldots, X_n)$.

1. *Choose a starting point* x_0*. Set* $x = x_0$*.*
2. *At* x*, calculate the local center of mass* $\mu^x = \sum_{i=1}^n w_i X_i$*, where* $w_i = K_H(X_i - x)X_i / \sum_{i=1}^n K_H(X_i - x)$*.*
3. *Compute the* 1^{st} *local eigenvector* γ^x *of*
$$\Sigma^x = \sum_{i=1}^n w_i (X_i - \mu^x)(X_i - \mu^x)^T$$
4. *Step from* μ^x *to* $x := \mu^x + t_0 \gamma^x$*.*
5. *Repeat steps 2. to 4. until the* μ^x *remain constant. Then set* $x = x_0$*, set* $\gamma^x := -\gamma^x$ *and continue with 4.*

- The sequence of the local centers of mass μ^x makes up the local principal curve (LPC).

Algorithm for LPCs

- Given: A data cloud $X = (X_1, \dots, X_n)$.

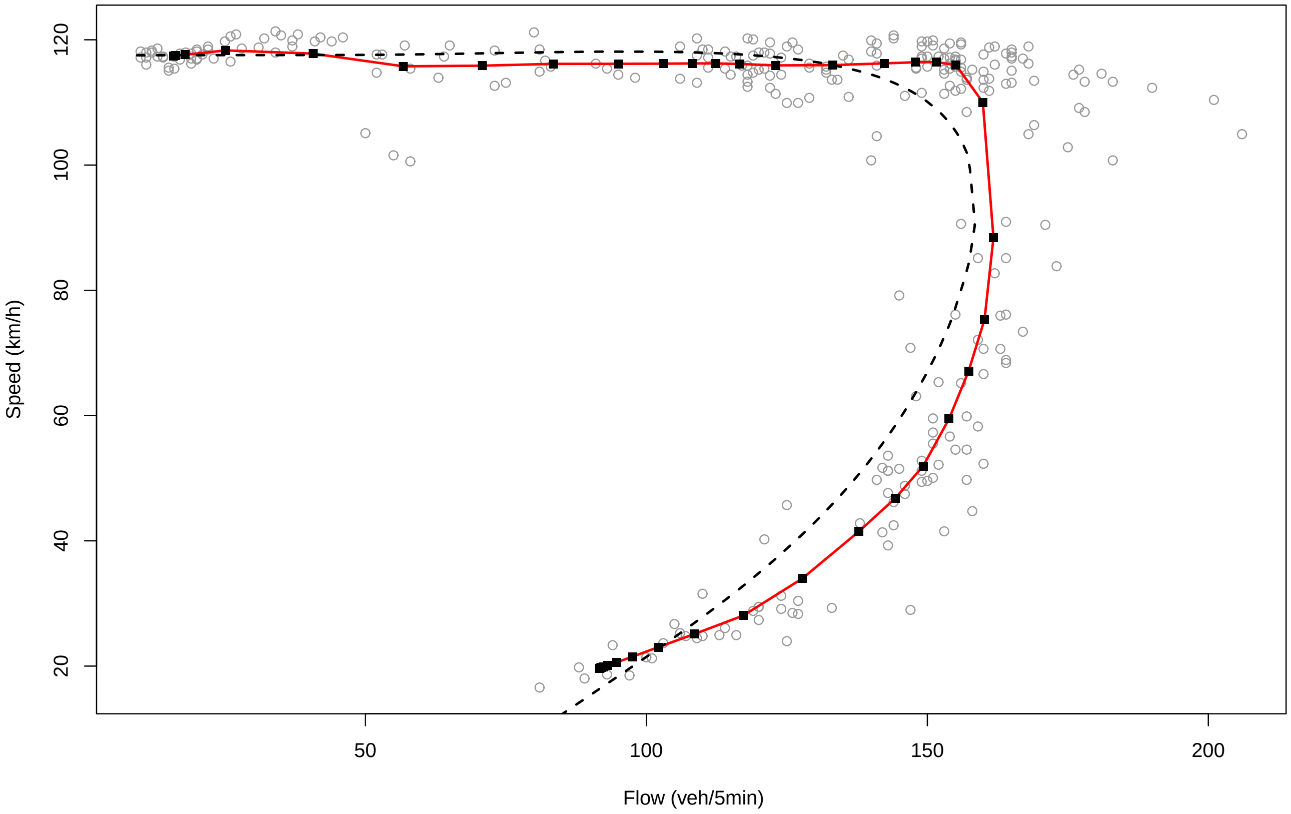
1. *Choose a starting point* x_0. *Set* $x = x_0$.
2. *At* x, *calculate the local center of mass* $\mu^x = \sum_{i=1}^n w_i X_i$, *where* $w_i = K_H(X_i - x)X_i / \sum_{i=1}^n K_H(X_i - x)$.
3. *Compute the* 1^{st} *local eigenvector* γ^x *of*
$$\Sigma^x = \sum_{i=1}^n w_i(X_i - \mu^x)(X_i - \mu^x)^T$$
4. *Step from* μ^x *to* $x := \mu^x + t_0\gamma^x$.
5. *Repeat steps 2. to 4. until the* μ^x *remain constant. Then set* $x = x_0$, *set* $\gamma^x := -\gamma^x$ *and continue with 4.*

- The sequence of the local centers of mass μ^x makes up the local principal curve (LPC).

- Need “signum flipping” of γ^x at every loop in order to maintain direction of curve.

Application on traffic data

- LPC (red curve, h=12) with local centers of mass μ^x (black squares). For comparison, also a HS curve is shown (black, dashed).



Some theory for LPCs

- For the 'mean shift', $\mu^x - x$, it is known that, asymptotically (i.e., $n \longrightarrow \infty$ and each entry of $H \longrightarrow 0$)

$$\mu^x - x \overset{a}{\sim} H\nabla f(x)/f(x).$$

- For the local PCA step, one can show that

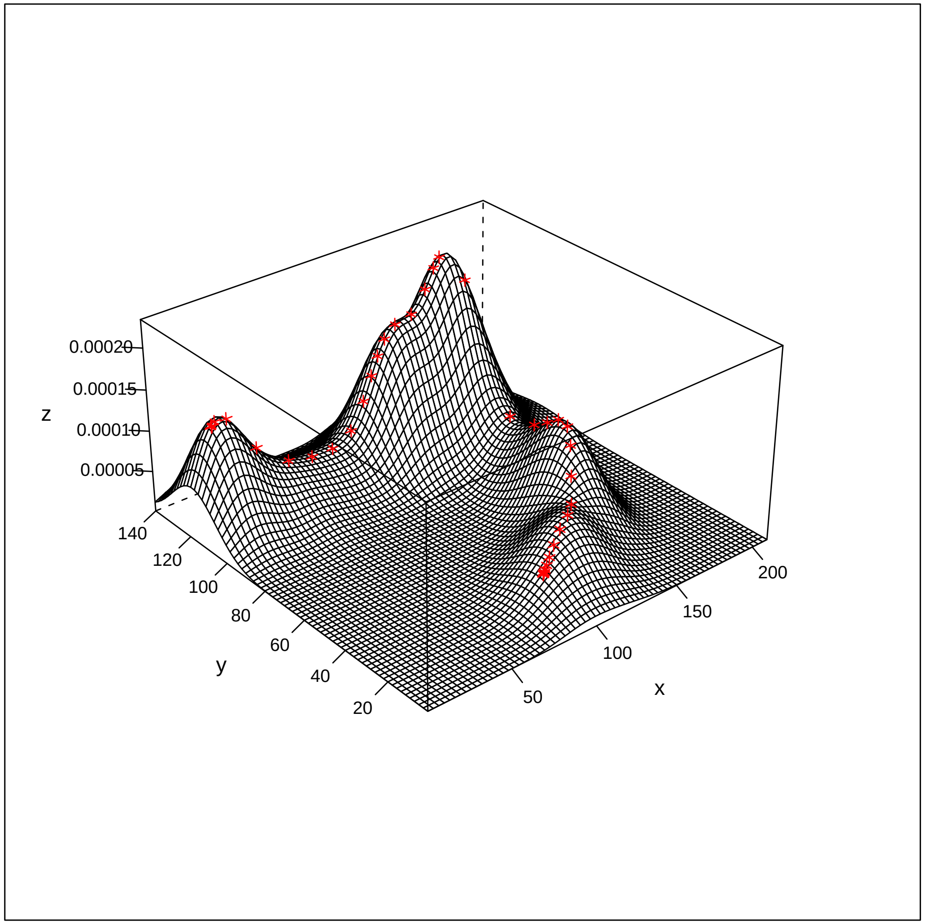
$$\gamma^x \overset{a}{\sim} -\frac{H\nabla f(x)}{||H\nabla f(x)||}$$

- Combining these, and using $H = \text{diag}(h^2)$, the distance between two local centers of mass, say μ^{x_j} and $\mu^{x_{j+1}}$, is given by

$$\mu^{x_{j+1}} - \mu^{x_j} \overset{a}{\sim} \left[\frac{1}{f(x_{(j)})}h^2 \pm \frac{1}{||\nabla f(x_{(j)})||}t_0\right]\nabla f(x_{(j)})$$

Some theory for LPCs (cont.)

- Hence, the LPC always turns in direction of the gradient, which implies that it attempts to follow the density ridge:



Kernel density estimate:

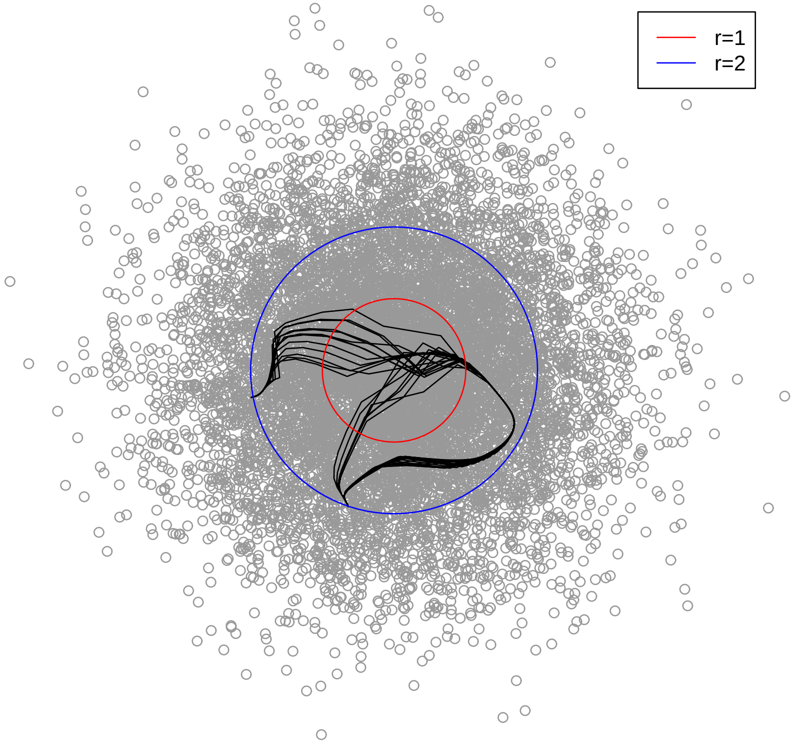
$$\hat{f}(x) = \frac{1}{n|H|} \sum_{i=1}^{n} K\left(H^{-1}(X_i - x)\right)$$

Some theory for LPCs (cont.)

- Secondly, we observe that the LPC stops when
$$f(x) = \frac{h^2}{t_0}||\nabla f(x)||$$
- Special case: $X \sim N(0, \sigma^2 \boldsymbol{I})$. Then $f(x) = c||\nabla f(x)||$ iff $x = \frac{1}{c}\sigma^2$.

Some theory for LPCs (cont.)

- Secondly, we observe that the LPC stops when
$$f(x) = \frac{h^2}{t_0}||\nabla f(x)||$$
- Special case: $X \sim N(0, \sigma^2 \boldsymbol{I})$. Then $f(x) = c||\nabla f(x)||$ iff $x = \frac{1}{c}\sigma^2$.



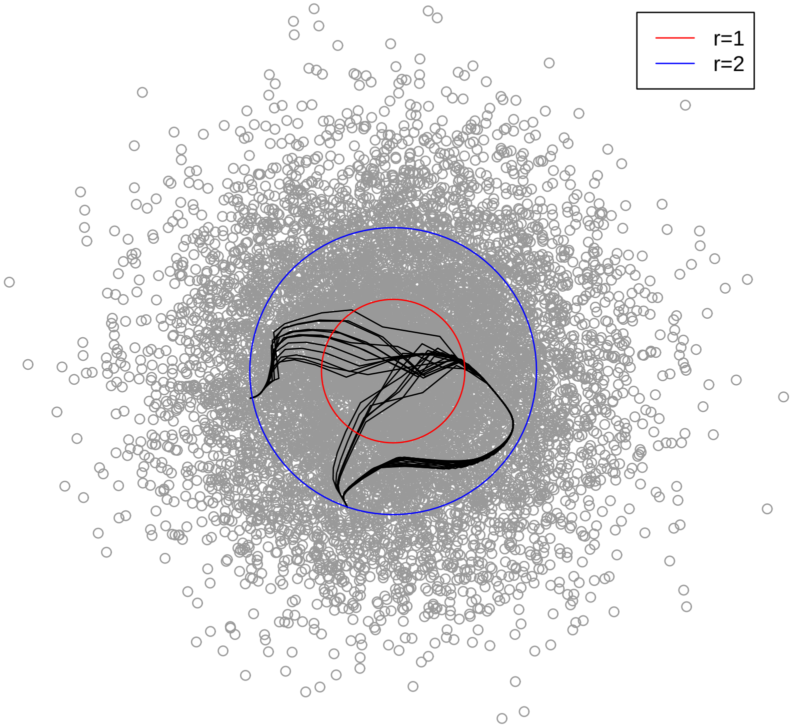
- Simulation: BVN with $\sigma^2 = 2$.
- 20 LPCs with $h = 1$, $t_0 = 1$ started within circle of radius $r = 1$.
- All of them converge to blue circle $r = \sigma^2 = 2$.

Some theory for LPCs (cont.)

- Secondly, we observe that the LPC stops when

$$f(x) = \frac{h^2}{t_0}||\nabla f(x)||$$

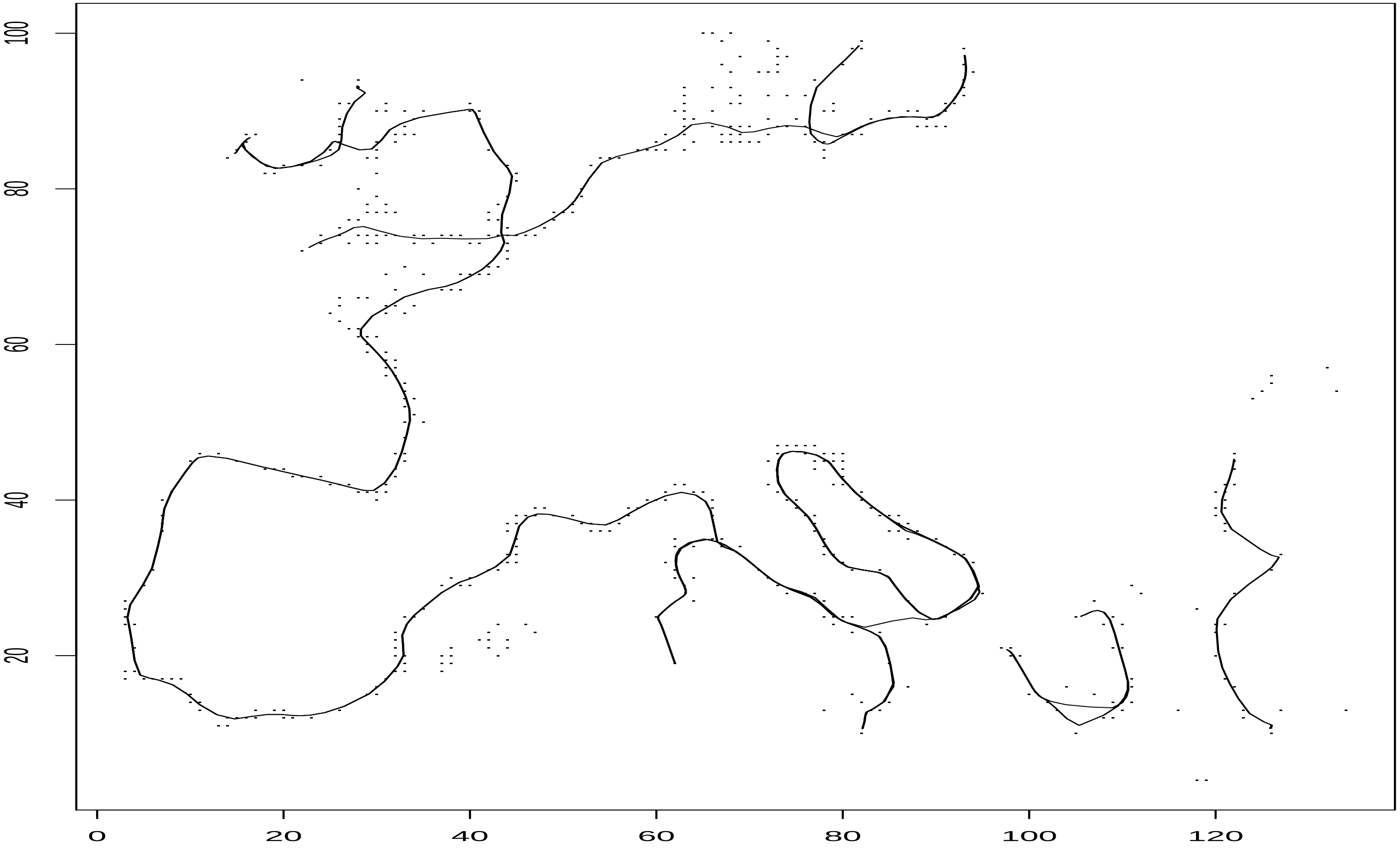
- Special case: $X \sim N(0, \sigma^2 \boldsymbol{I})$. Then $f(x) = c||\nabla f(x)||$ iff $x = \frac{1}{c}\sigma^2$.



- Simulation: BVN with $\sigma^2 = 2$.
- 20 LPCs with $h = 1$, $t_0 = 1$ started within circle of radius $r = 1$.
- All of them converge to blue circle $r = \sigma^2 = 2$.
- Can be exploited for boundary extension (M. Zayed).

Another descriptive toy example

- LPC through European Coastal Resorts (using multiple starting points):



- nice, but ...

Is this everything?

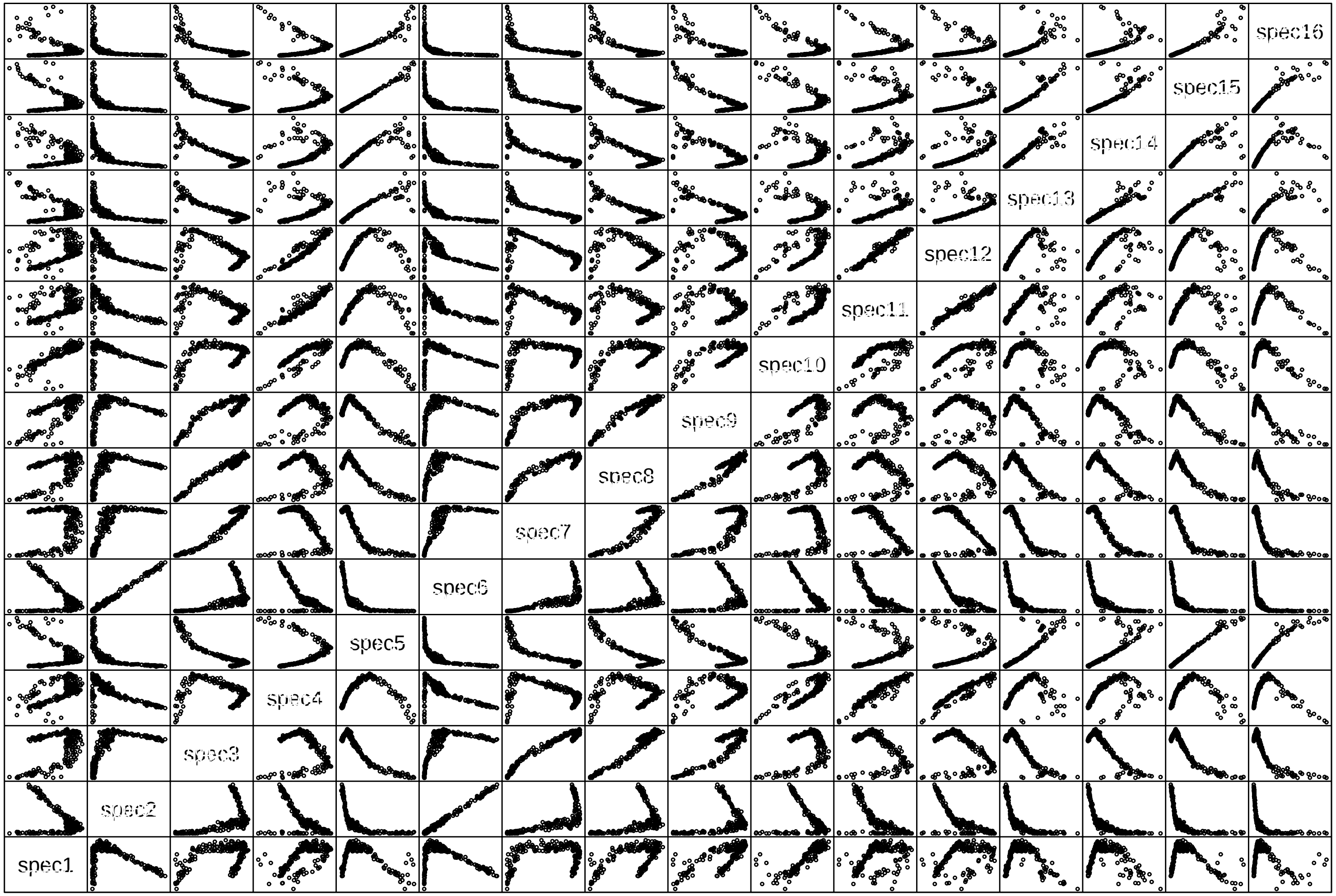
- Most principal curve papers stop about here (after some analysis of goodness of fit, theoretical properties, etc.)
- Surprisingly, the literature has rarely proceeded with exploiting a principal curve once it's there.
- The value of their parametric counterpart, principal components, also brings to bear only when they are used for data compression or regression.
- So, why not do the next step?

Motivation: GAIA data

- GAIA is an astrophysics mission of the European Space Agency (ESA) which will undertake a detailed survey of over 10^9 stars in our Galaxy and extragalactic objects.
- Satellite to be launched in 2012.
- Aims of the mission (among others)
  - Classify objects (star, galaxy, quasar,...)
  - Determine astrophysical parameters (“APs”: temperature, metallicity, gravity) from spectroscopic data (photon counts at certain wavelength bands).
- Work is led by the group “Astrophysical parameters” based at MPIA Heidelberg, being part of the DPAC (Data Processing and Analysis Consortium) which is responsible for the general handling of data from the GAIA mission.
- Yet, one has to work with simulated data generated through complex computer models.

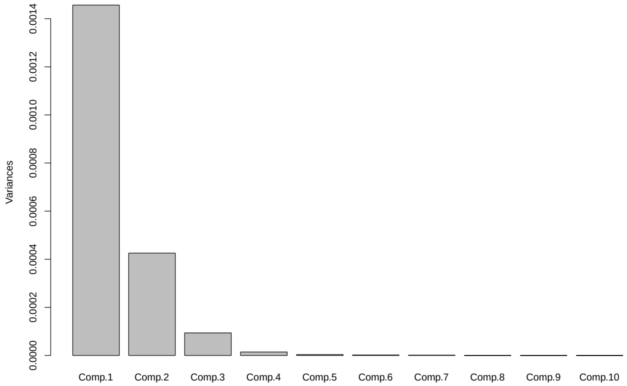
GAIA data

- Photon counts $(n = 8286)$ simulated from APs:



Dimension reduction

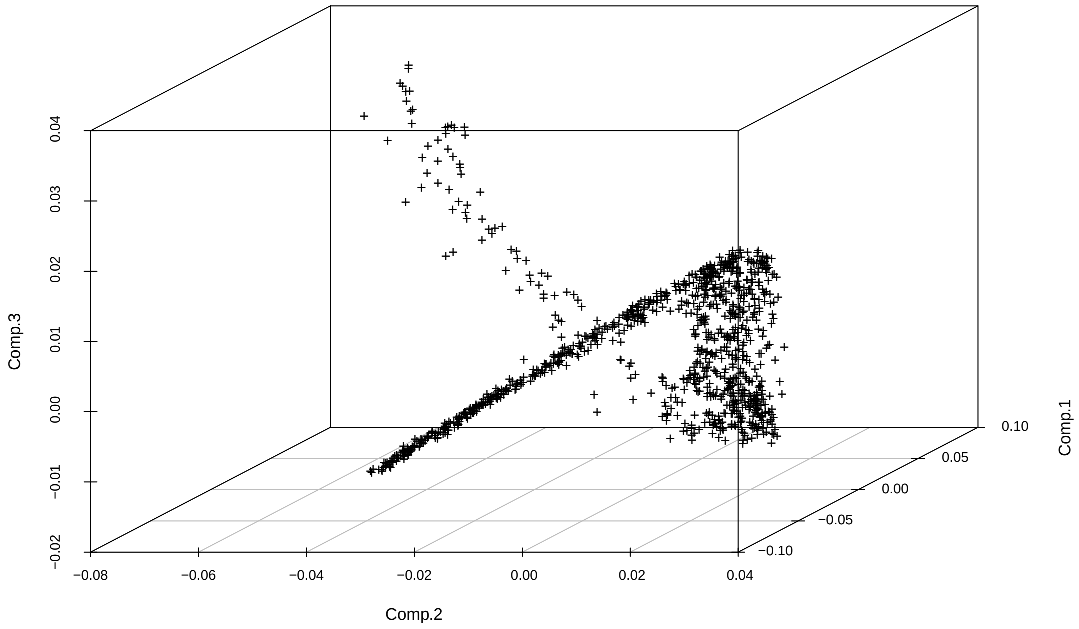
- We observe that the 16 variables contribute very similar, partially redundant information.
- In fact, when the satellite is up, even up to 80 wavelength bands are going to be considered.
- Hence, there is a need for dimension reduction techniques.
- Look at scree plot:



- Three principal components appear to be sufficient.

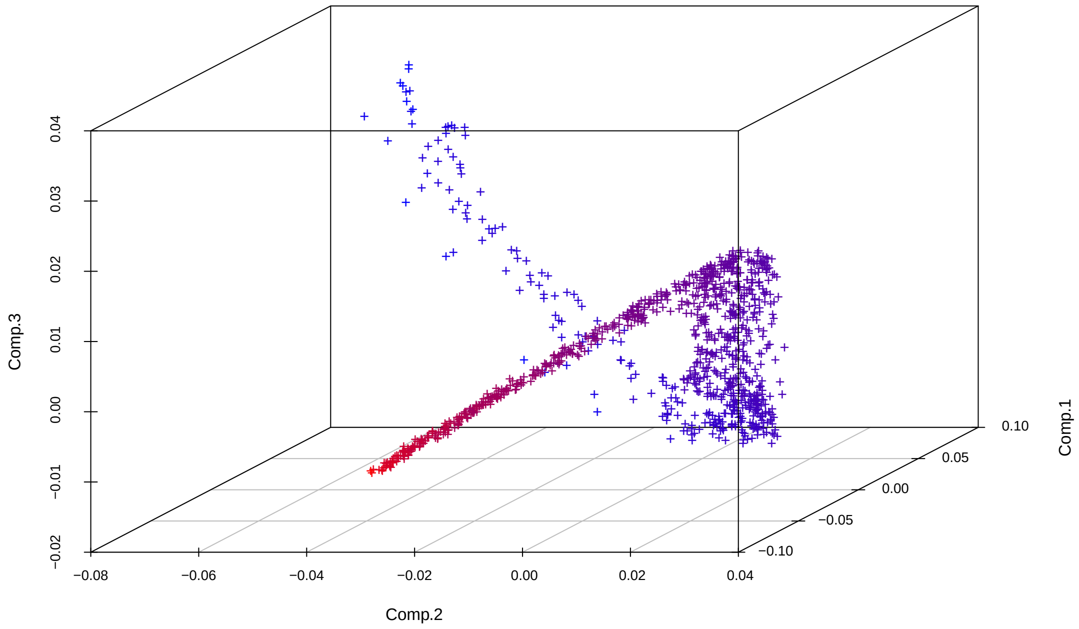
Principal component scores

- We plot the the first three principal component scores.



Principal component scores (cont.)

- We plot the the first three principal component scores and shade higher temperatures red.



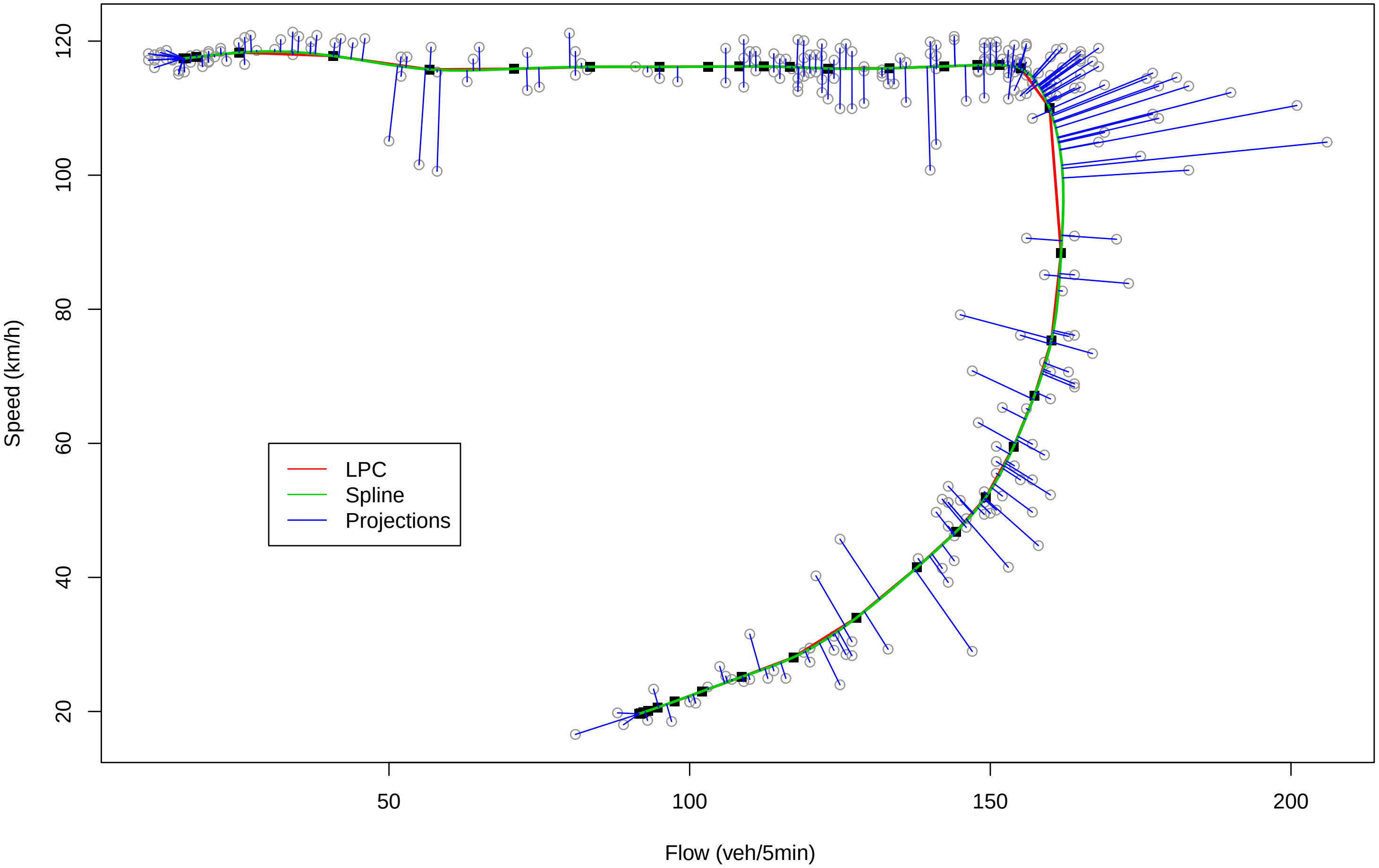
- Actually, we seem to need only *one* parameter if we were able to lay a smooth curve through the data cloud.
- We would need to be able to parametrize such a curve, and to project the data onto it.

Parametrization and Projection

- Unlike HS curves, LPCs do not have a natural parametrization, so it has to be computed retrospectively.
- Define a preliminary parametrization $s \in \mathbb{R}$ based on Euclidean distances between neighboring local means $\mu \in \mathbb{R}^d$.
- For each component μ_j, $j = 1, \ldots, d$, use a natural cubic spline to construct functions $\mu_j(s)$, yielding together a function $(\mu_1, \ldots, \mu_d)(s)$ representing the LPC (no smoothing involved here!).
- Recalculate the parametrization along the curve through the arc length of the spline function.
- Each point $x_i \in \mathbb{R}^d$ is projected on the point of the curve nearest to it, yielding the corresponding projection index t_i.

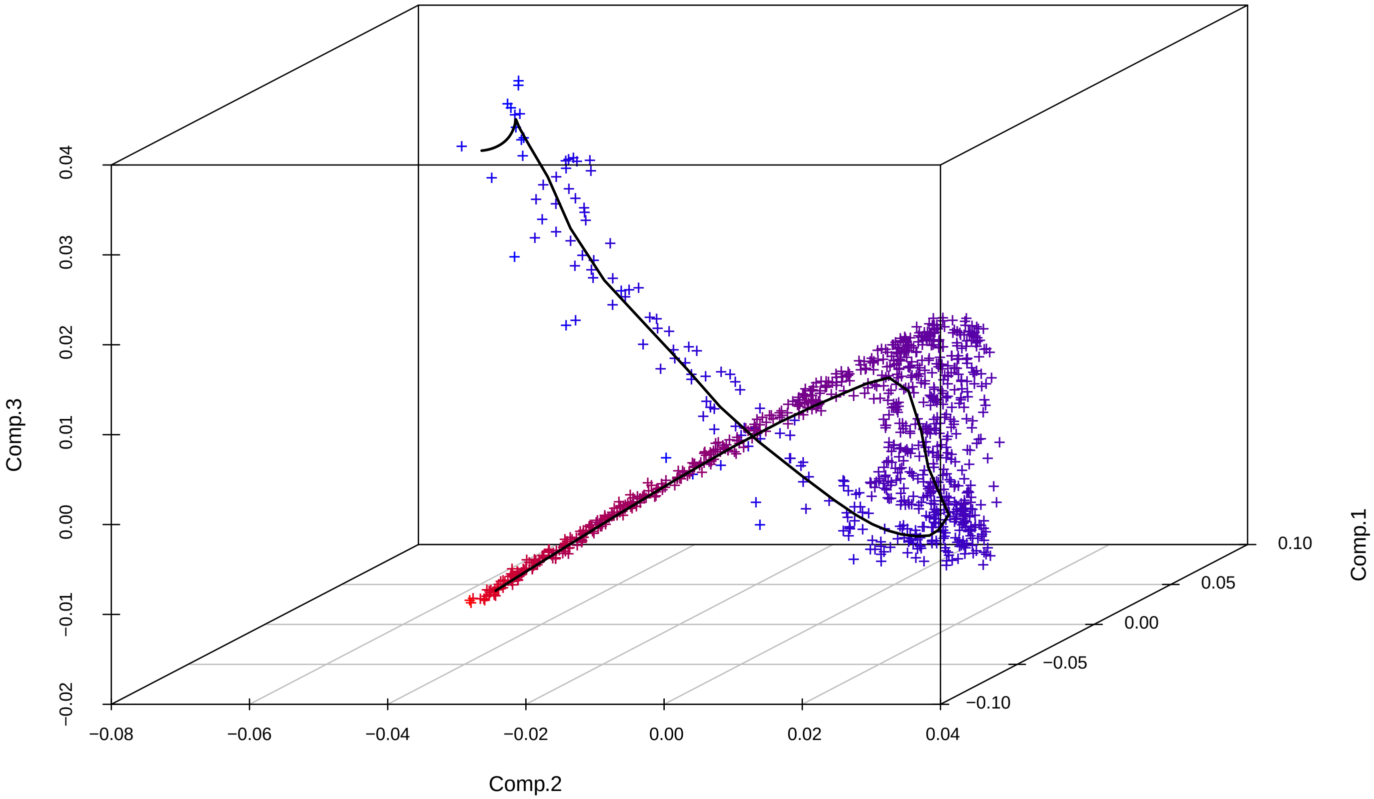
Illustration: traffic data

Original LPC, Spline, and projections for speed-flow data:



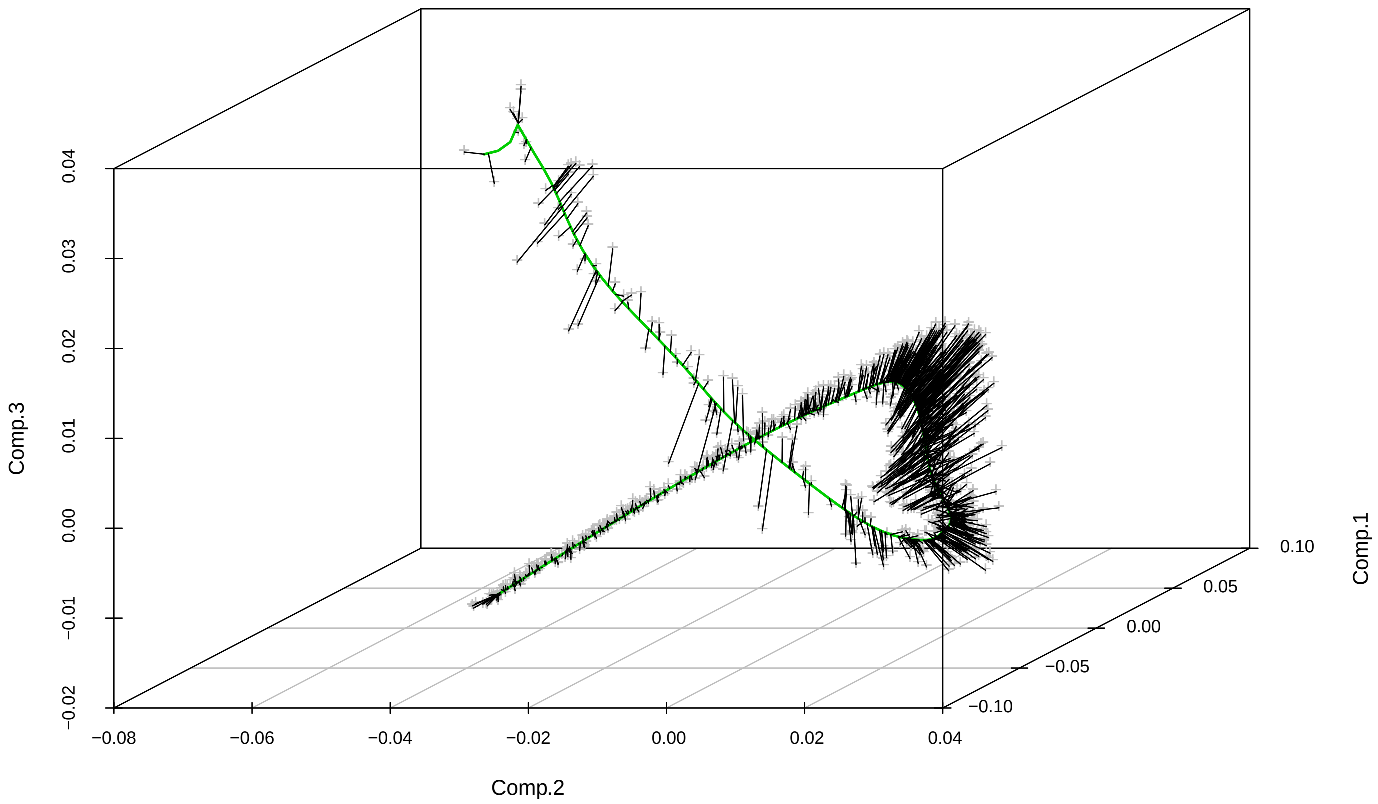
Back to GAIA data

- LPC through first three principal component scores of photon counts



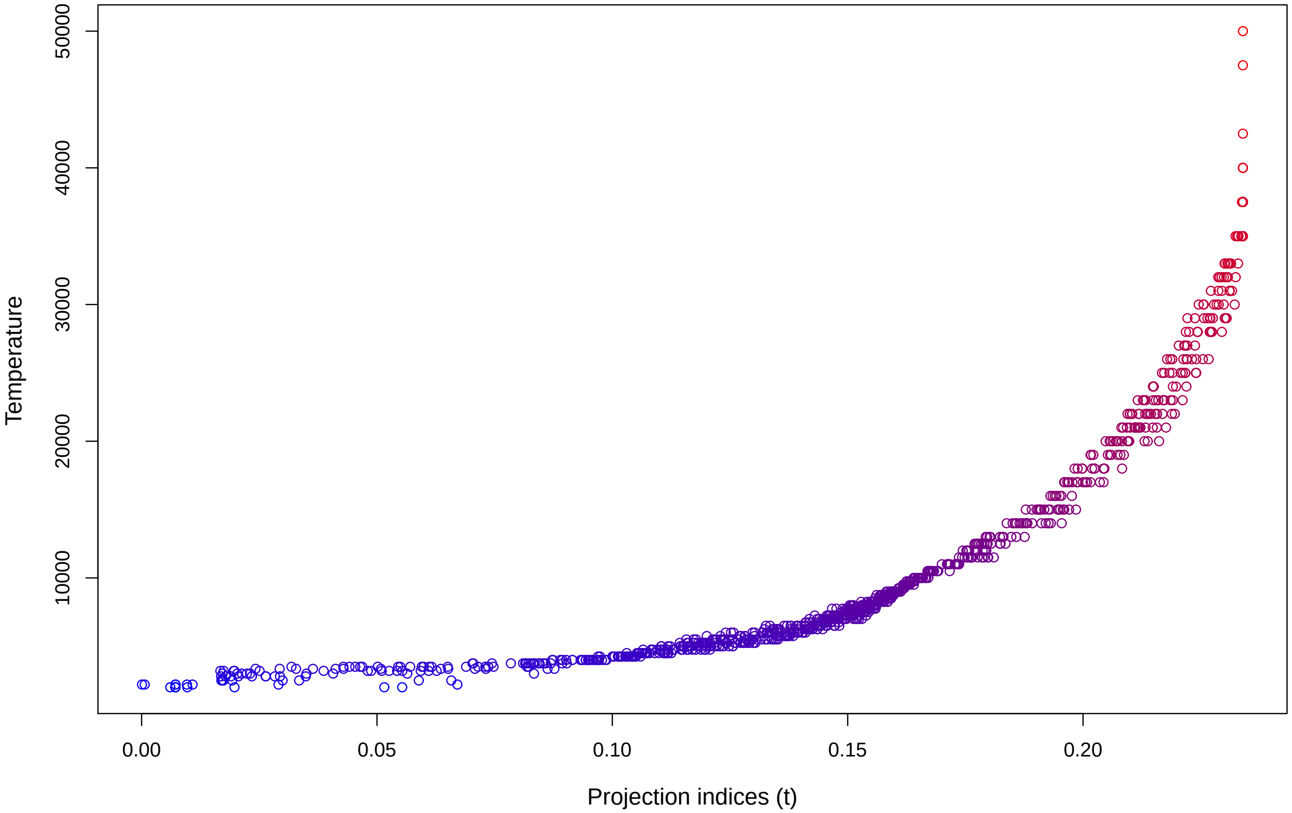
Back to GAIA data (cont.)

- LPC (in spline representation) through PC scores, with vertical projections:



Regression

- We want to predict stellar temperature from 16-d spectral data, using the projection indices of the spectra as predictors.

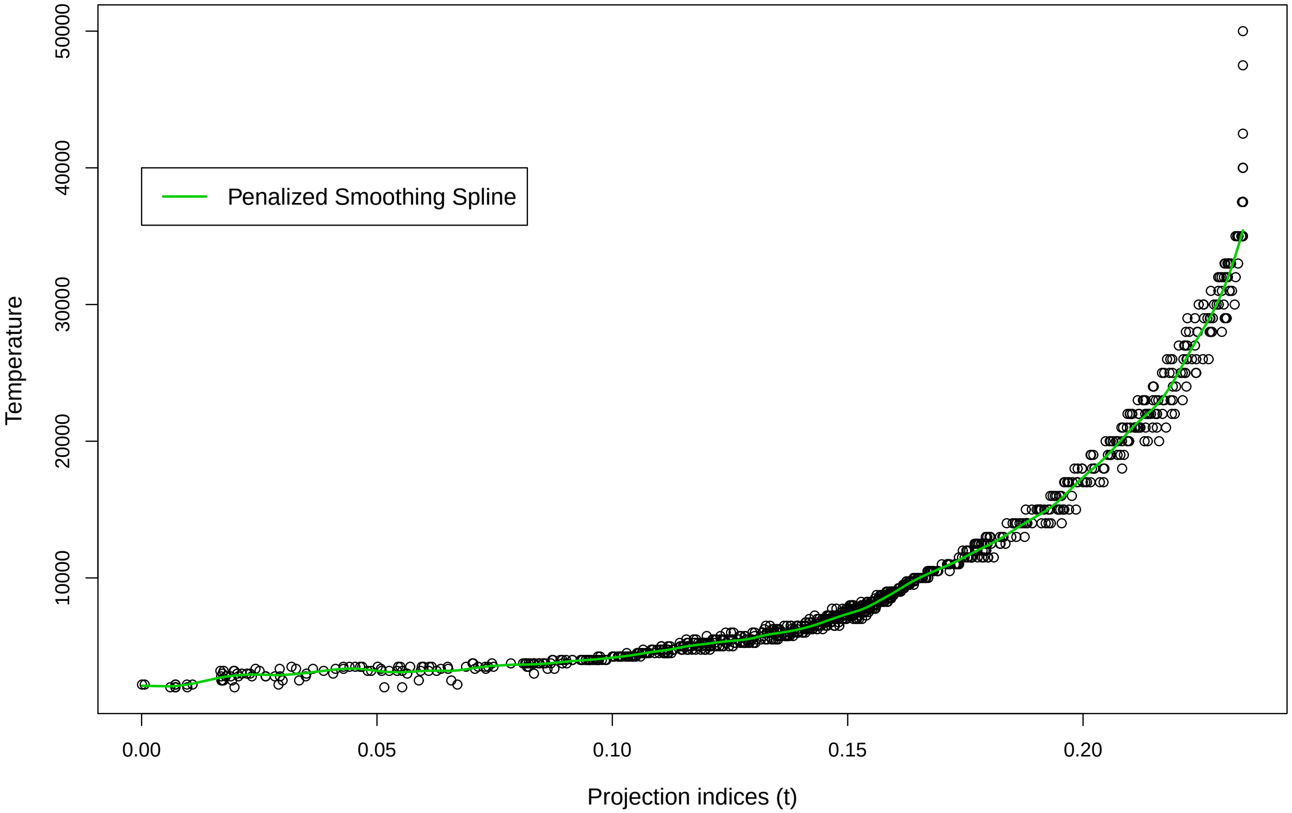


- This is now a simple one-dimensional regression problem,

$$y_i = g(t_i) + \varepsilon_i.$$

Regression

- We want to predict stellar temperature from 16-d spectral data, using the projection indices of the spectra as predictors.

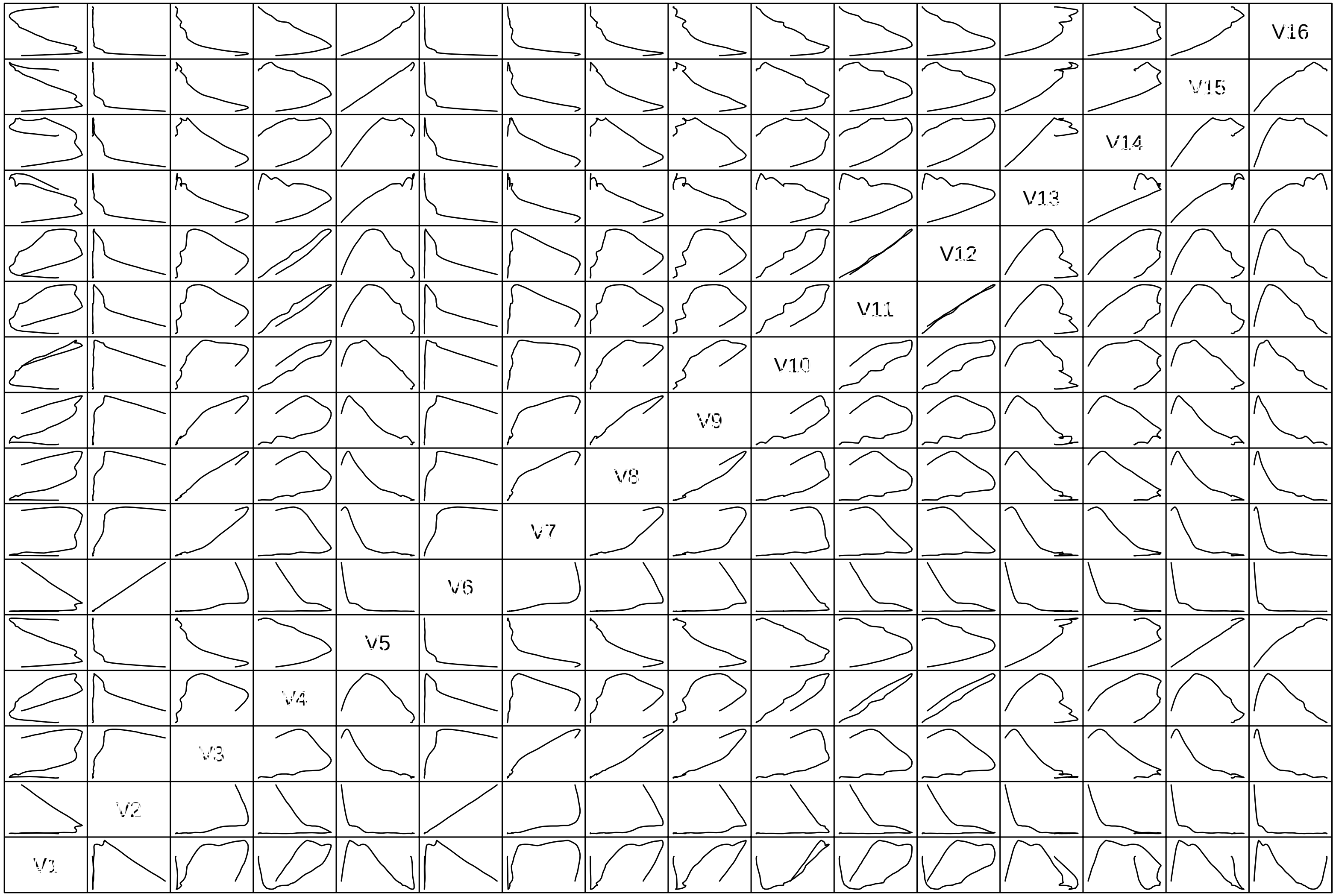


- This is now a simple one-dimensional regression problem,

$$y_i = g(t_i) + \varepsilon_i.$$

Shortcut

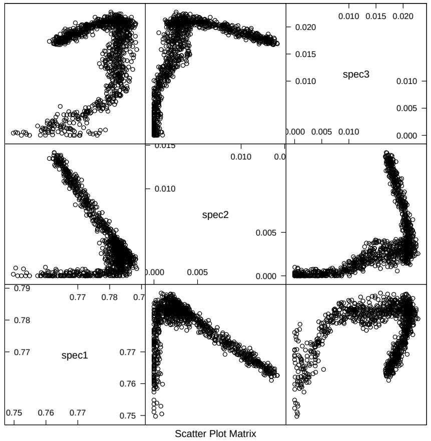
- LPC fitted *directly* through 16- dimensional space:



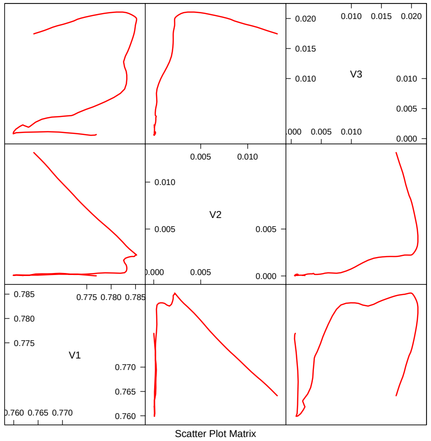
Shortcut (cont.)

- Zoom into the the first three dimensions:

Data



LPC



- Direct data compression with LPCs works in principle, but is potentially “dangerous” as data gets sparse in high dimensions and remote parts of the predictor space maye be missed.

Prediction

- For a new observation x_{new} (i.e., here, a new set of spectra), prediction proceeds as follows:
  - Project x_{new} onto the LPC, giving t_{new}.
  - Compute $\hat{y}_{new} = \hat{g}(t_{new})$ from the fitted regression model.
- Comparison: We sample $n' = 1000$ test data from the remaining $8286 - 1000$ observations and observe the prediction error:

prediction error $/10^3$	LM	PC+LM	PC+AM	PC+LPC	LPC
average($\hat{\varepsilon}_i^2$)	4593	4967	1732	1359	1320
median($\hat{\varepsilon}_i^2$)	1049	1124	104	43	69

where $\hat{\varepsilon}_i$ is the difference between true and predicted temperature.

Prediction

- For a new observation x_{new} (i.e., here, a new set of spectra), prediction proceeds as follows:
  - Project x_{new} onto the LPC, giving t_{new}.
  - Compute $\hat{y}_{new} = \hat{g}(t_{new})$ from the fitted regression model.
- Comparison: We sample $n' = 1000$ test data from the remaining $8286 - 1000$ observations and observe the prediction error:

prediction error $/10^3$	LM	PC+LM	PC+AM	PC+LPC	LPC
average($\hat{\varepsilon}_i^2$)	4593	4967	1732	1359 [91]	1320 [211]
median($\hat{\varepsilon}_i^2$)	1049	1124	104	43 [3]	69 [23]

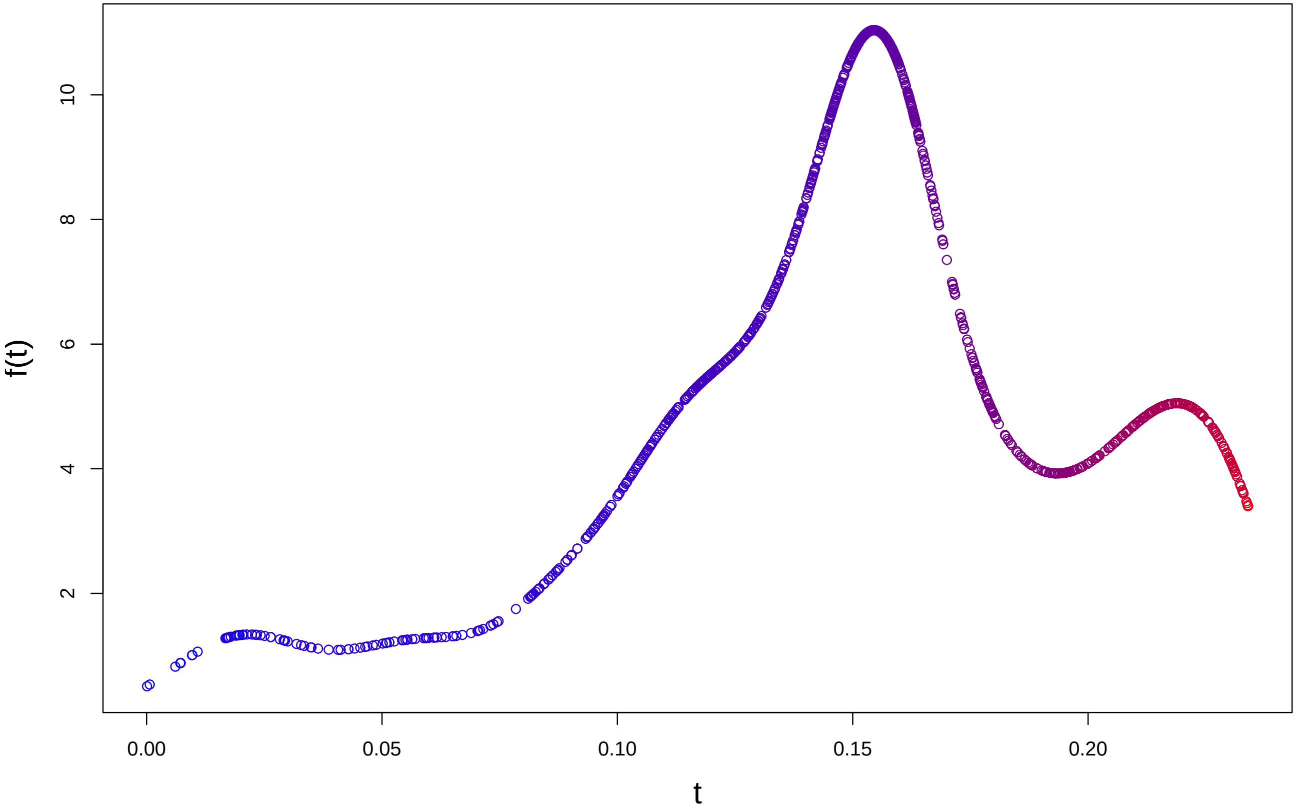
where $\hat{\varepsilon}_i$ is the difference between true and predicted temperature.

- Note: For the LPC methods, the highest density point has been used as starting point. The IQR of the prediction errors using 100 *random* starting points are given in [squared brackets].

Density estimation

- Having now the projection indicies, t_i, of the data, X_i, these can be easily used for other purposes such as "density estimation along the principal curve"

$$\hat{f}(t) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{t_i - t}{h}\right)$$



Density estimation (cont.)

- Can we reconstruct the full density, $f(x)$, from $f(t)$?

$$\begin{aligned} f(x) &= \int f(x,t)dt = \int f(x|t)f(t)\,dt \\ &\overset{(?)}{\approx} f(x|t_x)f(t_x) \approx \frac{1}{h}\phi\left(\frac{||x-m(t_x)||}{h}\right)\hat{f}(t_x) \end{aligned}$$

  where $t_x \in \mathbb{R}$ is the PI of $x \in \mathbb{R}^d$, and $\phi \sim N(0,1)$.

- For traffic data,

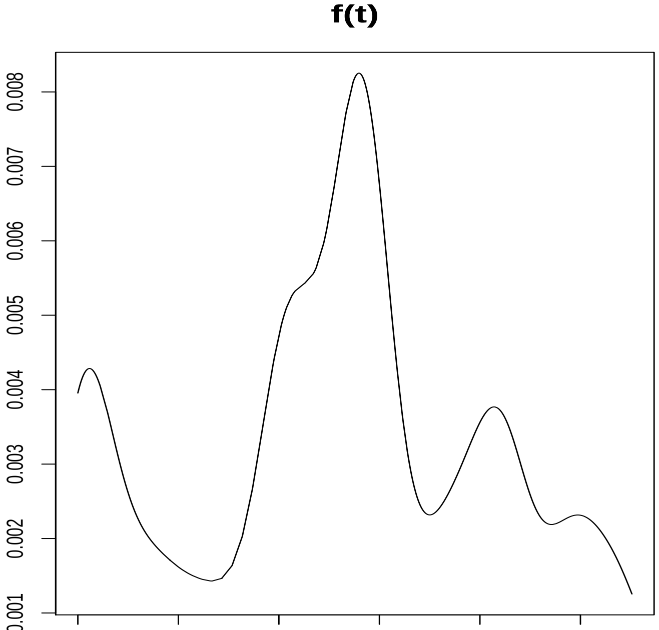
Density estimation (cont.)

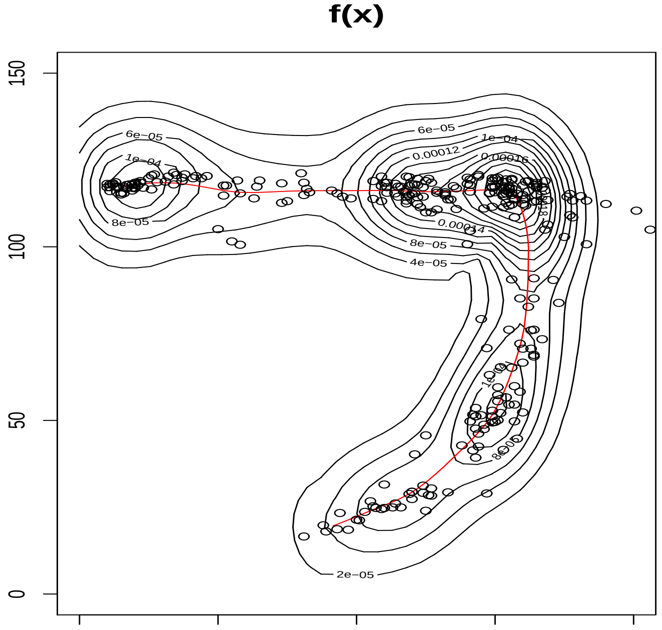
- Can we reconstruct the full density, $f(x)$, from $f(t)$?

$$\begin{aligned} f(x) &= \int f(x,t)dt = \int f(x|t)f(t)\,dt \\ &\overset{(?)}{\approx} f(x|t_x)f(t_x) \approx \frac{1}{h}\phi\left(\frac{||x - m(t_x)||}{h}\right)\hat{f}(t_x) \end{aligned}$$

where $t_x \in \mathbb{R}$ is the PI of $x \in \mathbb{R}^d$, and $\phi \sim N(0,1)$.

- For traffic data,





Density estimation (cont.)

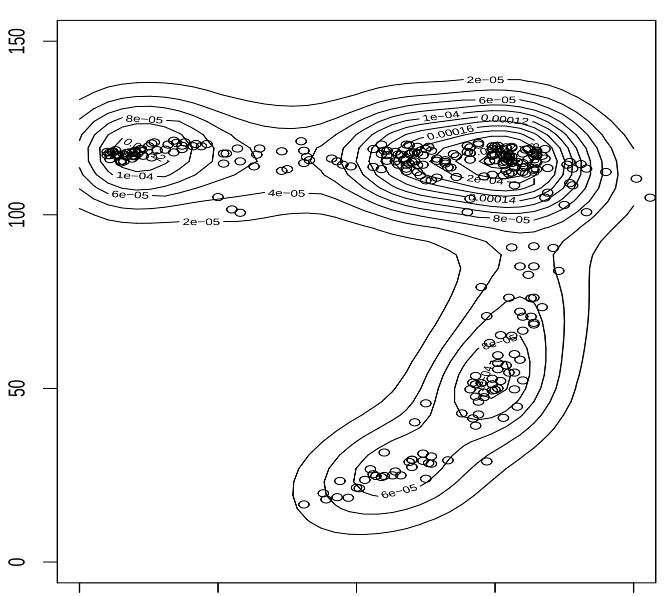
- Can we reconstruct the full density, $f(x)$, from $f(t)$?

$$\begin{aligned} f(x) &= \int f(x,t)dt = \int f(x|t)f(t)\,dt \\ &\overset{(?)}{\approx} f(x|t_x)f(t_x) \approx \frac{1}{h}\phi\left(\frac{||x-m(t_x)||}{h}\right)\hat{f}(t_x) \end{aligned}$$

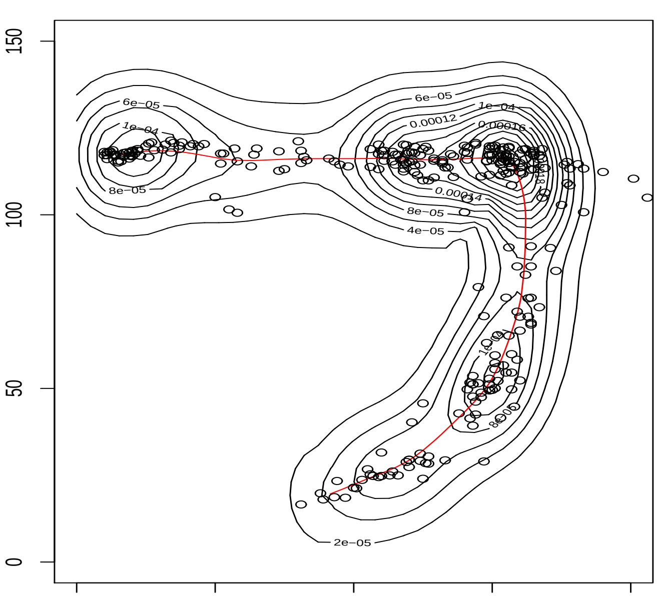
where $t_x \in \mathbb{R}$ is the PI of $x \in \mathbb{R}^d$, and $\phi \sim N(0,1)$.

- For traffic data,

kernel density estimator

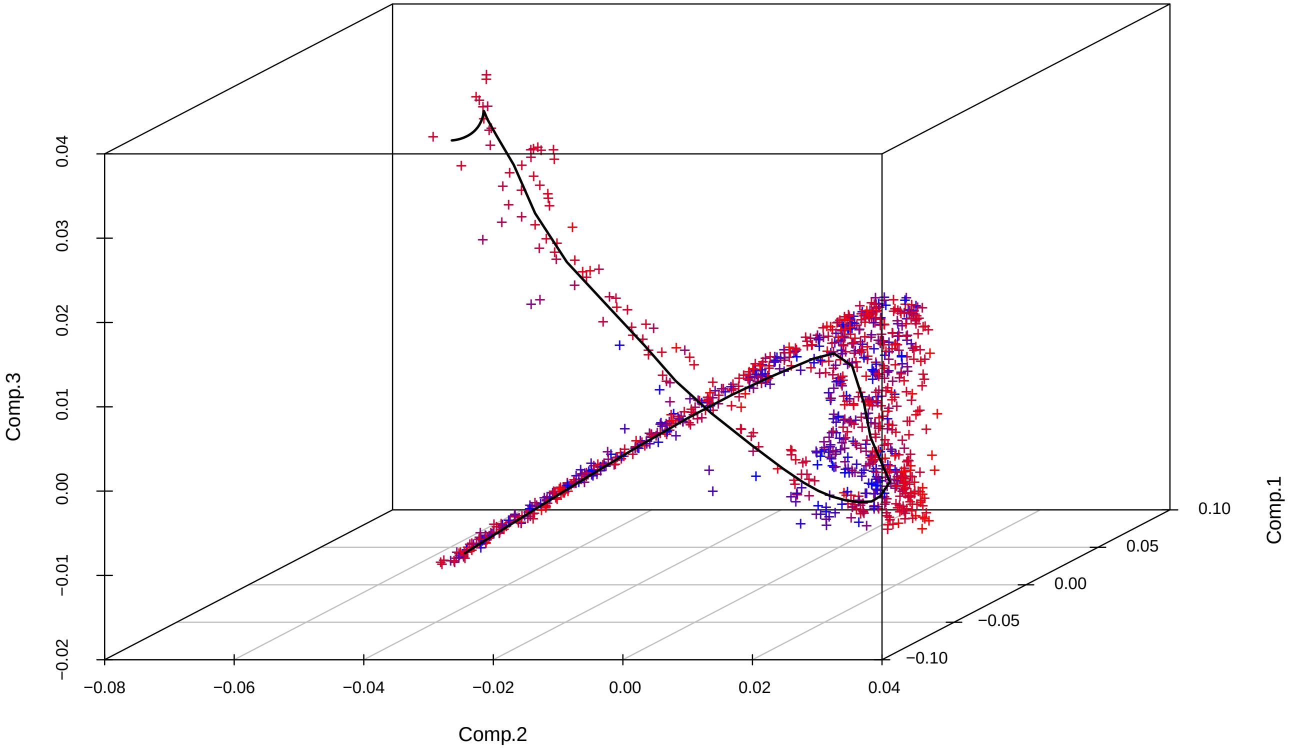


f(x)



Limits of one-dimensional data summaries

- Look at “metallicity”



- The relevant information seems to be orthogonal to the principal curve!
- Would a *principal surface* help?

Local principal surfaces

- Instead of points x, we work with the "building block" triangles Δ.
- Local PCA is only used to determine the initial triangle, say Δ_0.
- Then, the algorithm iterates

  (1) For a given triangle Δ, we glue further triangles at each of its sides $j = 1, 2, 3$.

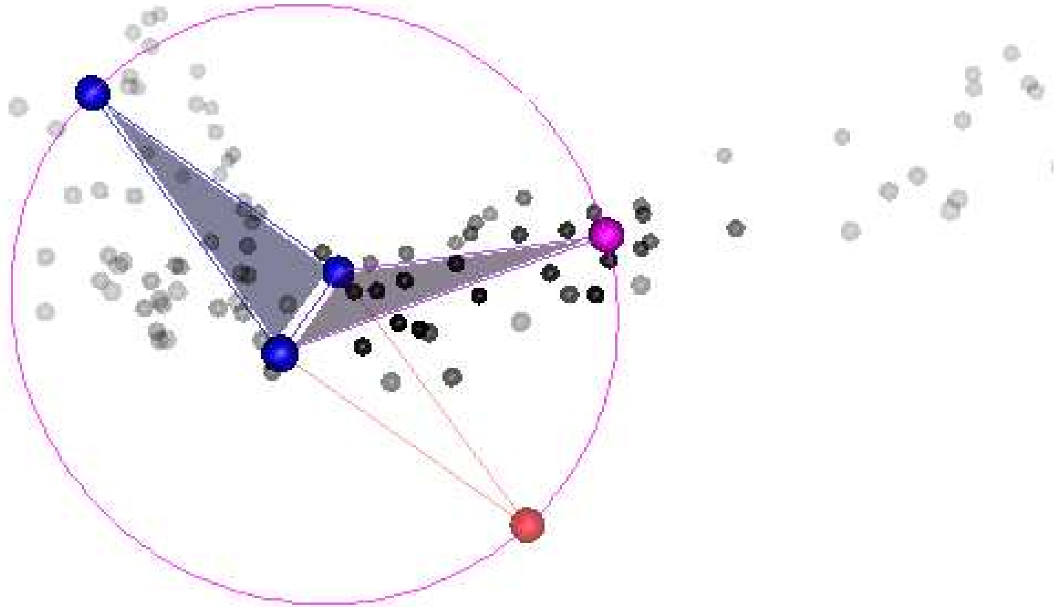
  (2) For $j = 1, 2, 3$, adjust the free triangle vertex via the mean shift. We dismiss the new triangle if

    - the new vertex falls into a region of small density, or
    - the new vertex is too close to an existing one (Delaunay triangulation).

  until all sides of all triangles (including the new ones) have been considered.

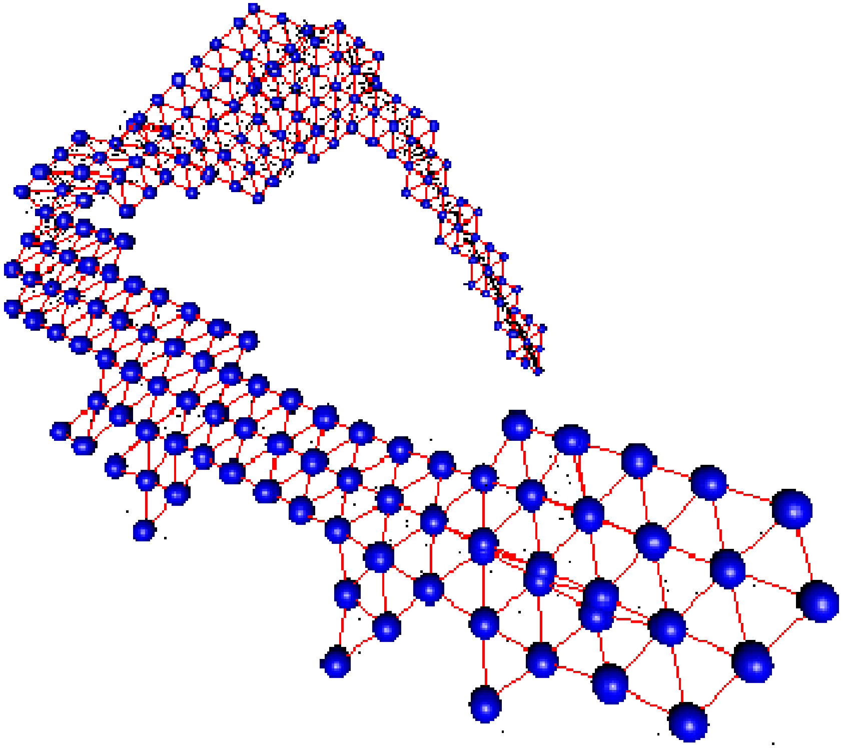
Local principal surfaces (cont.)

- Illustration: Constrained mean shift on a circle (enforcing equiliteral triangles):



Local principal surface for GAIA data

- Local principal surface (LPS) for PC scores based on training data set with $n = 1000$:



Regression on the surface

- Then, how to use this surface for regression?
- It seems hard to define a meaningful 2-dim. parametrization on the surface.
- However, we may use *distances* instead: For each triangle, we can count the distance d to all other triangles through the smallest number of triangle borders that have to be crossed to walk from one to the other.
- Assign local weights via discrete distance-based kernel

$$\kappa(d) = e^{-d/\lambda}$$

The parameter $\lambda \in [0, \infty)$ steers the degree of smoothing on the manifold: the higher λ, the smoother.

Regression on the surface (cont.)

The entire fitting process is summarized as follows:

(I) Fit a LPS as explained above, yielding a surface with, say, R triangles.

(II) Assign each data point $\boldsymbol{x}_i, i = 1, \ldots, n$ to their nearest triangle.

(III) For each triangle $r = 1, \ldots, R$, compute the mean $\bar{y}_r$ over the response values of all data points assigned to it.

(IV) Compute all pairwise distances $d_{r,s}$ between all triangles on the surface.

(V) Use the discrete kernel $\kappa(\cdot)$ to smooth over the manifold. The smoothed response value g_r on triangle r is given by

$$g_r = \frac{\sum_s \kappa(d_{r,s}) \bar{y}_s}{\sum_s \kappa(d_{r,s})}.$$

Simulation study

Prediction errors for $n' = 1000$ test data. The LPS is fitted with $\lambda = 1$.

- Temperature

prediction error $/10^3$	LM	PC+LM	PC+AM	PC+LPC	PC+LPS
average($\hat{\varepsilon}_i^2$)	4593	4967	1732	1320	1227
median($\hat{\varepsilon}_i^2$)	1049	1124	104	44	47

- Metallicity

prediction error	LM	PC+LM	PC+AM	PC+LPC	PC+LPS
average($\hat{\varepsilon}_i^2$)	2.601	3.084	2.849	3.070	3.067
median($\hat{\varepsilon}_i^2$)	1.287	1.821	1.671	1.859	1.323

Conclusion

- Once the principal curve is parametrized, data may be compressed by projecting onto the curve. Depending on the problem at hand, the compressed data may be further processed, for instance via
  - (nonparametric) regression (if response is continuous);
  - classification (if response is discrete);
  - density estimation.
- Extension to local principal surfaces (LPS) works by considering the building block "triangles". The approach considered here only projects data "onto the nearest triangle" rather than "onto the nearest point". Work on the latter is in process, which would enable to fit a continuous, rather than stepwise, *regression* surface.
- Extension to local principal manifolds (LPM) by considering "tetrahedrons" or "simplices" instead of triangles.....

References

Hastie & Stuetzle (1989): Principal Curves. *JASA* **84**, 502–516.

Hastie, Tibshirani, & Friedman (2001): The Elements of Statistical Learning. Springer.

Tibshirani (1992): Principal Curves Revisited. *Statistics and Computing* **2**, 183–190.

Kégl, Krzyzak, Linder,& Zeger (2000): Learning and Design of Principal Curves. *IEEE Transactions Patt. Anal. Mach. Intell.* **24**, 59–74.

Delicado (2001): Another Look at Principal Curves and Surfaces, *Journal of Multivariate Analysis* **77**, 84–116.

References (cont.)

Einbeck, Tutz & Evers (2005): Local principal curves. *Statistics and Computing* **15**, 301–313.

Einbeck, Evers & Bailer-Jones (2008): Representing complex data using localized principal components with application to astronomical data. In Gorban et al. (Eds): Principal Manifolds for Data Visualization and Dimension Reduction; *Lecture Notes in Computational Science and Engineering* **58**, 180–204.

Einbeck, Evers & Powell (2010): Data compression and regression through local principal curves and surfaces. *International Journal of Neural Systems*, **20**, 177–192.

Einbeck & Evers (2010): **LPCM** – Local principal curve methods (R package version 0.40-2).