Localized regression on principal manifolds

Jochen Einbeck[1] and Ludger Evers[2]

[1] Department of Mathematical Sciences, Durham University, Durham DH1 3LE, England, jochen.einbeck@durham.ac.uk
[2] Department of Statistics, University of Glasgow, Glasgow G12 8QQ, Scotland

Abstract: We consider nonparametric dimension reduction techniques for multivariate regression problems in which the variables constituting the predictor space are strongly nonlinearly related. Specifically, the predictor space is approximated via “local” principal manifolds, based on which a kernel regression is carried out.

Keywords: Smoothing; principal curves and surfaces; localized PCA.

1 Introduction

This article deals with the problem of multivariate regression for situations where the (possibly high-dimensional) predictor space features complex dependency patterns. As an example, consider oceanographic data extracted from the World Ocean Database, which include measurements on the water temperature (serving as the response variable, Y), and the three covariates X_1=salinity, X_2=water depth, and X_3=oxygen content (Fig. 1 left). Obviously, the three covariates are highly and nonlinearly related and contain partially redundant information. Potential modelling strategies include a full interaction model $Y = m(X_1, X_2, X_3) + \epsilon$, which becomes the more difficult the more covariates are involved, or an additive model $Y = m(X_1) + m(X_2) + m(X_3) + \epsilon$, which ignores the interaction between the variables.

Neither of these methods exploits the fact that the covariates occupy a space of lower intrinsic dimensionality than 3. Formulating the problem more generally: We are given a regression problem with response Y and predictor space $X = (X_1, \ldots, X_p)^T$. We aim for a two-step strategy which would (1) approximate X nonparametrically by a curve, surface, or, more generally, a low-dimensional manifold of dimension $d < p$, and (2) use the compressed data as a $d-$dimensional predictor henceforth. In this sense, this article provides an extension of principal component regression, being nonparametric both in the compression and the regression step. We assume that the intrinsic dimensionality of the manifold, d, is given, e.g. from visual inspection of the data cloud. Dimensionality estimation is beyond the scope of this paper; an overview on such methods is given in Camastra (2003).

2 Methodology

2.1 The case $d = 1$

We are given independent replicates $x_1, \ldots, x_n \in \mathbb{R}^p$ drawn from the random vector X, i.e. $x_i = (x_{i1}, \ldots, x_{ip})^T$. For the compression step (1), we use the local principal curve algorithm (LPC; Einbeck et al., 2005), which can be summarized as follows. Let w_i^x denote an appropriate (bell-shaped) weight function centered at $x \in \mathbb{R}^p$. Beginning at some starting point $x = x_0 \in \mathbb{R}^p$, we calculate $\mu^x = \sum_{i=1}^n w_i^x x_i$, and then iterate

(i) Compute the first local eigenvector γ^x of $\Sigma^x = (\sigma_{jk}^x)_{(1 \leq j,k \leq p)}$, 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 step size z, step from μ^x to $x := \mu^x + z\gamma^x$;

(ii) Calculate the local center of mass μ^x;

until the distance between neighboring values of μ^x becomes negligible. The resulting series of μ^x, which defines the local principal curve, is subsequently connected through a cubic spline and parametrized by its arc length. Each data point is then projected to its nearest point on the curve, and the compressed data correspond to their projection index (PI). This is illustrated in Fig. 1 (left). Details on the parametrization and projection are found in Einbeck et al. (2010). In the regression step (2), we regress the response versus the PIs, using any univariate nonparametric smoother (e.g., local linear). This is illustrated in Fig. 1 (right).

2.2 The case $d \geq 2$

The use of localized principal components in (i) is by no means the only possible option. If we replaced γ^x by the direction of, say, the vector connecting the previous and the current local center of mass, then step (ii) would adjust the principal curve again towards the “middle” of the (local) data distribution. This slightly modified algorithm has, just like the original LPC algorithm, line segments as geometric building blocks in step (i). We exploit this idea for the extension of LPCs to local principal manifolds (LPMs). As the basic building block we will now use a triangle ($d = 2$), tetrahedron ($d = 3$), or simplex ($d \geq 4$). Although the algorithm that we are going to propose can in principle be applied using any $2 \leq d < p$, we will describe it for ease of presentation for the special case $d = 2$, in which case the resulting object is a local principal surface (LPS).

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 Δ at the free edges. Suppose that the current triangle Δ has the vertices δ_1, δ_2, and δ_3, and that the edge (δ_2, δ_3) is a free edge beyond which we want to extend the surface:

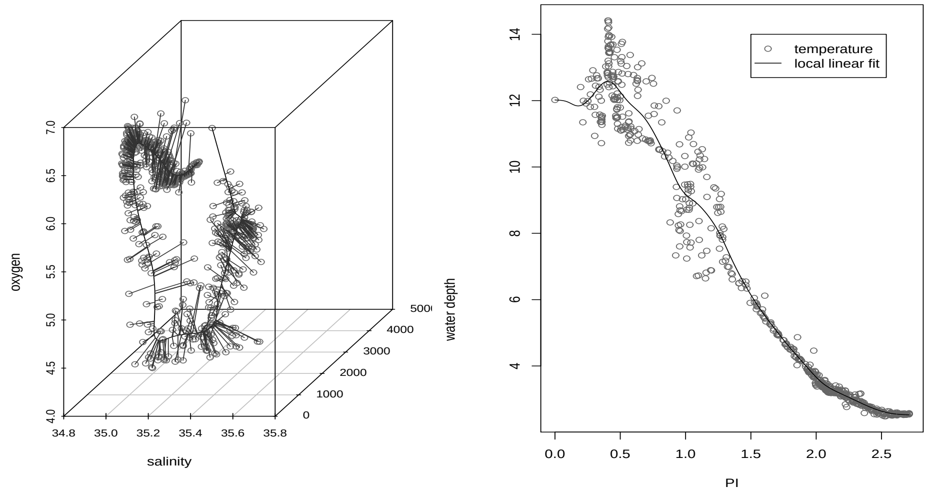

FIGURE 1. Left: 3d- scatterplot (grey circles) of salinity (measured on the 'Practical Salinity Scale'), water depth (metres), and oxygen content (millilitre/litre of water). Solid curve: cubic spline representation of local principal curve, with orthogonal projections; right: water temperatures plotted vs. projection indices.

(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. The bottom right point in Fig. 2 (left) illustrates this preliminary vertex.

(ii) Compute δ_4 from $\tilde{\delta}_4$ as a constrained local center of mass, which enforces that the triangle with vertices δ_2, δ_3, and δ_4 is equilateral. Fig. 2 (left) shows the weights of the observations (darker grey corresponds to higher weights), with the circle representing the constraint. The new vertex δ_4 is shown in the top right. The newly-created triangle is dismissed if an already existing vertex lies in its circumsphere 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), or if the new vertex falls into a region of small density.

The initial triangle is placed in the plane spanned by the first two local principal components obtained at a (manually or randomly chosen) starting value x_0. Steps (i) and (ii) correspond to their counterparts in the LPC algorithm. The checks for dismissal of vertices in (ii) ensure that branching triangles "meet" again and do not form many parallel surfaces.

We now apply the LPS algorithm to the oceanographic data. The fitted surface, which features 177 triangles with an average count of 3.63 data

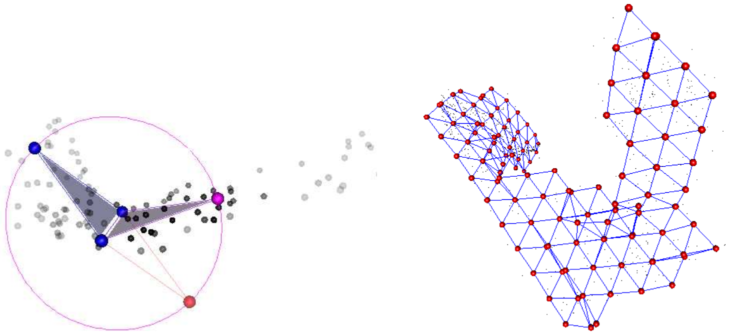

FIGURE 2. Left: Illustration of the LPS algorithm; right: Fitted LPS for oceanographic data.

points per triangle, is shown in Fig. 2 (right): it nicely captures the shape of the data cloud.

It is hard to find a full bivariate parametrization of the LPS. Therefore, we use a simple kernel regression. For each pair of triangles we define the (discrete) "distance" d 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 can be obtained by applying Dijkstra's algorithm to the neighborhood graph, and is thus cheap to compute. In order to assign local weights, we define the discrete distance-based kernel $\kappa(d) = e^{-d/\lambda}$, where λ is a smoothing parameter. Special cases are $\lambda = 0$, corresponding to no smoothing at all, and $\lambda \longrightarrow \infty$, where 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'. Though formulated here in the special case $d = 2$, both the estimation of the manifold, as well as the kernel regression on it, extend straightforwardly to higher intrinsic dimensions $d > 2$ by using the appropriate geometric building block.

In order to study the performance of this technique, we split the $n = 643$ observations into a training set of size 500 and a test set of size 143. We include in our study the additive model (AM) as well as localized regression on a local principal curve (LPC) or surface (LPS). The training data are used to learn these nonparametric models. The smoothing parameters for the smooth terms in the additive model and the local smoother on the principal curve are calibrated so that a total of ≈ 16 degrees of freedom is used in each model. For the regression on the surface, we compare three different choices of the smoothing parameter λ. The results of this study

		AM	LPC	LPS		
				$\lambda = 0.2$	$\lambda = 1$	$\lambda = 2$
Training	mean	0.08946	0.32606	0.04335	0.07380	0.14444
error	median	0.01538	0.00650	0.00143	0.00655	0.01455
Test	mean	0.15494	0.30962	0.11090	0.11569	0.17471
error	median	0.02855	0.00877	0.00395	0.01009	0.02059

TABLE 1. Mean and median prediction errors for the training and test data; using AM-, LPC- and LPS-based regression, respectively.

are displayed in Table 1. As expected, the LPC-based regression is inferior to the additive model in terms of the mean prediction error (i.e., the mean of squared distances between predicted and true temperature). The poor performance of the LPC-based technique is due to the branched shape of the response data seen in Fig. 1 (right). The LPS-based approach clearly outperforms the additive model for $\lambda \leq 1$, though for $\lambda = 0.2$ considerable overfitting (undersmoothing) appears to be present, which is reflected in test errors that are about three times larger than the training errors. The choice $\lambda = 2$ leads to larger prediction errors; here we have oversmoothed. Considering the *median* instead of the mean prediction error, the performance of all investigated methods improves drastically (relative to the additive model), which can be explained with an increased robustness of the median to very poor predictions, which can occasionally happen for the LPC/LPM- based approaches especially in the boundary regions.

3 Conclusion

We have presented an entirely nonparametric approach to modelling data which feature a low-dimensional non-linear latent structure. Just like the local principal curves (LPC) algorithm, this local principal manifolds algorithm (LPM) is based on the simple geometric idea of locally approximating the data by connected simplices.

Of course, not every data set will have such a low-dimensional structure. The majority of data sets probably does not, but there are still surprisingly many datasets which do have such a structure. Once the algorithm has established the low-dimensional latent structure, one can use it to define new, data-dependent topologies, which often give a better representation of the dynamics underlying the data than the standard Euclidean distance in the original data space. This implied dimension reduction can, for example, be exploited when studying regression problems, as illustrated in the example shown in the preceding section. Other applications include classification or density estimation on the manifold.

Acknowledgments: We wish to thank Benedict Powell, Durham University, for retrieving, preparing, and sharing his insight into the oceanographic data set.

References

Camastra, F. (2003). Data dimensionality estimation methods: a survey. *Pattern recognition* **36**, 2945–2954.

Einbeck, J., Tutz, G., and Evers, L. (2005). Local principal curves. *Statistics and Computing* **15**, 301–313.

Einbeck, J., Evers, L., and Hinchliff, K. (2010). Data compression and regression based on local principal curves. In Fink et al. (Eds): Advances in Data Analysis, Data Handling, and Business Intelligence, pp. 701–712, Heidelberg: Springer.