Exploring Multivariate Data Structures with Local Principal Curves

Jochen Einbeck, Gerhard Tutz, and Ludger Evers

Institut für Statistik, Ludwig-Maximilians-Universität München,
Akademiestr.1, D-80799 München, Germany

Abstract. A new approach to find the underlying structure of a multidimensional data cloud is proposed, which is based on a localized version of principal components analysis. More specifically, we calculate a series of local centers of mass and move through the data in directions given by the first local principal axis. One obtains a smooth "local principal curve" passing through the "middle" of a multivariate data cloud. The concept adopts to branched curves by considering the second local principal axis. Since the algorithm is based on a simple eigendecomposition, computation is fast and easy.

1 Introduction

Principal components analysis (PCA) is a well established tool in dimension reduction. For a set of data $\mathbf{X} = (X_1, \ldots, X_n)^T$ with X_i in $\mathbb{R}^d$ the principal components provide a sequence of best linear approximations to that data. Specifically, let Σ be the empirical covariance matrix of $\mathbf{X}$, then the principal components decomposition is given by

$$\Sigma = \Gamma \Lambda \Gamma^T \tag{1}$$

where $\Lambda = \text{diag}(\lambda_1, \ldots, \lambda_d)$ is a diagonal matrix containing the ordered eigenvalues of Σ, with $\lambda_1 \geq \ldots \geq \lambda_d$, and Γ is an orthogonal matrix. The columns of $\Gamma = (\gamma_1, \ldots, \gamma_d)$ are the eigenvectors of Σ. The first eigenvector γ_1 maximizes the variance of $\mathbf{X}\gamma$ over all $\gamma \in \mathbb{R}^d$ with $||\gamma|| = 1$, the second eigenvector γ_2 maximizes the variance of $\mathbf{X}\gamma$ over all $\gamma \in \mathbb{R}^d$ with $||\gamma|| = 1$ which are orthogonal to γ_1, and so on. For illustration, we consider the location of scallops near the NE coast of the United States (Fig. 1; the data are included in the S+ SpatialStats Package). The first and second principal component axes, $g_j(t) = \mu + t\gamma_j \qquad (j = 1, 2, \, t \in \mathbb{R})$, with $\mu = \frac{1}{n}\sum_{i=1}^{n} X_i$, are also depicted. The principal axes unveil nicely the main directions in which the scallops spread out: the first from SW to NE, and the second from NW to SE. In the

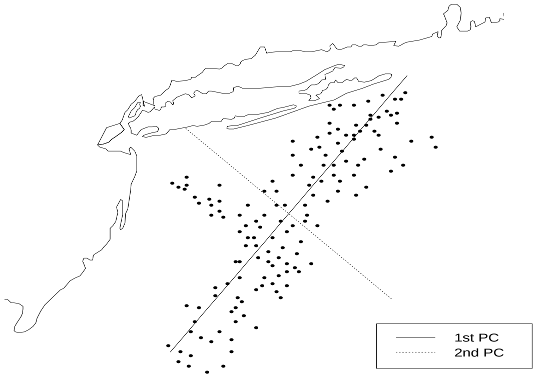

Fig. 1. First and second principal component through scallops near the NE coast of the USA.

data cloud we clearly see two fields of scallops: one along the first principal axis and the other one along the second principal axis. The crossing of the axes is not positioned at the junction of the fields, since the default centering in PCA is at the over-all-center of mass of the data. Intuitively, one might determine the position of the crossing of the two fields by the point on the first principal axis where the spread on the second principal axis is maximal. In the following, we will go one step further and abandon the assumption of linearity, i.e. not only linear structures shall be described, but any form of multivariate curvaceous, possibly branched, connected or disconnected data structures. The goal is to find smooth nonparametric *local principal curves* passing through a data cloud. Therefore, it can be seen as a competitor to the principal curve algorithms from Hastie & Stuetzle (1989), Tibshirani (1992), Kégl et al. (2000), and Delicado (2001). Only the latter one is also based on the concept of localization. However, Delicado does not use local principal components, but rather local *principal directions*, which however cannot be calculated by a simple eigendecomposition. Principal directions are defined as vectors orthogonal to the hyperplane that locally minimize the variance of the data points projected on it. For a comparison of the principal curve algorithms we refer to Einbeck et al. (2003).

2 Local principal curves

Assume a data cloud $\mathbf{X} = (X_1, \ldots, X_n)^T$, where $X_i = (X_{i1}, \ldots, X_{id})^T$. We propose the following algorithm to find the local principal curve passing through $\mathbf{X}$:

Algorithm 1 (Local principal curves)

1. *Choose a set* $S_0 \neq \emptyset$ *of starting points. This may be done randomly, by hand, or by choosing the maximum/maxima of a kernel density estimate.*
2. *Draw without replacement a point* $x_0 \in S_0$. *Set* $x = x_0$.
3. *Calculate the local center of mass*

$$\mu^x = \frac{\sum_{i=1}^n K_H(X_i - x) X_i}{\sum_{i=1}^n K_H(X_i - x)}$$

at x, *where* $K_H(\cdot)$ *is a* $d-$*dimensional kernel function and* H *a multivariate bandwidth matrix. Denote by* μ_j^x *the* j*-th element of* μ^x.

4. *Estimate the local covariance matrix* $\Sigma^x = (\sigma_{jk}^x)$ *at* x *via*

$$\sigma_{jk}^x = \sum_{i=1}^n w_i (X_{ij} - \mu_j^x)(X_{ik} - \mu_k^x)$$

with weights $w_i = K_H(X_i - x) / \sum_{i=1}^n K_H(X_i - x)$, *and* H *as in step 3. Let* γ^x *be the first column of the loadings matrix* Γ^x *computed locally at* x *in analogy to equation (1).*

5. *Update* x *by setting*

$$x := \mu^x + t_0 \gamma^x,$$

where t_0 *determines the step length.*

6. *Repeat steps 3 to 5 until the border of the data cloud is reached. This is the case when the sequence of* μ^x *remains approximately constant. Then set again* $x = x_0$, *set* $\gamma^x := -\gamma^x$ *and continue with step 5.*
7. *Repeat steps 2 to 6 as long as the set* S_0 *is not empty.*

The local principal curve (LPC) is given by the sequence of the μ^x. Note that, in step 5, one has to make sure that the orientation of the local eigenvector $\gamma_{(i)}^x$ after a number i of loops is the same as the local eigenvector $\gamma_{(i-1)}^x$ one loop before, and has to change its signum if $\gamma_{(i-1)}^x \circ \gamma_{(i)}^x < 0$, where $\circ$ denotes the scalar product.

In the sequel, we will extend the algorithm and look at local principal components of higher order. Let the term "k-th local eigenvalue" denote the k-th largest eigenvalue of Σ^x. The $k-$th local eigenvalues λ_k^x $(k \geq 2)$ are useful indicators for branching points.

Definition 1 (Branches of order θ and depth ϕ).

- The order θ of a branch of a LPC is the order of the local principal component which launched it. In other words, $\theta = k$ means that this branch of the LPC was induced by the k-th local eigenvalue. LPC's according to Algorithm 1 lead for all $x_0 \in S_0$ to a branch with $\theta = 1$.
- The depth ϕ of a branch is the number of junctions (plus 1) between the starting point and the branch. Thus, a branch of depth $\phi = \ell$ ($\ell \geq 2$) is launched by a high k-th ($k \geq 2$) local eigenvalue on a branch with $\phi = \ell - 1$. Algorithm 1 always yields curves of depth $\phi = 1$.
- Denote the maximum values of θ and ϕ used to construct a LPC by θ_{max} and ϕ_{max}, resp.

Obviously, $\theta_{max} = 1$ implies $\phi_{max} = 1$; $\theta_{max} \geq 2$ implies $\phi_{max} \geq 2$; and vice versa. The case $\theta_{max} \geq 3$ might be interesting for highdimensional and highly branched data structures. However, for the most applications it should be sufficient to have only one possible bifurcation at each point. Thus, we extend Algorithm 1 only to the case $\phi_{max} \geq 2, \theta_{max} = 2$:

Algorithm 2 (LPC with $\phi_{max} \geq 2, \theta_{max} = 2$)

```
Let $0 \leq \rho_0 \leq 1$ be a suitable constant, e.g. $\rho_0 = 0.5$.
1. Construct a local principal curve $\alpha$ according to Algorithm 1. Compute the relation
       $$\rho^x = \frac{\lambda_2^x}{\lambda_1^x}$$
   for all points $x$ which were involved in the construction of $\alpha$.
2. Iterate for all $\phi = 2, \dots, \phi_{max}$:
   a. Let $\zeta_1, \dots, \zeta_m$ denote all points $x$ belonging to branches of depth $\phi - 1$ with $\rho^x > \rho_0$. If this condition is fulfilled for a series of neighboring points, take only one of them.
   b. Iterate for $j = 1, \dots, m$:
      i. Compute the second local eigenvector $\gamma_2^{\zeta_j}$.
      ii. Set $x := \mu^{\zeta_j} + 2t_0\gamma_2^{\zeta_j}$ and continue with Algorithm 1 at step 3. Afterwards, set $x := \mu^{\zeta_j} - 2t_0\gamma_2^{\zeta_j}$ and continue with Algorithm 1 at step 3.
```

The factor 2 employed in 2.b.ii) for the construction of starting points of higher depth shall prevent that branches of second order fall immediately back to the branch of first order. In order to avoid superfluous or artificial branches one can apply a very simple form of pruning: If starting points of depth $\phi \geq 2$ fall in regions with negligible density, simply dismiss them.

3 Simulated data examples

The performance of the method shall be illustrated by means of some simulated examples. Our simulated data clouds resemble letters, keeping in mind that the recognition of hand-written characters is a possible application of principal curves (Kegl & Krzyzak, 2002). We consider noisy data structures in the shape of a "C", "E", and "K". In all examples, the set of starting points S_0 contains only one element x_0 which was chosen randomly. For the "C" only one branch of depth $\phi = 1$ is needed and thus Algorithm 1 is applied. The letter "E" requires to compute branches of depth up to $\phi = 2$. The letter "K" is even a little more complicated, and depending on the position of x_0 one needs branches of depth $\phi = 2$ or $\phi = 3$. Table 1 shows the setting of the simulation, and the parameter values used in the algorithm. We apply a bandwith matrix $H = h^2 \cdot I$, where I is the 2-dimensional identity matrix.

Table 1. Parameters for simulation and estimation of characters.

	Simulation		Estimation			
	σ	n	ϕ_{max}	θ_{max}	$h = t_0$	ρ_0
"C"	0.01	60	1	1	0.1	–
	0.1	60	1	1	0.15	–
"E"	0.01	100	2	2	0.1	0.4
	0.1	100	2	2	0.1	0.4
"K"	0.01	90	3	2	0.08	0.4
	0.07	90	2	2	0.15	0.4

The results are depicted in Fig. 2. The large amount of tunig parameters might give the impression that finding an appropriate curve might be quite cumbersome. In practice, however, there is only one crucial smoothing parameter: the bandwidth h. The parameter t_0 has certainly to be chosen as well, but it turned out to be a sensible choice setting it equal to the bandwidth. The parameters θ_{max} and ϕ_{max} depend directly on the data structure. The parameter ρ_0 does not play any role when $\theta_{max} = 1$, and will usually be situated in the small interval between 0.3 and 0.6. We illustrate the detection of branching points by means of the "E" with small noise. Fig. 3 shows the flow of the second local eigenvalue starting from the right bottom end of the "E" and rising to the right top end of it. One sees that the peaks are distinct and well localized, and thus useful for the detection of a bifurcation.

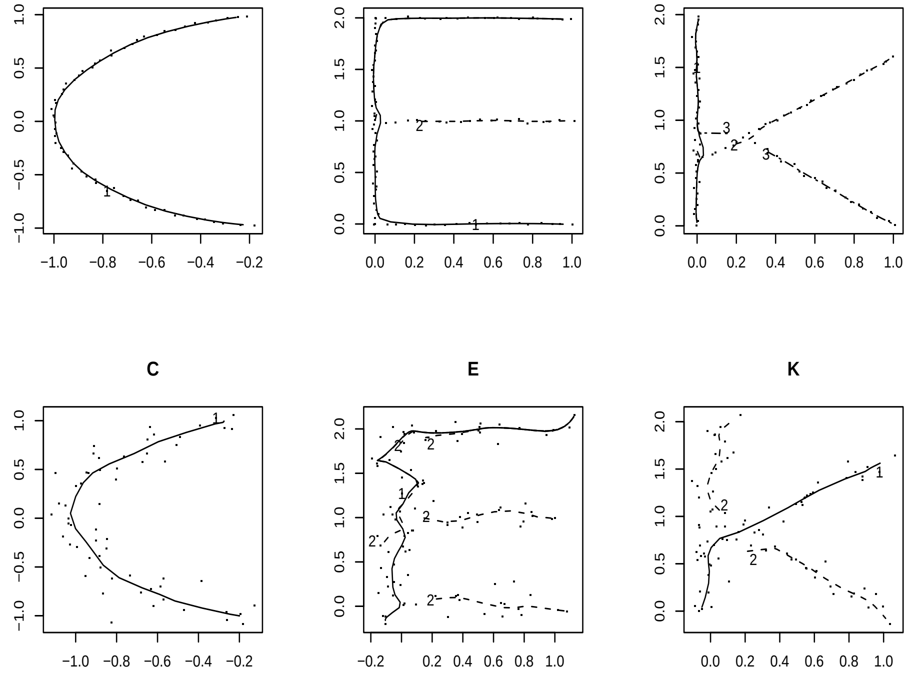

Fig. 2. LPC through letters with small (top) and large noise (bottom). Data points are depicted as "·". Branches of depth $\phi = 1$ are symbolized by a solid line, branches of depth $\phi = 2$ by a dashed line, and branches of depth $\phi = 3$ by a dashed-dotted line. The numbers indicate the starting points for branches of the corresponding depth.

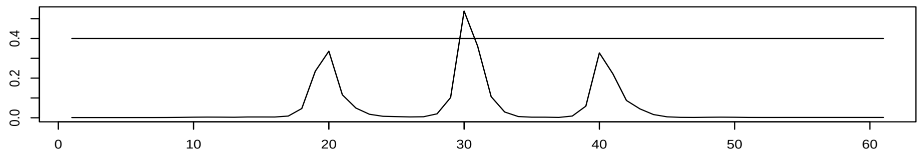

Fig. 3. Flow diagram of $\rho^x = \lambda_2^x/\lambda_1^x$ from the right bottom to the right top of the "E" with small noise. The horizontal line symbolizes the threshold $\rho_0 = 0.4$.

4 Real data examples

We return to the scallops example from the introduction. From the structure of these data it is immediately clear that one needs curves of second order and depth, i.e. $\theta_{max} = 2$ and $\phi_{max} = 2$. Fig. 4 shows that the results of Algorithm 2 can differ for different starting points. Thus, it is natural to ask what the constructed curves represent. Scallops are known to like shallow ocean water. This suggests that the resulting local principal curves follow the ridges of underwater mountains. This hypothesis is confirmed by looking at contour plots from that area (Fig. 4 right). Obviously the left one of the two pictures represents nicely the underwater ridges: One small one from NW to SE (corresponding to the branch with $\phi = 2$), and one larger one from SW to NE (corresponding to the branch with $\phi = 1$). Certainly, the gap between the two branches is not a real feature but is due to the factor 2 employed in step 2.b.ii) in Algorithm 2.

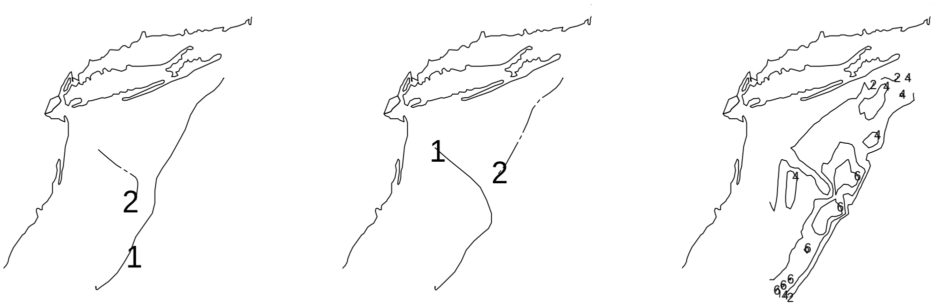

Fig. 4. Left, Middle: LPC of scallops data with bandwidth $h = 0.15$. Branches of depth $\phi = 1$ are launched by starting points "1" and branches of depth $\phi = 2$ start at points "2". Right: Contour plot of underwater plateaus. The numbers indicate the depth: High numbers mean shallow water. In all three pictures the NE coast line of the USA is plotted for orientation.

The scallops data are highly noisy, but not very far from linearity. We will provide one more real data example with data having small noise, but having a very complex nonparametric structure. The data are coordinates of European coastal resorts (taken from Diercke, 1984). Suppose one wants to reconstruct the European coast line given these sites. The European coast does not have mentionable ramifications, thus we use $\phi_{max} = \theta_{max} = 1$, but choose 10 starting points randomly. A typical result is shown in Fig. 5. Taking into account that Algorithm 1 does not have the notion about the shape of Europe that humans have, the coast is reconstructed nicely, although it failed to describe areas with very few data, as Albania, and highly chaotic regions as Schleswig-Holstein and Southern Denmark.

5 Conclusion

We demonstrated that local principal components can be effectively used to explore the structure of multivariate complex data structures. The method

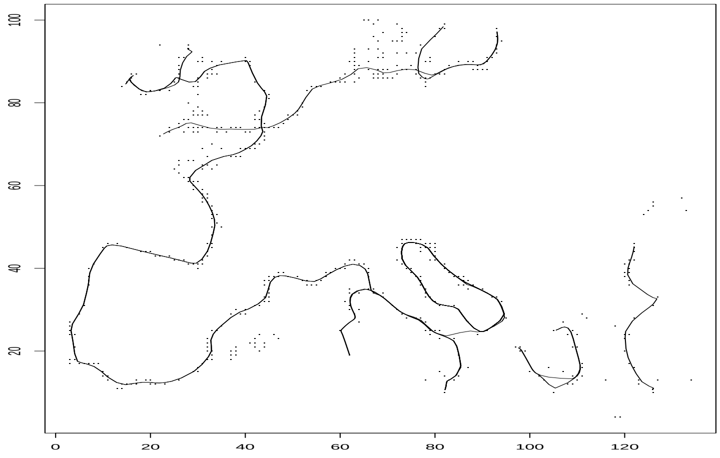

Fig. 5. LPC (solid line) through European coastal resorts (·). The positions are given on a digitalized 101 × 135 grid, and the applied bandwidth is $h = 2$, meaning that about 2 digits in each direction are considered for construction of the curve.

is especially useful for noisy spatial data as frequently met in geostatistics. The next step should be to reduce the dimensionality of the predictor space in a multivariate regression or classification problem by employing the local principal curve as low-dimensional, but highly informative predictor.

References

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

DIERCKE (1984): Weltatlas, Westermann Verlag GmbH, Braunschweig, Germany.

EINBECK, J., TUTZ, G., and EVERS, L. (2003): Local Principal Curves. *SFB386 Discussion Paper No. 320*, LMU München.

HASTIE, T. and STUETZLE, L. (1989): Principal Curves. *JASA, 84, 502-516.*

KÉGL, B., and KRZYZAK, A., (2002): Piecewise Linear Skeletonization using Principal Curves. *IEEE Trans. Patt. Anal. Mach. Intell., 24, 59-74.*

KÉGL, B., KRZYZAK, A., LINDER, T. and ZEGER, K. (2000): Learning and Design of Principal Curves, *IEEE Trans. Patt. Anal. Mach. Intell., 22, 281-297.*

TIBSHIRANI, R. (1992): Principal Curves Revisited. *Statistics and Computing, 2 , 183-190.*