International Journal of Neural Systems, Vol. 0, No. 0 (April, 2000) 00–00

DATA COMPRESSION AND REGRESSION THROUGH LOCAL PRINCIPAL CURVES AND SURFACES

JOCHEN EINBECK
Department of Mathematical Sciences, Durham University
Durham DH1 3LE, England
E-mail: jochen.einbeck@durham.ac.uk

LUDGER EVERS
Department of Statistics, University of Glasgow
Glasgow G12 8QQ, Scotland
E-mail: ludger@stats.gla.ac.uk

BENEDICT POWELL
Department of Mathematical Sciences, Durham University
Durham DH1 3LE, England
E-mail: benedict.powell@durham.ac.uk

Received (to be inserted
Revised by Publisher)

We consider principal curves and surfaces in the context of multivariate regression modelling. For predictor spaces featuring complex dependency patterns between the involved variables, the intrinsic dimensionality of the data tends to be very small due to the high redundancy induced by the dependencies. In situations of this type, it is useful to approximate the high-dimensional predictor space through a low-dimensional manifold (i.e., a curve or a surface), and use the projections onto the manifold as compressed predictors in the regression problem. In the case that the intrinsic dimensionality of the predictor space equals one, we use the local principal curve algorithm for the the compression step. We provide a novel algorithm which extends this idea to local principal surfaces, thus covering cases of an intrinsic dimensionality equal to two, which is in principle extendible to manifolds of arbitrary dimension. We motivate and apply the novel techniques using astrophysical and oceanographic data examples.

1. Introduction

Nowadays, we are confronted with data of ever increasing complexity. There are three main manifestations of this complexity. Firstly, it is not unusual to observe sample sizes of formerly unthinkable magnitudes. Although this never posed a methodological problem, such data sets could not be handled in the past due to data storage and computational limitations; however with advances in modern technology the sample size in itself does not constitute a problem any more.

The second manifestation of complexity is more severe. Often, not only the number of observations collected is large, but also the number of variables involved. This problem, sometimes referred to as "$p \gg n$", is challenging not only from a computational point of view, but also from a methodological point of view. Consider the example of variable selection: the number of possible subsets of a set of p variables is 2^p, which is even for a moderately large number like $p = 20$ already more than a million.

The third manifestation of complexity has to do with the intrinsic structure of the data themselves.

Advances in science and modern technology have enabled us to look deeper than ever into formerly inaccessible structures, yielding data with complex dependency patterns. Whilst this might appear as a curse at first sight, it can actually be a blessing: the complexity of high-dimensional data is often due to the high redundancy of the variables involved. Exploiting this redundancy allows avoiding many of the pitfalls of high-dimensional data analysis.

In Astrophysics for example, an issue of current research is to extract information on stellar parameters from photon counts collected at many different wavelengths, paired with huge numbers (thousands or millions) of observations.[1,2] Figure 1 shows a scatterplot matrix of photon counts recorded at a subset of 16 different wavelengths. Most variables are very strongly related. However, this relationship is non-linear, so that the association between these variables would not be captured using the correlation coefficient. We will show that exploiting this lower-dimensional latent structure of the data allows for building better models for predicting the stellar parameters.

Clearly, for situations of this type — but also for much simpler problems — it is inefficient to operate with a full interaction model of type $Y = m(X_1, \ldots, X_p) + \text{noise}$. Here, Y is the response variable, for instance the stellar temperature, and $(X_1, \ldots, X_p)$ are the predictors, corresponding here to the photon counts at different wavelengths. Statisticians have developed a huge range of tools in order simplify the full interaction model so that it is more tractable. Common simplifications are, in decreasing order of complexity, project pursuit regression, the additive model, the partially linear model, or, most simply, the multivariate linear model.[3] Due to the exponentially increasing difficulty of the model selection process mentioned above, a second string of research has looked for alternative ways of simplifying the model, and this family of methods is known under the term *dimension reduction*. These methods aim to compress the space of predictors $X = (X_1, \ldots, X_p)$ before the actual model is fitted, i.e. we have a two-stage strategy:

1 Find a dimension-reducing mapping $f : \mathbb{R}^p \longrightarrow \mathbb{R}^d$, with $d < p$, giving *compressed* data $T = f(X) \in \mathbb{R}^d$
2 Base further inference on a regression model for Y using T instead of the X as covariates.

The best-known example of a dimension-reducing mapping is *principal component analysis* (PCA). Other examples of such a technique include auto-associative neural networks and self-organizing maps.[4] In this article we will explain how principal curves and surfaces can be used as a dimension-reducing mapping.

We will start with reviewing principal components, which, in combination with linear regression, is often referred to as *principal component regression* (PCR). Here, the function f projects $X \in \mathbb{R}^p$ onto the d-dimensional space spanned by the principal components corresponding to the largest d eigenvalues $\lambda_1, \ldots, \lambda_d$ of $\Sigma = \mathrm{Cov}(X)$:

$$f : \mathbb{R}^p \longrightarrow \mathbb{R}^d,\ X \mapsto (\gamma_1, \cdots, \gamma_d)^T (X - m),$$

$i = 1, \ldots, n$, where $m = E(X)$ and $\gamma_1, \ldots, \gamma_d$ are the corresponding eigenvectors.

Several alternative mappings have been proposed, which have in common with PCA that they can be written as an affine transformation of X, i.e. there is some $d \times p$ matrix B and a d-dimensional vector c such that $f(X) = BX + c$. Members of this family of methods include sliced inverse regression[5] and parametric inverse regression.[6]

In our context, the word "compressing" means nothing else than "projecting". That is, each data point will be projected onto the nearest point on the dimension-reduced subspace. In projecting data onto this subspace, we have to be prepared to lose some information compared to the original "raw" data, which may impact on the accuracy of our fitted model. However, there is also a huge potential gain compared to the model based on the raw data: if we have reduced the dimension in step 1, we may be able to use a far more flexible and accurate model in step 2. For instance, instead of a linear model with many variables, we may use a one or two-dimensional non-parametric smoother. In other words, there is some trade-off to be made between the loss of information in the projection step and the gain in precision in the estimation step. What the best trade-off will be, will largely depend on how meaningful the projections in step 1 are. If the predictor space features a strongly non-linear shape, then the projections onto a linear subspace (such as in PCA) may be of limited use. To illustrate this point more clearly, assume we are given

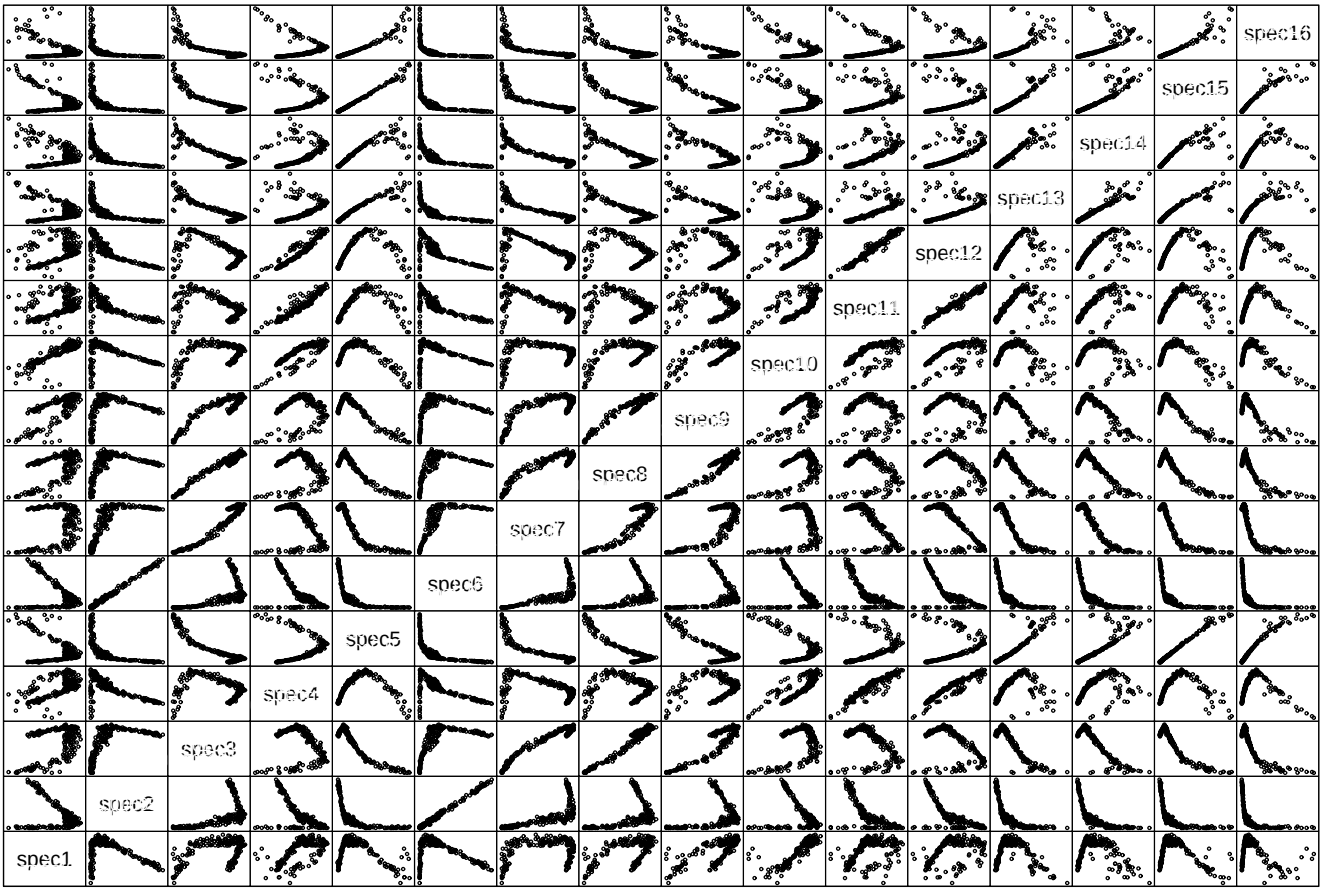


Fig. 1. Pairwise matrix scatterplot of photon counts (fluxes) obtained at 16 different wavelengths. The data were simulated through computer models within the Gaia project.[2]

a spiral-shaped bivariate predictor space as in figure 2 (left panel). The dashed line shows the first principal component line through this data cloud, which explains about 54% of the total variance. Clearly, the projection indices (PIs) of the data projected orthogonally onto this line will be uninformative for the actual position of the data point within this cloud, just as it would be the case for any other linear approximation of this data. In order to capture the intrinsic structure of this data, one has to fit a curve through it nonparametrically. The statistical term for such a smooth curve "through the middle of a data cloud" is a *principal curve*.[7] The solid line in Figure 2 (left panel) shows such a curve fitted using the technique of *local principal curves*.[8] Visually, the curve provides a good one-dimensional summary of this data set. In order to use this curve for dimension reduction purposes, one has to be able to parametrize this curve, or at least to project data points onto it. The projections onto the local principal curve are shown in Figure 2 (right panel), and the resulting projection indices are informative for the position of the data points within the cloud. Whether these projection indices are more informative for a (hypothetical) response variable than the straight line projections, is, of course, a question that we cannot answer in this example, but we would hope that this will be the case. We will see three examples in Section 3 where this turns out to be the case.

An important concept that we will refer to is that of *intrinsic dimensionality*. We consider this term as being equivalent to the *topological dimensionality*, which is the basis dimension of the local linear approximation of the hypersurface on which the data resides, i.e. the tangent space.[9]

For instance, the data in figure 2 appear to have a topological dimension of one as they could be locally approximated by a tangent to the curve in each

local neighborhood along the curve. This paper will focus on data which feature a topological dimension of one or two, in which cases we will use local principal curves and surfaces, respectively, in the compression step. These terms should be separated from the notion of *structural dimensionality* as advocated for instance by Cook[10], which is the dimension of the central subspace, i.e. the smallest linear subspace which contains all relevant information about the response.

We proceed in the following section with setting up the local principal curve methodology that we shall be using to handle situations with intrinsic dimensionality equal to 1. We provide several real data examples and a comparison with other dimension reduction techniques in Section 3, and extend our methodology towards two-dimensional nonparametric data summaries (in form of principal surfaces) in Section 4. We finish with a conclusion in Section 5.

2. Dimension reduction via principal curves

2.1. *Local principal curves*

Local principal curves (LPC)[8] are based on the idea that, at each point $x \in \mathbb{R}^p$ along a principal curve, the localized first principal component line forms the best one-dimensional linear approximation to the curve. They can be seen as a simple and fast approximation to the mathematically and computationally more demanding concept developed earlier by Delicado.[11] Assume we are given data $x_1, \ldots, x_n \in \mathbb{R}^p$ of which we think as n independent replicates drawn from the random vector $X = (X_1, \ldots, X_p)^T$, i.e. $x_i = (x_{i1}, \ldots, x_{ip})^T$.

Beginning at some starting point $x = x_0 \in \mathbb{R}^p$, LPCs proceed through the data cloud, alternating between the following two steps:

(i) Calculate a localized center of mass $\mu^x = \sum_{i=1}^n w_i^x x_i$, where $w_i^x = K_H(x_i - x)X_i / \sum_{j=1}^n K_H(x_j - x)$.

(ii) Compute the first local eigenvector γ^x of $\Sigma^x = (\sigma_{jk}^x)_{(1\le j,k\le p)}$, where $\sigma_{jk}^x = \sum_{i=1}^n w_i^x (x_{ij} - \mu_j^x)(x_{ik} - \mu_k^x)$ and μ_j^x denotes the $j-$th component of μ^x. Using a predetermined step size z, step from μ^x to $x := \mu^x + z\gamma^x$.

The sequence of the local centers of mass μ^x makes up the local principal curve. Here, $K_H(\cdot) = |H|^{-1/2}K(H^{-1/2}\cdot)$, with a multivariate kernel K and a positive definite bandwidth matrix $H = \mathrm{diag}(h_1^2, \ldots, h_p^2)$. Just as for usual PCA, it is recommendable to use input variables $X_1, \ldots, X_p$ which are operating on similar scales, which can be achieved e.g. by dividing by their range or standard deviation. In this case, it is common to use bandwidths $h \equiv h_1 = h_2 = \ldots = h_p$, and to choose $z = h$ as well. The LPC algorithm has been extended to disconnected[8] and branched[12] curves, which can be easily implemented using suitable multiple starting points. Crossings can be handled conveniently using an angle penalization.[8] As in each iteration only points in the local neighborhood are considered, the algorithm is quite flexible, and, at the same time, robust to outliers.

2.2. *Parametrization, projection, and feature extraction*

For a fitted LPC consisting of L local centers of mass $\mu^{x_\ell} \equiv \mu^\ell = (\mu_1^\ell, \ldots, \mu_p^\ell)^T$, $\ell = 1, \ldots, L$, we seek a curve $\{g(t),\ t \in I_g\}$ which interpolates the local centers of mass. This curve can be parametrized by a function

$$g : I_g \longrightarrow \mathbb{R}^p,\ t \mapsto (g_1(t), \ldots, g_p(t))^T,$$

where $I_g \subset \mathbb{R}$ denotes the domain of g. The parameter t corresponds to the projection index. Firstly, one end point is chosen to be the origin corresponding to $t = 0$. This is an arbitrary choice and we will use the convention that t increases in the direction of γ^{x_0}. Technically, the curve is parametrized in three steps:

(i) Compute a discrete, preliminary parametrization $(s_\ell)_{(1\le\ell\le L)}$, with the same origin as t, by adding up the Euclidean distances between subsequent $\mu^\ell, \ell = 1, \ldots, L$.

(ii) For each dimension of the covariate space $j = 1, \ldots, p$, interpolate the points $(s_\ell, \mu_j^\ell)_{1\le\ell\le L}$ by a cubic spline, yielding graphs $(s, \mu_j(s))$. Putting them together, one obtains a continuous and differentiable spline function $(\mu_1, \ldots, \mu_p)^T(s) \equiv \mu(s)$.

(iii) For each value of s within the support of the spline function, recalculate the parameter using the arc length,

$$t = \int_0^s \sqrt{(\mu_1'(u))^2 + \ldots + (\mu_p'(u))^2}\, du,$$

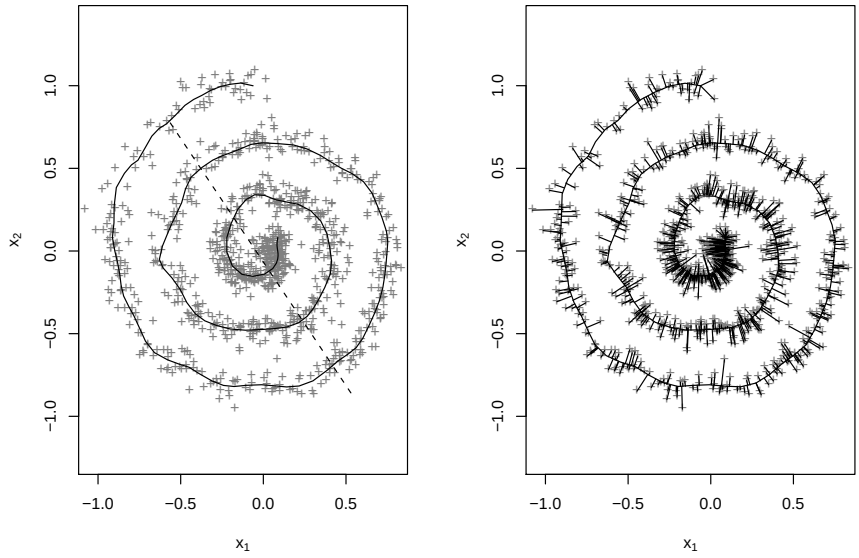


Fig. 2. Left: illustrating example comparing the principal component line (dashed) to a principal curve (solid) as dimension-reducing mapping; right: orthogonal projections onto the principal curve.

and set $g(t) = \mu(s)$.

It should be noted that no smoothing is involved in (ii) — the μ^ℓ are just interpolated.

Once this parametrization is established, each data point x_i, $i = 1, \ldots, n$, can be projected onto the curve by finding the point on the curve which is nearest to it (in terms of Euclidean distances), yielding the projection index t_i. More formally, the dimension reducing mapping is given by

$$T \equiv f(X) = \sup_{t \in I_g} \{||x - g(t)|| = \inf_{\tau \in I_g} ||x - g(\tau)||\}. \quad (1)$$

This definition goes back to the original principal curve paper[7]: Hastie and Stuetzle use the projection indices both in the definition of principal curves and in the algorithm for fitting them. However they did not make any further use of the projection indices. More recently, Ming-Ming et al.[13] emphasized the significance of the function $f(\cdot)$ as a *feature extractor* for X. The logical next step is to base further inference about the response variable of a regression model on the extracted features $t_i \equiv f(x_i), i = 1, \ldots, n$.

2.3. *Regression and prediction*

In order to link the extracted feature T to the response Y, we proceed by fitting a univariate regression model

$$y_i = m(t_i) + \epsilon_i,\ i = 1, \ldots, n.$$

The function $m : \mathbb{R} \longrightarrow \mathbb{R}$ could in principle be specified parametrically, for instance $m(t_i) = a + bt_i$. An example for this will be provided in Section 3.1. However, in the vast majority of situations where we have to cope with data structures which are sufficiently complex to justify application of the techniques mentioned above, we will also expect the response to be non-trivially related to the extracted feature, so that typically $m(\cdot)$ will need to be modelled nonparametrically. Univariate nonparametric smoothing is a standard procedure and well-studied routines performing this job are readily available. For instance, smoothing splines, local polynomials, but also feed-forward neural networks could be used here.

Assume finally that we have a new observation $x_{new} \in \mathbb{R}^d$ available and wish to predict the yet unobserved response y_{new}. This is now achieved in two steps:

(i) Using (1), project x_{new} onto the LPC g. This gives a projection index t_{new}.
(ii) Compute $\hat{y}_{new} = \hat{m}(t_{new})$ from the fitted nonparametric smoother.

We will give some examples illustrating these techniques in the next section.

3. Data examples

3.1. *New Zealand Horse mussels*

We consider data consisting of measurements of the shell height (H), shell length (L), shell width (W), shell mass (S), and the edible muscle mass of the mussels in gram (M) of 172 horse mussels. We will use the edible muscle mass (M) as the response variable. These data were repeatedly analyzed in the context of dimension reduction.[10,6] The latter reference also performs a test based on the singular values of the standardized matrix of inverse regression coefficients to demonstrate that the structural dimension of the predictor space can be taken to be equal to one. There is no theoretical justification which would allow us to conclude that the topological and structural dimension should necessarily be the same. Nevertheless, visual inspection of the four-dimensional mussel characteristics (figure 3, top panel), seems to give sufficient evidence to allow us to work with an intrinsic dimension of $d = 1$. A local principal curve is fitted, with the result shown in figure 3 (bottom panel): it matches closely the appearance of the raw data.

We proceed with projecting the predictors onto this curve, and plotting the response against the projection indices. The resulting scatterplot is shown in figure 4 (left), which shows clearly a linear relationship between muscle mass and the projection index. The resulting linear regression line $y = 1.037 + 0.113T$ has a residual standard error of 4.108 on 80df, and the coefficient of determination R^2 takes the value 0.879. For comparison, Bura and Cook[6] derived another one-dimensional summary of the predictor space via parametric inverse regression. Specifically, they propose to define a new variable, say C, as

$$C = 0.028H - 0.029L - 0.0593\log(S) + 0.804\log(W).$$

From the right plot in figure 4 it is evident that a simple linear regression of M against C is not adequate here. Therefore, we employ a quadratic model, yielding the regression curve $y = -2.230 - 3.832C + 0.964C^2$ with a residual standard error of 6.051 on 79df and corresponding $R^2 = 0.7401$. This curve is shown in figure 3 (right). Clearly, the fit based on the LPC performs superior in all aspects, and, in contrast to parametric inverse regression[6], the method does not require "visual inspection of the scatterplot matrix" in order to "decide what functions of Y fit the data best".

One may have doubts on the stability of the LPC-based result, as the fitted local principal curve depends (slightly) on the position of the starting point x_0. To check this, we ran the LPC algorithm 100 times, each time selecting a starting point at random from the cloud. The mean of the residual standard errors of the 100 linear regression models was 4.1159 with a standard deviation of 0.0515, indicating that the estimated line is very stable and that the differences in the fitted local principal curve only play a marginal role. More care is, of course, needed if the predictors are highly scattered in space. An example for such a situation will be provided below.

3.2. *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. A satellite is to be launched in 2012, which will collect spectra (photon counts at certain wavelengths) from objects all over the universe. The aims of the mission, among others, are to classify objects (as star, galaxy, quasar,...), and to learn about stellar properties in form of certain astrophysical parameters ("APs": temperature, metallicity, gravity, etc.).

Until the satellite will be launched, one has to work with simulated data generated by a complex computer model. In total, 68 different wavelengths are considered in the scope of the Gaia project, but for simplicity, we will consider in this paper only a subset of 16 different wavelengths showing variance in the three astrophysical parameters temperature, metallicity and gravity. Temperature is a "strong" parameter: it accounts for most of the variance across the data set.[14] Gravity and metallicity, in contrast, are "weak" parameters. The parameters have a correlated impact on the data, e.g. at high temperatures, varying the metallicity has a much smaller impact on spectra than it does at low temperatures. The data are simulated to the typical noise

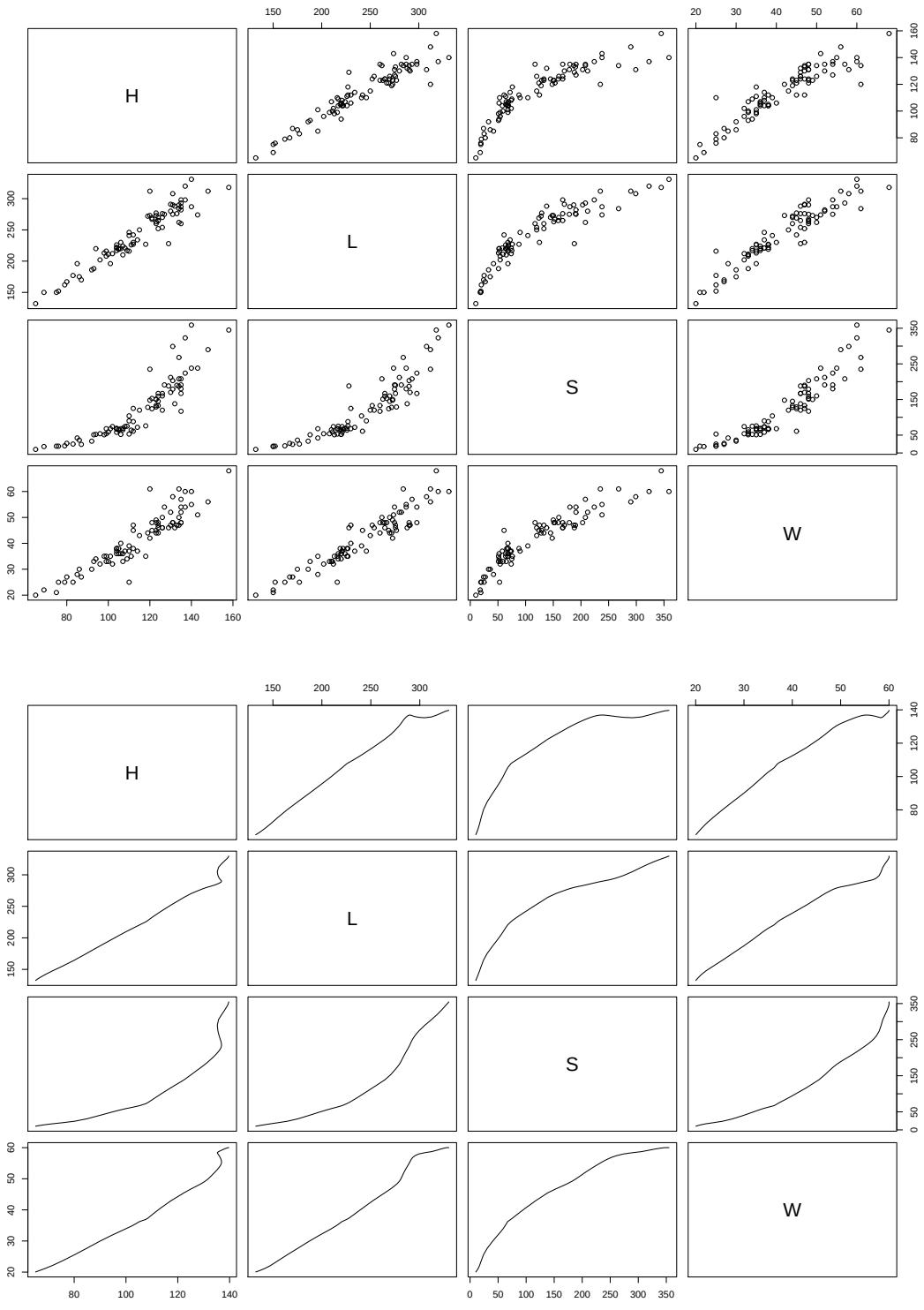


Fig. 3. Scatterplot matrix of horse mussel data (top panel); local principal curves (bottom panel). It should be emphasized the fitted LPC is *one* curve through four-dimensional space; what we are seeing here are the two-dimensional pairwise projections onto the respective coordinate axes.

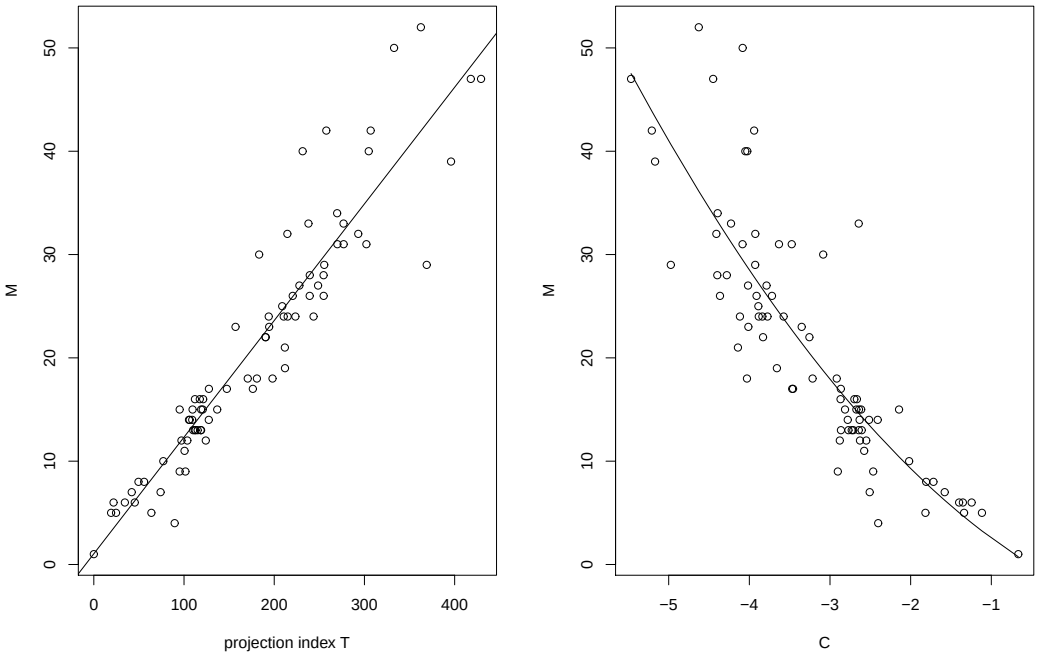


Fig. 4. Left: plot of mussel muscle mass M vs. projection indices T with regression line through the origin; right: plot of M versus the values of Bura and Cook's[6] linear combination of predictors (C) obtained via parametric inverse regression.

properties for such data, in our case Gaussian white noise.

In our setting, the photon counts form the predictor space and the APs form the response space. Note that this is opposite to the direction of simulation. A consequence is that the regression problem may be degenerate, i.e., one set of photon counts may be associated with two different APs. We focus here on the temperature, which features the least amount of degeneracy.

Approaching the data naively, one could consider fitting a multiple linear regression model, with the photon fluxes at the 16 wavelengths as regressors. However, this leads to a useless model due to the multi-collinearity induced by the high redundancy of the photon counts.[15] Obviously there is the potential for dimension reduction in this data set. To get a deeper insight into the structure of the data, we plotted the first three principal component scores against each other, yielding the data cloud depicted in figure 5 (a). Data points corresponding to higher temperatures are shaded in red. One can see that the position within the curved data cloud is informative for the temperature. Next we will fit the local principal curve, which is shown in figure 5 (b) as a solid line, with the local centers of mass represented as sky blue squares. The fitted spline function is depicted in figure 5 (c). It is clear that it is almost indistinguishable from the original LPC (and precisely coincides with it at the position of the local centers of mass). Projections onto the curve are illustrated in figure 5 (d). A scatterplot of the temperature against the projection indices is provided in figure 6, and the fitted smoothing spline is shown as a green solid curve. This spline curve provides the fitted output of the originally 16-dimensional regression problem.

Next, we perform a small simulation study to get an impression of the relative performance of the proposed technique. We sample $n' = 1000$ test data from the remaining $8286 - 1000$ observations and observe the prediction errors, $\hat{\varepsilon}_i$ = "true minus predicted temperature". The average prediction error of the test data as well as the training data are summarized in Table 1. Considering firstly the

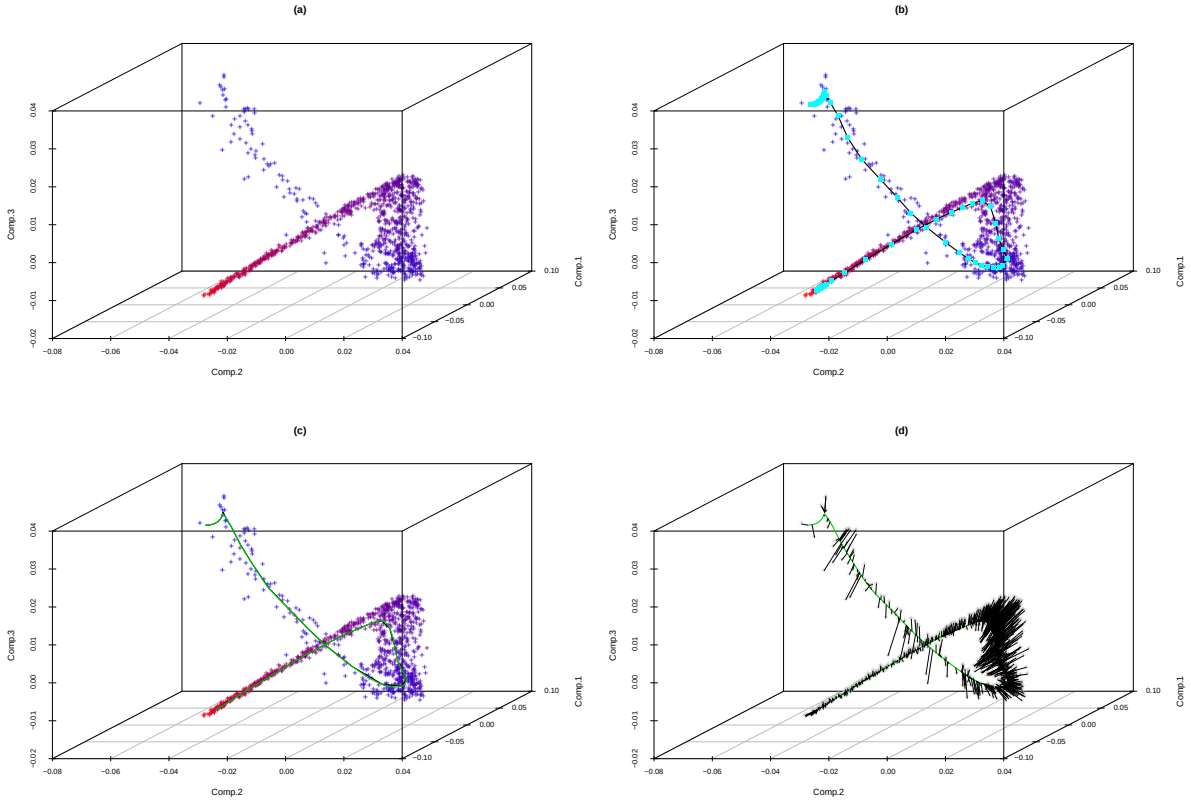


Fig. 5. (a) 3D scatterplot of principal component scores. Red data points correspond to high temperatures and blue data points to low temperatures. (b) The same plot with a local principal curve (solid), and local centers of mass plotted as light blue squares. (c) The cubic spline constructed via the algorithm in Section 2.3 is overlaid over the LPC. (d) Projections (black) onto the cubic spline (green) through PC scores (grey).

parametric methods, we observe that, unsurprisingly, PC/LM performs almost as well as LM. The additive model PC/AM beats the parametric models significantly, which is particularly evident for the medians of squared residuals. Next, we turn to LPC-based regression techniques. Note that PC/LPC stands for extracting the principal components (PC), fitting the LPC, and smoothing the response vs. the projection index, where the third step is notationally omitted for convenience. As the starting point of the LPC algorithm, we choose the point of highest density. Comparing PC/AM and PC/LPC to each other, we observe that the latter performs generally better than the former, where the improvement is larger for the medians than for the means. This can be explained through the very hot points at the left boundary of figure 6 which impact more severely on the mean than on the medians. We will attempt to improve these results even further in Section 4. We also compare our results to nonparametric regression based on a local principal curve fitted *directly* through the 16-dimensional space of spectra. The corresponding test errors given in Table 1 in the

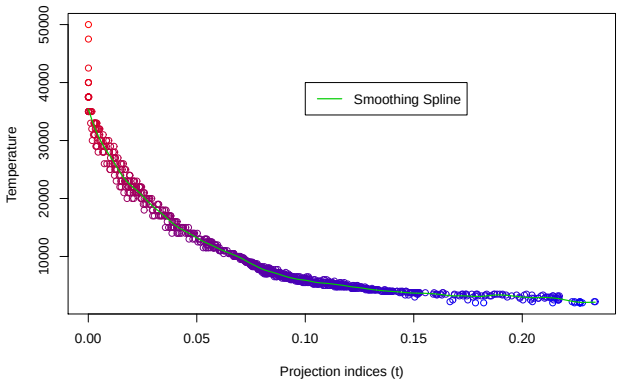


Fig. 6. Scatterplot of stellar temperatures versus PIs.

column 'LPC' indicate a very slightly improved result compared to the two-step compression PC/LPC. However, it has recently been reported that direct local principal curve regression for high-dimensional predictor spaces (say, $p > 4$ or 5) should better be avoided, or at least performed with care.[15] The reason for this is that the dependence of the LPC on the starting point, and, at the same time, the risk of missing remote data patterns, increases for data of high dimension. To shed some light on this statement, we repeated the two LPC-based regression approaches each a 100 times, but now choosing each time a starting point *at random* from the data cloud. The interquartile range of the 100 test errors observed is provided in the squared brackets. It is clearly seen that direct LPC regression behaves far less reliably than the regression based on the compressed scores.

3.3. *Sea water temperature*

The oceanographic data comes via the World Ocean Database[16], held by the American National Oceanographic Datacenter, whose data is publicly available online*. The sample studied here consists of observations over nine days in May 2000 taken by the German vessel, Gauss, in the North Atlantic. The shape of the temperature vs. depth plot is well documented in introductory oceanographic literature.[17] It shows high temperature and high variability near the surface, and a pronounced drop typically from around 1000m to 2000m known as the pycnocline: a transition stage between surface waters and bottom waters. The oxygen levels near the ocean surface also tend to be high, due to photosynthetically active plant-life there. Further down sunlight is reduced so oxygen is not produced but is still absorbed by respiring organisms. An oxygen peak at 2000m coincides with the upper surface of the previously mentioned fresh cold deep water whose presence largely is due to sea ice melt water from the poles.

One can see that the simple trends between the variables tend to break down at the surface, because of disturbances from the atmosphere, and also at the pycnocline. The variability in the second region is partially explained by considering contours of water density given its temperature and salinity. At this boundary we have a meeting of warmer saltier water (from evaporation at the surface) and colder fresher water. Whether the change in temperature, or change in salinity, dominates in its effect on the density gradient, and therefore whether the layers mix, is dependent on the water properties at the boundary.

As all variables operate on different scales, we first standardize the data by dividing each variable by their range. An LPC is fitted through the data cloud using the bandwidth $h = 0.11$. The local principal curve (as interpolated by splines) is depicted along with projections in figure 8. The curve seems to do a fairly good job, though variation around it still appears to be quite high. The question relevant for our developments is whether the projection index is informative for the target variable, water temperature. Therefore, we coloured the segments representing the projections by their associated (true, observed) temperature values. If the projection indices are meaningful for the temperature, then the colour saturation of red and blue colours should vary continuously and smoothly with the projection index. One observes that this is largely the case for the blue (cold) branch of the cloud, but something less clear occurs in the red (warm) part. Here "purple" (moderately warm) segments from one side of the curve project closely to red (warm) segments from the other side of the curve. Obviously, there is relevant information on the temperature which is not captured through the projection indices. The consequence of this can be observed in Figure 9: For the warmer regions, the plot of water temperature against the projection index features two almost parallel strings, with the upper and lower one corresponding to data on each side of the LPC. The black line is a fitted local-linear smoother, which describes the right part of the curve very well, but does not describe the left part equally well. This suggests that a (one-dimensional) curve cannot capture all the relevant information, which appears to reside in a two-dimensional surface.

One approach which allows for accessing the information orthogonal to a principal curve was proposed very recently by Ming-Ming et al.[13] They define a "second-order feature extractor" through the directed distance (i.e., distances on one side of the curve are counted negatively, and on the other

*`http://www.nodc.noaa.gov/OC5/WOD/pr_wod.html`

		LM	PC/LM	PC/AM	PC/LPC		LPC		PC/LPS		
									$\lambda = 0.1$	$\lambda = 1$	$\lambda = 10$
Training	mean	3845	4227	1199	821		793		669	753	9573
error	median	982	1073	100	46		45		22	36	1355
Test	mean	4593	4967	1732	1359	[91]	1320	[211]	1064	1227	10666
error	median	1049	1124	104	44	[3]	43	[23]	35	47	1339

Table 1. Squared prediction error/10^3 for temperature. (LM= Linear Model, PC=Principal Components, AM=Additive Model, LPC=Local Principal Curve, LPS=Local Principal Surface). The results under PC/LPS will be explained in Section 4. For all reported test errors, the starting point of the LPC or LPS was chosen to be the highest density point. The IQR of the test errors obtained through LPCs using 100 *random* starting points are provided in squared brackets.

side of the curve positively), which gives together with the first order features (the projection) a two-dimensional feature space, onto which the response can be regressed. We do not pursue this approach further in this paper, firstly because the concept of "different sides of a curve" is potentially ambiguous, and secondly as we are aiming for a more general handling of this problem by extending local principal curve methodology directly to higher-dimensional nonparametric data summaries which could be generally described as "local principal manifolds". A first but essential step to this is the extension towards local principal surfaces, which is the topic of the next section.

4. Local principal surfaces

Before we generalize local principal curves to surfaces, let us first of all go back to the local principal curve algorithm presented above. It had two important building blocks: the local first eigenvector, which is responsible for extrapolating the current curve, and the local mean, which is responsible for adjusting this extrapolated value. We will refer to this second step as *mean shift*.[18] It turns out, as we will explain below, that this mean shift is the much more important of the two steps.

The first local principal component at x is the line through μ^x which minimizes the weighted distance between data and line, with weights w_i^x as defined in part (i) of the algorithm. In other words, γ^x defines the locally optimal line, i.e. the most relevant direction to which one can turn from μ^x. However, this choice is, despite its optimality properties, by no means the only possible option.[8] In turns out that it is only important that a movement is made "into the direction of the data cloud", and the mean shift will subsequently do the job of adjusting the principal curve again towards the "middle" of the (local) distribution of the data cloud. Most importantly, if we were to replace the first local eigenvector γ^x by the direction of the previous step $\mu^\ell - \mu^{\ell-1}$, we would obtain an algorithm very similar to the local principal curve algorithm. This modified algorithm has, just like the original local principal curve algorithm, line segments as geometric building blocks. Continuing this geometric interpretation, the modified algorithm can be viewed as extending the curve by attaching a new line segment obtained by extending (or reflecting over) the last line segment and adjusting its free vertex by applying the mean shift.

We exploit this geometric view for the extension of local principal curves to local principal surfaces (LPS). The basic building block of the local principal surface algorithm is a triangle (or, if we want to estimate a r-dimensional manifold, a simplex with $r+1$ vertices). Given a triangle Δ on the boundary, we extend the surface by attaching new triangles to its "free" edges. The triangles are obtained by reflecting the current triangle Δ, or to be more precise by reflecting it at the "free" edge. In more detail, we determine the new triangle using the following steps. Suppose that the current triangle has the vertices δ_1, δ_2, and δ_3, and suppose that the edge (δ_2, δ_3) is a free edge beyond which we want to extend the surface:

(i) A preliminary vertex $\tilde{\delta}_4$ is obtained by attaching an equilateral triangle to the edge (δ_2, δ_3) such that δ_1, δ_2, δ_3, and $\tilde{\delta}_4$ all lie on the same plane. Figure 10 (a) illustrates this initial step, the preliminary vertex $\tilde{\delta}_4$ is shown in red.

(ii) Compute δ_4 from $\tilde{\delta}_4$ by carrying out a con-

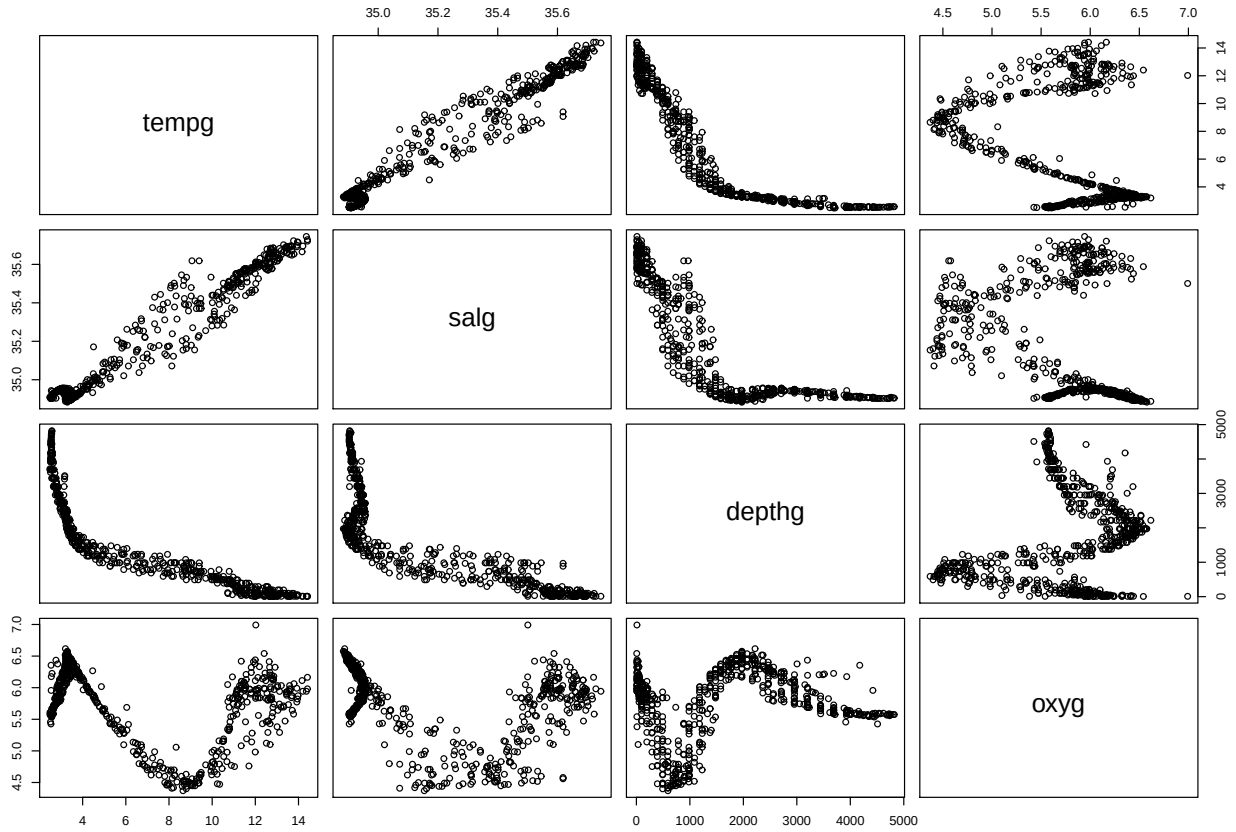


Fig. 7. Scatterplot matrix of the pre-standardized oceanographic data. Salinity is measured according to the PSS (practical salinity scale) as the ratio of the electrical conductivity against a standard solution; oxygen in millilitres per litre of water; temperature in degrees Celsius; and depth in metres. Variables are suffixed with the letter g for convenience of coding.

strained mean shift which enforces that the triangle with vertices δ_2, δ_3, and δ_4 is equilateral. Figure 10 (b) shows the weights of the observations (darker grey corresponds to higher weights) in the mean shift, with the circle in Figure 10 (c) representing the constraint. The newly obtained vertex δ_4 is shown in purple. The use of an angle penalty[8] can be beneficial in this step.

(iii) The newly-created triangle is dismissed if the Delaunay condition is violated, which is the case if an already existing vertex lies in the circumsphere of the newly created triangle or if the new vertex δ_4 lies in the circumsphere of an existing triangle. In the former case δ_4 is replaced by the already existing offending vertex. Figure 10 (d) illustrates this check. The newly-created triangle is also dismissed if the new vertex falls into a region of small density.

Step (iii) is an important ingredient of the algorithm, as these checks make sure that the branching triangles "meet" again and form a single surface instead of many parallel surfaces. Checking the density at the new vertex δ_4 is the only stopping criterion used by the algorithm and ensures that the algorithm does not extend in directions in which there is only little, or even no data. Enforcing the Delaunay condition can occasionally yield to neighbouring triangles not being connected. Thus a post-processing step is used to connect neighbouring triangles with free edges, which are not already connected. These

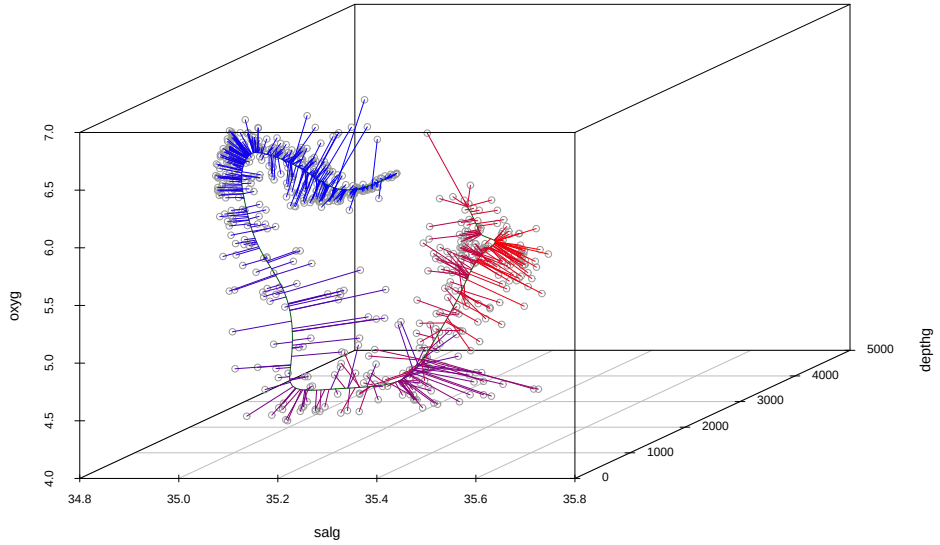


Fig. 8. Spline representation of LPC (green curve) through oceanographic data (grey circles), with the latter projected onto the former (the more "red" the colour of the segments, the higher the temperature associated to the data).

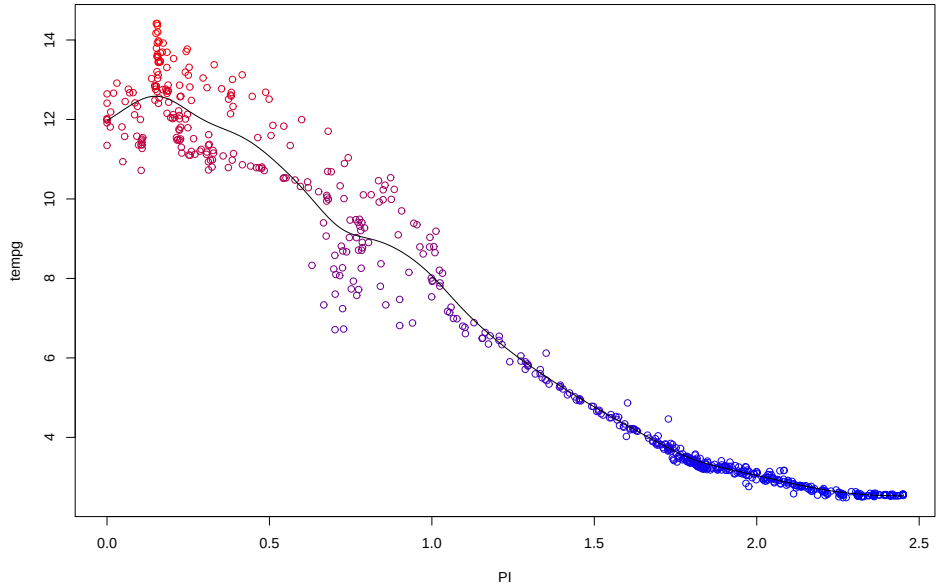
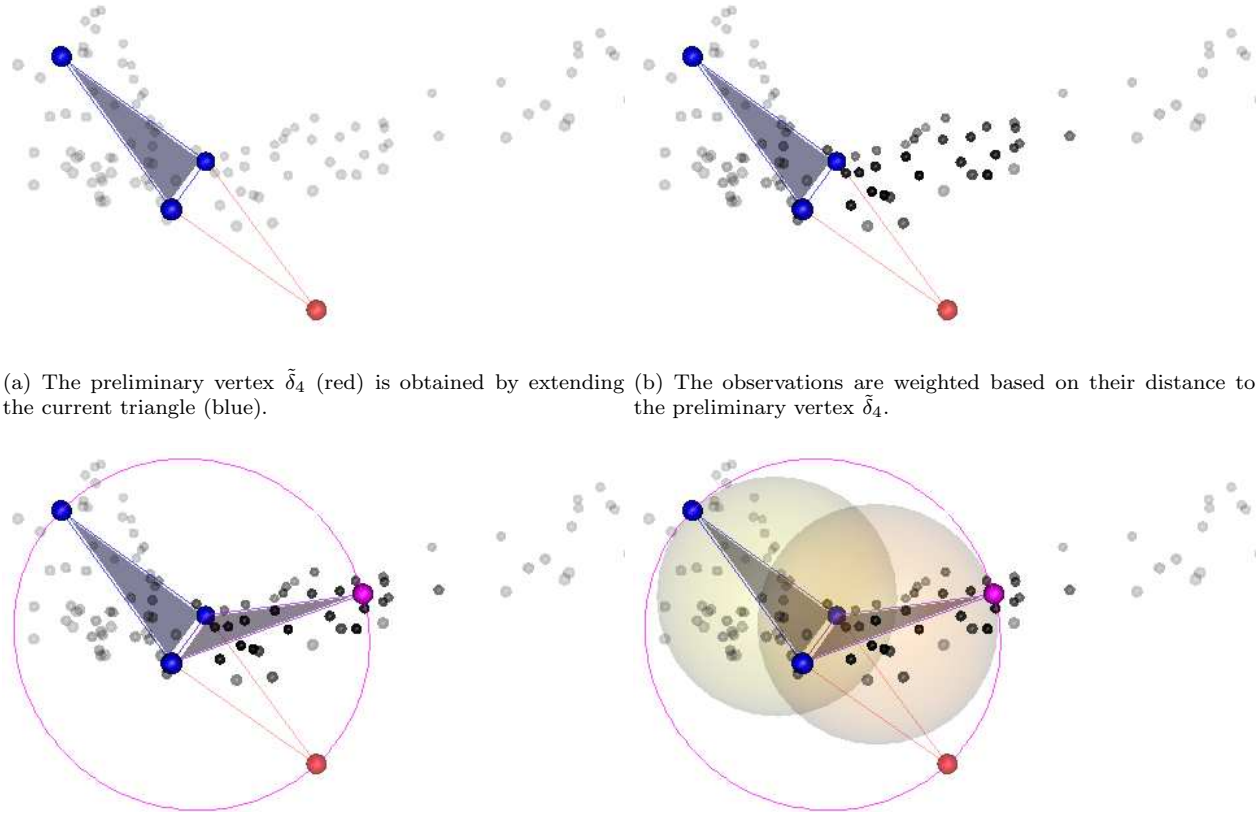


Fig. 9. Water temperature versus projection indices with local linear smoother (black curve). Again, red data points correspond to higher temperatures.



(a) The preliminary vertex $\tilde{\delta}_4$ (red) is obtained by extending the current triangle (blue).

(b) The observations are weighted based on their distance to the preliminary vertex $\tilde{\delta}_4$.

(c) The vertex δ_4 (purple) is obtained using a constrained mean-shift.

(d) Checking of the Delaunay condition.

Fig. 10. Illustration of the local principal surface algorithm for a three-dimensional toy problem.

triangles are then not necessarily equilateral.

The algorithm is initialized like the local principal curve algorithm. The first two local principal components are computed based on a (manually or randomly chosen) starting value x_0. The initial triangle is placed in the plane spanned by the first two local principal components. We now apply this algorithm to the oceanographic data. The fitted surface is shown in figure 11: it nicely captures the shape of the data cloud.

To demonstrate how powerful the information contained in the surface is, we combine it with a very simple local kernel regression with a discrete bivariate kernel. More precisely, for each pair of triangles we define the (discrete) "distance" d between them as the smallest number of triangle borders that need to be crossed to proceed from one triangle on the surface to the other one. This distance is cheap to compute and can for example be obtained by applying Dijkstra's algorithm to the neighborhood graph. In order to assign local weights, we define the discrete distance-based kernel $\kappa(d) = e^{-d/\lambda}$, where λ is a smoothing parameter. Important special cases are $\lambda = 0$, in which case $\kappa(0) = 1$ and $\kappa(d) = 0$ for $d > 1$, i.e. no smoothing at all, and $\lambda \longrightarrow \infty$, in which case $\kappa(d) = 1$ for all $d \geq 0$, i.e. the estimated response function is constant.

The smoothed response value $\hat{y}_\Delta$ on triangle Δ is then given by

$$\hat{y}_\Delta = \frac{\sum_{\Delta'} \kappa(d_{\Delta,\Delta'}) \bar{y}_{\Delta'}}{\sum_{\Delta'} \kappa(d_{\Delta,\Delta'})},$$

where $\bar{y}_{\Delta'}$ is the mean of all observations for which Δ' is the closest triangle, and $d_{\Delta,\Delta'}$ is the discrete distance between the triangles Δ and Δ'.

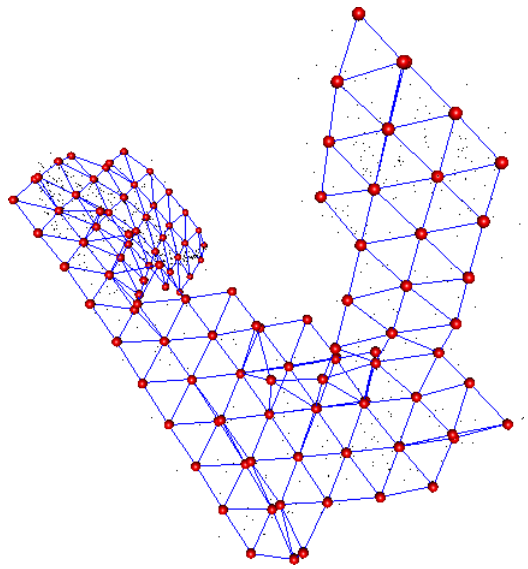


Fig. 11. LPS for the oceanographic data.

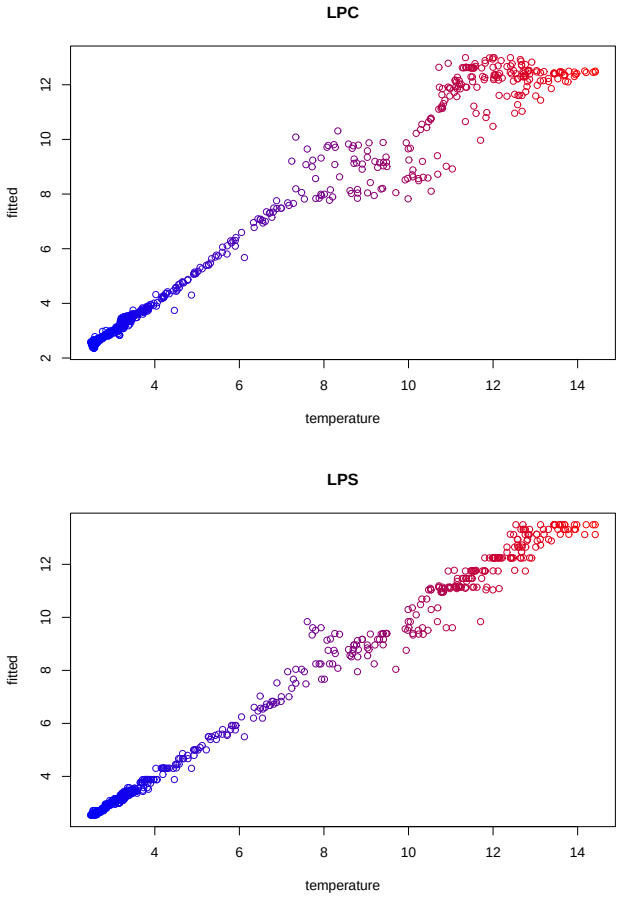


Fig. 12. Top: LPC-based fitted values vs. true temperature values; bottom: LPS-based fitted values vs. true temperature values.

This model is admittedly rather crude, but has the advantage that it does not require finding a parametrization of the fitted local principal surface. We will however see that, despite its simplicity, this model allows us to improve our predictions obtained using the LPC algorithm.

For the oceanographic data, we obtain the LPS shown in figure 11, which features 177 triangles with an average count of 3.63 data points per triangle. We compute the fitted values as outlined above and plot them versus the true temperatures in figure 12. It is clearly seen that, when using the projections onto the LPS, the inconvenient branched structure which was observed for the LPCs disappears.

We also fitted the surface for the stellar temperature data with smoothing parameters $\lambda = 0.1$, $\lambda = 1$, $\lambda = 10$. The result is provided in Table 1. The message to be taken from this is that the prediction error does improve (compared to the LPC-based method) when accounting for the two-dimensional nature of the shape of the data. However this new technique is sensitive to the choice of the smoothing parameter λ. For too small λ, overfitting is inevitably present. This can be alleviated by increasing λ, which decreases effectively the degrees of freedom used for the regression fit. Note that, for the data at hand, overfitting does not seem to constitute much of a problem since the average test errors are even for small smoothing parameters almost of the same magnitude as the average training errors.

5. Conclusion

In this article we have presented a novel approach to regression based on exploiting the structure of the covariate space by fitting a local principal curve or surface to the covariate space. The data examples studied showed that such a strategy can be very successful. In all the examples the method based on local principal curves and surfaces outperformed the competing methods.

However this does not always need to be the case. Firstly, the data might not exhibit a manifold structure at all. But even if the data lies to a large extent on a low-dimensional structure, it might be that the information relevant to the response variable of interest is not represented in the manifold structure.

From this point of view local principal curves and surfaces are no different to principal components. For instance, when replacing the "strong" response variable temperature by the "weak" variable metallicity in the Gaia example, all methods considered in this paper give relatively poor results, with values of R^2 around 0.2. This is simply a very hard estimation problem and any form of dimension reduction cannot do much about this. An entirely different approach to this problem based on forward modelling was recently provided by Bailer-Jones.[14]

We conclude with pointing out a connection to the elastic net algorithm of Gorban and Zinovyev.[19] Both the local principal curve algorithm and the local principal surface algorithm cannot update the location of an already created line segment or triangle. However one can view both the local principal curve and the edges of the local principal surface as some sort of elastic net and thus postprocess the estimated curve or surface with the elastic net algorithm. This could be beneficial in order to smooth out minor irregularities on the fitted surface as visible for instance in the bottom of figure 11. Furthermore this allows for estimating the local principal curve or manifold in a low-dimensional "pilot" space and using the elastic net algorithm for embedding the curve or surface in the original data space.

Acknowledgments

We are grateful to Coryn Bailer-Jones, MPIA Heidelberg, for providing the Gaia data and explaining their astrophysical context.

References

1. C.A.L. Bailer-Jones. Determination of stellar parameters with GAIA. *Astrophysics and Space Science*, 280:21–29, 2002.
2. C.A.L. Bailer-Jones, K.W. Smith, C. Tiede, R. Sordo, and A. Vallenari, Finding rare objects and building pure samples: Probabilistic quasar classification from low resolution Gaia spectra. *Monthly Notices of the Royal Astronomical Society*, 391:1838-1853, 2008.
3. J. Fan and I. Gijbels. *Local Polynomial Modelling and its Applications*. Chapman and Hall, London, 1996.
4. N. Keeratipranon, F. Maire, and H. Huang. Manifold learning for robot navigation. *International Journal of Neural Systems*, 16: 383-392, 2006.
5. K. Li. Sliced inverse regression. *J. Amer. Statist. Assoc.*, 86:316–327, 1991.
6. E. Bura and R. D. Cook. Estimating the structural dimension of regressions via parametric inverse regression. *Journal of the Royal Statistical Society, Series B*, 63:393–410, 2001.
7. T. Hastie and W. Stuetzle. Principal curves. *J. Amer. Statist. Assoc.*, 84:502–516, 1989.
8. J. Einbeck, G. Tutz, and L. Evers. Local principal curves. *Statistics and Computing*, 15:301–313, 2005.
9. F. Camastra. Data dimensionality estimation methods: a survey. *Pattern recognition*, 36:2945–2954, 2003.
10. R.D. Cook. *Regression Graphics — Ideas for studying regressions through graphics*. Wiley, New York, 1998.
11. P. Delicado. Another look at principal curves and surfaces. *Journal of Multivariate Analysis*, 77:84–116, 2001.
12. J. Einbeck, G. Tutz, and L. Evers. Exploring multivariate data structures with local principal curves. In C. Weihs and W. Gaul, editors, *Classification - The Ubiquitous Challenge*, pages 257–263, Springer, Heidelberg, 2005.
13. S. Ming-Ming, Y. Jian, L. Chuan-Cai, and Y. Jing-Yu. Similarity preserving principal curve: an optimal one-dimensional feature extractor for data representation. *IEEE Transactions on Neural Networks*, to appear, 2010.
14. C.A.L. Bailer-Jones. The ILIUM forward modelling algorithm for multivariate parameter. *Monthly Notices of the Royal Astronomical Society*, to appear, 2010.
15. J. Einbeck, L. Evers, and K. Hinchliff. Data compression and regression based on local principal curves. In A. Fink, B. Lausen, W. Seidel, and A. Ultsch, editors, *Advances in Data Analysis, Data Handling and Business Intelligence*, pages 701–712, Springer, Heidelberg, 2009.
16. T. Boyer, J. Antonov, H. Garcia, D. Johnsonn, R. Locarnini, A. Mishonov, M. Pitcher, O. Baranova, and I. Smolyar. (2006). World ocean database 2005. In *NOAA Atlas NESDIS 60*. Washington, D.C.
17. T. Garrison. *Essentials of Oceanography (Fifth Edition)*. Brooks/Cole, Belmont, Canada, 2009.
18. D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Machine Intell.*, 24:603–619, 2002.
19. A.N. Gorban and A.Y. Zinovyev. Elastic principal graphs and manifolds and their practical applications. *Computing*, 75:359–379, 2005.