Local Principal Curves

Jochen Einbeck

Department of Mathematics, NUI Galway

jochen.einbeck@nuigalway.ie

Leicester — 24th to 26th August 2006

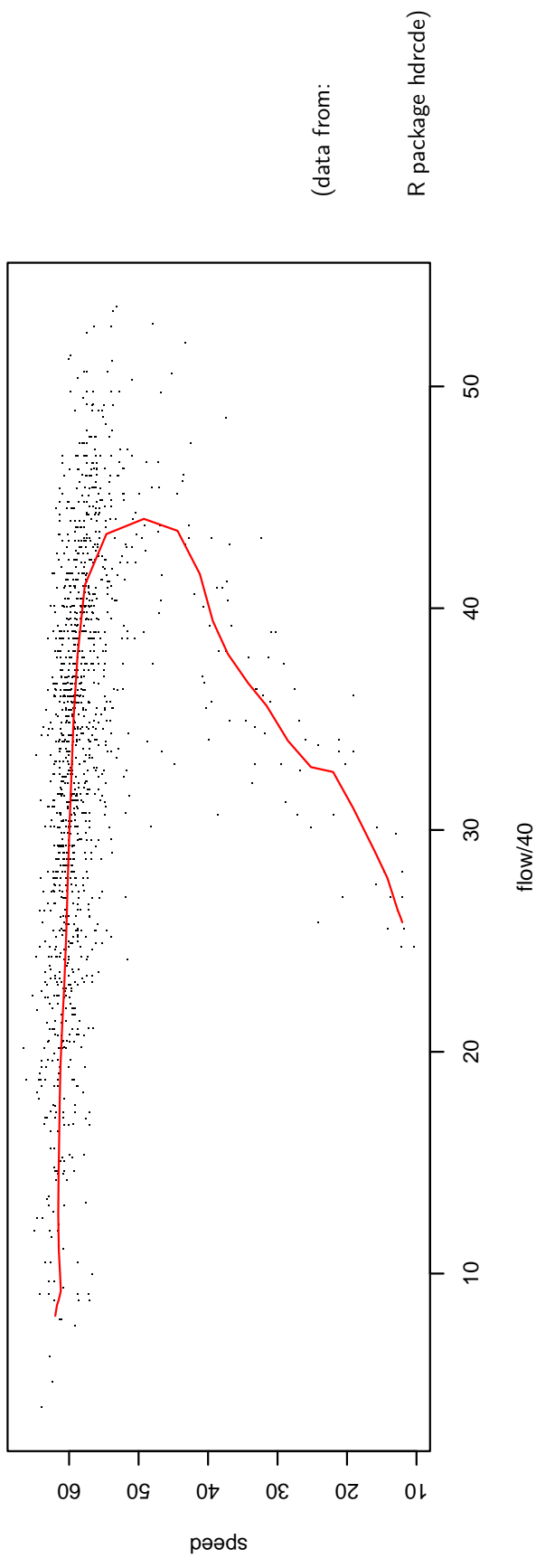joint work with

Descriptive Definition

Principal Curves are smooth curves passing through the 'middle' of a multidimensional data

cloud $X = (X_1, \ldots, X_n)$, where $X_i \in \mathbb{R}^d$.

Example: Speed-Flow diagram.

(data from:

R package hdrcde)

X: traffic flow in cars/hour,   Y: speed in Miles/hour

recorded on a Californian "freeway".

Types of principal curves

There exist a variety of definitions of principal curves, which essentially vary in what is understood of the "middle" of a data cloud. The algorithms associated to these definitions can be divided into two major groups:

- Global ('top-down') algorithms start with an initial line and try to dwell out this line or concatenate other lines to the initial line until the resulting curve fits well through the data cloud.

- Local ('bottom-up') algorithms estimate the principal curve locally moving step by step through the data cloud.

Principal curve definitions associated to 'top-down' - approaches

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

Self-consistent curves $m : I \longrightarrow \mathbb{R}^d$ are obtained as critical points of the distance function

$$\Delta(m) = E\left(\inf_t \|X - m(t)\|^2\right) \quad (1)$$

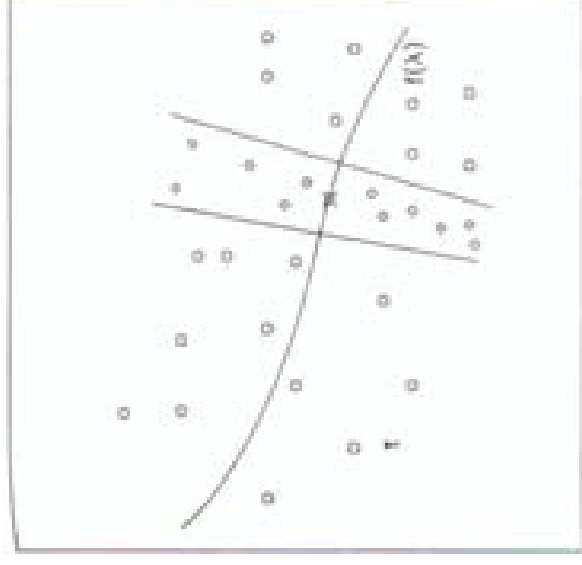and generalize linear principal components in a natural way.

Kégl, Krzyzak, Linder & Zeger (KKLZ, 2000) define a principal curve as the curve minimizing the average squared distance (1) over all curves with bounded length L.

Figure 1 Each point on a principal curve is the average of the points that project there.

(Picture from: Hastie & Stützle, 1989)

Tibshirani (1992) defines principal curves such that for data generated as

$$X = m(t) + \epsilon \quad \text{with} \quad E(\epsilon) = 0$$

curve m is also principal curve of the data cloud X.

Properties of 'top-down' algorithms:

- Starting from the first principal component line of the whole data set, the principal curve is estimated iteratively with EM-like algorithms.

- Quite fast and computationally stable.

- Dependance on an initial line leads to a lack of flexibility, as an initial unsuitable assignment of projection indices can often not be correct in the further run of the algorithm (particularly for HS).

- Estimation of branched or disconnected data clouds difficult

$(\longrightarrow$ *principal graphs*, Kégl & Krzyżak, 2002)

Delicado (2001) defines principal curves 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. He estimates 'PCOPs' using a fixed point algorithm moving smoothly through the data cloud.
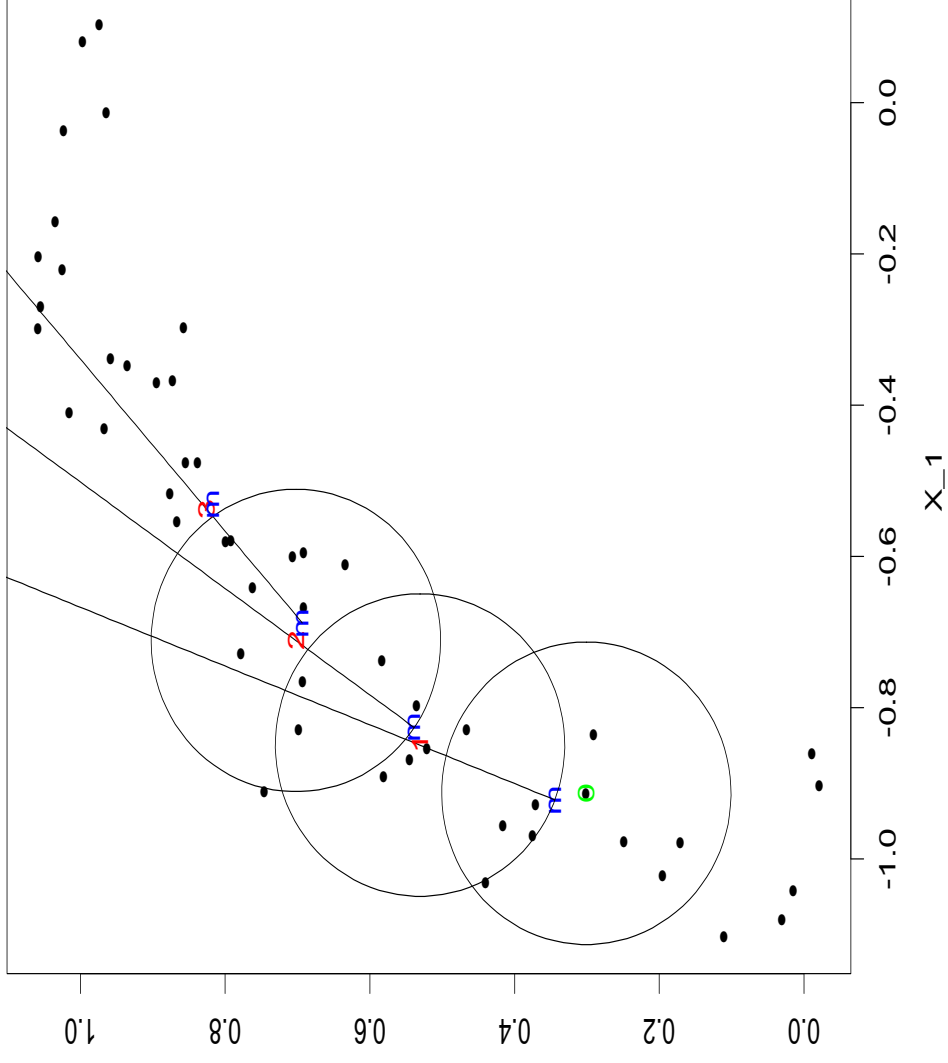

- Works fine for most (not too complex) data sets.

- Mathematically elegant

- However, quite complicated and computationally demanding.

- Requires a cluster analysis at every point of the principal curve.
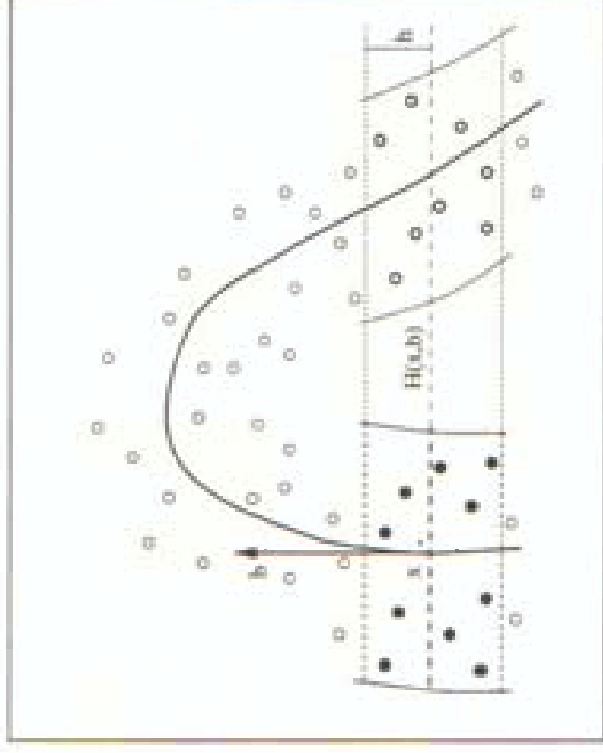
(Picture from: Delicado, 2001)

Idea: Calculate alternately a local center of mass and a first local principal component.

0: starting point,

m: points of the LPC,

$1, 2, 3$: enumeration of steps.

Given: A data cloud $X = (X_1, \ldots, X_n)$, where $X_i = (X_{i1}, \ldots, X_{id})$.

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 1st local eigenvector γ^x of $\Sigma^x = (\sigma^x_{jk})_{(1 \leq j, k \leq d)}$, where*

$$\sigma^x_{jk} = \sum_{i=1}^n w_i (X_{ij} - \mu^x_j)(X_{ik} - \mu^x_k).$$

4. *Step from μ^x to $x := \mu^x + t_0 \gamma^x_1$.*

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).

Background

- LPCs can be seen as a simplified version of Delicado's 'PCOPs'. Both algorithms can be shown to differ essentially by the type of weighting and centering used in Σ^x. But Delicado's Σ^x depends on the 'principal direction' b, ruling out a simple eigenanalysis as for LPCs.

- A local principal curve approximates the density ridge. For instance, speed-flow data:
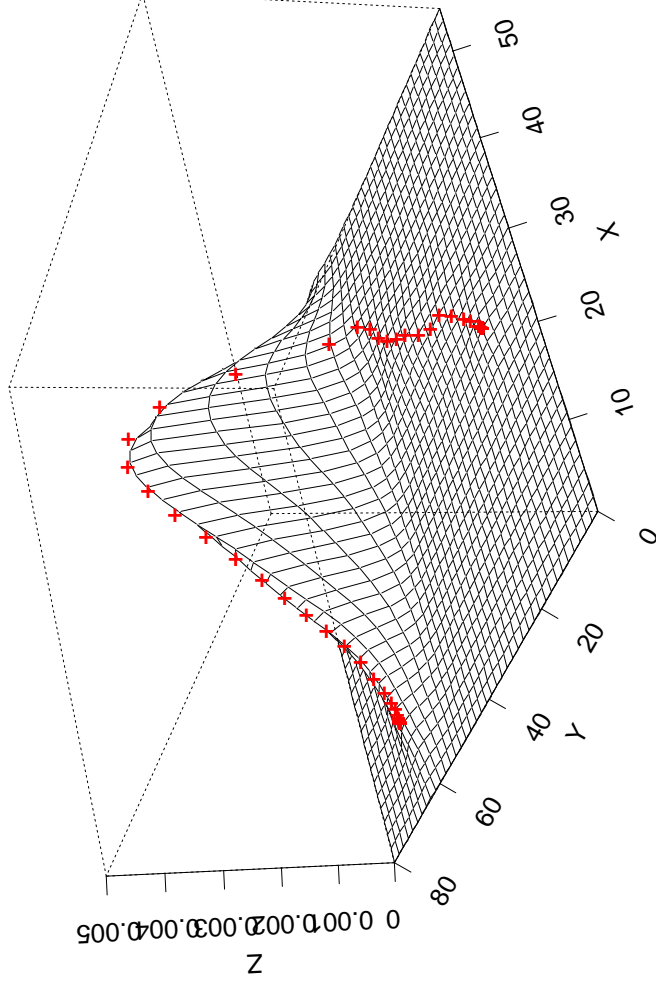
Kernel density estimate:

$$\hat{f}_K(x) = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{X_i - x}{h}\right)$$

Comaniciu & Meer (2002):

'Mean Shift' $\mu^x - x \sim \nabla \hat{f}_K(x)$

Technical Details

- "Signum flipping": Check in every cycle if

$$\gamma_{(i-1)}^x \circ \gamma_{(i)}^x > 0.$$

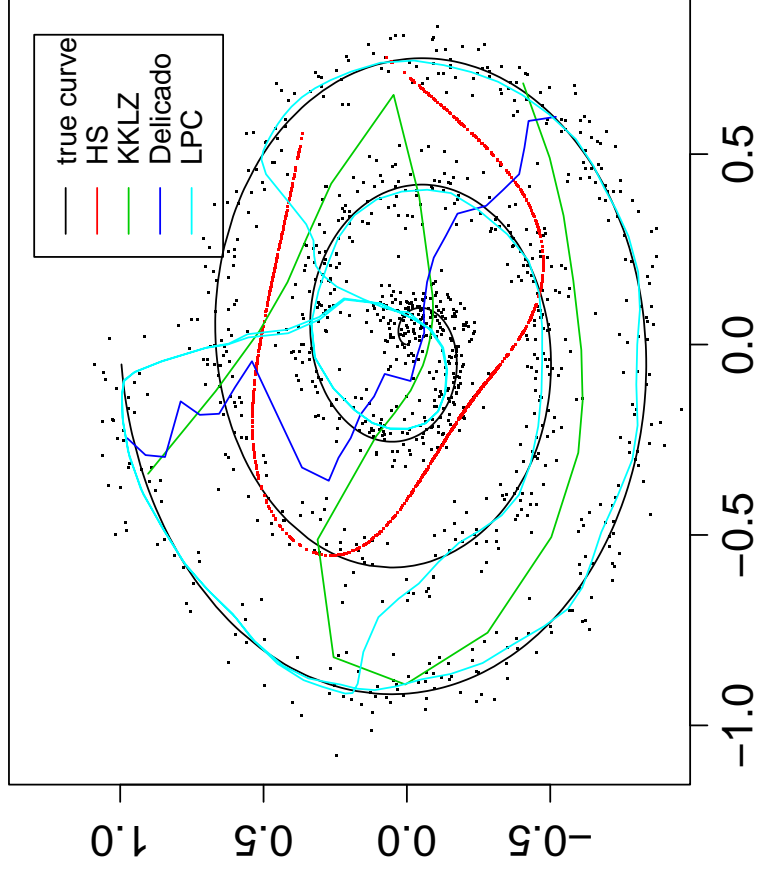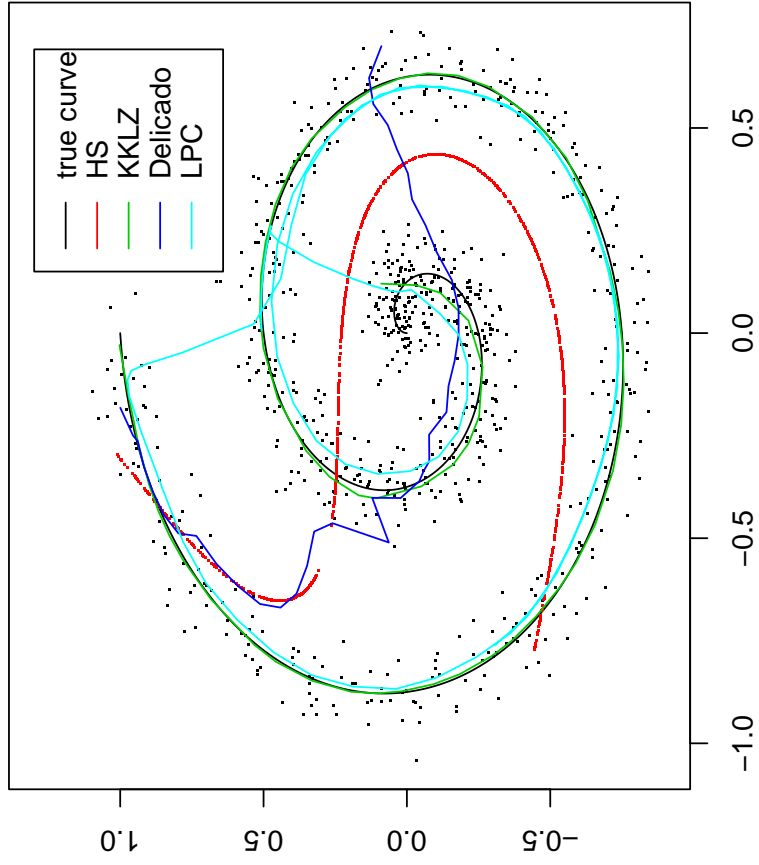Otherwise, set $\gamma_{(i)}^x := -\gamma_{(i)}^x$.

- Angle penalization, to hamper the principle curve from bending off at crossings.

- Use multiple initializations if data cloud consists of several branches (e.g. using a random generator).

Simulated Examples

Spirals with small noise
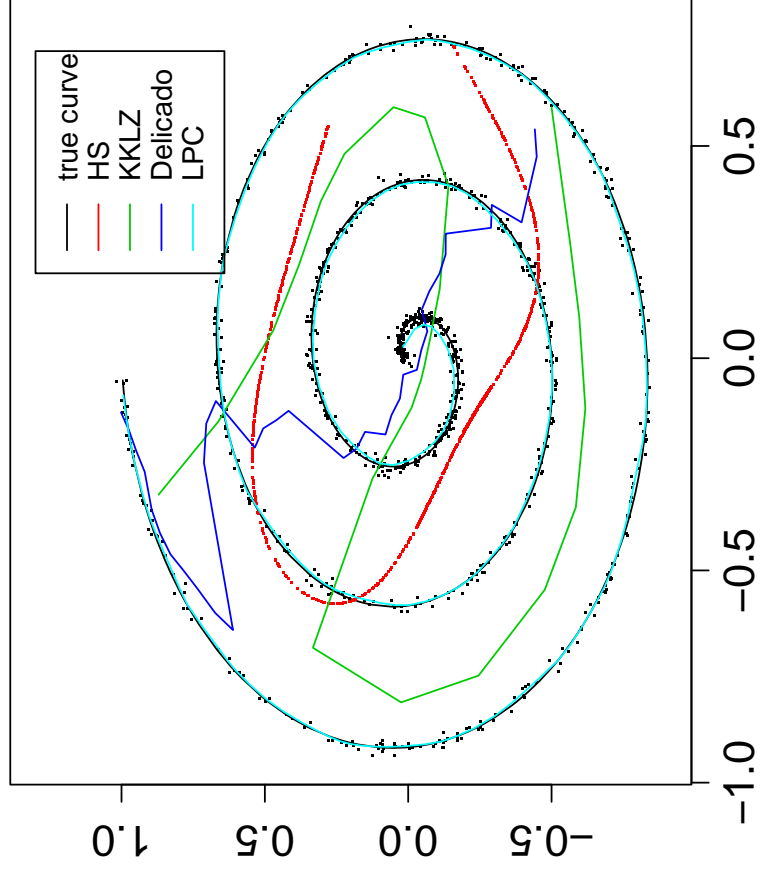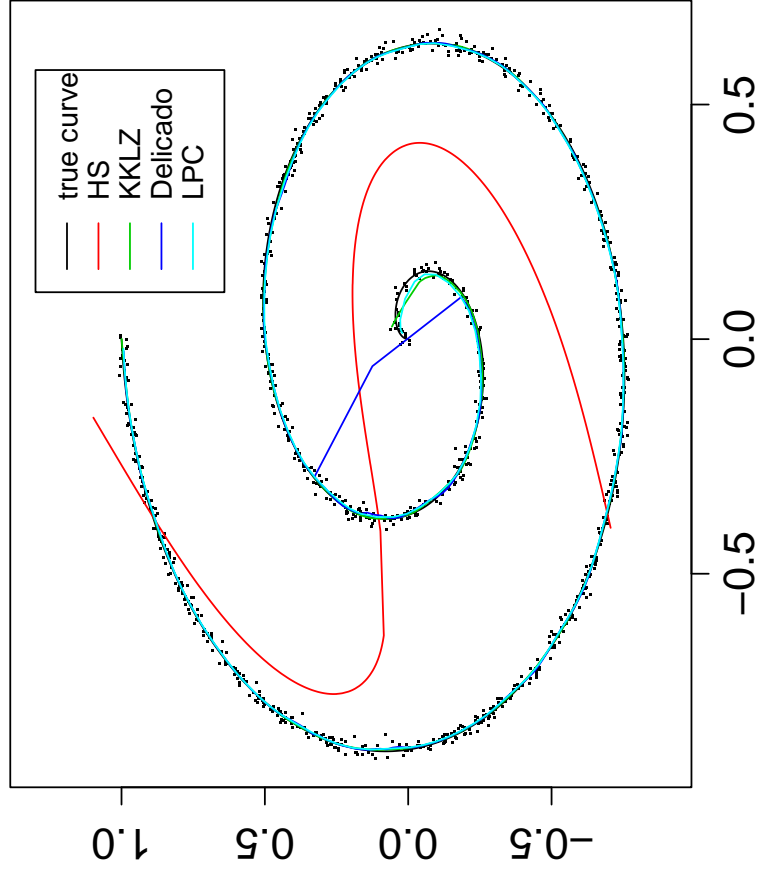
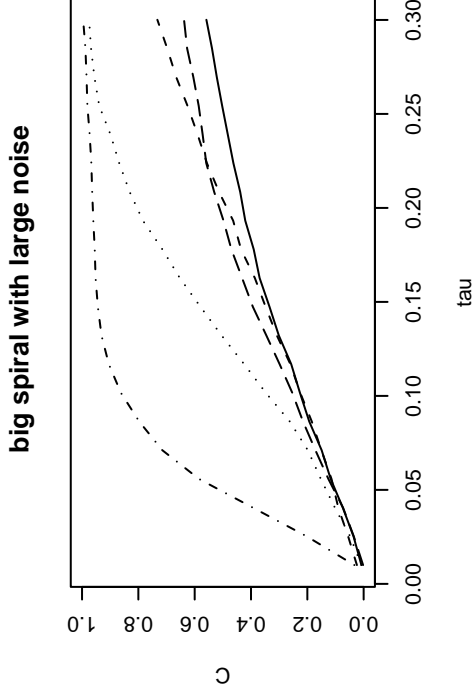Spirals with large noise

Measuring performance: Coverage
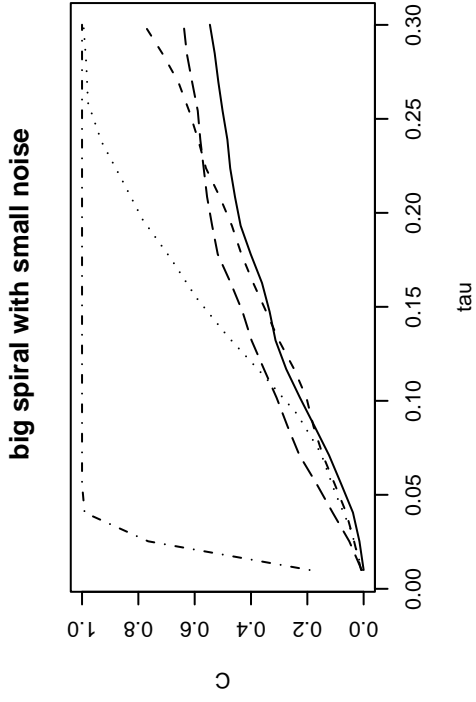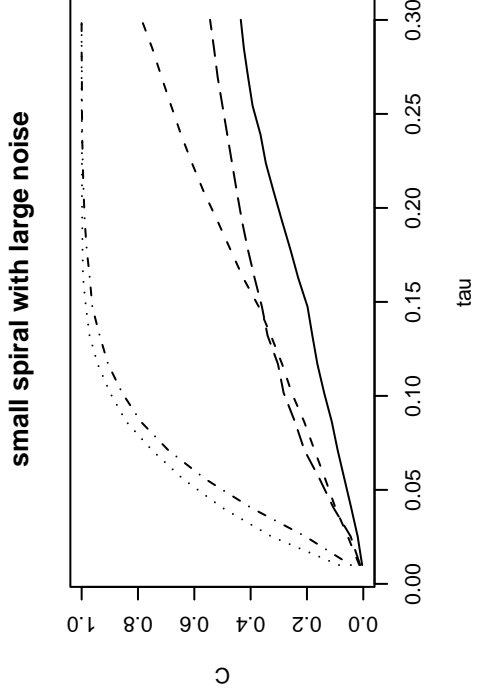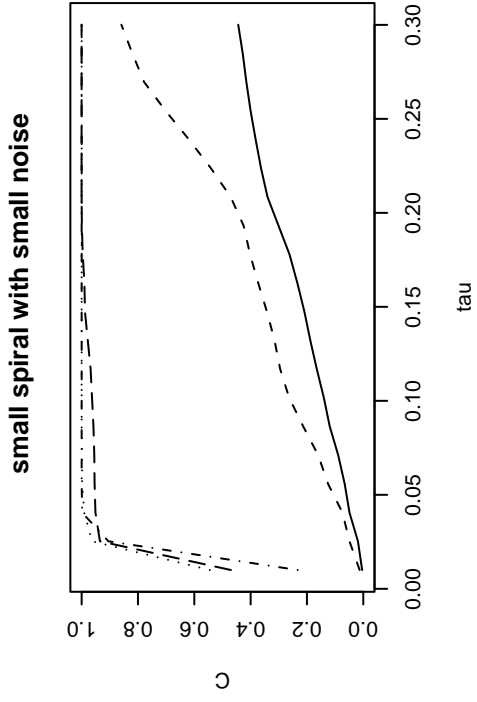
The coverage of a principal curve is the fraction of all data points found in a certain neighborhood of the principal curve.

Formally, for a principal curve m consisting of a set P_m of points, the coverage is given by

$$C_m(\tau) = \#\{x \in X | \exists p \in P_m \text{ mit } \|x - p\| \leq \tau\}/n$$

- The coverage can also be interpreted as empirical distribution function of the residuals.

- The area between $C_m(\tau)$ and the constant 1 corresponds to the mean length of the observed residuals.

Coverage for spiral-data

small spiral with small noise

small spiral with large noise

big spiral with small noise

big spiral with large noise

PCA
HS
KKLZ
Delicado
LPC

Residual mean length relative to principal components (A_C):

A_C	small spiral		big spiral	
	small noise	large noise	small noise	large noise
HS	0.72	0.77	0.92	0.92
KKLZ	0.03	0.20	0.50	0.65
Delicado	0.05	0.85	0.87	0.92
LPC	0.05	0.24	0.08	0.29

● The closer to 0, the better the performance

● the quantity $R_C = 1 - A_C$ can be interpreted in analogy to R^2 used in regression analysis

Bandwidth selection with self-coverage

Idea: A bandwidth suitable for computation of a principal curve m should also be able to cover adequately the data cloud. This motivates to define the self-coverage,
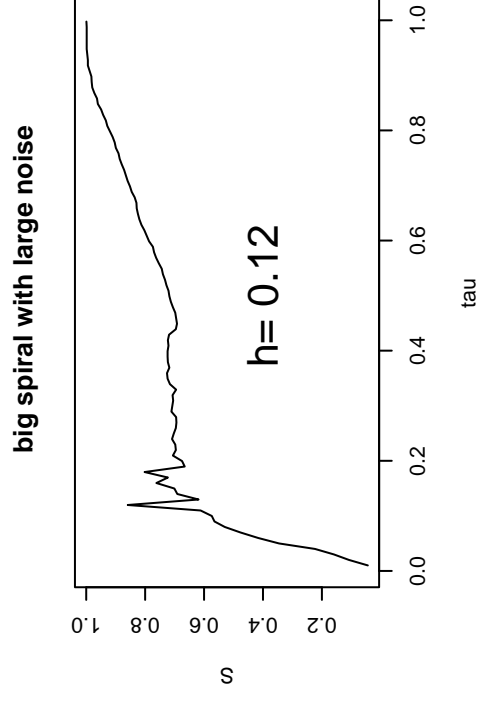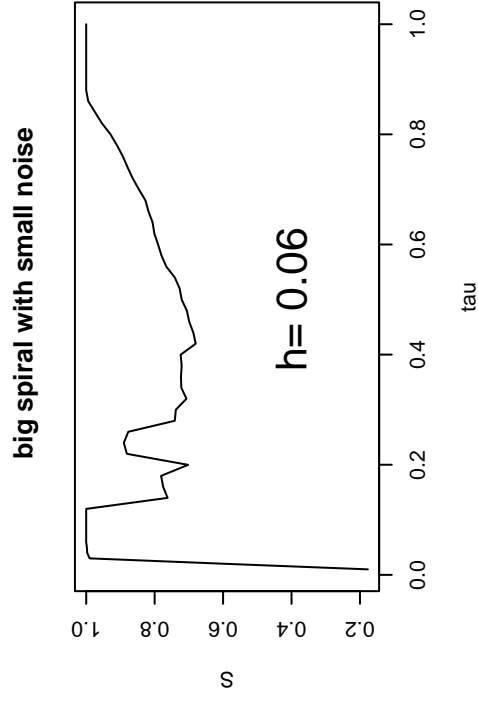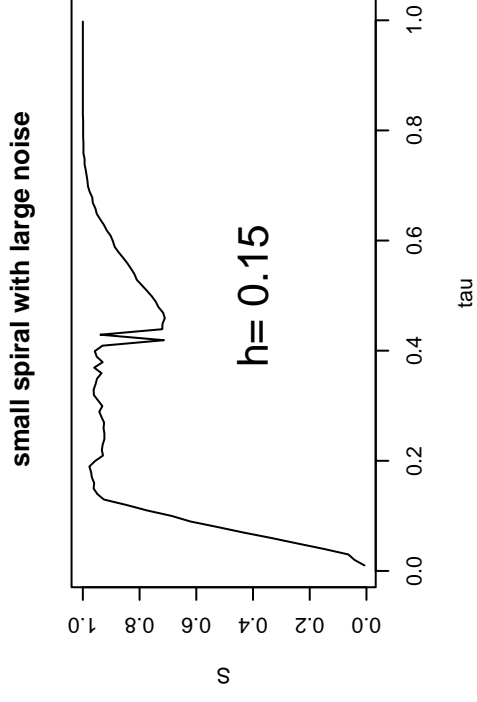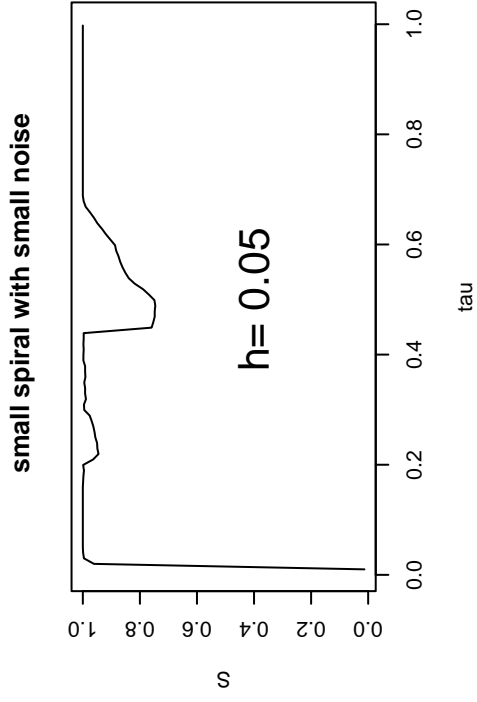
$$S(\tau) = C_{m(\tau)}(\tau) = \frac{\#\{x \in X | \exists p \in P_{m(\tau)} \text{ mit } \|x - p\| \leq \tau\}}{n},$$

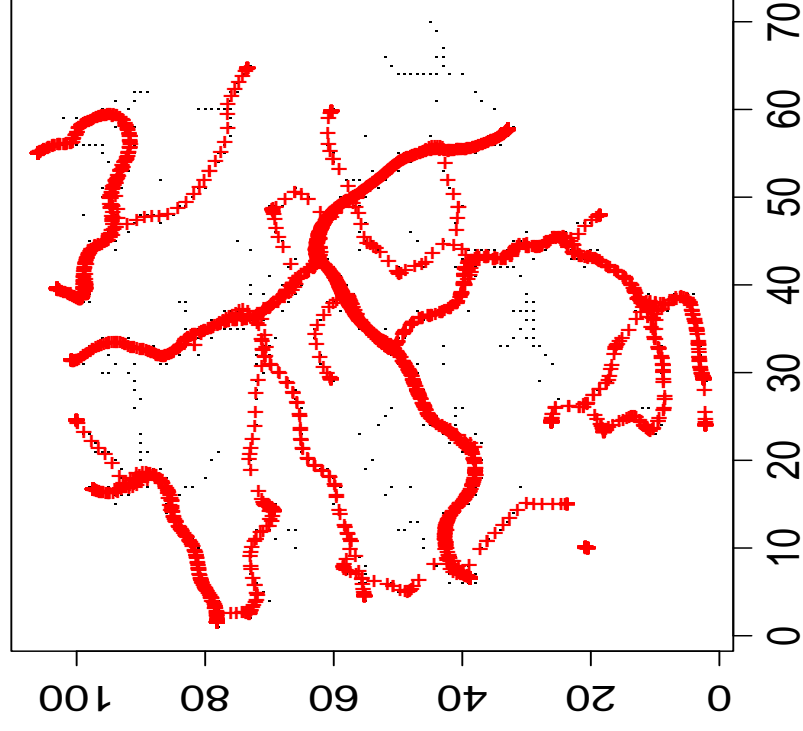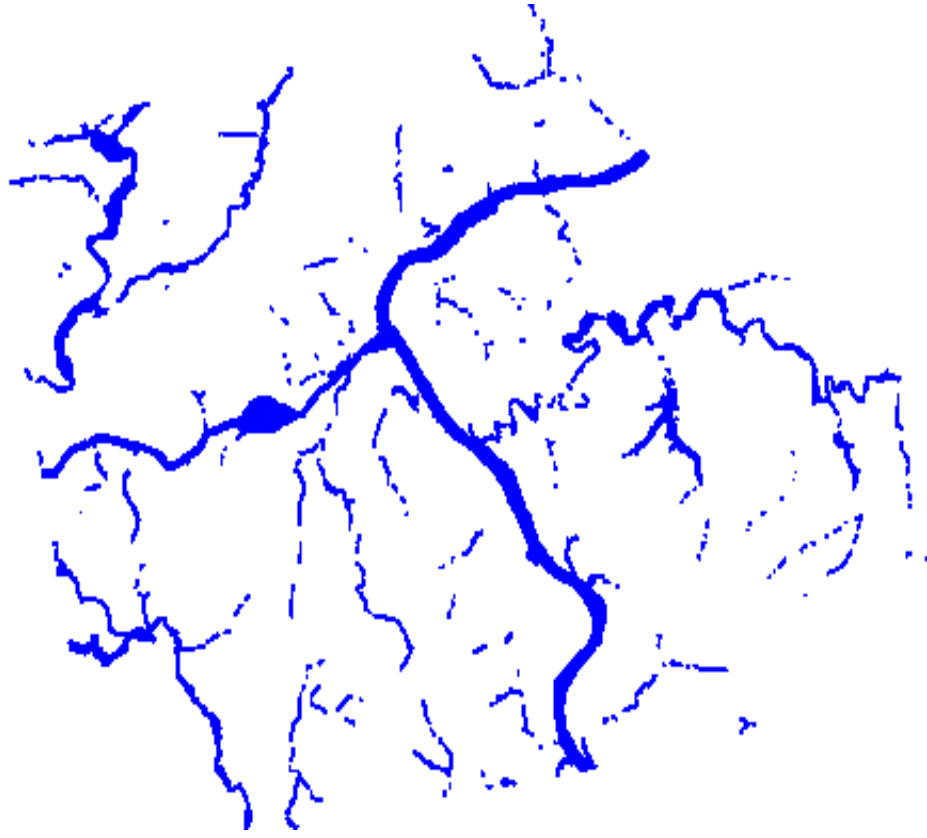where $P_{m(\tau)}$ is the set of points belonging to a principal curve $m(\tau)$ calculated with bandwidth τ. Then

$$h = \text{first local maximum of } \quad S(\tau)$$

is a suitable bandwidth.
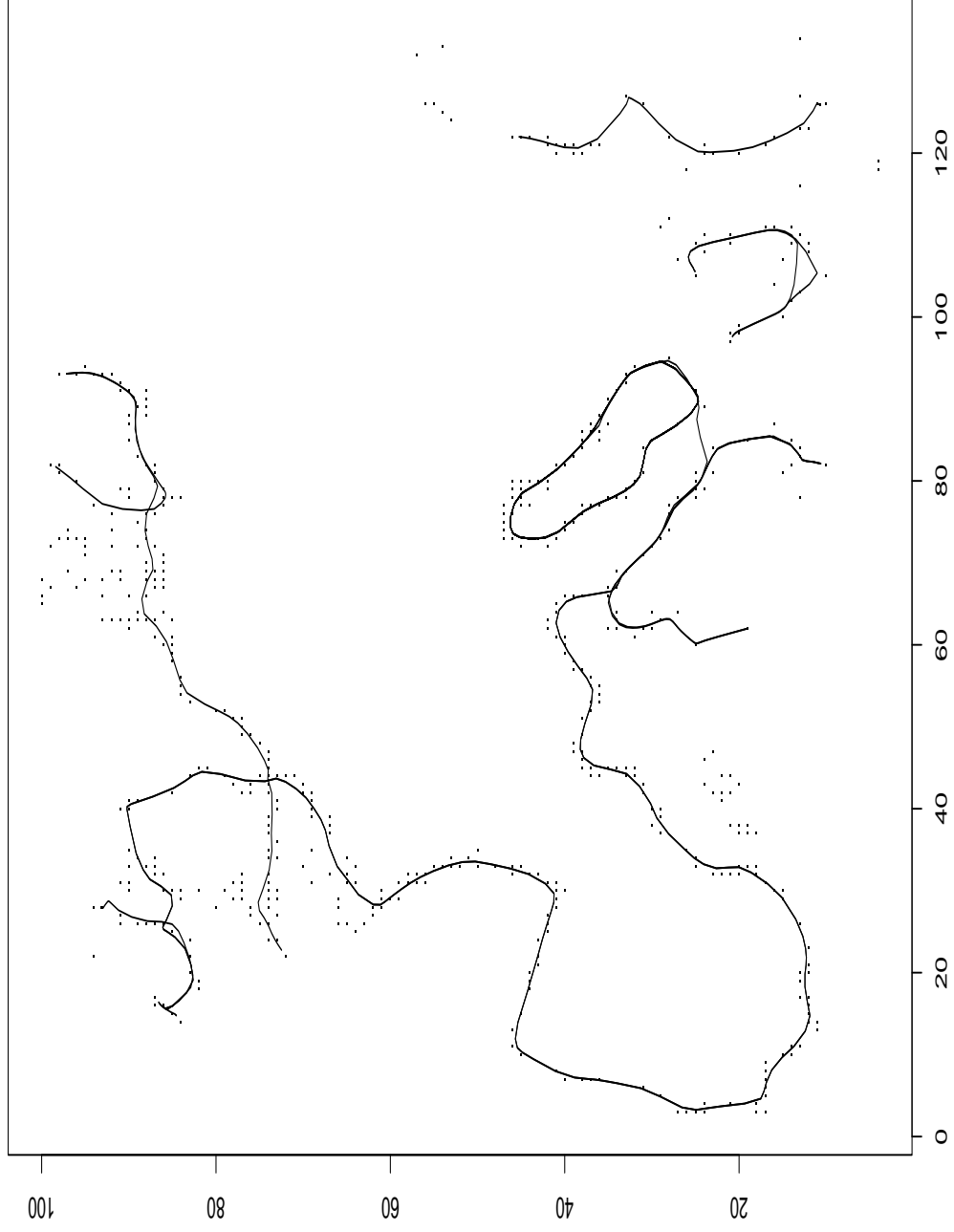
Self-coverage for spiral-data

Real data example: Floodplains in Pennsylvania

LPC with multiple (50) initializations.

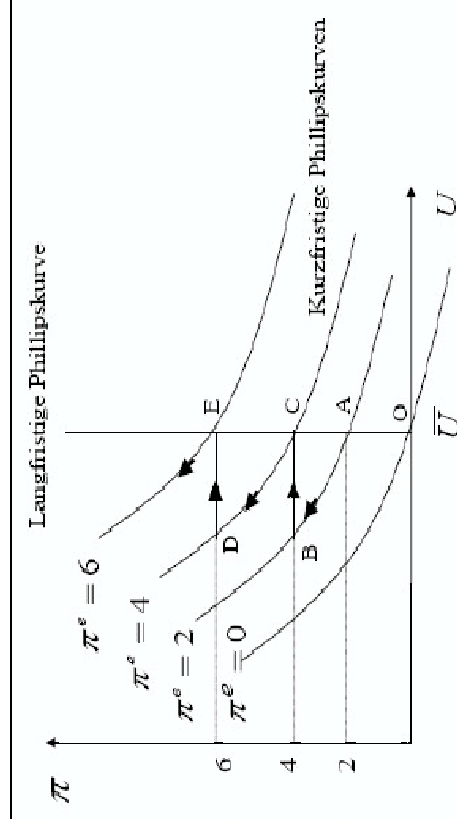Further example: Coastal Resorts in Europe

3D example: Phillips curves

Price index and unemployment in the

Dependance between inflation (price index) and un-

employment rate over time.

Usually just seen as a two-dimensional problem

(infl/rate):

USA, 1995-2005, with LPC:

(Picture from: Prof. Eisen, University of Frankfurt)
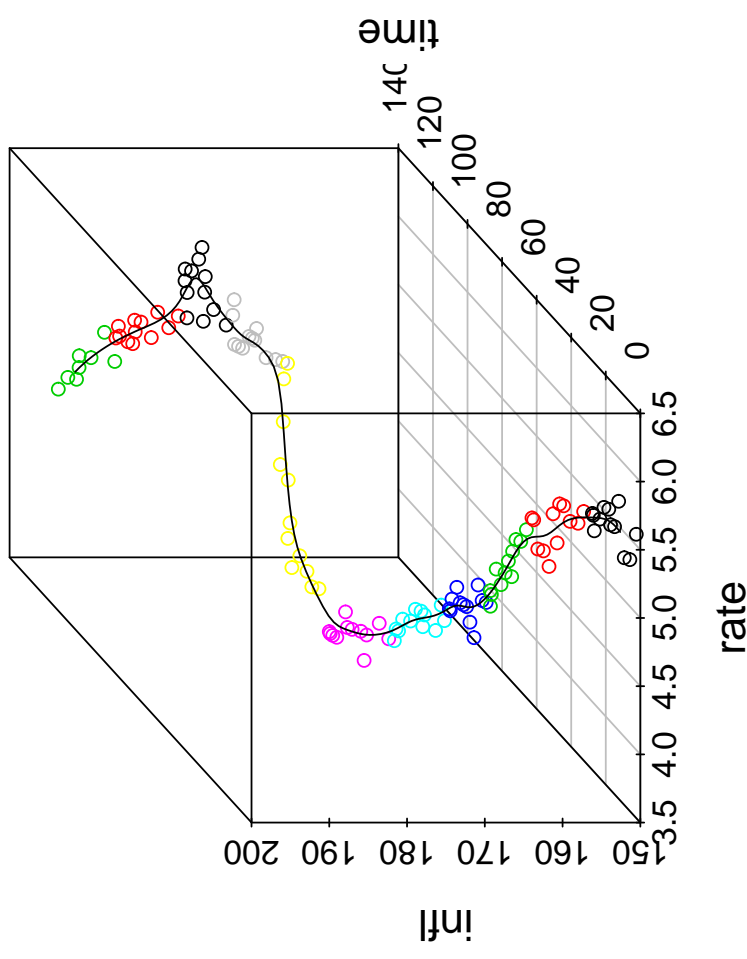
## Higher-order-LPC's

Consider the second local eigenvalue λ_2^x, i.e. the second largest eigenvalue of Σ^x: If this value is large at a certain point of the original LPC, a new LPC is launched in direction of the second local eigenvector γ_2^x. Every bifurcation raises the depth of the LPC tree.

Example
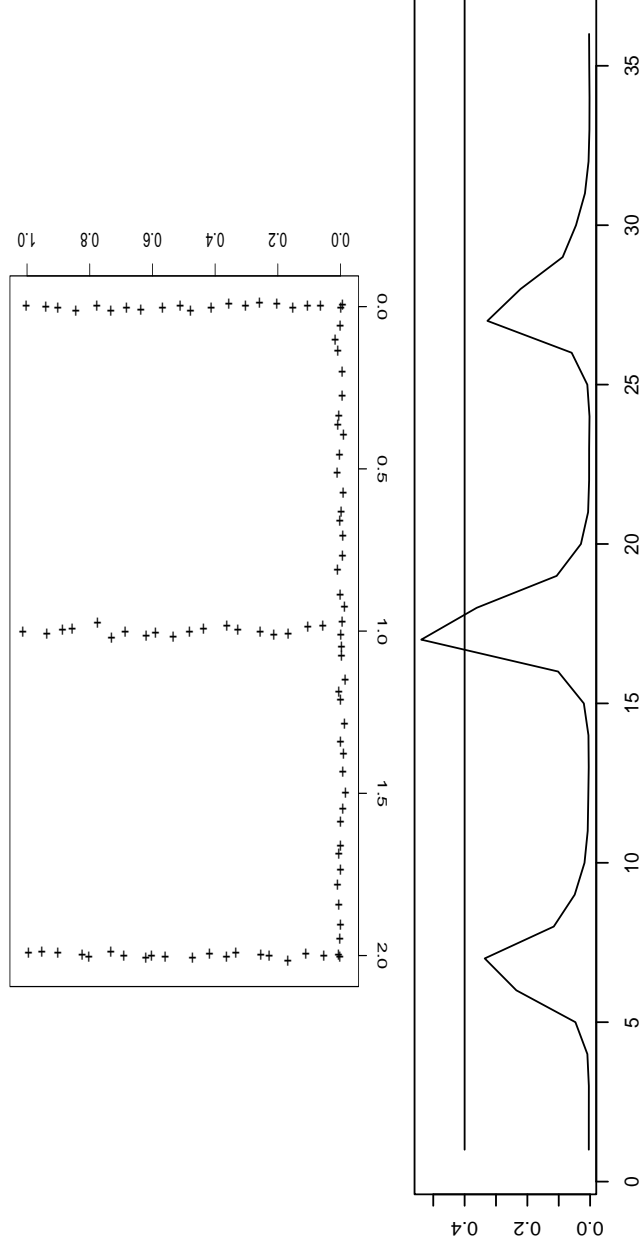
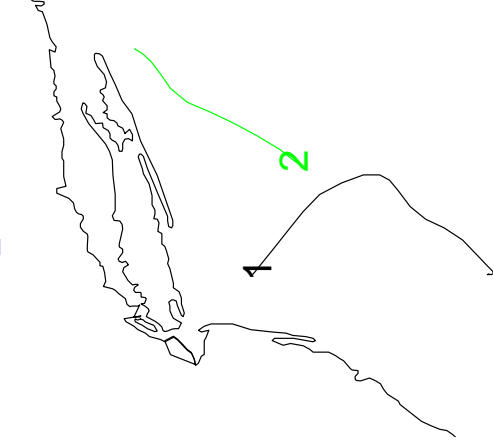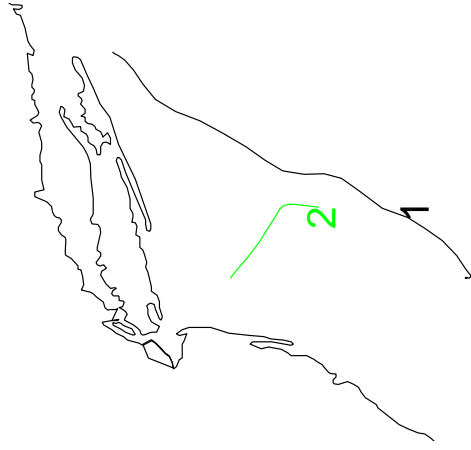

Simulated E and flow diagram of relation λ_2^x/λ_1^x .

LPC's through simulated letters (C,E,K)
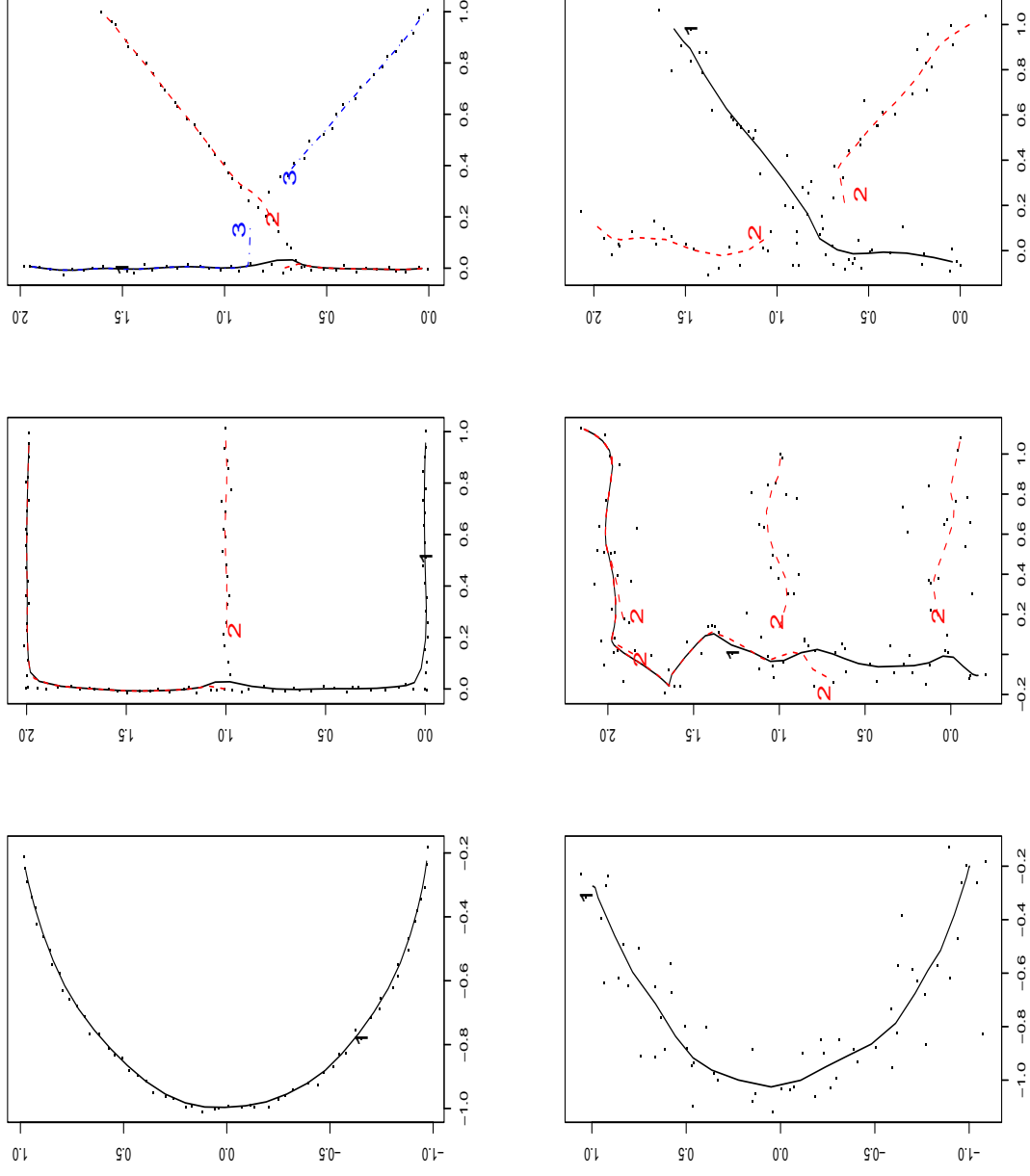
LPC's and corresponding starting points with depth 1, 2, 3.

Example: Scallops
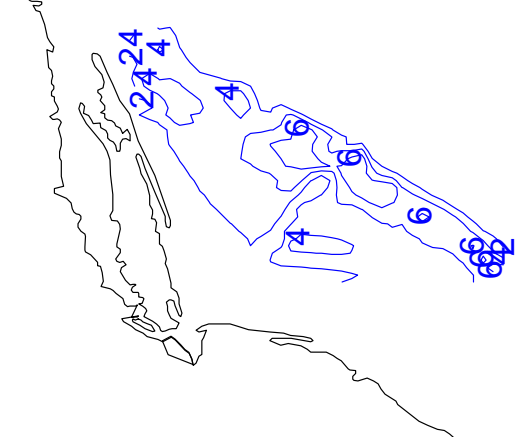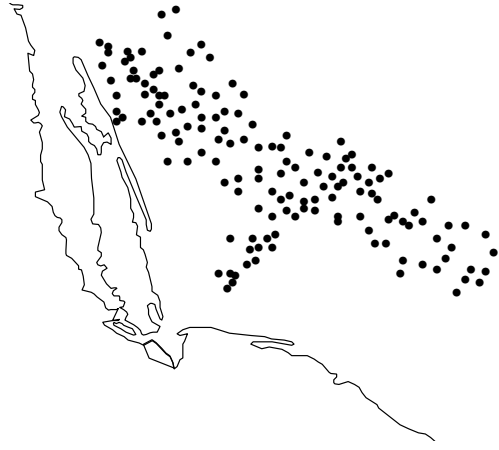
Top left: Scallops

Top right: Water depth

Bottom left, right: Two LPC's

1, 2:   Branches of depth 1, 2.

Example: Bladder cancer microarray data

- $m = 40$ samples

- $n = 3036$ gens

- Analysis in sample space

- Consider subspaces of dimension 10:

```
> dim(sbladder)

[1]   40  3036

> pairs(sbladder[,11:20])

> lpc(sbladder[,11:20], h=1, t0=1, plotlpc=2)
```

Pairs plots vs. local principal curves

The plot to the right shows two-dimensional projections of LPCs calculated in 10-dimensional subspaces.

What's the worth of it?

- Graphical summary

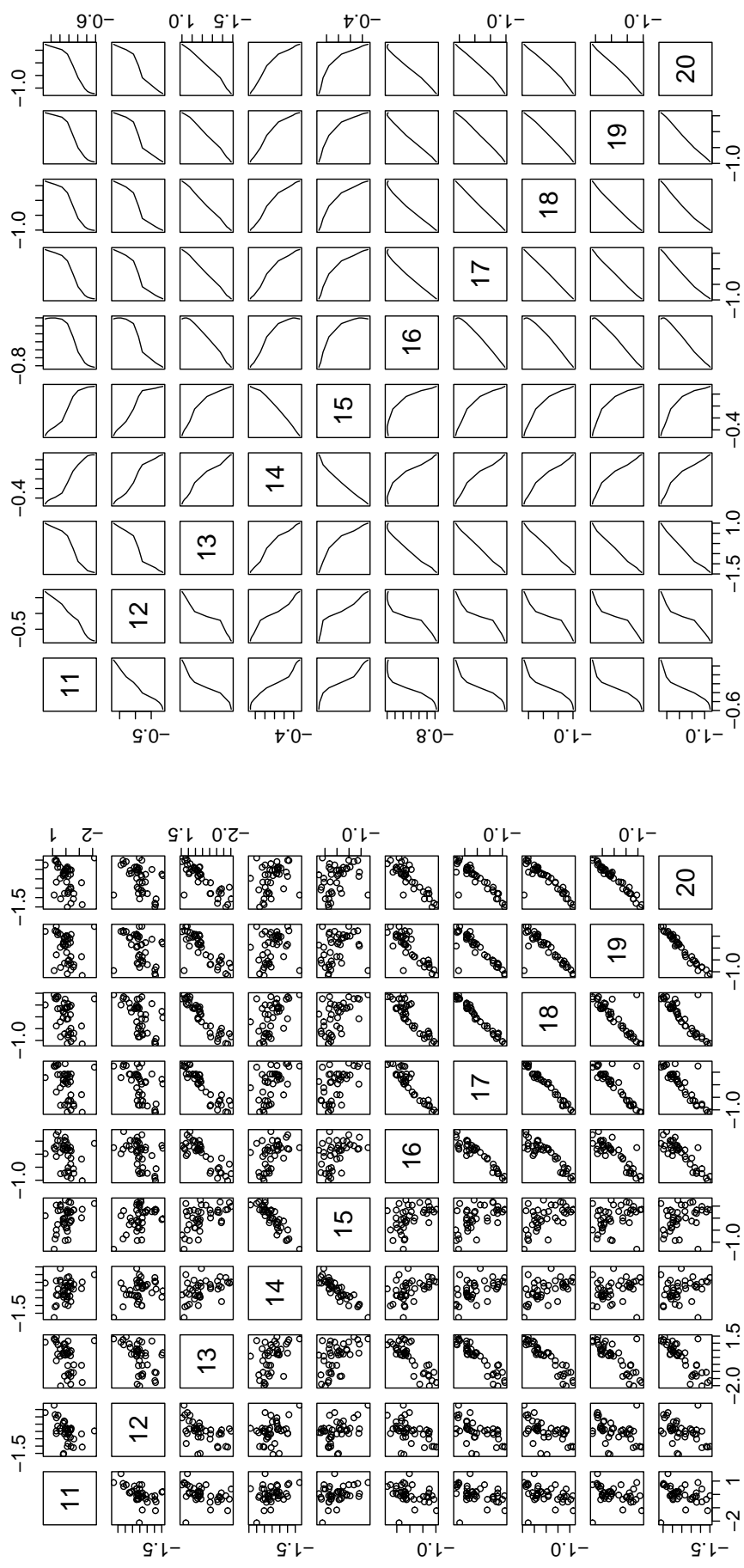- Speed: In this example, LPC is 30% faster calculated (and displayed) than pairs needs only to *plot the data*. This difference rises, and gets increasingly relevant, with increasing sample size.

- Identify groups of genes.

- Find different 'tunnels' through the data' cloud (there might be more than one!). LPC's always go the way of 'lowest resistance', which can lead to different solutions depending on where one starts (*Advantage or disadvantage??*). Given the starting point, LPCs are deterministic.

- works up to ca. 100 dimensions

- ?????

Conclusions

- LPCs work well in a variety of data situations, and seem to be more suitable for some noisy complex structures than its competitors.

- The price to be paid for the increase in flexibility is an increase in variability. Always compute several LPCs to confirm the first run!

- Bandwidth selection works by means of a coverage measure.

- LPCs are not based on a statistical model and hence there is no 'true' principal curve.

- R Code and all data examples available at:

  http://www.nuigalway.ie/maths/je/material/Software/lpc.html

- General drawback of principal curves: Principal curves are not suitable for prediction of Y for given $X = x$.

Outlook: Multi-valued regression (Einbeck & Tutz, 2006)

Goal: Estimate a multifunction $r : \mathbb{R} \longrightarrow \mathbb{R}$ rather than a regression function

Idea: Consider the conditional densities, e.g. for speed-flow data:
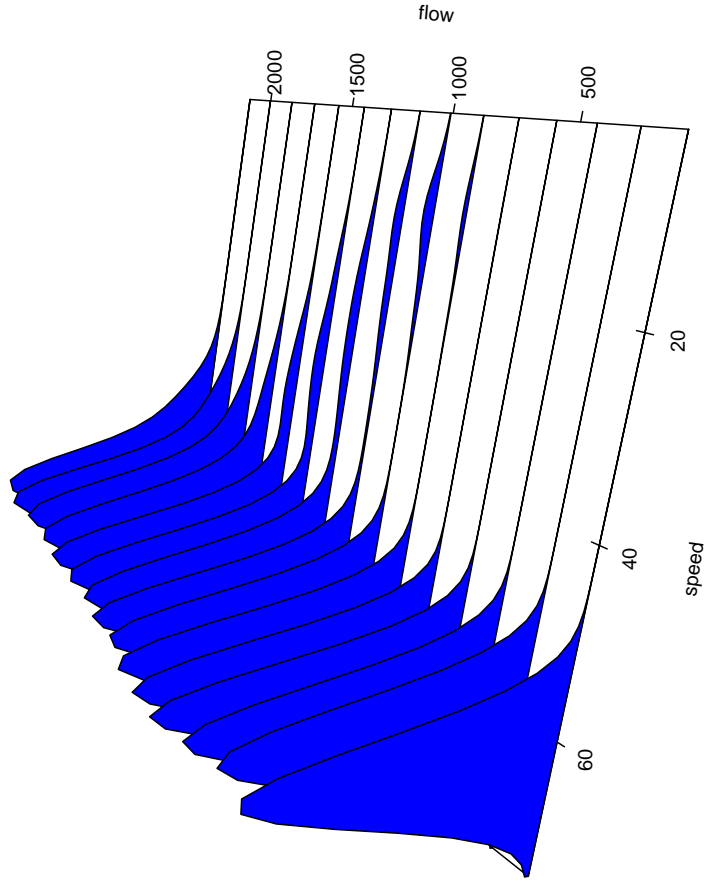
For instance, conditional density at a flow = 1400.
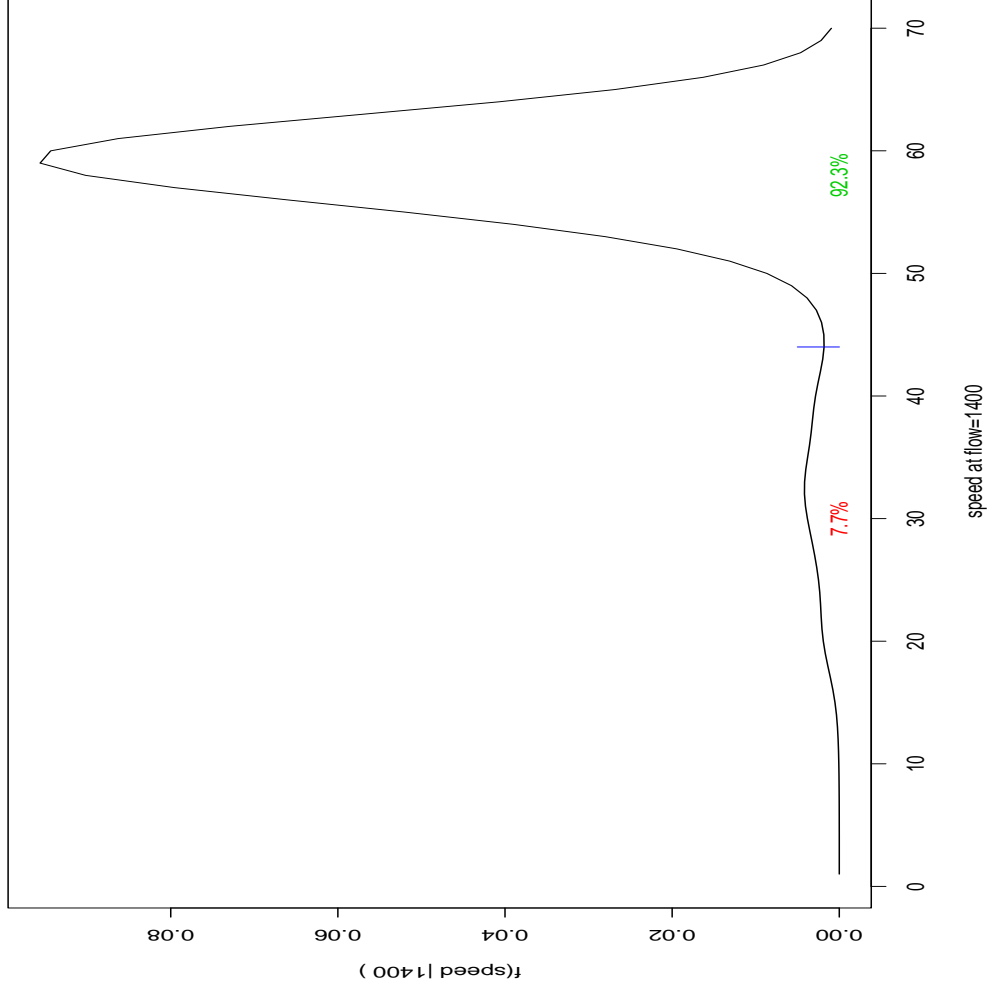
- For estimation of $r(x)$, compute the modes of the estimated conditional densities $\hat{f}(y|x)$.

- The area between a mode and the neighboring 'antimode' serves as estimated probability, that, given x, a value on the corresponding branch is attained.

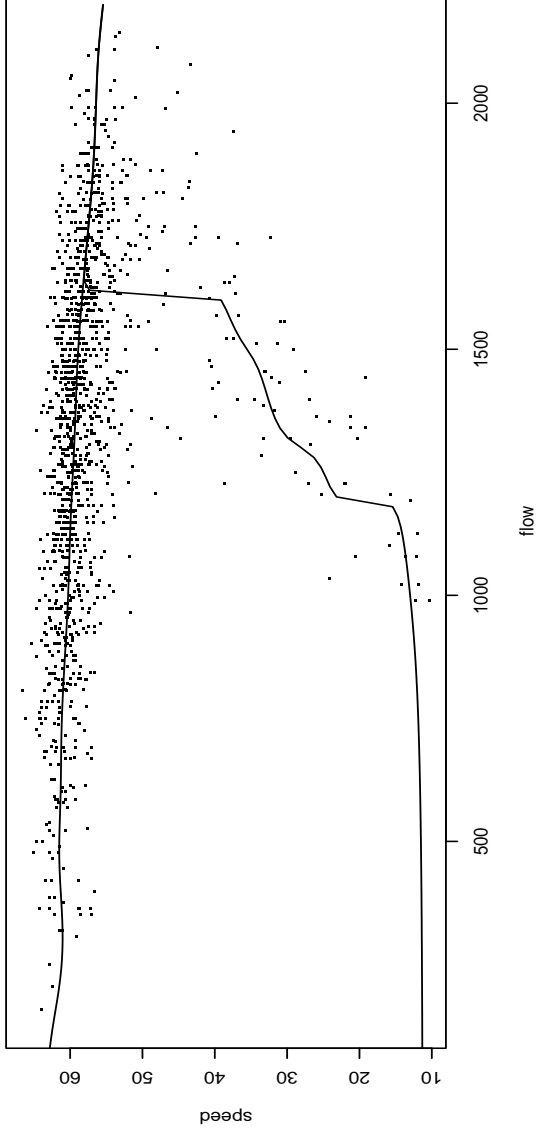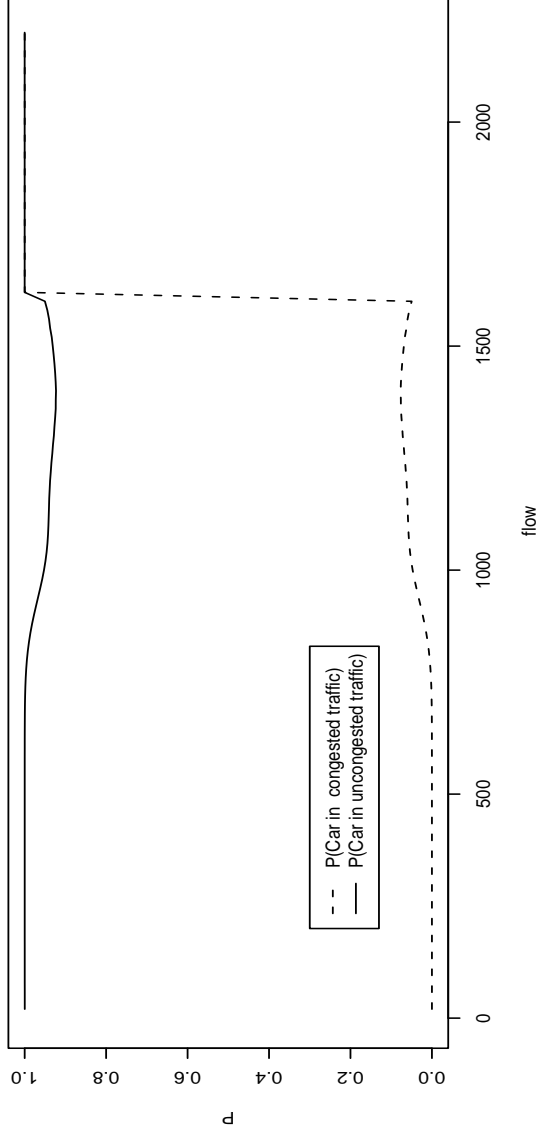Multi-valued regression curve

Relevance assessment

P(Car in congested traffic)
P(Car in uncongested traffic)

Estimation of conditional modes

We are interested in all local maxima of the estimated conditional densities

$$\hat{f}(y|x) = \frac{\hat{f}(x,y)}{\hat{f}(x)} = \frac{\sum_{i=1}^{n} K_1\left(\frac{x-X_i}{h_1}\right) K_2\left(\frac{y-Y_i}{h_2}\right)}{h_2 \sum_{i=1}^{n} K_1\left(\frac{x-X_i}{h_1}\right)}$$

with kernels K_1, K_2 and bandwidths h_1, h_2. We assume that a profile $k(\cdot)$ for kernel K_2 exists

such that $K_2(\cdot) = c_k k((\cdot)^2)$ holds. One calculates

$$\frac{\partial \hat{f}(y|x)}{\partial y} = \frac{2c_k}{h_2^3} \sum_{i=1}^{n} K_1\left(\frac{x-X_i}{h_1}\right) k'\left(\left(\frac{y-Y_i}{h_2}\right)^2\right)(y-Y_i) \overset{!}{=} 0$$

and obtains

$$y = \frac{\sum_{i=1}^{n} K_1\left(\frac{x-X_i}{h_1}\right) G\left(\frac{y-Y_i}{h_2}\right) Y_i}{\sum_{i=1}^{n} K_1\left(\frac{x-X_i}{h_1}\right) G\left(\frac{y-Y_i}{h_2}\right)}. \tag{2}$$

with $G(\cdot) = -k'((\cdot)^2)$.

- Gives conditional mean shift procedure.

- The right side of (2) is just the "Sigma-Filter" used in digital image smoothing.

Literature

Comaniciu & Meer (2002): Mean shift: A robust approach towards feature space analysis. *IEEE Trans. Pattern Anal. Machine Intell.* **24**, *603-619.*

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

Kégl, Krzyżak, Linder & Zeger (2000): Learning and design of principal curves. *IEEE Transactions Patt. Anal. Mach. Intell.* **22**, 281–297.

Kégl & Krzyżak (2002): Piecewise linear skeletonization using 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.

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

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

Einbeck, Tutz & Evers (2005b): Exploring multivariate data structures with local principal curves. In: Weihs, C. and Gaul, W. (Eds.): *Classification - The Ubiquitous Challenge.* Springer, Heidelberg.

Einbeck & Tutz (2006): Modelling beyond regression functions: An application of multimodal regression to speed-flow data. *Journal of the Royal Statistical Society, Series C (Applied Statistics)*, 55, 461–475.